the range of lower costs, the k-NN and MC curves practically coincide, meaning that k-NN does not offer a significant advantage over the simplistic solution. In contrast, ONN outperforms both MC and k-NNs. Zero test error is achieved for a subset of cost 0.09.

Table II shows the number of kept specification tests from each test group in the case of ONN. The third column refers to the best non-RF subset, which achieves an error of 13 mispredicted devices. The forth column shows the additional RF specification tests that suffice to reduce this error down to zero. Note that some specification tests in the best non-RF subset have been eliminated on account of their correlation to the selected RF specification tests. In particular, the digital tests as well as the tests to examine the lock of the voltage control oscillator (VCO) have been deduced to be redundant. Intuitively, the latter tests have been eliminated on account of their correlation to the tests of the mixer.

V. Conclusion

Analysis of historical production test data from an RF device reveals that specification tests comprise significant redundancy. This redundancy can be exploited to build prediction models for reaching pass or fail decisions based on a reduced-size set of specification tests. To this end, advanced machine learning methods, such as the combination of a multi-objective GA and an ONN, achieve excellent results and demonstrate great potential for reducing test cost through specification test compaction. For a specification test subset of cost $0.09 \times C(S)$, where $C(S)$ is the cost of the complete specification test suite, our results indicate that the ONN can predict correctly the pass or fail label of all devices in our data set. Moreover, this result is considered to be statistically significant since the data set was collected from four different lots within a period of six months.

References

[1] S. D. Huss and R. S. Gyurcsik, "Optimal ordering of analog integrated circuit tests to minimize test time," in *Proc. ACM/IEEE Des. Autom. Conf.*, 1991, pp. 494–499.

[2] L. Milor and A. L. Sangiovanni-Vincentelli, "Minimizing production test time to detect faults in analog circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits and Systems*, vol. 13, no. 6, pp. 796–813, Jun. 1994.

[3] J. B. Brockman and S. W. Director, "Predictive subset testing: Optimizing IC parametric performance testing for quality, cost, and yield," *IEEE Trans. Semicond. Manuf.*, vol. 2, no. 3, pp. 104–113, Mar. 1989.

[4] S. Biswas, P. Li, R. D. Blanton, and L. Pileggi, "Specification test compaction for analog circuits and MEMS," in *Proc. Des., Autom. Test Eur.*, 2005, pp. 164–169.

[5] S. Biswas and R. D. Blanton, "Statistical test compaction using binary decision trees," *IEEE Des. Test Comput.*, vol. 23, no. 6, pp. 452–462, Jun. 2006.

[6] H.-G. D. Stratigopoulos, P. Drineas, M. Slamani, and Y. Makris, "Non-RF to RF test correlation using learning machines: A case study," in *Proc. IEEE VLSI Test Symp.*, 2007, pp. 9–14.

[7] C. M. Bishop, *Neural Networks for Pattern Recognition*. London, U.K.: Oxford University Press, 1995.

[8] H.-G. Stratigopoulos and Y. Makris, "Error moderation in low-cost machine-learning-based Analog/RF testing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 27, no. 2, pp. 339–351, Feb. 2008.

[9] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recog. Lett.*, vol. 10, pp. 335–347, 1989.

[10] K. Deb, A. Pratap, A. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionary Comput.*, vol. 6, no. 2, pp. 182–197, Feb. 2002.

[11] S. Biswas and R. D. Blanton, "Test compaction for mixed-signal circuits using pass-fail test data," in *Proc. IEEE VLSI Test Symp.*, 2008, pp. 299–308.

Pattern Sensitive Placement Perturbation for Manufacturability

Shiyan Hu, Pratik Shah, and Jiang Hu

***Abstract*—The gap between VLSI technology and fabrication technology leads to strong refractive effects in lithography. Consequently, it is a huge challenge to reliably print layout features on wafers. The quality and robustness of lithography directly depend on layout patterns. It becomes imperative to consider the manufacturability issue during layout design such that the burden of lithography process can be alleviated. In this paper, three algorithms, namely, cell flipping algorithm, single row optimization approach and multiple row optimization approach, are proposed to tune any existing cell placement to be lithography friendly. These algorithms are based on dynamic programming and graph theoretic approaches, and can provide different tradeoff between critical dimension (CD) variation reduction and wirelength increase. Using lithography simulations, our experimental results demonstrate that over 15% CD variation reduction can be obtained in post-OPC stage by the new approaches while only less than 1% additional wire is introduced.**

***Index Terms*—Boundary pattern, CD variation, manufacturability, physical design, placement.**

I. Introduction

As VLSI technology enters the nano-scale regime, demands for minimum feature sizes have outpaced the advances in lithography hardware solutions. This imposes a great challenge on manufacturing reliability. Currently, semiconductor industry heavily relies on *resolution enhancement techniques* (RETs) for improving printability. Roughly speaking, printability refers to the difficulty in obtaining a good match between the intended image and the printed image in lithography process. Printability is often measured by *critical dimension* (CD) accuracy, which refers to the size of thin features (e.g., gate length) which are difficult to print reliably. Thus, achieving high CD accuracy means that the printed patterns well match the desired ones. Prevailing RETs include optimal proximity correction, phase shift mask, off-axis illumination, and sub-resolution assist features [1].

