Fast Substrate Noise-Aware Floorplanning with Preference Directed Graph for Mixed-Signal SOCs

Minsik Cho, Hongjoong Shin and David Z. Pan
Dept. of ECE, The University of Texas at Austin, Austin, TX 78712
thyeros@cerc.utexas.edu, unishin@cerc.utexas.edu, dpan@ece.utexas.edu

Abstract— **In this paper, we introduce a novel substrate noise estimation technique during early floorplanning, based on the concept of Block Preference Directed Graph (BPDG) and the classic Sequence Pair (SP) floorplan representation. Given a set of analog and digital blocks, the BPDG is constructed based on their inherent noise characteristics to capture their preferred relative orders for substrate noise minimization. For each sequence pair generated during floorplanning evaluation, we can measure its violation against BPDG very efficiently. We observe that by simply counting the number of violations obtained in this manner, it correlates remarkably well with accurate but computation-intensive substrate noise modeling. Thus, our BPDG-based model has high fidelity to guide the substrate noise-aware floorplanning and layout optimization, which become a growing concern for mixed-signal/RF system on chips (SOC). Our experimental results show that the proposed approach is over 60x faster than conventional floorplanning with even very compact substrate noise models. We also obtain less area and total substrate noise than the conventional approach.**

I. Introduction

Continuing demand for data and telecommunication application is driving tighter integration of many heterogeneous functions into a single system-on-chip. These components can be pre-designed IP cores of different natures such as sensitive front-end RF circuits, high-precision analog/mixed-signal circuits, and high-performance digital circuits. Therefore, the interference between these heterogeneous components has to be considered during layout planning and optimization [1]. A key interference is the substrate noise caused by large amount of switching activities in high speed digital cores, to the analog/RF components. It may degrade the reliability and performance of these sensitive analog/mixed-signal/RF IPs [2]. The problem is becoming a growing concern due to higher clock frequency, more accurate analog precision, deeper technology scaling, and the integration of front-end RF with digital blocks [3]–[5]. Many effects that corrupt RF signal such as DC offset, oscillator pulling and pushing, local oscillator leakage can be traced to the substrate-coupled noise [2].

Therefore, fast yet accurate evaluation and optimization of substrate noise in physical design has become a crucial part of mixed-signal SOC designs, in order to avoid expensive over-design and excessive design iterations. A key step of such layout optimization is the floorplanning stage [6]. Although many works have been done in modeling and simulation of substrate noise [2], [3], [5], [12]–[15], there is not much in the literature on substrate noise optimization in early floorplanning stage. Lin et al. [7] proposed an optimization technique that incorporates substrate noise minimization into optimization loops. This technique, however, requires detailed and expensive circuit simulations to estimate the coupled substrate noise. Blakiewicz et al. [8] proposed a floorplanning algorithm with substrate noise as a cost function. A more scalable substrate model with frequency-dependent sensitivity function of analog and digital blocks is used, but still it requires significant computational overhead to evaluate the substrate noise cost function during floorplanning.

In this paper, we propose a novel concept of block preference directed graph (BPDG) to overcome the modeling bottleneck for substrate noise-aware floorplanning. Our BPDG-based model has high fidelity compared with accurate but much more expensive substrate noise modeling. Thus, it is suitable to guide MS-SOC floorplanning. The major contributions of this paper include the following.

- We introduce the novel concept of block preference directed graph (BPDG) to represent the preferred relative block locations in floorplanning. In BPDG, all the preferences are decided to minimize the substrate noise, and each preference is specified as a directed edge.
- We propose a fast substrate noise estimation algorithm by combining BPDG and sequence pair. We simply count how many preferences in BPDG are not held in a sequence pair with simple bitwise-OR operation.
- We show that our approach has surprisingly high fidelity to the substrate noise calculated by a most recent, accurate substrate noise model [5].
- We propose a fast substrate noise-aware floorplanning algorithm with BPDG and sequence pair. Our experimental results show the proposed approach is significantly (at least 60x) faster than a conventional simulation-based, substrate noise-aware floorplanning.

The rest of the paper is organized as follows. In Section II, preliminaries are described. In Section III, the concept of block preference directed graph is introduced. The fast substrate noise estimation algorithm is proposed in Section IV, and the overall floorplanning flow is described in Section V. Experimental results are discussed in Section VI. Section VII concludes this paper with future work.

