Fast Substrate Noise Aware Floorplanning for Mixed Signal SOC Designs

Minsik Cho and David Z. Pan

***Abstract*—In this paper, we introduce a novel substrate noise estimation technique during early floorplanning for mixed signal system-on-chip (SOC), based on block preference directed graph (BPDG). Given a set of analog and digital blocks, BPDG is constructed based on their inherent noise characteristics to capture the preferred relative locations for substrate noise minimization. For each instance of floorplan in sequence pair or B^*-tree, we efficiently count the number of violations against BPDG which correlates remarkably well with accurate but computation-intensive substrate noise modeling. Thus, our BPDG-based model can guide fast substrate noise-aware floorplanning and layout optimization for mixed signal SOC. Our experimental results show that the proposed approach is significantly faster than conventional full-blown substrate model-based floorplanning.**

***Index Terms*—B^*-Tree, floorplanning, mixed signal system-on-chip (SOC), physical synthesis, preference directed graph, sequence pair, substrate noise.**

I. Introduction

Increasing demand for wireless and telecommunication application is driving tighter integration of multiple heterogeneous components [e.g., front end RF circuit, mixed signal circuits, and high speed digital positioning system (DPS)] into a single system-on-chip (SOC). As such components can degrade the performance or cause failure by interfering with each other, it has to be optimized during layout planning [1]. A major interference is the substrate noise caused by large amount of switching activities in high speed digital cores to analog/RF components, degrading the reliability and performance of these sensitive analog/mixed signal/RF Intellectual Properties (IPs) [2]. Such substrate noise is becoming a growing concern due to higher clock frequency, more accurate analog precision, deeper technology scaling, and tighter integration of analog blocks with digital blocks [3]–[5].

Although abundant amount of works have been done in modeling and simulation of substrate noise [2], [3], [5]–[11], none of them are suitable to guide substrate noise optimization in floorplanning due to high computational expense or limited application. Therefore, there is not much in the literature on substrate noise optimization in early floorplanning stage. Mitra *et al.* [12] presented a substrate-aware mixed signal macrocell placement with electrothermal-like substrate model [13]. Lin *et al.* [14] incorporated substrate noise minimization into placement based on a semi-empirical model [11]. Kao *et al.* [15] presented a constraint driven placement to address substrate noise in mixed signal designs. The substrate noise estimation techniques in these works, however, either suffer from low accuracy or high complexity. Blakiewicz *et al.* [16] proposed a floorplanning algorithm with a more scalable substrate noise model, but still it requires significant computational overhead to evaluate the substrate noise as a floorplanning cost.

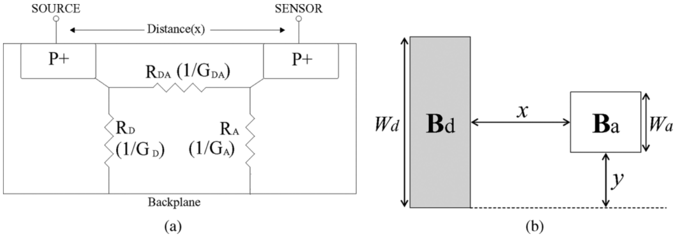

Fig. 1. Compact substrate noise model in [5]. (a) Macromodel for the substrate based two-port lumped resistor network; (b) two different size blocks with separation x and relative position y.

In this paper, we propose a novel concept of *block preference directed graph* (BPDG) to overcome the modeling bottleneck for substrate noise-aware floorplanning. Using the proposed theorems to compare floorplan instance in sequence pair or B^*-tree against BPDG, BPDG-based substrate model shows high fidelity to accurate but much more expensive substrate noise modeling [5]. Thus, it can efficiently guide substrate noise-aware floorplanning for mixed signal SOC.

II. Preliminaries

A. Sequence Pair, B^*-Tree, and Block Alignment

Sequence pair [17] specifies the geometric relations between each pair of blocks using two sequences. For example, (..A..B.., ..A..B..) means that a block A is to the left of a block B, and (..B..A.., ..A..B..) implies that A is below B. A sequence pair can be translated into a floorplan by horizontal and vertical constraint graph [17]. Conditions for block alignments in sequence pair are studied in [18] and [19].

B^*-tree [20] is an ordered binary tree for floorplan representation. Given an admissible floorplan, B^*-tree keeps the relationship between two blocks B_p and B_c, by setting B_c as either the left child or the right child of B_p, depending on the geometric locations. A skewed B^*-tree is proposed in [21] to satisfy block alignment conditions.