RETs are effective in improving CD accuracy. However, increasingly shrinking features on the die and increasing complexity of the design overstretch the capability of RETs. This problem aggravates when RETs are applied to the layouts which are not lithography friendly. Furthermore, RETs often complicate photomark shapes and introduce considerable amount of additional cost to photomask fabrication, which makes RETs expensive to apply. To attack the above issues, efforts are needed in all process and design stages. With respect to physical design, manufacturability-aware methodologies would be performed to reduce the burden of manufactures. Our purpose is that with more lithography-friendly layout, the tasks of manufacturers would become significantly easier and RETs become less expensive to apply.

Previous works on RET-aware physical designs include [2] and [3] on routing problems, and regular fabric and restricted design rules on

Manuscript received November 29, 2007; revised November 18, 2008. First published June 30, 2009; current version published May 26, 2010. This work was supported in part by Intel. Software tool Calibre LFD is donated by Mentor Graphics.

S. Hu is with the Department of Electrical and Computer Engineering, Michigan Technological University, Houghton, MI 49931 USA (e-mail: shiyan@mtu.edu).

P. Shah and J. Hu are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: pratikshah505@ece.tamu.edu; jianghu@ece.tamu.edu).

Digital Object Identifier 10.1109/TVLSI.2009.2017268

1063-8210/$26.00 © 2009 IEEE

emerging lithography friendly design methodologies. In contrast to the above, this work focuses on placement and considers to improve the printability of a given placement by various optimizations. The work [4] studies the similar problem. However, it only performs spacing optimization between cells and does not allow changing relative locations and orientations of cells. These turn out to be important to obtain a high-quality lithography-friendly placement as indicated by our experiments. Compared to [5], CD variation but not edge placement error (EPE) is used in this paper since the former better captures the variation effects than the latter. In addition, a more effective pruning technique for flipping algorithm is presented.

In this paper, several manufacturability-driven new cell placement algorithms are proposed. They are dynamic programming based cell flipping algorithm, single row based optimization approach and multiple row-based optimization approach. To measure printability, CD variation is used and it is computed using Calibre LFD which considers OPC effect. To measure perturbation, wirelength increase is used. Our experimental results demonstrate that over 15% CD variation reduction can be obtained in post-OPC stage by the new approaches with only less than 1% additional wire.

II. Preliminaries

Distortions in lithography process can be measured by a generic function *pattern dependent manufacturability cost*, or *manufacturability cost* in short, denoted by $\eta(\cdot)$. $\eta(\cdot)$ is defined on *pattern* which is associated with cells, i.e., a pattern could be cells or part of cells. In this paper, we are interested in the pattern spanning only two horizontally adjacent cells. Denote by C a cell, by C^l the left side of C, and by C^r the right side of C. If a pair of cells C_i, C_j are adjacently placed in a row, a pattern $P\left(C_i^r, C_j^l\right)$ associated to them could refer to the part spanning over C_i^r and C_j^l. For each pattern P, we have a manufacturability cost $\eta(P)$.

This manufacturability cost can refer to many instances such as CD variations, edge placement error (EPE), image log slope (ILS), process window, or combination of the above [1]. CD variation [1] refers to the CD difference between the printed PV Band of the feature and the desired feature CD which can be computed by averaging the CD difference at a number of sample points along the feature length. EPE [1] refers to the difference (absolute value) in placement between the printed PV band and the expected feature which can be computed by averaging the difference at a number of sample points along the feature length. The printability error is smaller with smaller CD variation and EPE. ILS [1] refers to the amount of allowable dose variation. It is inversely proportional to the dose sensitivity of the edge position. An edge can be better printed if its ILS is larger. Process Window [1] refers to a combinational effect of the allowable exposure latitude and the depth-of-focus. The printability error is smaller with larger depth-of-focus and exposure latitude. To illustrate our approaches, CD variation is used as the manufactilibity cost.