This work is supported in part by SRC under contract 2005-TJ-1321, IBM Faculty Award, and equipment donations from Intel.

0-7803-9451-8/06/$20.00 ©2006 IEEE.

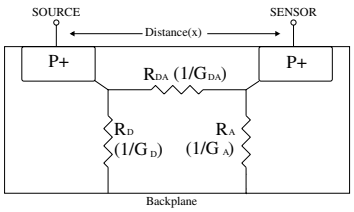


Fig. 1. Macromodel for the substrate

II. Preliminaries

A. Sequence Pair and Block Alignment

A sequence pair [9] is a pair of sequences of *n* elements representing a list of *n* blocks. Two sequences specify the geometric relations between each pair of blocks. 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 [9]. Conditions for block alignments in sequence pair are studied in [10]. H/V alignment constraints and abutting constraint between blocks are introduced and applied for performance-aware floorplanning.

B. Substrate Noise Model

Several techniques have been proposed to model and analyze substrate noise accurately in integrated circuit level [11]–[13]. However, these techniques require the detailed implementation information in transistors and time-intensive transistor-level simulation. In this paper, we use compact substrate coupling model [5] to evaluate an instance of floorplan from conventional approach, and to verify the final floorplan. The model in [5] is known to be highly scalable with dimensions and separations.

A two-port lumped resistor network, modeling substrate is illustrated in Fig. 1. 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. The resistances, R_{DA}, R_A and R_D can be derived from the scalable macromodel, which is based on Z-parameters.

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

where $\triangle = G_A G_{DA} + G_D G_{DA} + G_A G_D$ and any Z_{ij} can be calculated with equations in [5], [14], [15].

The coupling gain (propagation gain) of the substrate can be calculated from the value of resistors in the two-port lumped network shown in Fig. 1. The coupling gain of *i-th* digital block to *j-th* analog block, $CG_{i,j}$ is given by:

$$CG_{i,j} = \frac{R_A}{R_A + R_{DA}} = \frac{G_{DA}}{G_{DA} + G_A} = \frac{Z_{12}}{Z_{22}} \quad (2)$$

Although $CG_{i,j}$ exhibits frequency-dependent characteristics, it is constant under a few gigahertz [4]. In this paper, we assume that the bands of interest are within this limit. The quantity of the substrate noise can be estimated using frequency-dependent characteristics of noise source and sensor block, and a simple analytical formula based on $CG_{i,j}$. The substrate noise of *j-th* analog block from switching of *i-th* digital block, $N_{i,j}$ can be approximated by [8]:

$$N_{i,j} = (CG_{i,j}) \cdot \sqrt{\int_0^\infty (S_i(f) \cdot H_j(f))^2 df} \quad (3)$$

where $S_i(f)$ and $H_j(f)$ are Power Spectral Density (PSD) of noise source and transfer function of noise sensor respectively. Also, the total noise from all digital blocks is:

$$N_{total} = \sum_i \sum_j N_{i,j} \quad (4)$$

As shown in Eqn. (3), $CG_{i,j}$ is scaled by average power of noise with regard to the frequency. The frequency-dependent noise generated by a digital block, $S_i(f)$ is shaped by the transfer function of the noise sensor. The integration of the shaped power of noise represents the quantity of noise injected into analog block, when $CG_{i,j}$ is equal to 1. We use a piecewise-linear approximation of PSD to estimate $S_i(f)$, and Power/Ground bounce limits to determine its parameters.

III. BPDG: Block Preference Directed Graph

The substrate noise model in Section II-B is one of the most compact models with high scalability and accuracy. However, it is still computationally expensive to perform a substrate noise estimation even with such an efficient model 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], [14], [15]), whereas area and wire-length can be calculated approximately. Furthermore, computing noise itself with Eqn. (2, 3) is not computationally trivial.

For fast substrate noise estimation, a new concept of *block preference directed graph*, BPDG is introduced and described in this section. BPDG represents preferred relative locations of blocks to guide substrate noise-aware floorplanning. BPDG construction consists of three steps.

1) A table of substrate noises ($N_{i,j}$) between all analog and digital blocks is constructed.
2) Analog block orderings and digital block orderings are created separately with the substrate noise table.
3) BPDG is constructed by finding common orders from block orderings.