B. Substrate Noise Model

The substrate noise model in [5] is known to be highly scalable and accurate, so that it can guide floorplanning [22]. However, such compact model is still expensive in the design optimization inner loop during floorplanning. Consider a two-port lumped resistor network, modeling substrate as illustrated in Fig. 1(a). The resistance R_{DA} models the coupling between two blocks, and R_A and R_D model the coupling from the blocks to the backplane. Then, the following can be derived from the macromodel:

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} = \frac{1}{\triangle} \begin{bmatrix} G_D + G_{DA} & G_{DA} \\ G_{DA} & G_A + G_{DA} \end{bmatrix} \quad (1)$$

where $\triangle = G_A G_{DA} + G_D G_{DA} + G_A G_D$ and each Z_{ij} is as in [5] and [10]. For example, Z_{11} and Z_{12} are as follows:

$$Z_{11} = (K_1 \cdot A + K_2 \cdot P + K_3)^{-1} \quad (2)$$

$$Z_{12} = \alpha e^{-\beta x} = (ay^2 + by + 1) Z_0 \cdot e^{-\beta x} \quad (3)$$

where A and P are the area and perimeter of the block, respectively. K_1, K_2, K_3, and β are process dependent parameters and constant for given process. As shown in Fig. 1(b), y represents the vertical relative position of two blocks, and a and b are the coefficient of symmetry and relative position of a merged block. Z_0 is equal to the value of Z_{11} of a single merged block ($x = y = 0$).

The coupling gain of the substrate can be calculated from the value of resistors in the two-port lumped network shown in Fig. 1(a). The

Manuscript received October 21, 2006; revised May 30, 2007 and September 13, 2007. Current version published November 19, 2008. This work is supported in part by the SRC under Contract 2005-TJ-1321, by the IBM Faculty Award, and the equipment donations from Intel.

The authors are with the Electrical and Computer Engineering Department, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: thyeros@cerc.utexas.edu; dpan@ece.utexas.edu).

Digital Object Identifier 10.1109/TVLSI.2008.2001734

TABLE I
SUBSTRATE NOISE TABLE

	D_1	D_2	D_3	D_4	D_5	D_6
A_1	5	2	6	3	10	1
A_2	2	1	3	10	8	5
A_3	3	8	7	11	9	12

D_1: $A_1 \leftarrow A_3 \leftarrow A_2$ D_2: $A_3 \leftarrow A_1 \leftarrow A_2$

D_3: $A_3 \leftarrow A_1 \leftarrow A_2$ D_2: $A_3 \leftarrow A_2 \leftarrow A_1$

D_5: $A_1 \leftarrow A_3 \leftarrow A_2$ D_6: $A_3 \leftarrow A_2 \leftarrow A_1$

(a)

A_1: $D_6 \leftarrow D_2 \leftarrow D_4 \leftarrow D_1 \leftarrow D_3 \leftarrow D_5$

A_2: $D_2 \leftarrow D_1 \leftarrow D_3 \leftarrow D_6 \leftarrow D_5 \leftarrow D_4$

A_3: $D_1 \leftarrow D_3 \leftarrow D_2 \leftarrow D_5 \leftarrow D_4 \leftarrow D_6$

(b)

Fig. 2. Block orderings of Table I. (a) Analog block orderings. (b) Digital block orderings.

coupling gain of ith digital block to jth analog block under a few gigahertz [4], $CG_{i,j}$ can be given as

$$CG_{i,j} = \frac{R_A}{R_A + R_{DA}} = \frac{G_{DA}}{G_{DA} + G_A} = \frac{Z_{12}}{Z_{22}}. \tag{4}$$

Finally, the total noise from all digital blocks is

$$N_{\text{total}} = \sum_{i,j} N_{i,j} = \sum_{i,j} CG_{i,j} \cdot \sqrt{\int_0^\infty S_i(f) \cdot |H_j(f)|^2 \, df} \tag{5}$$

where $S_i(f)$ and $H_j(f)$ are power spectral density (PSD) of noise source and transfer function of noise sensor, respectively.

III. BPDG

The substrate noise model in Section II-B is highly compact, scalable, and accurate. However, it is still expensive to compute substrate noise during simulated annealing-based floorplanning, because every noise estimation after a movement requires the accurate location of every block (substrate noise is exponentially sensitive to geometric distance [5], [10]), whereas area and wirelength can be calculated approximately. For fast substrate noise estimation, we introduce and describe a new concept of BPDG in this section. BPDG represents preferred relative locations of blocks to guide substrate noise-aware floorplanning. BPDG construction consists of two steps as in the following subsections.

