3D IC Floorplanning with Leakage-aware Decoupling Capacitor Insertion

ABSTRACT

Decoupling capacitor (decap) insertion is a popular method for reducing power supply noise in integrated circuits. In 3D integrated circuits, allowing circuit modules to access decaps in other device layers can reduce the area overhead of inserting decaps. In order to achieve this goal, we devise the Effective Distance formulation to analyze how circuit modules are affected by non-neighboring decaps. In addition, we propose a new algorithm for allocating whitespace in 3D floorplan to decap while taking effective decap distance into consideration. Our algorithm, which is based on generalized network flow, also has the flexibility to assign multiple decap oxide thicknesses for leakage power reduction. The generalized network flow-based decap allocation algorithm is integrated with a decap-aware 3D floorplanner. Experimental results show that our floorplanner combined with our decap allocator can significantly reduce decap budget and leakage with a small increase in area and wirelength.

1. INTRODUCTION

Three dimensional (3D) integrated circuits are an emerging technology with great potential to improve performance. In a 3D integrated circuit, transistors may be fabricated on top of other transistors, resulting in multiple layers of active components. These transistors may then be wired to other transistors on the same device layer, to transistors on different device layers, or both, depending on the process technology. Several different approaches to fabricating 3D integrated circuits or 3D-compatible transistors have been taken [1, 2, 3]. These techniques vary in terms of the maximum number of device layers and the maximum density of interconnects between these layers. The wafer-bonding approach in [3] joins discrete wafers using a copper interconnect interface, and permits multiple wafers and multiple 3D interconnects. 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.

Signal integrity is a very important issue in VLSI 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. Inserting decoupling capacitors (decaps) is one method of alleviating power supply noise. Numerous decap allocation algorithms have been proposed. In [4], quadratic programming was used to insert decaps into standard cell placements. Conjugate gradients were used in [5] for decap allocation. [6] and [7] refine conjugate gradient methods for decap allocation. In [8], linear programming was used to allocate whitespace for decap after floorplanning. A power integrity-aware floorplanning for 2D ICs was presented in [9].

This paper presents the first 3D floorplanning algorithm for power supply noise and leakage reduction. We perform 3D module placement while minimizing the amount of decap needed to suppress the noise. This is followed by a simultaneous decap insertion and thickness assignment step that minimizes the area cost and leakage power. The decap allocation problem in a 3D IC has a couple of additional factors not present in the 2D case. Having multiple layers creates the possibility of allowing circuit modules to access decaps on other layers. To make this happen, we analyze the effect of placing a decap at varying distances from a circuit module. From this analysis, we formulate the *effective decap distance* model to allow circuit modules to access non-neighboring decaps, including decaps on other layers. The effective distance model is integrated into our *generalized min-cost network flow* based decap allocation algorithm. Often the existing whitespace in a floorplan is insufficient to supply the needed decap. In those cases a floorplan can be expanded to add additional whitespace. In 3D IC's, 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.

As VLSI technology continues to scale down, noise tolerances will become tighter. This will increase the amount of decap required to bring power supply noise within the tolerances. Technology scaling reduces the oxide thickness of on-chip capacitors. This has the benefit of increasing the capacitance per unit area of decaps. Unfortunately, thinner oxides can significantly increase the leakage current of decaps. This problem is addressed in [10] by performing wire sizing of the power/ground network after decap insertion. Reverse scaling of capacitor oxide thicknesses was explored in [11]. This paper also uses reverse scaling of capacitor oxide thicknesses. Our network flow-based decap allocation algorithm can automatically assign multiple decap oxide thicknesses to reduce leakage while suppressing noise.

The remainder of this paper is organized as follows. The problem formulation is presented in Section 2. Our 3D floorplanning, decap insertion, and thinkness assignment algorithms are presented

DAC'06, July 24–28, 2006, San Francisco, California, USA.

in Section 3. Experimental results are provided in Section 4, and conclusions are in Section 5.

