Power Supply Noise-aware 3D Floorplanning for System-On-Package

Eric Wong, Jacob Minz, and Sung Kyu Lim
School of Electrical and Computer Engineering
Georgia Institute of Technology
{ewong, jrminz, limsk}@ece.gatech.edu

Abstract

We develop the first power supply noise-aware floorplanning for 3D System-On-Package. Our 3D floorplanner aims at reducing the amount of decoupling capacitance (decap) needed to suppress the simultaneous switching noise. We perform footprint-aware decap insertion to minimize the area overhead of the 3D structure. Our effective decap distance concept allows functional blocks to access decaps in other layers.

I. Introduction

A major problem with 3D System-On-Package (SOP) integration is the power supply noise coupling between the various dissimilar blocks constituting the system. The noise is primarily generated by the high-speed digital processor and coupled through the power distribution network. The noise coupling occurs when the maximum power supply noise is transmitted through interconnections and vias at resonance in the power/ground planes of the package. Due to the low noise floor required for analog circuits, at frequencies below the substrate resonance frequencies, considerable power supply noise occurs in the form of simultaneous switching noise through the common inductive impedance of the power/ground return current path. Moreover, the amount of decoupling capacitors (decap) needed to alleviate the problem is often prohibitive and occupies a significant portion of the packaging area. Post-floorplanning or post-route 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. We develop the first power supply noise-aware floorplanning for 3D System-On-Package. Our floorplanner aims at reducing the amount of decoupling capacitance (decap) to suppress SSN in a 3D SOP design without compromising traditional design metrics such as area and wirelength.

II. SOP Placement Algorithm

A. Overview of the Algorithm

Simulated Annealing is a very popular approach for floorplanning due to its high quality solutions and flexibility in handling various constraints. We extend the existing 2D Sequence Pair scheme [1] to represent our 3D floorplanning solutions. Simulated Annealing procedure starts with an initial multi-layer placement along with its cost in terms of area, wirelength, and decap. In our perturbation scheme, we swap a random pair of blocks from the the same or two 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 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 based on its SSN is computed so that the overall SSN constraint is satisfied. (3) white space detection and insertion: we first detect and allocate existing white space (decap) to the blocks. In case the existing white space is not enough to suppress the SSN, more white space is added by expanding the floorplan in the X and Y direction for decap implementation. (4) decap allocation: white spaces (decaps) are allocated to the blocks that need them so that the utilization of white space is maximized.

The decap cost of a given 3D floorplan is the area increase due to white space (decap) insertion to suppress the SSN noise. Due to the runtime overhead involved, we use the first two steps to estimate the decap budget during the high temperature region. During the low temperature region, however, we perform all four steps to accurately compute the decap cost (area increase). 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. We examine a pre-determined number of candidate solutions at each temperature. The temperature is decreased exponentially, and the annealing process terminates when the freezing temperature is reached.

0-7803-9220-5/05/$20.00©2005 IEEE.

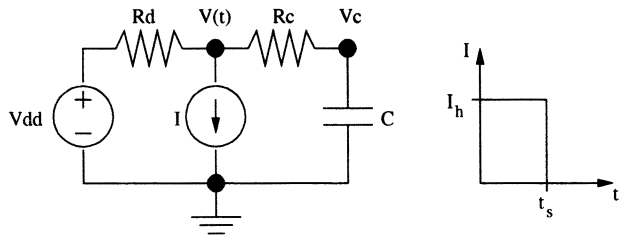

Fig. 1. The circuit used for effective distance formulation. V_{dd} is the power pin. I is the current demand of the block. C is the decap. R_d is the resistance between the block and power pin. R_c is the resistance between the block and decap. The graph shows the assumed switching current.

B. 3D Power Supply Noise Modelling

We use a 3D grid to model the power/ground (P/G) network for 3D SOP. 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 [2] 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.

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 $C^k = Q^k / V_{tol}$, $1 \leq k \leq M$, where 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.

C. Decap Modelling with Effective Distance

We introduce the concept of *Effective distance* 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, $\gamma_{eff}(d_c)$, is the amount of decap needed at distance d_c to get the same noise reduction as a unit decap adjacent to the block. This section formulates *effective distance* based on the circuit model shown in Figure. 1.