A. Block Ordering

Since substrate noise is heavily related to the distance between blocks, the nominal distance is assumed to normalize the effect of distance. With such fixed distance, the substrate noise purely depends on frequency coupling and geometric properties like area and perimeter [5], [10]. Under such condition, a substrate noise on an analog block A_j due to a digital block D_i, $N_{i,j}$ can be computed from (5). Table I shows an example of substrate noise table between digital blocks $(D_1 - D_6)$ and analog blocks $(A_1 - A_3)$.

Based on the substrate noise table, analog blocks can be sorted for each digital block by the *descending* order of substrate noise. Consider the example in Table I. Analog block A_1, A_3, and A_2 can be ordered by the substrate noise from D_1, as $N_{1,1} > N_{1,3} > N_{1,2}$ $(5 > 3 > 2)$. The other orderings can be obtained in the same manner as in Fig. 2(a). These orderings push more noise-sensitive analog blocks to the head of block orderings. In a similar way, digital blocks can be sorted for each analog block by the *ascending* order of substrate noise. All digital block orderings are shown in Fig. 2(b). This pushes less aggressive digital blocks to the head of block orderings.

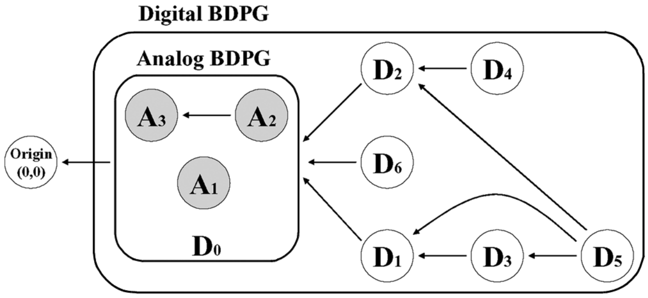

Fig. 3. BPDG built from Table I.

B. BPDG Construction

The key idea behind BPDG is to make less aggressive digital blocks and less sensitive analog blocks interfaced by finding common block orders in order to minimize the substrate noise. Algorithm 1 shows how to construct an analog BPDG and a digital BPDG with analog and digital block orderings. The reason to create a virtual vertex in Algorithm 1 is to force analog blocks isolated from digital blocks, which is common in real mixed signal design. Consider the final BPDG in Fig. 3. Since A_3 is before A_2 for all analog block orderings in Fig. 2(a), vertices A_3 and A_2 are inserted into G_a (analog BPDG), and connected with a directed edge. Again, vertices D_1 and D_3 are inserted into G_d (digital BPDG), as D_1 is before D_3 for all digital block orderings in Fig. 2(b).

Algorithm 1 BPDG Construction

```
Input Analog, Digital block orderings $O_a$ and $O_d$
Analog BPDG $G_a \leftarrow \phi$, Digital BPDG $G_d \leftarrow \phi$
for each common order between $A_i$ and $A_j$ in all $O_a$ do
    Add a directed edge from $A_j$ to $A_i$ to $G_a$
end for
for each common order between $D_i$ and $D_j$ in all $O_d$ do
    Add a directed edge from $D_j$ to $D_i$ to $G_d$
end for
Add a virtual vertex $D_0$ for $G_a$ to $G_d$, and insert directed edges
    from all vertices without successors to $D_0$
Output $G_d$
```

IV. SUBSTRATE NOISE ESTIMATION WITH BPDG

In this section, we propose theorems to efficiently compare a floorplan instance in sequence pair or B^*-tree against a BPDG. Our theorems count the number of violations in the instance against the preferences in BPDG. Intuitively, more violations indicate more noise, as an edge from a block B_a to a block B_b in the BPDG means that B_b must be closer to the origin (left-bottom corner) than B_a for substrate noise minimization.

A. Sequence Pair With BPDG

In [18], the concept of *strictly ahead* is defined for block alignment in a floorplanning with sequence pair. When there is no block between B_a and B_b in a floorplan, B_a is strictly ahead of B_b. Fig. 4(a) shows a floorplan where B_a is strictly ahead of B_1, B_2, B_3 and B_4. In fact, *strictly ahead* is a necessary condition for two blocks to be abutted. We extend *strictly ahead* definition for easier explanation of this section.

Definition α: Given a block B_a and a sequence pair, all the blocks which are both strictly ahead of B_a and below (to the left) B_a form a **strictly below set** (**strictly left set**) of B_a.

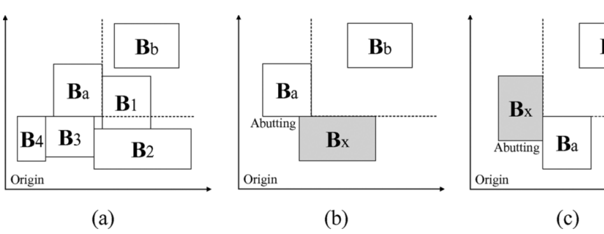

Fig. 4. Floorplan examples.