2. PROBLEM FORMULATION

The following are given as the input to the 3D floorplanning problem: (i) a set of blocks that represent the circuit modules, (ii) width, height, and maximum switching currents for each block, (iii) a netlist that specifies how the blocks are connected, (iv) the number of placement layers in the 3D IC, (v) the oxide thicknesses available for decap fabrication, (vi) the location of the power/ground pins, and (vii) the power supply noise tolerance. Let W denote the total wirelength of the 3D floorplan. Let A denote the final footprint area of the 3D floorplan. Let D denote the total amount of decoupling capacitance required to suppress the power supply noise under the given tolerance value. The first goal is to find the location of each block in the floorplan such that $w_1 \cdot A + w_2 \cdot W + w_3 \cdot D$ is minimized, where w_1, w_2, and w_3 are weighting factors for the three objectives. The second goal is to allocate whitespace for decaps and assign oxide thicknesses to the decaps, while limiting the leakage current caused by the decaps. If existing whitespace cannot fill all of the decap demand, then the floorplan will be expanded to add additional whitespace. This expansion is to be minimized.

3. 3D FLOORPLANNING ALGORITHM

3.1 Overview of the Algorithm

Simulated Annealing is a popular approach for floorplanning due to its high quality solutions and flexibility in handling various constraints. We extend the existing 2D Sequence Pair scheme [12] to represent 3D floorplans. Specifically, k sequence pairs are used to represent the block placements of k device layers. This representation only encodes relative block positions for blocks in the same layer. However, it is straightforward to figure out the inter-layer position relationships by computing the block coordinates. Simulated Annealing starts with an initial multi-layer placement along with its cost in terms of area, wirelength, and decap. The floorplan is perturbed by swapping random pairs of blocks from the same layer or from different layers. The following steps are performed to measure the decap cost for a given 3D floorplanning solution:

1. SSN noise analysis: the amount of simultaneous switching noise (SSN) for each block is computed based on the location of the blocks and power pins.

2. decap budget calculation: the amount of decap needed for each block is computed based on its SSN so that the overall SSN constraint is satisfied.

If the new cost is lower than the old one, the solution is accepted; otherwise the new solution is accepted based on some probability that is dependent on temperature of the annealing schedule. A pre-determined number of candidate solutions are examined at each temperature. The temperature is decreased exponentially, and the annealing process terminates when the freezing temperature is reached.

After floorplanning, decaps are inserted based on the decap budget calculated during floorplanning. First, the existing whitespace in the floorplan is detected. Then, a generalized network flow graph is constructed. Solving the generalized flow network allocates whitespace for decap and assigns oxide thicknesses to the decaps. If not all of the decap budgets of the blocks are filled, then the footprint aware area expansion is performed on the floorplan to add extra whitespace. After expansion, generalized network flow based decap allocation is performed again. Iteration between decap allocation and floorplan expansion is performed until the decap demands of all of the blocks are satisfied.

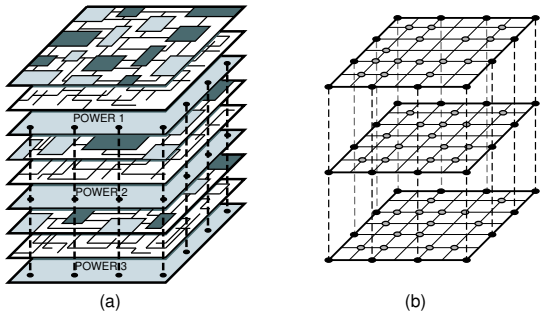

Figure 1: Illustration of 3D power supply modeling. (a) multi-layer power supply network, where multiple sets of device, routing, and power supply layers are stacked together, (b) 3D-grid modeling. where black and gray nodes respectively denote power supply and consumption nodes.

3.2 3D Power Supply Noise Modeling

Our grid-based 3D P/G network modeling is shown in Figure 1. Each P/G layer in the multi-layer structure is represented as a mesh. The edges in the mesh have inductive and resistive impedances. The mesh contains power-supply points 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 current source* for a block is defined as the voltage source supplying significantly more power to the block than any other neighboring sources. The *dominant path* for a block is the path from the dominant supply to the block causing the most drop in voltage. It has been shown experimentally in [8] that the shortest path between the dominant current source (nearest Vdd 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 $R_{P_{jk}}$ and $L_{P_{jk}}$ denote the resistance and inductance of P_{jk}. After the current paths and their values have been determined for all blocks, the SSN for B_k is given by

$$V^k_{noise} = \sum_{P_j \in T^k} (i_j \cdot R_{P_{jk}} + L_{P_{jk}} \frac{di_j}{dt})$$

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 and its rate of change are the resistive and inductive components of the path. An illustration is shown in Figure 2.