The following subsections illustrate each step with detailed examples in Table I and Fig. 2, 3, 4.

A. Substrate Noise Table Construction

Since substrate noise is heavily related to the distance between blocks, we assume that the nominal distance is fixed to normalize the effect of distance. With such fixed distance, the substrate noise between a digital block and an analog block purely depends on frequency coupling and geometric properties like area and perimeter [5], [14], [15]. Under such condition, for each digital block D_i and analog block A_j, a substrate noise on A_j due to D_i, $N_{i,j}$ can be computed from Eqn. (3). Table I shows an example of substrate noise table of between digital blocks (D_1, D_2, D_3, D_4, D_5, D_6) and analog blocks (A_1, A_2, A_3).

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

B. Analog Block Ordering

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 $N_{1,1} = 5 > N_{1,3} = 3 > N_{1,2} = 2$. The other five orderings can be obtained in the same manner, as shown in Fig. 2. Basically, this ordering pushes more noise-sensitive analog blocks to the head, and less sensitive ones to the tail of block orderings.

$$D_1 : A_1 \leftarrow A_3 \leftarrow A_2 \quad , \quad D_2 : A_3 \leftarrow A_1 \leftarrow A_2$$
$$D_3 : A_3 \leftarrow A_1 \leftarrow A_2 \quad , \quad D_4 : A_3 \leftarrow A_2 \leftarrow A_1$$
$$D_5 : A_1 \leftarrow A_3 \leftarrow A_2 \quad , \quad D_6 : A_3 \leftarrow A_2 \leftarrow A_1$$

Fig. 2. Analog block orderings

C. Digital Block Ordering

In similar way, digital blocks can be sorted for each analog block by the *ascending* order of substrate noise. Again considering the example in Table I, digital block D_6, D_2, D_4, D_1, D_3 and D_5 can be ordered such that the substrate noise on A_1 is increasing. All digital block orderings are shown in Fig. 3. This pushes less aggressive blocks to the head and more aggressive blocks to the tail of block orderings.

$$A_1 \ : \ D_6 \leftarrow D_2 \leftarrow D_4 \leftarrow D_1 \leftarrow D_3 \leftarrow D_5$$
$$A_2 \ : \ D_2 \leftarrow D_1 \leftarrow D_3 \leftarrow D_6 \leftarrow D_5 \leftarrow D_4$$
$$A_3 \ : \ D_1 \leftarrow D_3 \leftarrow D_2 \leftarrow D_5 \leftarrow D_4 \leftarrow D_6$$

Fig. 3. Digital block orderings

D. BPDG Construction

The two key ideas behind BPDG construction are: first, to find common block order patterns in order to minimize the substrate noise; second, to make less aggressive digital blocks and less sensitive analog blocks interfaced. An analog BPDG and a digital BPDG are constructed with analog and digital block orderings by Algorithm 1. 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. 4 as an example. Since A_3 is before A_2 for all analog block orderings in Fig. 2, 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) with a directed edge from D_3 to D_1, as D_1 is before D_3 for all digital block orderings in Fig. 3. Note that A_1 does not have any edge, as there is no common order regarding A_1 in Fig. 2, and D_6 only has an edge to D_0 for analog-digital separation. Lastly, G_a and G_d are merged via virtual vertex D_0.

Algorithm 1 BPDG Construction

Input: Analog, Digital block orderings O_a and O_d
1: Analog BPDG $G_a \leftarrow \phi$, Digital BPDG $G_d \leftarrow \phi$
2: **for** each analog block A_i, A_j, i ≠j **do**
3: **if** A_i is before A_j in all O_a **then**
4: Add a directed edge from A_j to A_i to G_a
5: **end if**
6: **end for**
7: **for** each digital block D_i, D_j, i ≠j **do**
8: **if** D_i is before D_j in all O_d **then**
9: Add a directed edge from D_j to D_i to G_d
10: **end if**
11: **end for**
12: Add a virtual vertex D_0 for G_a to G_d
13: Add directed edges from all root vertices to D_0
Output: G_d