Definition β: Given a block B_a and a sequence pair, any block in a strictly below or left set of B_a and abutting to B_a is a **reference block** of B_a.

In Fig. 4(a), B_2, B_3, and B_4 are in the strictly below set of B_a, because they are strictly ahead of B_a as well as below B_a, and B_3 is a reference block of B_a. One intuitive property of the reference block is stated in Theorem 1 referring to [18].

Theorem 1: If a block B_a has a non-empty strictly below/left set S, a reference block B_x must exist in S under a completely packed floorplan.

Proof: For any floorplan, it can be always converted into the completely packed floorplan by shifting the blocks toward left/bottom direction. For a completely packed floorplan, any block cannot be moved, as it is abutted and blocked by another block B_x which is a reference block of B_a. ■

Based on Theorem 1, the relative locations of two blocks can be determined. Consider Fig. 4(b), where B_a is to the left of B_b and B_x is a reference block of B_a. It can be proved that if a block such as B_x exists below B_b, it is guaranteed that B_a has a shorter distance to the origin (0,0) than B_b. This key idea to compare the relative locations of two blocks with a sequence pair is in Theorem 2 by extending Theorems in [18].

Theorem 2: Let S_b be a strictly below set, and S_l a strictly left set of B_a, respectively. A block B_a is guaranteed to be closer to the left bottom corner than a block B_b under a completely packed floorplan, if either of the following conditions is satisfied:

1) for any block B_s in S_b, if a sequence pair $(P, N) = (..B_a X_1 B_b X_2 B_s.., ..B_s Y_1 B_a..B_b..)$;
2) for any block B_s in S_l, if a sequence pair $(P, N) = (..B_s X_3 B_b X_4 B_a.., ..B_s Y_2 B_a..B_b..)$.

Proof: For case 1), B_b is to the right of B_a by (P, N). Also, B_b is above some reference B_x in S_b, because B_b is above some block in S_b as in Theorem 1. Thus, B_a is closer to the left bottom corner than B_b as in Fig. 4(b). Case 2) can be proved similarly with Fig. 4(c). ■

Thus, when a sequence pair (P, N) and a BPDG G are given, the preferred relative block location (an edge) in G can be examined with Theorem 2 to see if such preference is held in (P, N). Theorem 2 can be further simplified into Theorem 3 with the longest common string (LCS) search for speedup.

Theorem 3: A block B_a is guaranteed to have shorter distance to the left-bottom corner than a block B_b under a completely packed floorplan, if either of the following conditions is satisfied:

1) there is no block B_s satisfying $\mathrm{LCS}(X_1, Y_1) = \phi$ in a sequence pair $(P, N) = (..B_a X_1 B_s..B_b.., ..B_s Y_1 B_a..B_b..)$;
2) there is no block B_s satisfying $\mathrm{LCS}(X_2, Y_2) = \phi$ in a sequence pair $(P, N) = (..B_b..B_s X_2 B_a.., ..B_s Y_2 B_a..B_b..)$.

Proof: Consider a strictly below set S_b of B_a and a reference block of B_a, then B_x must be in S_b by Theorem 1. For case 1), if there exists a block in S_b (possibly the reference block B_x) between B_a and B_b in P, it automatically violates Theorem 2. If there does not exist any block of S_b between B_a and B_b in P, all blocks in S_b exist after B_b. Accordingly, B_b is to the right of B_a in (P, N) and above the reference block B_x included in S_b. Case 2) can be proved in the same manner. ■

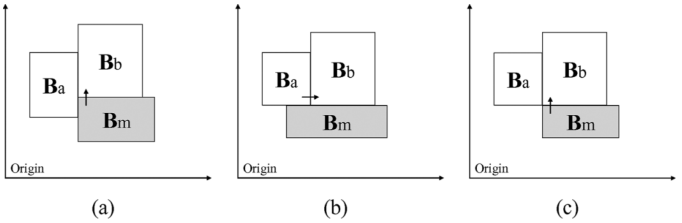

Fig. 5. Example of parent-children relationships by different block locations.

Therefore, we can apply Theorem 3 to efficiently compare geometric distances from the origin to any two blocks in a sequence pair conservatively *without other geometric information*. Note that in a real implementation, bitwise-OR can be used instead of LCS computation.

B. B^*-Tree With BPDG

In B^*-tree, the geometric relationship of blocks is stored in binary tree. As a block has information only on two child blocks and its parent block in B^*-tree, the following two properties can be identified and further extended to Theorem 4.