It is important to note the following facts. As a cell is associated with an orientation, when the orientation is changed (i.e., cell flipping happens), any pattern associated to this cell is in general also changed. In lithography process, adjacent cells with different spacing can have different printability. This is automatically handled using patterns as the same cell pair with different spacing will be treated as different patterns. Since the printability of a cell weakly depends on cells at other rows [6], it is safe to neglect it. Formally, $\eta(P) = 0$ for any P associated with cells in different rows. This is why we are only interested in pattern associated with horizontally adjacent cells. Given an initial cell placement, our goal is to reduce the manufacturability cost η through postprocessing optimizations. Online evaluation of manufacturability cost for each pattern is time consuming and not necessary. A better idea is to compute $\eta(P)$ for each possible pattern P offline and store them in a lookup table for future usage. Various advantages can be obtained with this. Online lithography simulations, which are very computationally expensive, are avoided. This is a key difference between our approach and those in [2] and [3]. There exists fast lithography simulations [3]. However, as many runs of online simulations have to be performed during circuit optimizations, such approaches can still be improved. As performing a simulation is much slower than obtaining a number in a lookup table, one could use lookup table based optimization for speedup. As the lookup table is built offline, full-fledged expensive lithography simulations can be performed. Further, OPC can be performed to the circuit pattern before computing the manufacturability cost. It is well known that OPC effects are very difficult to model, however, our lookup table-based methodology can easily handle them.

In this paper, CD variation is used as the manufactuability cost to illustrate our approaches. In our experiments, the cell library consists of 22 cells. Our lookup table for manufacturability cost is built as follows. For each pair of cell types, for each possible cell orientation, for each representative spacing between cells, an CD variation is obtained by Mentor Graphics Calibre LFD, which is a commercial tool for performing lithography simulations. Thus, CD variations in our lookup table are accurate in contrast to fast estimation on CD variation. RETs are often performed to improve the printability in practice. To capture this, OPC is performed on the pattern using Calibre LFD before computing the manufacturability cost by simulations. Note that other RETs can also be applied.

As manufacturability cost between cells in different rows is negligible, row-based placement approaches are designed in this paper. Define *row η cost* for a row of cells as the sum of η between all adjacently placed cell pairs, and define the *total η cost* for the whole placement as the sum of all row η costs. Three types of placement perturbation are considered. The first type is to change orientations but not locations of cells, which is called *Cell Flipping*. The second type allows both cell re-location and cell flipping. To introduce small perturbation to the original placement, each cell is only allowed to move within a small range around its original location and when a row is adjusted, all other rows must be fixed. It is called *Single Row Optimization*. The third type is the same as single row optimization except that several neighboring rows are optimized simultaneously, which is called *Multiple Row Optimization*. Our problem is formulated as follows.

Perturbation Constrained Lithography-Driven Cell Placement Problem: Given a cell placement, to perform post-processing optimizations, which can be cell flipping, single row optimizations or multiple row optimizations, such that the total manufacturability cost is reduced subject to the constraint α on wirelength perturbation.

III. Cell Flipping

Cell flipping algorithm works under the dynamic programming framework and is carried out row by row. In a row, the location of each cell is fixed and the orientation of each cell is to be determined. We define a *partial* cell flipping solution to be an incomplete determination for cell orientations. A partial solution becomes *complete* when the orientations of all cells are determined. A cell is *processed* if its orientation has been determined. A set of partial solutions $\mathcal{S}$ keep being updated during dynamic programming. Each solution $S \in \mathcal{S}$ is associated with a (CE, CW) pair, where CE denotes the cumulative η cost for all processed cells, and CW denotes the cumulative wirelength. CW is computed using the widely-used metric *half-perimeter wirelength* (HPWL) on all those nets which do not span on any unprocessed cell.

Suppose that a cell C to the right of the current partial solution S is to be processed, i.e., we are to decide the orientation of C. A new solution S' will be formed for each possible cell insertion (flipped C and

unflipped C). Because all cells are processed according to the topological order, when C is processed, the cumulative η cost can be updated by $CE(S') = CE(S) + \eta\left(P\left(C^r_{\text{last}}, C^l\right)\right)$, where C_{last} is the last processed cell. The cumulative wirelength is updated by recomputing HPWL of all nets not spanning on any unprocessed cell. Note that for a net, CW considers all the cells the net connects, which may be in different rows.

