Decoupling Capacitor Planning and Sizing For Noise and Leakage Reduction*

Eric Wong, Jacob Minz, and Sung Kyu Lim

School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA 30332
{ewong, jrminz, limsk}@ece.gatech.edu

ABSTRACT

Decoupling capacitor (decap) is a popular means to reduce 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. In this paper, we devise the Effective Decap Distance model to analyze how functional blocks are affected by non-neighboring 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 wirelength when integrated into 2D and 3D floorplanners.

Categories and Subject Descriptors

B.7.2 [**Design Aid**]: Integrated circuits

General Terms

Algorithms, Design, Performance

Keywords

Power supply noise, decoupling capacitors, leakage power reduction, 3D floorplanning

1. INTRODUCTION

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. Decoupling capacitors (decaps) are often inserted to serve as local reservoir of current to meet the sudden current demands. 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 [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 non-neighboring decaps.

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 [2] by performing wire sizing of the power/ground network after decap insertion. Another possible solution for the leakage is to use thicker oxides. However, that reduces capacitance and increases the area required to implement the decaps. Using dual oxide thicknesses for decap fabrication was proposed in [3]. Although dual oxide thickness decaps may increase manufacturing costs due to the additional mask, the benefits include decap leakage reduction and decap area reduction.

The contributions of the paper are as follows: First, 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 non-neighboring decap access when used in 2D floorplanning. Second, 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. Third, having multiple device layers creates the possibility of allowing circuit modules to access decaps on other layers in 3D IC. We show that the effective distance model and our decap allocation/sizing schemes work very effectively for 3D floorplanning.

2. PRELIMINARY

2.1 Problem Formulation

The following are the inputs to the Decoupling Capacitor Planning and Sizing (DCPS) 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 oxide thicknesses available for decap fabrication, (v) the location of the power/ground pins, and (vi) the power supply noise constraint, (vii) decap leakage power constraint. The

*This research has been supported by the grants from MARCO C2S2 and NSF CAREER Award under CCF-0546382.

ICCAD'06, November 5–9, 2006, San Jose, CA, USA.
Copyright 2006 ACM 1-59593-389-1/06/0011 ...$5.00.

goal of DCPS problem is to find (i) the location of the blocks and whitespace, (ii) assignment of whitespace to blocks, (iii) 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 are and wirelength of the floorplan, and D denote the total amount of decoupling capacitance required. 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.

2.2 Overview of the Algorithm

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 2D floorplans. In addition to the area and wirelength objectives, the following steps are performed to measure the decap cost for each candidate 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.

After floorplanning is completed, decaps are inserted based on the decap budget calculated from the final floorplan. 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 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.

3. EFFECTIVE DISTANCE MODELING

3.1 Power Supply Noise Modeling

We use a 2D mesh as in [1] to model our P/G network. 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 [1] 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

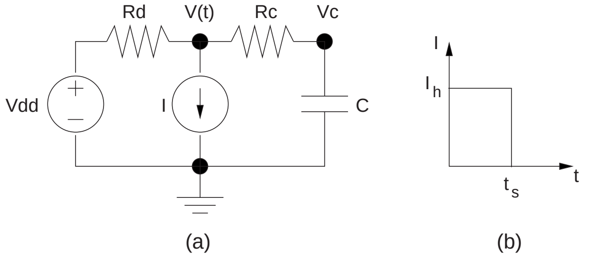

Figure 1: (a) circuit used for effective distance formulation, (b) switching current of the block.

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

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.

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:

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

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. If k denotes the number of blocks, this $m \times n$ mesh-based decap analysis takes $O(kmn)$, 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 SA-based floorplan perturbation. The worst-case complexity still remains at $O(kmn)$, but the runtime can be significantly reduced if the perturbation causes minor change in the floorplan.