We analyzed the circuit shown in Figure 1 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, I represents the current demand of the block.The power pin and the decap are located at distances d_d and d_c from the block. R_d and R_c represent the resistances of the block to the power pin and the decap, which depend linearly 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:

$$V(t) = V_{dd} - V_{noise} + V_{noise} \frac{R_d}{R_c + R_d} \cdot e^{\frac{-t}{(R_c+R_d)C}} \tag{1}$$

where $V_{noise} = R_d I_h$. This equation can be solved for C to find the amount of decap needed by the block.

$$C = \frac{-t_s}{(R_c + R_d)[\ln \frac{(V_{noise} - V_{tol})}{V_{noise}} + \ln \frac{R_c + R_d}{R_d}]} \tag{2}$$

where V_{tol} is the noise tolerance of the block. This equation only holds when $V_{noise} > V_{tol}$ and $R_c < \frac{R_d V_{tol}}{V_{noise} - V_{tol}}$. 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 the block and its decap. Effective distance $\gamma_{eff}(d_c)$ can be defined as:

$$\gamma_{eff}(d_c) = \frac{C(d_c)}{C(0)} = \frac{d_d \cdot \ln \frac{V_{noise} - V_{tol}}{V_{noise}}}{(d_c + d_d)[\ln \frac{V_{noise} - V_{tol}}{V_{noise}} + \ln \frac{d_c + d_d}{d_d}]} \tag{3}$$

To find the actual decap allocated to a block, the base decap budget C^k is calculated and multiplied by $\gamma_{eff}(d_c)$.

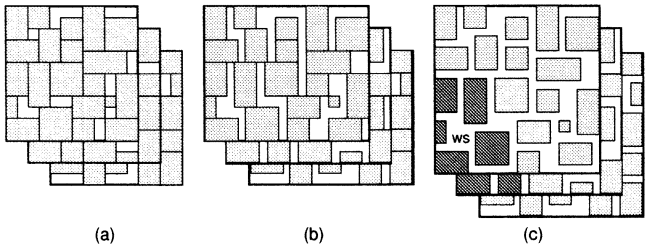

Fig. 2. Illustration of 3D decap allocation. (a) 3D placement, (b) X-expansion, (c) XY-expansion, where the darker blocks denote the neighboring blocks of the decap (= white space) inserted. Note that blocks from other layers can utilize the white space for decap insertion.

D. Whitespace Detection and Insertion

The white space present in the 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 the floorplan is highly desirable. This is done by using the longest path tree calculation based on the vertical constraint graph. All nodes at the n^{th} level in the tree are at an edge distance of n from the source node. Each level is ordered by the horizonal constraints. The white spaces at level n are detected by comparing the *upper* boundary of blocks at level n and the *lower* boundary of the blocks at level $n+1$. If the boundaries are not incident on each other, then there is whitespace. 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 2. 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. *Footprint aware* insertion 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 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*.

Note that we may have to iterate between white space insertion and allocation if the current expansion does not satisfy all the decap demands. 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.

E. Flow-based Decap Allocation

In this work, the decap allocation problem is modelled by *generalized network flow*. Generalized network flow problems generalize traditional network flow problems by adding a *gain factor* $\gamma(e) > 0$ for each arc e. For each unit of flow that enters the arc, $\gamma(e)$ units must exit. For traditional network flows, the gain factor is one. Capacity constraints and node conservation constraints are satisfied by the generalized networks, as in traditional network flows. This model accurately captures the decap allocation problem with *effective distance*, formulated in this paper. Generalized network flow is a well studied problem, but elegant exact and approximate algorithms have only been proposed recently [3].

The nodes on the left represent the blocks. The capacities of the incoming edges are the decap demands of the blocks. The costs of these edges are zero and the gains are unity. The nodes on the right represent the whitespace modules. The capacities of the outgoing edges are the areas of the whitespace modules. The gains are unity, and the costs are set to one. If a circuit module is close enough to draw decap from a whitespace module, they are connected with an edge of infinite capacity, zero cost, and gain factor γ_{eff} to represent the effectiveness of the whitespace, based on distance. The gain factor of the edge between a block and a white space is the amount of area needed to satisfy unit decap. A min-cost maximum flow in this generalized network, allocates the maximum possible decap and uses the minimum white-space area.