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 time. The decap budget can then be calculated as:

$$C^k = Q^k / V_{tol},\ 1 \leq k \leq M \quad (1)$$

where V_{tol} is the noise tolerance of the block, and M denotes the total number of blocks. This base decap budget is for the case where there is no resistance between a block and its decap.

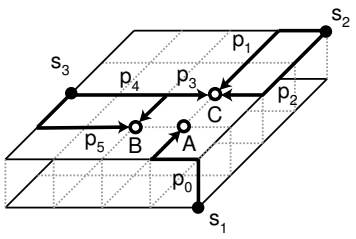

Figure 2: Illustration of SSN calculation. The dominant current source for block A is s_1, which is not located in the same layer. The dominant (shortest) path p_0 carries $I_A/6$ amount of current, where I_A denotes the current demand of A. The block C draws current from s_2 and s_3 using p_1, p_2, 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.

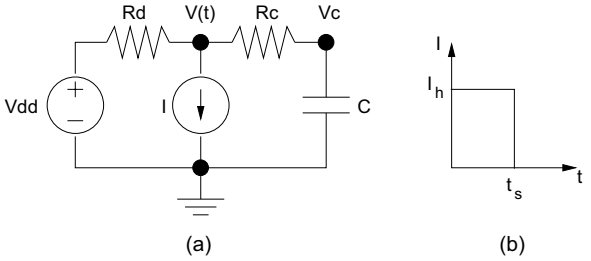

Figure 3: (a) circuit used for effective distance formulation. V_{dd} is the power pin. I is the current demand of the block. C is the decap. R_d is the resistance between the block and power pin. R_c is the resistance between the block and decap. (b) switching current of the block.

3.3 Decap Modeling with Effective Distance

A recent work on decap-aware floorplanning for 2D ICs [8] only assigns decaps to blocks when they are adjacent to each other. However, blocks can potentially draw current from *all* nearby decaps, including the ones that are not adjacent. This restriction may result in excessive decap insertion and thus unnecessary floorplan area expansion. We introduce the concept of *effective distance* to overcome this limitation and to make use of non-adjacent white spaces for decap allocation. A decap placed far away from a block is less effective at reducing noise. Effective distance, $\gamma_{eff}(R_c)$, 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.

The circuit shown in Figure 3 as analyzed to find a relationship between 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

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

where $V_{noise} = R_d \cdot I_h$ (see Figure 4). This equation can be solved

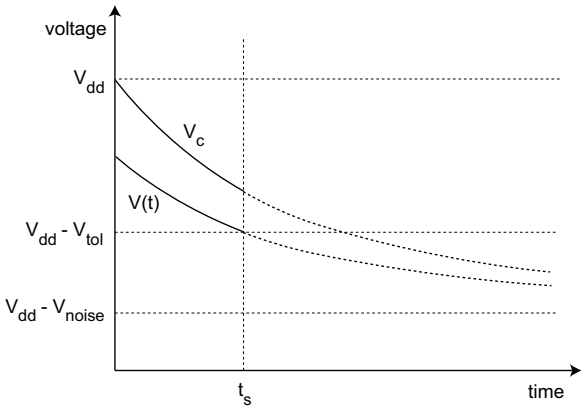

Figure 4: The voltage of the circuit module $V(t)$ and the voltage of the capacitor V_c during switching. V_{dd} is the voltage of the power pin. V_{tol} is the maximum noise the block can handle. V_{noise} is the SSN.

for C to find the amount of decap needed by the block.

$$C = \frac{-t_s}{(R_c + R_d)[\ln\frac{(V_{noise}-V_{tol})}{V_{noise}} + \ln\frac{R_c+R_d}{R_d}]}$$

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

$$R_{max} = \frac{R_d \cdot V_{tol}}{V_{noise} - V_{tol}}$$

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

$$\gamma_{eff}(R_c) = \frac{C(R_c)}{C(0)} = \frac{R_d \cdot \ln\frac{V_{noise}-V_{tol}}{V_{noise}}}{(R_c + R_d)[\ln\frac{V_{noise}-V_{tol}}{V_{noise}} + \ln\frac{R_c+R_d}{R_d}]}$$

To find the actual decap allocated to a block, the base decap budget C^k is calculated from Equation (1) and multiplied by $\gamma_{eff}(R_c)$. 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. Figure 5 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 $0.7R_{max}$ where 50% extra decap would be needed.