Property α: A parent block B_p is guaranteed to have shorter distance to the left-bottom corner than the right child block B_r, but **not always** than the left child block B_l.

Property β: If a parent block B_p is the root, B_p is guaranteed to have shorter distance to the left-bottom corner than the left child block B_l and the right child block B_r.

Theorem 4: A block B_a is guaranteed to have shorter distance to the left-bottom corner than a block B_b under a completely packed floorplan, if a block B_m is the left child of B_a, and B_b is the right child of B_m.

Proof: Let the coordinate, width and height of a block B_* be (x_*, y_*, w_*, h_*). As B_m is the left child of B_a, $x_m = x_a + w_a$ and $y_m + h_m > y_a$. Also, as B_b is the right child of B_m, $x_b = x_m$ and $y_b = y_m + h_m$. Thus, $x_b > x_a$ and $y_b > y_a$. This case is described in Fig. 5(a). ■

Due to Property α, the relative location between two blocks B_a and B_b cannot be determined if B_b is in the left subtree of B_a. But, if a floorplan satisfies a whitespace condition between blocks as in Theorems 5 and 6, we can compare the relative distances of a parent block and child blocks to the origin.

Theorem 5: If a left child block, B_b is abutting to and right above a block B_m, a parent block, B_a is guaranteed to have shorter distance to the left-bottom corner than B_b, as long as there is no whitespace between B_a, B_b, and B_m.

Proof: Let the coordinate, width and height of a block B_* be (x_*, y_*, w_*, h_*). If $y_a \neq y_b$ and $x_a = x_b + w_b$, B_b is not the left child of B_a, but the right child of the block below B_b in a floorplan without whitespace, as B*-tree construction is performed in the depth first search (DFS) order [20]. This case is corresponding to Fig. 5(a). If $y_a = y_b$ and $x_a = x_b + w_b$, B_a has shorter distance to the left-bottom corner than B_b, independently of whether B_b is the left child of B_a or not. This case is illustrated in Fig. 5(b) and (c). ■

Theorem 6: A parent block B_a is guaranteed to have shorter distance to the left-bottom corner than any block B_b in the subtree T_a which has B_a as a root, as long as there is no whitespace between any two blocks in T_a.

Proof: Any parent has shorter distance to the left-bottom corner than the right child by Property α. Also, any parent has shorter distance to the left-bottom corner than the left child by Theorem 5. By

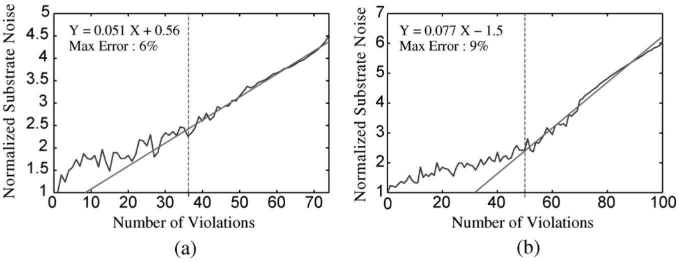

Fig. 6. Number of violations versus substrate noise. (a) Total 76 violations. (b) Total 100 violations.

recursively applying Property α and Theorem 5, any block B_b in T_a has longer distance to the left-bottom corner than B_a. ■

We can apply Property α, β and Theorem 4, 6 to find violations using DFS in the B^*-tree. As long as any of Property α, β and Theorem 4, 6 is satisfied with a preference in a BPDG, we can conclude that the preference is *not* violated. B^*-tree is not as efficient as sequence pair for BPDG due to both repeated DFS search and multiple conditions to check. This is because B^*-tree by nature does not provide an easy way to calculate the relation of any arbitrary two blocks (each block in B^*-tree only knows about the other three blocks, its parent and two children).

```
Algorithm 2 Fast Substrate Noise-Aware Floorplanning

Input: Analog BPDG $G_a$, Digital BPDG $G_d$
1: Floorplanning with analog blocks with $G_a$
2: Inflate the analog floorplan and make a virtual block $B_v$
3: Floorplanning with digital blocks and $B_v$ with $G_d$
```

V. Fast Substrate Noise-Aware Floorplanning

Our floorplanning algorithm efficiently examines discrepancy between a BPDG and an instance of sequence pair or B*-tree for fast substrate noise estimation using the theorems in Section IV. The overall algorithm is described in Algorithm 2.

We first floorplan the analog blocks only based on an analog BPDG, as the preference in the analog BPDG will result in a substrate-noise robust analog floorplan. The packed analog floorplan is treated as a virtual block, and inflated to insert whitespace around analog blocks as a guard ring. Finally, the virtual block is packed together with all the other digital blocks to get the final floorplan.