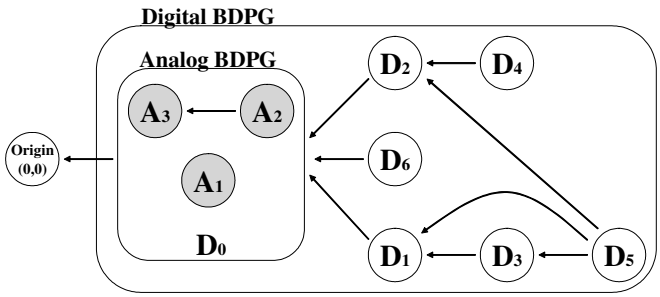


Fig. 4. Block preference directed graph (BPDG)

IV. SUBSTRATE NOISE ESTIMATION WITH BPDG

The BPDG in Section III can be used to estimate substrate noise quickly by comparing it against a sequence pair which is one of the most popular floorplan representations. In this section, a theorem which returns the number of violations against preferences in a BPDG from a sequence pair is presented, and its high fidelity to substrate noise is shown. The number of violations is highly correlated to substrate noise quantity; intuitively, more violations indicate more noise, because each directed edge from a block B_a to a block B_b in the BPDG means that B_b is preferred to be closer to the origin (left-bottom corner of floorplan) than B_a to reduce the substrate noise.

A. Sequence Pair with BPDG

In [10], 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. 5(a) shows a floorplan with several blocks. In this example, 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 (only B_1 and B_3 are abutted to B_a). In the following, we introduce several definitions based on *strictly ahead* for easier explanation of this section.

DEFINITION α. Given a block B_a and a sequence pair (P, N), all the blocks which are both strictly ahead of B_a and below B_a form a **strictly below set** of B_a.

DEFINITION β. Given a block B_a and a sequence pair (P, N), all the blocks which are both strictly ahead of B_a and to the left of B_a form a **strictly left set** of B_a.

DEFINITION γ. Given a block B_a and a sequence pair (P, N), any block in a strictly below/left set of B_a and abutting to B_a is a **reference block**.

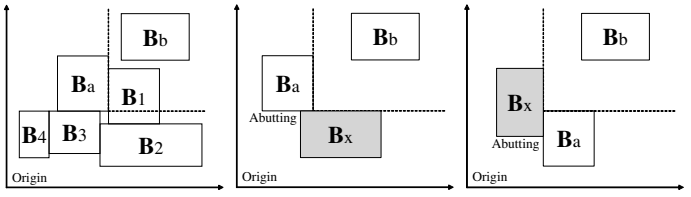
In Fig. 5(a), B_2, B_3 and B_4 are strictly below set of B_a, because they are strictly ahead of B_a as well as below B_a. Also, B_3 is a reference block of B_a. One intuitive property of the reference block is stated in Lemma α referring to [10].

Lemma α: *If a block B_a has a strictly below/left set S, there must exist some reference block B_x in S under a completely packed floorplan.*

Based on Lemma α, the relative locations of two blocks can be determined. Consider a specific floorplan in Fig. 5(b) where B_a is to the left of B_b, and B_x is a reference block of B_a. It can be easily 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 location of two blocks **conservatively** with a sequence pair is presented as Theorem 1 by extending Theorem 1 and 3 in [10]. Note that the conditions 1) and 2) of Theorem 1 are corresponding to Fig. 5(b) and (c) respectively.

Theorem 1: *Let S_b be a strictly below set of B_a and S_l a strictly left set of B_a. 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 following conditions is satisfied.*

1) *for any block B_s in S_b, a sequence pair (P, N) is $(..B_aX_1B_bX_2B_s.., ..B_sY_1B_a..B_b..)$.*
2) *for any block B_s in S_l, a sequence pair (P, N) is $(..B_sX_3B_bX_4B_a.., ..B_sY_2B_a..B_b..)$.*



(a) (b) (c)

Fig. 5. Floorplan examples

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 1 to see if such preference is held in (P, N). Theorem 1 can be further simplified into Theorem 2 with the longest common string (LCS) search by narrowing down the size of subsequences to scan.