3.4 Whitespace Detection and Insertion

The white space present in a floorplan can be used to fabricate decap. If the existing white space is insufficient or unreachable by modules needing decap, then white space insertion through floorplan expansion may be necessary. Hence detection of all existing white spaces in a floorplan is highly desirable. This is done by using the longest path tree calculation based on the vertical constraint graph. All nodes at the i^{th} level in the tree are at an edge distance of

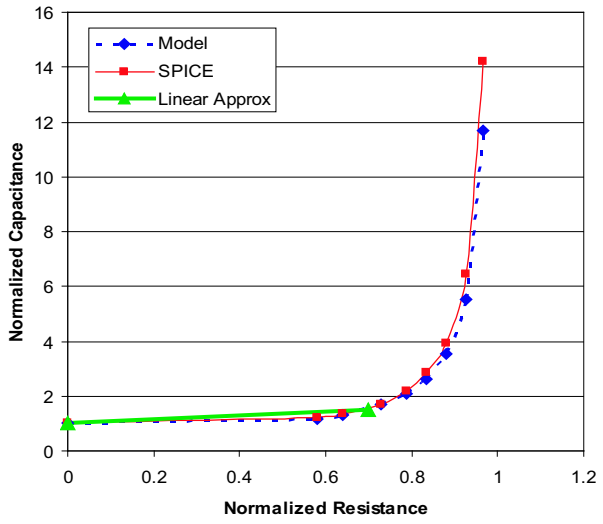

Figure 5: SPICE modeling on decap requirement as a function of resistance R_c, which is normalized with respect to R_{max}. Normalized capacitance is equivalent to γ_{eff}.

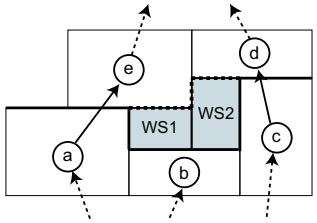

Figure 6: Whitespace detection. Blocks a, b, c are in the lower level. Blocks d, e are in the next level. The bold line is the lower boundary, while the dotted line is the upper boundary. $ws1, ws2$ are the detected white spaces.

i from the source node. Each level is ordered by the horizonal constraint graph. The white spaces at level i are detected by comparing the *upper* boundary of blocks at level i and the *lower* boundary of the blocks at level $i+1$. If the boundaries are not incident on each other, then there is whitespace. In Figure. 6, blocks a, b, c are in the same level and blocks d, e are in the next level. The algorithm compares the upper boundary of a, b, c, to the lower boundary of d, e. The mismatched boundaries allows the algorithm to find white spaces $ws1, ws2$. This algorithm is capable of detecting all white spaces, and runs in $O(n)$ time, given the ordered longest path tree, where n is the total number of blocks. Typically, longest path tree calculations from constraint graphs are used to convert sequence pairs into floorplans.

If sufficient decap cannot be allocated from the existing white space to suppress the SSN, then more white space is added by expanding the floorplan in the X and Y direction as illustrated in Figure 7. A naive approach is to look at the additional decap needed for each layer and expand as necessary, splitting the X and Y expansion evenly. However, this does not take advantage of the 3D structure. 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 bottle-neck 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.

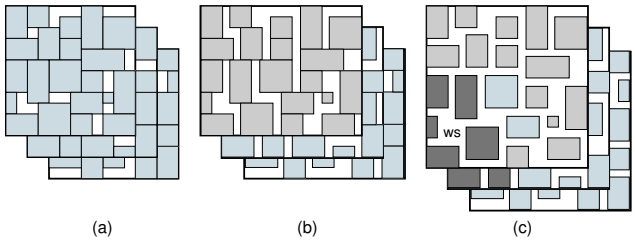

Figure 7: Illustration of 3D decap allocation. (a) 3D placement, (b) X-expansion, (c) XY-expansion, where the darker blocks denote the neighboring blocks of the decap (= white space) inserted. Note that blocks from other layers can utilize the white space for decap insertion.