If the flow in the source edges are saturated, then the decap demands of all the circuit modules can be met. Assigning cost to the sink edges minimizes the use of the whitespace. If the flow in some source 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 to add additional whitespace. In the 3D environment, the smaller layers will be expanded first to avoid increasing the footprint area of the entire package. This expansion can also help circuit modules on other layers since the effective distance formulation allows circuit modules to draw decap from other layers.

TABLE I

AREA/WIRELENGTH-DRIVEN VS DECAP-DRIVEN 3D FLOORPLANNING. EFFECTIVE DECAP DISTANCE AND FOOTPRINT-AWARE DECAP INSERTION SCHEMES ARE USED FOR BOTH.

	area/wirelength-driven FA ED				decap-driven FA ED			
ckt	wire length	decap	area before	area after	wire length	decap	area before	area after
n50	21541	17	22185	22185	25784	2	25258	25258
n50b	21065	11	23944	23944	22266	3	23828	23828
n50c	18505	10	18340	18340	21449	1	18720	18720
n100	54264	80	35148	35148	65470	52	36860	36860
n100b	40848	86	33990	34302	57145	53	33998	33998
n100c	53240	83	33286	33286	66365	50	36966	36966
n200	155370	235	67599	67751	173997	219	52948	54211
n200b	166159	244	68021	68636	182016	231	53352	54632
n200c	152917	242	63612	63917	171901	233	52675	52974
ratio	1.000	1.000	1.000	1.000	1.178	0.574	0.968	0.970
time	677				1894			

TABLE II

IMPACT OF THE NUMBER OF PLACEMENT LAYERS. EFFECTIVE DISTANCE AND FOOTPRINT AWARENESS ARE USED.

	single layer				2 layers				4 layers			
ckt	wire length	decap	area before	area after	wire length	decap	area before	area after	wire length	decap	area before	area after
n50	50764	66	83790	84959	36806	38	50052	50104	25784	2	25258	25258
n50b	45422	58	80032	81097	35662	27	45346	45346	22266	4	23828	23828
n50c	43081	61	66515	67089	35287	20	39780	39780	21449	1	18720	18720
n100	126017	137	128904	131217	92714	122	59787	60877	65470	52	36860	36860
n100b	106440	142	101985	106269	68810	116	60977	61080	57145	53	33998	33998
n100c	117112	145	130800	133164	88819	126	60977	62120	66365	50	36966	36966
n200	407024	279	198628	205829	249744	264	98820	102958	173997	219	52948	54211
n200b	354319	284	249066	256141	269754	271	102718	105962	182016	231	53352	54632
n200c	398234	288	190437	197270	245559	275	98245	101436	171901	233	52675	52974
ratio	1.000	1.000	1.000	1.000	0.718	0.757	0.524	0.519	0.499	0.401	0.282	0.277
time	1424				3222				1895			

III. EXPERIMENTAL RESULTS

The proposed algorithms were implemented using C++/STL. GSRC benchmarks were run on the program. Table I compares area/wirelength driven floorplanning and decap driven floorplanning. For the large (200 block) circuits, decap driven floorplans have better area than the area/wirelength driven floorplans. However, this improvement comes at the expense of wirelength. Having a 3D structure has many benefits over 2D. The following observations can be made from Table II. The wirelength *decreases* by 28% when going from a single to double layered floorplan, and *decreases* by 50% for a floorplan with four layers. The decap amount *decreases* by 24% and 60% for double and quadruple layered floorplans, respectively. Original area *decreases* by 48% and 72% when increasing layers to two and four. The reduction in expanded area after decap allocation is slightly greater. This suggests that 3D structures offer greater flexibility in decap allocation. Decap decreases because the compact 3D structure allows for shorter paths from blocks to power pins. For the 2D floorplans, there is a much larger area expansion for decap allocation since footprint awareness is unavailable.

IV. CONCLUSIONS

Our 3D System-On-Package floorplanner aims at reducing the amount of decoupling capacitance (decap) needed to suppress the simultaneous switching noise without compromising traditional design metrics such as area and wirelength. We performed footprint-aware decap insertion to allow functional blocks to access decaps in other layers. We introduced the concept of effective decap distance to facilitate the non-neighboring decap access.

REFERENCES

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

[2] 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*, pp. 81–92, 2002.

[3] K. Wayne and L. Fleischer, "Faster approximation algorithms for generalized flow," in *Proc. 10th Annual ACM/SIAM Symposium on Discrete Mathematics*, 1999.