Theorem 2: *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 following conditions is satisfied.*

1) *there is no block B_s satisfying LCS(X_1, Y_1)=ϕ in a sequence pair (P, N)=$(..B_aX_1B_s..B_b.., ..B_sY_1B_a..B_b..)$.*
2) *there is no block B_s satisfying LCS(X_2, Y_2)=ϕ in a sequence pair (P, N)=$(..B_b..B_sX_2B_a.., ..B_sY_2B_a..B_b..)$.*

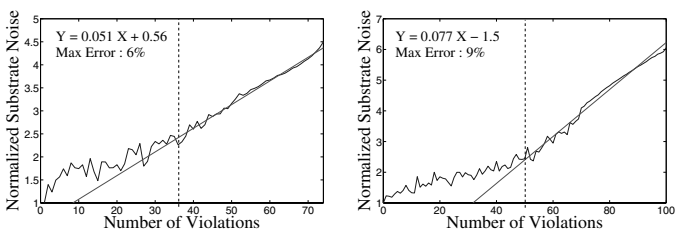
The following sequence pairs show examples with the BPDG in Fig. 4. Note that the blocks one need to pay attention to are marked with *, and we highlight one violation, even though there can be more.

- $(D_0D_6D_1D_2^*D_3D_4^*D_5,\ D_4^*D_0D_6D_1D_2^*D_3D_5)$
  This case has $D_2 \leftarrow D_4$ violation, because D_2 is after D_4 in the second sequence which does not match either one of required sequence pair patterns in Theorem 1.
- $(D_4D_5D_1^*D_2D_6D_0^*D_3^*,\ D_0^*D_5D_4D_1^*D_2D_6D_3^*)$
  This case has $D_1 \leftarrow D_3$ violation. D_0 is below D_1 and $LCS(D_2D_6,\ D_5D_4) = \phi$ But, D_0 is before D_3 in the first sequence which violates the required sequence pair pattern in condition 1) of Theorem 1.

The significance of Theorem 1 and 2 is that a geometric distance from the origin to any two blocks in a sequence pair can be compared conservatively *without other geometric information*. Thus, whether an edge (preference) in a BPDG is held in a sequence pair can be checked extremely efficiently. Note that in a real implementation, bitwise-OR can be used instead of LCS computation, since we are only interested in whether there is a common sequence.

B. Fidelity of BPDG

In order to measure the fidelity of BPDG-based model for substrate noise estimation, *ami33* from MCNC benchmarks [16] was simulated with carefully generated noise characteristics. Fig. 6 shows the normalized substrate noise on all analog blocks by the number of violations counted by Theorem 2 with different total number of violations. It shows that normalized substrate noise increases near linearly as the



(a) Total 76 violations (b) Total 100 violations

Fig. 6. Number of violations vs. Substrate noise

TABLE II
EXPERIMENTAL RESULTS

Name	Algorithm Description	Cost[a] Function	Input (node)	Area (mm^2)	White Space(%)	Normalized Noise	CPU (sec)	Overhead (%) CPU	Overhead (%) Area
parq	Pure Parquet with Sequence Pair	$\frac{A}{A_r}$	ami33	1.19	3.2	821.1	0.8	0.0	0.0
			ami49	36.7	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[b]	BPDG with Sequence 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	37.9	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	Substrate Noise Model with Sequence 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	38.4	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

[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**, each side of the virtual analog block is inflated by 0.6% as a whitespace(guard ring) insertion.

number of violations increases. Notice that the range over 50% of maximum violations shows high fidelity with less than 6% error in Fig. 6(a), and 9% in Fig. 6(b). Since the typical number of violations during simulated annealing falls in this high fidelity range, the number of violations in sequence pair is a good indicator of substrate noise. Thus, by comparing BPDG of Section III against sequence pair, substrate noise can be estimated very fast with high fidelity.

- Our approach estimates substrate noise without accurate geometric information such as x and y coordinates. As a result, all the efforts to compute the accurate locations of all blocks can be saved.
- Our approach needs only integer and bitwise-OR operations whereas model-based noise estimation requires floating point operations and transcendental functions like $\exp(x)$.

V. FAST SUBSTRATE NOISE-AWARE FLOORPLANNING

Our floorplanning algorithm takes advantage of the BPDG-based fast substrate noise estimation, efficiently examining discrepancy between BDPG in Section III and sequence pair in Section IV. Also, using block inflation, we insert whitespace around analog blocks as a guard ring. Note that our white space allocation is done as a preprocessing to minimize area overhead. The overall algorithm is described in Algorithm 2.