VI. Experimental Results

We implement the proposed algorithm in C++ by modifying a simulated annealing-based floorplanner, Parquet [23].[1] We perform experiments on Pentium4 Linux machines, and modify MCNC[1] benchmarks (ami33, ami49) and two randomly generated larger benchmarks (n75 with 75 blocks, n100 with 100 blocks) by choosing about 30% of the blocks in each benchmark as analog blocks and carefully generating frequency characteristics of all the blocks. All blocks are soft with $0.5 < W/H < 2.0$ aspect ratio constraint. All process dependent parameters are the same as in [5] and [10].

To show the fidelity of BPDG, we plot the normalized total substrate noise by the number of violations from one benchmark circuit in Fig. 6. It shows that normalized substrate noise increases near linearly as the number of violations increases. The range over 50% of maximum violations shows high fidelity with less than 6% and 9% in Fig. 6(a) and (b), respectively. Since the typical number of violations during simulated annealing falls in this high fidelity range, the number of violations in sequence pair or B^*-tree can be a good indicator of the amount of substrate noise on analog blocks.

TABLE II
Experimental Results With Sequence Pair (-sp) and B^*-Tree (-bt)

name	algorithm description	cost[a] function	input (node)	area (mm^2)	ws[c] (%)	norm[c] noise	CPU (sec)	overhead(%) CPU	overhead(%) area
parq -sp	pure parquet with seq-pair	$\frac{A}{A_r}$	ami33	1.19	3.2	821.1	0.8	0.0	0.0
			ami49	36.70	3.6	1629.9	2.6	0.0	0.0
			n75	42.04	4.0	3559.9	8.6	0.0	0.0
			n100	18.86	5.1	4697.5	24.6	0.0	0.0
bpdg -sp[b]	BPDG with seq-pair	$0.6\frac{A}{A_r} + 0.4\frac{NV}{NV_r}$	ami33	1.24	6.9	121.2	0.9	15.5	3.8
			ami49	37.90	7.1	72.2	2.7	6.6	3.4
			n75	43.12	6.6	173.1	9.2	7.1	2.6
			n100	19.22	6.9	202.5	26.4	7.1	1.9
modl -sp	substrate noise model with seq-pair	$0.6\frac{A}{A_r} + 0.4\frac{SN}{SN_r}$	ami33	1.23	6.1	143.9	73.0	8782.5	2.8
			ami49	38.40	8.4	90.8	158.3	6103.3	4.6
			n75	44.08	9.1	322.4	666.9	7692.5	4.9
			n100	19.94	11.1	696.1	1956.3	7844.1	5.8
parq -bt	pure parquet with B*-tree	$\frac{A}{A_r}$	ami33	1.19	2.6	805.8	0.3	0.0	0.0
			ami49	36.38	2.6	1569.2	0.7	0.0	0.0
			n75	41.61	3.0	3451.3	1.8	0.0	0.0
			n100	18.62	3.7	4390.6	3.6	0.0	0.0
bpdg -bt[b]	BPDG with B*-tree	$0.6\frac{A}{A_r} + 0.4\frac{NV}{NV_r}$	ami33	1.23	6.0	115.7	0.5	72.8	3.5
			ami49	37.58	5.9	67.4	1.2	65.5	3.3
			n75	42.73	5.7	191.9	3.0	65.1	2.7
			n100	19.12	6.4	184.4	6.2	70.6	2.7
modl -bt	substrate noise model with B*-tree	$0.6\frac{A}{A_r} + 0.4\frac{SN}{SN_r}$	ami33	1.22	5.1	142.0	11.3	3580.8	2.4
			ami49	37.24	5.1	99.0	35.0	4891.4	2.4
			n75	42.69	5.6	277.2	128.0	6833.7	2.6
			n100	18.93	5.5	488.4	245.6	6694.0	1.7

[a] A, NV, and SN denote total area, the number of violations, and total substrate noise on analog blocks, respectively. A_r, NV_r, and SN_r are the reference values of A, NV, and SN, respectively.

[b] for **bpdg-sp** and **bpdg-bt**, each side of the virtual analog block is inflated by 0.6% as a whitespace(guard ring) insertion.

[c] *ws* means whitespace, and *norm noise* means normalized total substrate noise on analog blocks.

Table II summarizes various algorithms we experiment and the corresponding results. The number of violations is *counted* for **bpdg-sp** and **bpdg-bt** by the theorems in Section IV, whereas substrate noise is *computed* for **modl-sp** and **modl-bt** based on the substrate noise model, i.e., (5). Current floorplan instance's substrate noise on the analog blocks is obtained after every movement inside the simulated annealing loop. Each number in the table is generated by taking the average of numbers obtained over 250 floorplans. All experiments are scheduled by Parquet, and stopped after the same number of movements for each benchmark. The final noises for all algorithms are computed based on (5).