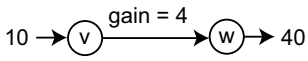

Figure 8: Example of a generalized network flow arc.

Note that there may be iteration between decap allocation and whitespace insertion before sufficient decap is allocated to all blocks. The XY-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. Simple expansion would set α and β equal to each other. In *footprint-aware* expansion, the X and Y slack of each layer are defined as $S_x = Footprint_{width} - Layer_{width}$. Then the equation $\beta/\alpha = S_y/S_x$ is used to make the white space insertion favor the direction with more slack. After each iteration, the α and β are increased until the decap demands are met.

3.5 Flow-based Decap Allocation

In a recent work, Linear Programming (LP) was used to formulate the decap allocation problem [8]. Solving LP is time consuming, which limits its utility for floorplans with lots of blocks. In this work, the decap allocation problem is modelled by *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 Figure 8). For the traditional network flows, the gain factor is one. Capacity constraints 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 [13, 14].

An example flow network for decap allocation is shown in figure 9. The nodes on the right represent the blocks. The capacities of the sink (t) edges are the decap demands 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 source (s) edges are the areas of the whitespaces. 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 whitespaces 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 of the edges are infinite. If a circuit module is close enough to draw decap from a whitespace module, the circuit

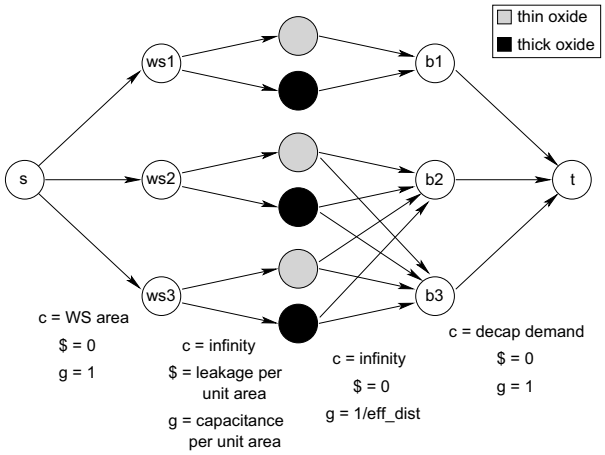

Figure 9: Generalized flow network for decap allocation. b1, b2, and b3 are blocks needing decap. ws1, ws2, and ws3 are whitespace. c = capacity, $ = cost, and g = gain.

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 $1/\gamma_{eff}$ to represent the effectiveness of the whitespace. Maximizing the flow in this generalized flow network allocates the maximum possible decap to blocks. Minimizing the cost in this generalized flow network minimizes the leakage of the decaps.

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 demands of the circuit modules. In this case the floorplan must be expanded for additional whitespace. In the 3D environment, the smaller layers will be expanded first to avoid increasing the footprint area of the entire floorplan. This expansion can also help circuit modules on unexpanded layers since the effective distance formulation allows circuit modules to draw decap from other layers.

Exact generalized min-cost max-flow algorithms are extremely slow, so we used an approximation algorithm [14]. This algorithm runs in $O(\epsilon^{-2} \cdot m^2)$, where ϵ is the error bound percentage from the maximum flow, and m is the number of edges. Since the amount of flow returned by the approximation algorithm can be ϵ percent below the max, it could under-allocate decap. Over-allocation would be preferred to under-allocation, so all of the decap demands are divided by $(1-\epsilon)$. For example, if a module has a decap demand of 100 and ϵ is set to 0.2, then anywhere from 80 to 100 would be allocated, assuming plentiful whitespace. If the decap demand were divided by $(1-\epsilon)$ to get 125 before sending it to the network flow, then the allocation would be between 100 and 125.

4. EXPERIMENTAL RESULTS

The 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 dual processor systems running linux. Nine GSRC benchmarks were used. The blocks were randomly assigned maximum current densities between $10^6 A/m^2$ and $10^7 A/m^2$. The values for capacitance and decap leakage were based on ITRS [15] values for 65nm and 90nm technology nodes for thin and thick oxide decaps, respectively. All floorplans have four placement layers.