Algorithm 2 Fast Substrate Noise-Aware Floorplanning

Input: Analog BPDG Ag, Digital BPDG Dg
1: Do floorplanning with analog blocks with Ag
2: Inflate the analog block floorplan
3: Make the analog floorplan as a virtual block B_v
4: Do floorplanning with digital blocks and B_v with Dg
Output: Final floorplan

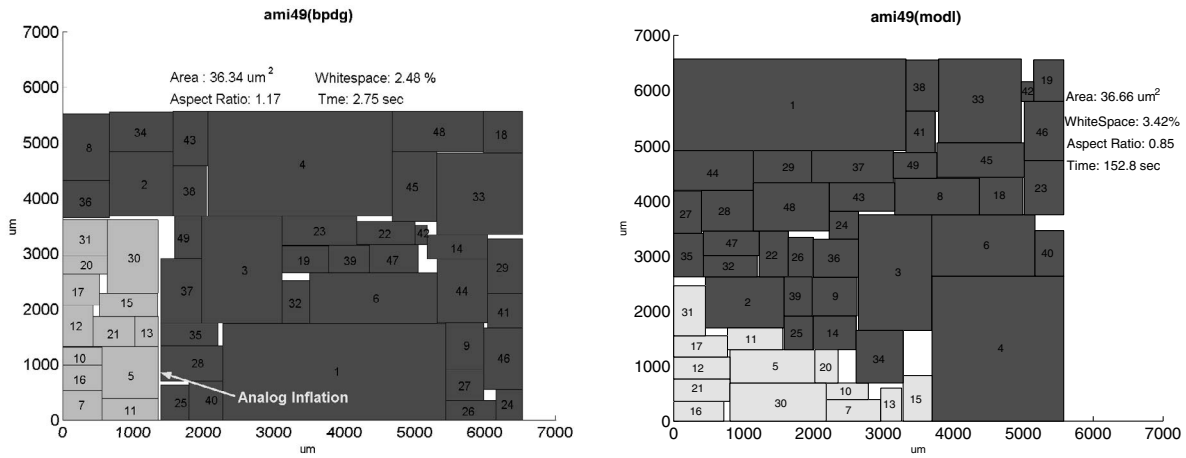
VI. EXPERIMENTAL RESULTS

We implemented the proposed algorithm in C++ by modifying Parquet [17] which is a simulated annealing-based floorplanning package available in [18]. In order to compare our approach (**bpdg** in Table II) with other approaches, we also implemented the conventional model-based, substrate noise-aware floorplanning algorithm (**modl** in Table II). After every movement inside the simulated annealing loop, to estimate current floorplan instance's substrate noise on the analog blocks, the number of violations by Theorem 2 was *counted* for **bpdg**, whereas substrate noise was *computed* for **modl** based on the substrate noise model, i.e., Eqn. (4) in Section II-B. The cost functions we used for each algorithm are summarized in Table II. Note that we disabled wirelength optimization, since real implementation of mixed-signal SOCs has sparse interconnection between analog blocks and digital blocks, which is not well reflected in MCNC benchmarks. However, our approach can be readily extended to include wirelength optimization, maintaining high computational efficiency.

All algorithms were tested on a Pentium4 Linux machines with MCNC [16] benchmarks (ami33, ami49) and two randomly generated larger benchmarks (n75 with 75 blocks, n100 with 100 blocks). About 30% of the blocks in each benchmark were chosen as analog blocks, and noise characteristics of all the blocks were carefully generated. All process dependent parameters were the same as in [5], [14] and [15].

Table II shows experimental results for all benchmarks with three algorithms. Each number in the table is generated by taking the average of numbers obtained over 250 floorplans. The simulated annealing of each floorplanning is scheduled by Parquet, and stopped after the same number of movements for each benchmark. The final noise quantities for all algorithms were computed based on Eqn. (4) for fair comparison.

The last two columns show the overhead of each algorithm in terms of cpu time and area with respect to **parq**. From the table, **parq** shows the best area and cpu time (thus, 0% overhead), but the worst noise for all benchmarks as expected. The cpu time of **bpdg** is significantly smaller than that of **modl** for all benchmarks; **bpdg** is approximately 60-80 times faster than **modl**. The area overhead of **bpdg** is slightly smaller for the three larger benchmarks as well than



(a) With proposed approach(**bpdg**)

(b) With model-based simulation approach(**modl**)

Fig. 7. Result of packing ami49

modl. Lastly **bpdg** shows less total substrate noise than **modl**. The reason why the proposed algorithm overall shows both smaller area and less substrate noise is that whitespace is more efficiently utilized. By making an analog floorplan inflated as a preprocessing step as in Section V, the substrate noise becomes less in the beginning of annealing, and this allows the simulated annealing engine to optimize the area further without increasing substrate noise. An analogy of this kind of effect can be found in congestion-aware placement [19].