From Table II, **parq-sp** shows the best area and cpu time (thus, 0% overhead), but the worst noise for all benchmarks as expected. The cpu time of **bpdg-sp** is significantly smaller than that of **modl-sp** for all benchmarks; **bpdg-sp** is 73× faster than **modl-sp** on average. The area overhead of **bpdg-sp** is slightly smaller for the three larger benchmarks as well than **modl-sp**. Last, **bpdg-sp** shows less total substrate noise than **modl-sp**. The same experimental simulations are performed again, but with B^*-tree (**parq-bt**, **bpdg-bt**, and **modl-bt**). It shows that the proposed approach, **bpdg-bt** is 33× faster **modl-bt** on average with highly comparable area overhead and less noise.

VII. Conclusion

We propose substrate noise-aware floorplanning for mixed signal SOC with fast substrate noise estimation powered by block preference directed graph (BPDG) with sequence pair and B^*-tree. The ex-

[1] [Online]. Available: http://vlsicad.eecs.umich.edu/BK/parquet

perimental results show that BPDG-based substrate noise estimation can speed up the sequence pair based floorplanning by 73× and the B^*-tree based floorplanning by 33×, comparing with the conventional *full-blown substrate noise simulation-based* floorplanning.

REFERENCES

[1] A. Nardi, H. Zeng, J. L. Garrett, L. Daniel, and A. L. Sangiovanni-Vincentelli, “A methodology for the computation of an upper bound on nose current spectrum of CMOS switching activity,” in *Proc. Int. Conf. Comput.-Aided Des.*, Nov. 2003, pp. 778–785.

[2] A. Koukab, K. Banerjee, and M. Declercq, “Modeling techniques and verification methodologies for substrate coupling effects in mixed-signal system-on-chip designs,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 6, pp. 823–836, Jun. 2004.

[3] M. V. Heijingen, M. Badarouglu, S. Donnay, G. G. E. Gielen, and H. J. D. Man, “Substrate noise generation in complex digital systems: Efficient modeling and simulation methodology and experimental verification,” *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1065–1072, Aug. 2002.

[4] H. Lan, Z. Yu, and R. W. Dutton, “A CAD-oriented modeling approach of frequency-dependent behavior of substrate noise coupling for mixed-signal IC design,” in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2003, pp. 195–200.

[5] B. E. Owens, S. Adluri, P. Birrer, R. Shreeve, S. K. Arunachalam, and K. Mayaram, “Simulation and measurement of supply and substrate noise in mixed-signal ICs,” *IEEE J. Solid-State Circuits*, vol. 40, no. 2, pp. 382–391, Feb. 2005.

[6] N. K. Verghese and D. J. Allstot, “Rapid simulation of substrate coupling effects in mixed-mode ICs,” in *Proc. IEEE Custom Integr. Circuits Conf.*, May 1993, pp. 18.3.1–18.3.4.

[7] R. Gharpurey and R. G. Meyer, “Modeling and analysis of substrate coupling in integrated Circuits,” *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 344–353, Mar. 1996.

[8] N. K. Verghese and D. J. Allstot, “Computer-aided design considerations for mixed-signal coupling in RF integrated circuits,” *IEEE J. Solid-State Circuits*, vol. 33, no. 3, pp. 314–323, Mar. 1998.

[9] J. P. Costa, M. Chou, and L. M. Silveria, “Efficient techniques for accurate modeling and simulation of substrate coupling in mixed-signal IC’s,” *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 5, pp. 597–607, May 1999.

[10] D. Ozis, T. Fiez, and K. Mayaram, “A comprehensive geometry-dependent macromodel for substrate noise coupling in heavily doped CMOS processes,” in *Proc. IEEE Custom Integr. Circuits Conf.*, May 2002, pp. 497–500.

[11] L. Deferm, C. Claes, and G. J. Declerck, “Two- and three-dimensional calculation of substrate resistance,” *IEEE Trans. Electron Devices*, vol. 35, no. 3, pp. 339–352, Mar. 1988.

[12] S. Mitra, R. A. Rutenbar, L. R. Carley, and D. J. Allstot, “Substrate-aware mixed-signal macro-cell placement in WRIGHT,” *IEEE J. Solid-State Circuits*, vol. 30, no. 3, pp. 529–532, Mar. 1995.

[13] S.-S. Lee and D. J. Allstot, “Electrothermal simulation of integrated circuits,” *IEEE J. Solid-State Circuits*, vol. 28, no. 12, pp. 1283–1293, Dec. 1993.