3.2 Decap Modeling with Effective Distance

A recent work on decap-aware floorplanning for 2D ICs [1] 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 whitespace for decap allocation. A decap placed far away from a block is less effective at reducing noise. A formal definition is as follows:

DEFINITION 1. *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 1 is 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

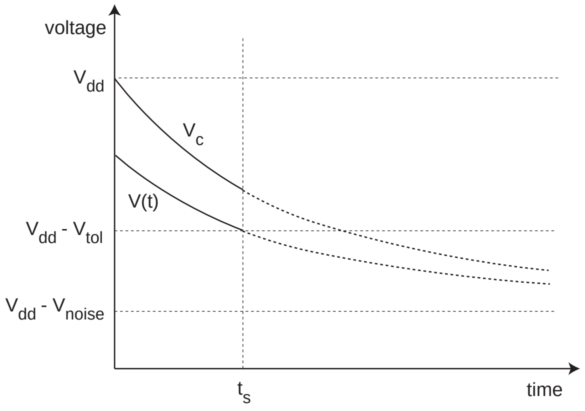

Figure 2: The voltage of the circuit module $V(t)$ and the voltage of the capacitor V_c during switching. V_{dd} is the voltage of the power pin. V_{tol} is the maximum noise the block can handle. V_{noise} is the SSN.

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 2). This equation can be solved for C to find the amount of decap needed by the block.

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

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 3 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.

4. DECAP PLANNING ALGORITHMS

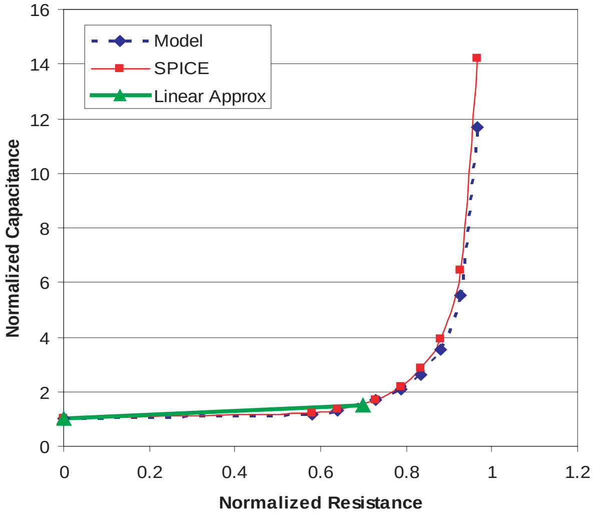

Figure 3: SPICE modeling on decap requirement as a function of resistance R_c, which is normalized with respect to R_{max}. Normalized capacitance is equivalent to γ_{eff}.

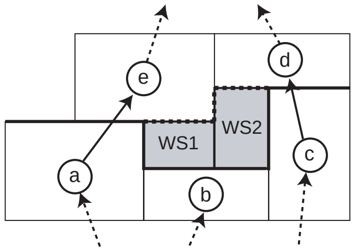

Figure 4: Whitespace detection. Blocks a, b, c are in the lower level. Blocks d, e are in the next level. The bold line is the lower boundary, while the dotted line is the upper boundary. $ws1, ws2$ are the detected whitespace.

4.1 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 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 i from the source node. Each level is ordered by the horizonal constraint graph. The whitespace 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. 4, 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 whitespace $ws1, ws2$. This algorithm is capable of detecting all whitespace, 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 the existing whitespace is not enough to suppress the SSN noise, more whitespace is added by expanding the floorplan in the

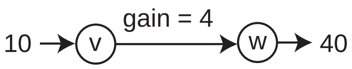

Figure 5: Example of a generalized network flow arc.

X and Y direction.

4.2 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 Figure 5). 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 [5, 6].

An example flow network for decap allocation is shown in Figure 6. The nodes on the right represent the blocks. The capacities of the edges connecting to the sink 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 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 $1/\gamma_{eff}$ 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.

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 [6]. 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. In our experiments, we set ϵ to 0.3.

5. 3D FLOORPLANNING

5.1 Motivation

Three dimensional (3D) integrated circuits are an emerging technology with great potential to improve performance and power. The wafer-bonding approach [7] 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 wirelength, which translates into less wire delay, less power, and greater performance.

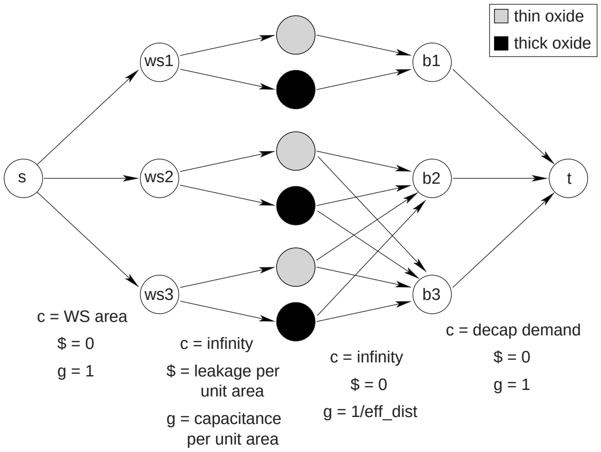

Figure 6: Generalized flow network for decap allocation. b1, b2, and b3 are blocks needing decap. ws1, ws2, and ws3 are whitespace. c = capacity, $ = cost, and g = gain.