VII. Conclusion

In order to cope with significant substrate noise impact on analog circuits from digital circuits, we propose substrate noise-aware floorplanning with fast substrate noise estimation powered by block preference directed graph (BPDG) and sequence pair. Compared with Parquet [17], the proposed approach has on average only 9% cpu time overhead, whereas naive model-based simulation approach shows over 6000% overhead. Also, the proposed approach shows smaller area overhead due to the efficient utilization of whitespace.

Since BPDG is a general concept for fast cost evaluation, it will be extend to deal with temperature or performance estimation in the future.

VIII. Acknowledgement

The authors would like to thank Prof. Karti Mayaram from Oregon State Univ. and Prof. Ranjit Gharpurey from the Univ. of Texas at Austin for helpful discussions.

References

[1] A. Nardi, H. Zeng, J. L. Garrett, L. Daniel, and A. L. S-Vincentelli, "A Methodology for the computation of an upper bound on noise current spectrum of CMOS swichting activity," in *Proc. Int. Conf. on Computer Aided Design*, 2003, pp. 778–785.

[2] A. Koukab, K. Banerjee, and M. Declercq, "Modeling Techniques and Verification Methdologies for Substrate Coupling Effects in Mixed-signal System-on-Chip designs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 6, 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 experiemental verification," *IEEE J. Solid-State Circuits*, vol. 37, 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. on Quality Electronic Design*, Mar 2003, pp. 195–200.

[5] B. 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, Feb 2005.

[6] T. Blalack, Y. Leclercq, and C. P. Yue, "On-chip RF isolation techniques," in *Proc. IEEE BCTM.*, 2002, pp. 205–211.

[7] C. Lin and D. Leenaerts, "A New Efficient Method Substrate-Aware Device-Level Placement," in *Proc. Asia and South Pacific Design Automation Conf.*, 2000, pp. 533–536.

[8] G. Blakiewicz, M. Jeske, M. Chrzanowska-Jeske, and J. S. Zhang, "Substrate Noise Modeling in Early Floorplanning of Mixed-Signal SOCs," in *Proc. Asia and South Pacific Design Automation Conf.*, Jan 2005, pp. 819–823.

[9] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," vol. 15, Dec 1996.

[10] X. Tang and D. Wong, "Floorplanning with Alignment and Performance Constraints," in *Proc. Design Automation Conf.*, Jun 2002.

[11] W. K. Chu, N. Verghese, K. S. H. Cho, H. Tsujikawa, S. Hirano, S. Doushoh, M. Nagata, A. Iwata, and T. Ohmoto, "A Substrate Noise Analysis Methodology for Large-Scale Mixed-Signal ICs," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2003.

[12] N. K. Verghese and J. J. Allstot, "Computer-aided design considerations in Mixed-signal coupling in RF integration circuits," *IEEE J. Solid-State Circuits*, vol. 33, Mar 1998.

[13] J. P. Costa, M. Chou, and L. M. Silveria, "Efficient techniques for accurate modeling and simulation of substrate coupling in Mixed-signal ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 5, pp. 597–607, May 1999.

[14] D. Ozis, T. Fiez, and K. Mayaram, "An Efficient Modeling Approach for Substrate Noise Coupling Analysis," in *Proc. IEEE Int. Symp. on Circuits and Systems*, 2002.

[15] ——, "Comprehensive geometry-dependent macromodel for substrate noise coupling in heavily doped cmos processes," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2002.

[16] http://www.cse.ucsc.edu/research/surf/GSRC/MCNC.

[17] S. N. Adya and I. L. Markov, "Fixed-outline Floorplanning : Enabling Hierarchical Design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 11, pp. 120–1135, Dec 2003.

[18] http://vlsicad.eecs.umich.edu/BK/parquet.

[19] U. Brenner and A. Rohe, "An effective congestion driven placemnet framework," in *Proc. Int. Symp. on Physical Design*, 2002, pp. 6–11.