[14] C. Lin and D. M. W. Leenaerts, “A new efficient method substrate-aware device-level placement,” in *Proc. Asia South Pac. Des. Autom. Conf.*, Jan. 2000, pp. 533–536.

[15] W. H. Kao and W. K. Chu, “Noise constraint driven placement for mixed signal designs,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2003, pp. 712–715.

[16] G. Blakiewicz, M. Jeske, M. Chrzanowska-Jeske, and J. S. Zhang, “Substrate noise modeling in early floorplanning of mixed-signal SOCs,” in *Proc. Asia South Pac. Des. Autom. Conf.*, Jan. 2005, pp. 819–823.

[17] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, “VLSI module placement based on rectangle-packing by the sequence-pair,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 15, no. 12, pp. 1518–1524, Dec. 1996.

[18] X. Tang and D. F. Wong, “Floorplanning with alignment and performance constraints,” in *Proc. Des. Autom. Conf.*, Jun. 2002, pp. 848–853.

[19] H. Xiang, X. Tang, and M. D. F. Wong, “Bus-driven floorplanning,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 11, pp. 1522–1530, Nov. 2004.

[20] Y.-C. Chang, Y.-W. Chang, G.-M. Wu, and S.-W. Wu, “B∗-Ttrees: A new representation for non-slicing floorplans,” in *Proc. Des. Autom. Conf.*, Jun. 2000, pp. 458–463.

[21] T.-C. Chen and Y.-W. Chang, “Modern floorplanning based on fast simulated annealing,” in *Proc. Int. Symp. Phys. Des.*, Apr. 2005, pp. 104–112.

[22] H. Han, “Synthesized compact models for substrate noise coupling in mixed-signal ICs,” Ph.D. dissertation, Dept. Electr. Eng., Stanford Univ., Stanford, CA, 2006.

[23] S. N. Adya and I. L. Markov, “Fixed-outline floorplanning: Enabling hierarchical design,” *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 12, pp. 120–1135, Dec. 2003.

Efficient Distributed On-Chip Decoupling Capacitors for Nanoscale ICs

Mikhail Popovich, Eby G. Friedman, Radu M. Secareanu, and Olin L. Hartin

***Abstract*—A distributed on-chip decoupling capacitor network is proposed in this paper. A system of distributed on-chip decoupling capacitors is shown to provide an efficient solution for providing the required on-chip decoupling capacitance under existing technology constraints. In a system of distributed on-chip decoupling capacitors, each capacitor is sized based on the parasitic impedance of the power distribution grid. Various tradeoffs in a system of distributed on-chip decoupling capacitors are also discussed. Related simulation results for typical values of on-chip parasitic resistance are also presented. The worst case error is 0.003% as compared to SPICE.**

***Index Terms*—Decoupling capacitors, power distribution systems, power noise.**

I. INTRODUCTION

Decoupling capacitors are widely used to manage power supply noise [1]. Decoupling capacitors are an effective way to reduce the impedance of power delivery systems operating at high frequencies [2]. A decoupling capacitor acts as a local reservoir of charge, which is released when the power supply voltage at a particular current load drops below some tolerable level. Since the inductance scales slowly [3], the location of the decoupling capacitors significantly affects the design of the power/ground (P/G) networks in high performance integrated circuits (ICs) such as microprocessors. At higher frequencies, a distributed system of decoupling capacitors is placed on-chip to effectively manage the power supply noise [4].

The efficacy of decoupling capacitors depends upon the impedance of the conductors connecting the capacitors to the current loads and power sources. As described in [5], a maximum parasitic impedance

Manuscript received March 07, 2007; revised August 28, 2007 and November 30, 2007. Current version published November 19, 2008. This work was supported in part by the Semiconductor Research Corporation under Contract 2004-TJ-1207, by the National Science Foundation under Contract CCR-0304574 and Contract CCF-0541206, by grants from the New York State Office of Science, Technology and Academic Research to the Center for Advanced Technology in Electronic Imaging Systems, and by grants from Intel Corporation, Eastman Kodak Company, and Freescale Semiconductor Corporation.

M. Popovich is with the QCT, Qualcomm Corporation, San Diego, CA 92121 USA (e-mail: mikhailp@qualcomm.com).

E. G. Friedman is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: friedman@ece.rochester.edu).

R. M. Secareanu and O. L. Hartin are with the Microwave and Mixed Signal Technologies Laboratory, Freescale Semiconductor, Tempe, AZ 85284 USA (e-mail: r54143@freescale.com; lee.hartin@freescale.com).

Digital Object Identifier 10.1109/TVLSI.2008.2001735