The decap allocation problem in a 3D IC has a couple of additional factors not present in the 2D 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 inter-layer, non-neighboring decap access. Second, in case the existing whitespace in a floorplan is insufficient to supply the needed decap, the floorplan needs be expanded to add additional whitespace. In 3D 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.

5.2 Footprint-aware Decap Insertion

We extend the existing 2D Sequence Pair scheme [4] 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 among the blocks in the same layer. However, it is straightforward to determine the inter-layer position relationships of the blocks by computing the block coordinates. We use a 3D mesh to model the P/G network in 3D ICs.

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. 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 whitespace insertion favor the direction with more slack. After each iteration, the α and β are increased until the decap demands are met.

6. EXPERIMENTAL RESULTS

Table 1: Comparison to an existing work.

	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 2: 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	4782	4567	1213	609
n50b	2397	1191	743	259
n50c	5861	4644	1082	43
n100	19549	16527	6272	3363
n100b	27453	27379	5957	4713
n100c	23075	23076	4370	3536
n200	35176	35158	8229	7304
n200b	43431	43383	11114	9648
n200c	40324	40313	8339	7863
ratio	1.000	0.898	1.000	0.636

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 dual processor systems running linux. To verify our floorplanner and noise analyzer, we performed 2D floorplanning on the MCNC benchmarks using the $0.25\mu m$ 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 1 shows the comparison of our 2D 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 lower than [1] because the current densities of the blocks are randomly assigned. Nevertheless, our decap aware floorplanner reduced the decap relative to our area/wirelength driven floorplanner, just as the noise aware floorplanner reduced the decap relative to the post floorplanner in [1].

Due to the small number of blocks in MCNC benchmarks, we used GSRC benchmarks for 3D floorplanning. The blocks were randomly assigned maximum current densities between $10^6 A/m^2$ and $10^7 A/m^2$. The values for wire resistance, inductance, decap capacitance, and decap leakage used for the 3D floorplans were taken from the ITRS for the $90nm$ technology node. The 3D floorplanning results are based on 4-die stacks.

Table 2 shows the impact of effective distance on 2D and 3D floorplans. We obtain floorplans with wire+area objective and insert decaps as a post-process. For both 2D and 3D 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 10.2% for 2D floorplans. The improvement from effective distance is more pronounced for the 3D floorplans at 36.4%. This is due to the fact that in 3D floorplans, effective distance can allow blocks to utilize decaps in other layers.

Table 3 compares area and wirelength-driven floorplanning to decap-driven floorplanning for 2D and 3D implementation. In both the 2D and 3D cases the decap-driven floorplanner was able to reduce the decap at the expense of area and wirelength. The reduction in decap for the 3D floorplans is greater than the reduction for 2D floorplans. This is due to the larger solution space for 3D floorplans. In several cases, the decap driven floorplanner was faster than the area/wirelength driven floorplanner. Even though the decap driven floorplanner must perform noise analysis during annealing, its reduced decap budget can reduce the number of iterations between the generalized min-cost network flow based decap allocation and area expansion.

Table 4 show the impact of dual oxide thickness decaps for 2D and 3D 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 below 8A. The flow-based decap allocator minimizes the area expansion by using as many thin oxide decaps as possible without violating the leakage constraint. For many of the 3D circuits the decap allocator chose to continue using all thin oxide decaps since the starting leakage was already low enough. On the other hand, the decap allocator assigned some thick oxide decaps to the smaller 2D 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.

7. CONCLUSIONS

We presented the effective distance model to analyze how functional blocks are affected by non-neighboring 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 wirelength when integrated into 2D and 3D floorplanner. Future work includes adapting whitespace redistribution techniques to further reduce the area expansion required for decap insertion.

8. REFERENCES

[1] 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*, 21(1):81–92, 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 and South Pacific Design Automation Conf.*, 2005.

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

[4] 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.

[5] 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.