Table 1 shows the results of adding footprint awareness and effective distance to the decap allocator. With simple decap allocation, which only allowed blocks to access neighboring whitespace for decap, there was an area expansion of 4.4%. Footprint-aware area expansion was only able to reduce the expansion by half a percent. Effective distance allowed blocks to access whitespace in other layers. Thus, effective distance made more effective use of existing whitespace and reduced area expansion to 2.4%. Combining both techniques reduced the area expansion to 1.7%.

Table 1: Impact of effective decap distance (ED) and footprint-aware (FA) decap insertion schemes on area/wirelength-driven floorplans.

	area	decap	area after decap insertion			
ckt	before	cost	simple	FA	ED	FA+ED
n50	63250	40	64715	63993	63417	63421
n50b	62738	37	63141	62825	62809	62752
n50c	68385	33	68515	68411	68385	68385
n100	56952	134	60227	59054	59039	57494
n100b	58512	163	61945	62958	60182	60285
n100c	61100	141	64947	63985	62892	62069
n200	52948	183	57710	58117	57294	57405
n200b	71022	230	74813	74304	72572	71479
n200c	64416	206	67005	66451	65520	64690
Δ area	-	-	4.4%	3.9%	2.4%	1.7%
time	-	-	1486	2007	1611	1839

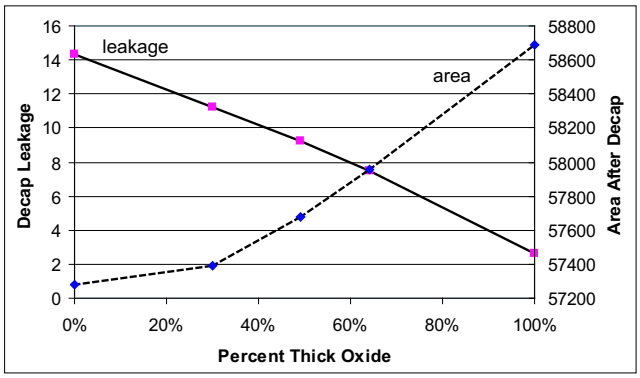

Figure 10: Decap insertion with dual oxide thicknesses for n100. Effective decap distance and footprint aware whitespace insertion are used. The area before decap insertion is 56925.

Figure 10 shows the effect of varying the proportion of thin and thick oxide decaps for the n100 benchmark. Only using thin oxide decaps minimized the area expansion but had high leakage due to the decaps. As more thick oxide decaps are used, the leakage decreases, but the area expansion increases. Using all thick oxide decaps resulted in the greatest area expansion, but had an over five fold decrease in decap leakage compared to only using thin oxide decaps.

Table 2 compares area and wirelength-driven floorplanning to decap-driven floorplanning. The decap-driven floorplanner reduced decap budget by almost 15% while only increasing area and wirelength by 2% and 7% respectively. The lower decap budget of the decap-driven floorplanner has the additional benefit of reducing leakage. Table 3 compares the two styles of floorplanners when dual oxide thicknesses for decaps are used. The availability of dual oxide thicknesses allowed for a great reduction in leakage compared to the single oxide thickness case.

Table 2: Area/wirelength-driven vs decap-driven floorplanning with thin oxide decaps. Effective decap distance and footprint-aware decap insertion schemes are used for both.

	area/wirelength-driven					decap-driven				
ckt	area before	wire length	decap cost	area after	decap leakage	area before	wire length	decap cost	area after	decap leakage
n50	63250	38957	40	63421	4.5	70752	40697	34	71168	3.8
n50b	62738	33974	37	62752	4.4	65835	39032	23	65841	2.5
n50c	68385	38452	33	68385	3.7	68105	41170	22	68169	2.4
n100	56952	83801	134	57494	12.4	63450	78597	123	64470	13.7
n100b	58512	58621	163	60285	16.7	63054	69096	139	63216	15.3
n100c	61100	70743	141	62069	14.5	65000	80710	124	65201	14.2
n200	52948	170813	183	57405	19.3	52948	171101	183	56964	19.2
n200b	71022	160395	230	71479	25.4	56203	181964	216	58021	22.3
n200c	64416	159061	206	64690	23.2	64770	153757	198	65020	22.7
ratio	1.000	1.000	1.000	1.000	1.000	1.024	1.069	0.853	1.022	0.881
time			1839					1586		