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

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

Table 3: Area/wirelength-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	run time	area before	wire length	decap cost	area after	decap leakage	run time
n50	219061	100058	81.3	223628	4.5	217	221195	98895	81.3	227417	4.1	208
n50b	223258	86165	76.3	224449	4.3	163	222642	84272	77.2	230380	3.8	198
n50c	235209	102424	72.1	239853	3.7	184	247835	100572	69.0	253117	3.6	216
n100	210308	196311	181.0	226835	9.4	1063	219235	212555	182.7	238177	8.5	966
n100b	187060	172316	209.8	214439	10.6	863	187856	173449	206.7	212638	10.3	933
n100c	195390	220524	183.5	218466	9.1	853	195390	220841	183.4	215453	8.9	922
n200	191296	408035	234.3	226454	11.8	3829	191296	408035	234.3	226454	11.8	3918
n200b	194636	447235	267.3	238019	13.5	3261	194636	447235	267.3	238019	13.5	3844
n200c	189385	397950	249.1	229698	12.7	3395	189385	397994	249.1	228514	12.6	4125
ratio	1.000	1.000	1.000	1.000	1.000	1.000	1.012	1.004	0.996	1.013	0.961	1.092

3D floorplanning

	area/wirelength-driven						decap-driven					
ckt	area before	wire length	decap cost	area after	decap leakage	run time	area before	wire length	decap cost	area after	decap leakage	run time
n50	59961	47562	36.7	60570	1.9	160	61503	47040	26.5	61588	1.7	141
n50b	59675	44969	31.9	59934	1.9	131	61344	45154	22.4	61956	1.3	140
n50c	62418	53238	28.4	62461	1.8	126	69841	53743	21.4	69920	1.4	138
n100	56952	84942	133.9	60315	6.4	830	56700	85647	132.9	59476	6.3	909
n100b	50358	71050	155.3	55071	7.9	773	50358	71050	155.3	55071	7.9	897
n100c	52416	82710	133.7	55952	6.9	786	52416	82818	133.5	55805	6.9	880
n200	52948	171582	183.6	60252	9.5	3560	52948	171582	183.6	60252	9.5	3999
n200b	53599	182486	216.3	63247	11.3	3058	53599	183360	212.2	62439	11.0	3698
n200c	52675	169359	196.1	60538	10.1	3645	52675	170136	194.5	59488	10.0	4414
ratio	1.000	1.000	1.000	1.000	1.000	1.000	1.019	1.002	0.905	1.014	0.917	1.106

Table 4: 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	221195	98895	81.3	227417	4.1	208	227416	4.1	208	100.0%	0.0%
n50b	222642	84272	77.2	230380	3.8	198	230381	3.7	228	93.3%	6.7%
n50c	247835	100572	69.0	253117	3.6	216	253117	3.5	215	92.8%	7.2%
n100	219235	212555	182.7	238177	8.5	966	238667	7.8	1382	89.1%	10.9%
n100b	187856	173449	206.7	212628	10.3	933	212688	7.8	944	69.5%	30.5%
n100c	195390	220841	183.4	215453	8.9	922	215455	7.8	1028	85.4%	14.6%
n200	191296	408035	234.3	226454	11.8	3918	226449	7.8	4152	52.5%	47.5%
n200b	194636	447235	267.3	238019	13.5	3844	238239	7.9	3786	43.1%	56.9%
n200c	189385	397994	249.1	228514	12.6	4125	228515	7.9	3899	47.7%	52.3%
ratio	-	-	-	1.000	1.000	1.000	1.001	0.817	1.077	-	-

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	61503	47040	26.5	61588	1.7	208	61588	1.7	139	100.0%	0.0%
n50b	61344	45154	22.4	61956	1.3	198	61956	1.3	140	100.0%	0.0%
n50c	69841	53743	21.4	69920	1.4	216	69920	1.4	137	100.0%	0.0%
n100	56700	85647	132.9	59476	6.3	966	59474	6.3	911	100.0%	0.0%
n100b	50358	71050	155.3	55071	7.9	933	55082	7.7	902	99.7%	0.3%
n100c	52416	82818	133.5	55805	6.9	922	55813	6.9	884	100.0%	0.0%
n200	52948	171582	183.6	60252	9.5	3918	60257	7.7	4470	70.3%	29.7%
n200b	53599	183360	212.2	62439	11.0	3844	62422	7.8	3780	56.4%	43.6%
n200c	52675	170136	194.5	59488	10.0	4125	59481	7.8	4651	70.1%	29.9%
ratio	-	-	-	1.000	1.000	1.000	1.001	0.920	1.020	-	-