Table 3: Area/wirelength-driven vs decap-driven floorplanning with dual oxide thickness decaps. Effective decap distance and footprint-aware decap insertion schemes are used for both.

	area/wirelength-driven					decap-driven				
ckt	area before	wire length	decap cost	area after	decap leakage	area before	wire length	decap cost	area after	decap leakage
n50	63250	38957	40	63421	3.2	64491	44709	27	64696	2.2
n50b	62738	33974	37	62752	4.3	69322	40821	24	69325	2.5
n50c	68385	38452	33	68385	3.5	75594	47206	18	75621	1.9
n100	56952	83801	134	57558	11.3	56925	86050	130	57393	11.1
n100b	58512	58621	163	60960	11.3	72988	75592	140	73039	7.0
n100c	61100	70743	141	62335	11.3	66582	87157	120	67403	9.3
n200	52948	170813	183	57398	11.0	52948	171101	183	56958	11.2
n200b	71022	160395	230	71671	11.1	56203	181964	216	59021	11.4
n200c	64416	159061	206	65116	11.1	55255	169427	199	56474	11.4
ratio	1.000	1.000	1.000	1.000	1.000	1.024	1.148	0.826	1.020	0.811
time			2819					1550		

5. CONCLUSIONS

A new algorithm for decap allocation-based on generalized network flow was presented for 3D ICs. Effective distance was formulated to allow blocks to utilize whitespace in other layers for decap. When effective distance was combined with footprint-aware floorplan expansion, the area overhead caused by decap insertion was greatly reduced. The generalized network flow-based decap allocation algorithm also incorporated dual oxide thickness decaps to reduce leakage. Future work includes investigating the benefits of more than two oxide thicknesses for decaps.

6. REFERENCES

[1] G. Roos, B. Hoefflinger, M. Schubert, and R. Zingg. Manufacturability of 3D-epitaxial-lateral-overgrowth CMOS circuits with three stacked channels. *Microelectron*, 1991.

[2] V. Subramanian, P. Dankoski, L. Degertekin, B. Khuri-Yakub, and K. Saraswat. Controlled two-step solid-phase crystallization for highperformance polysilicon TFTs. *IEEE Electron Device Letter*, 1997.

[3] A. Fan, A. Rahman, and R. Reif. Copper wafer bonding. *Electrochemical Solid-State Letter*, 1999.

[4] 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. on Physical Design*, pages 68–73, 2002.

[5] J. Fu, Z. Lou, X. Hong, Y.Cai, S. X.-D. Tan, and Z.Pan. A fast decoupling capacitor budgeting algorithm for robust on-chip power delivery. In *Proc. Asia and South Pacific Design Automation Conf.*, pages 505–510, Jan. 2004.

[6] Z. Qi, H. Li, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Fast decap allocation algorithm for robust on-chip power delivery. In *Proc. IEEE International Symposium on Quality Electronic Design*, 2005.

[7] H. Li, Z. Qi, S. X.-D. Tan, L. Wu, Y. Cai, and X. Hong. Partitioning-based approach to fast on-chip decap budgeting and minimization. In *Proc. ACM Design Automation Conf.*, Jun. 2005.

[8] S. Zhao, C. Koh, and K. Roy. Decoupling capacitance allocation and its application to power supply noise aware floorplanning. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pages 81–92, 2002.

[9] H. Chen, L. Huang, I. Liu, and M. Wong. Simultaneous power supply planning and noise avoidance in floorplan design. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pages 578–587, 2005.

[10] 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 and South Pacific Design Automation Conf.*, 2005.

[11] 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 Symposium on Integrated Circuits and Systems Design*, 2003.

[12] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani. Rectangle packing based module placement. In *Proc. IEEE Int. Conf. on Computer-Aided Design*, pages 472–479, 1995.

[13] N. Garg and J. Konemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 300–309, 1998.

[14] K. D. Wayne and L. Fleischer. Faster approximation algorithms for generalized flow. In *Proc. ACM/SIAM Symposium on Discrete Algorithms*, 1999.

[15] International technology roadmap for semiconductors. http://public.itrs.net.