During dynamic programming, some solutions are inferior to others and will be pruned for acceleration. For any two solutions S_1, S_2 *with the same set of processed cells*, we say that S_2 is *inferior to* S_1 if the following two conditions are satisfied. First, for wirelengths, we need $CW(S_2) \geq CW(S_1)$. Second, for manufacturability costs, one has to consider the effect due to the next cell. Denote by $\eta_{\text{next}}(S)$ the manufacturabiliy cost for the pattern formed by the current cell and the next cell (if available). Thus $\eta_{\text{next}} = \eta\left(P\left(C^r, C^l_{\text{next}}\right)\right)$, where C_{next} denotes the next cell. Consider that ith cell is being processed, and the $(i+1)$th cell is first set to be unflipped and then set to be flipped. In both scenarios, if $CE(S_2)+\eta_{\text{next}}(S_2) \geq CE(S_1)+\eta_{\text{next}}(S_1)$ consistently happens, S_1 must be at least as good as S_2 in terms of manufacturability cost. The reason is that given a complete solution achieving the best manufacturability cost constructed from S_2, one can always replace S_2 with S_1 to get a complete solution with no more manufacturability cost. Note that the above pruning technique is better than the one in [5] which only needs $CE(S_2) \geq CE(S_1)$ for manufacturability cost. Since the orientation of the ith cell certainly has impact on the following cell, the technique in [5] could prune potentially better solutions terms of manufacturability.

Whenever a solution becomes inferior, it is pruned from the solution set without further propagation. A solution S can also be pruned when it is infeasible, i.e., its cumulative wirelength is greater than the wirelength constraint of that row. For a row, the wirelength constraint is set to $(1+\alpha)\cdot L$, where L is the total wirelength of the row in the original placement and α is the maximum tolerable wirelength increase ratio.

IV. Single and Multiple Row Optimizations

Algorithmic Overview (Single Row Optimization)

In single row-based optimization, since small perturbation is desired, each cell is only allowed to be movable within a small range around its original position. To approximately implement this strategy, cells will be processed by groups. Every consecutive k cells form a *group* and optimizations (including relocation and cell flipping) are performed inside each group. Note that the groups are formed with the overlap of $k/2$ gates, which allows us to easily handle the boundary cells in groups. Only one group is optimized at a time and groups are optimized successively. To determine which groups to be optimized, a multidimensional descent based optimization approach is used. Such an approach has been successfully used in, e.g., [7] for gate sizing.

At the beginning, a set of groups called IMPROVABLEGROUPS are formed as follows. Each group in the row will be assigned with a *cost* which is equal to the possible η reduction for this group (computed by tentatively optimizing the group) when all other cells are fixed. Note that the cost corresponds to the score as mentioned above. As long as the cost for a group is positive (i.e., η reduction is possible), the group is included into IMPROVABLEGROUPS. We then compute the ratios of η reduction over wirelength increase for all groups in IMPROVABLEGROUPS. A subset of IMPROVABLEGROUPS, called OPTIMIZEDGROUPS, are then computed as an ordered (sorted according to the ratios) set of groups with ratios greater than a threshold as they may provide cumulative improvement in cost with small wirelength increase. Optimizations are then performed to each group in OPTIMIZEDGROUPS in turn. Subsequently, group costs are reevaluated since they are overlapped, new IMPROVABLEGROUPS are formulated, and the above process is repeated until convergence.

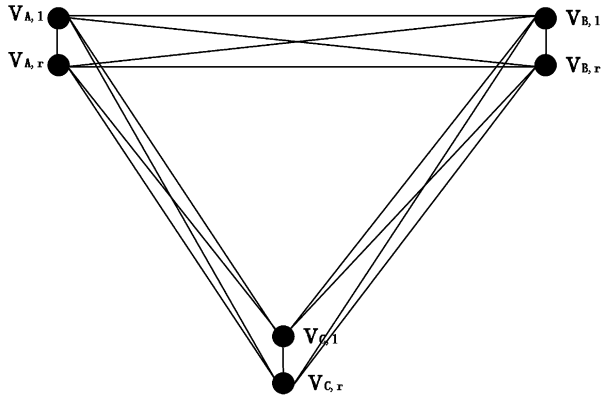

Fig. 1. Graph G corresponding to a placement of three cells A, B, C.

The remaining question is how to compute η reduction for a group of cells. Precisely, our goal is to compute a new cell placement (for this group of cells) with reduced η cost subject to the constraint on wirelength increase. For this, we will first compute the "best η" solution without considering wirelength constraint and then gradually reduce its wirelength.

A. Unconstrained Manufacturability-Driven Placement

When wirelength is not considered, for a group of cells, the cell placement achieving the minimal manufacturability cost can be obtained through reduction to the *minimum cost Hamiltonian path problem*. This formulation allows us to perform simultaneous cell flipping and relocation in group optimizations involved in single and multiple row optimizations. A graph $G = (V, E)$ is to be constructed as follows. Each cell C in the group will be mapped to two nodes $v_{C,l}$ and $v_{C,r}$ in G, where $v_{C,l}$ corresponds to the left side of unflipped C and $v_{C,r}$ corresponds to the right side of unflipped C. There is an edge between $v_{C,l}$ and $v_{C,r}$ with weight 0. For any two nodes v_i and v_j which belong to different cells, there is an edge (v_i, v_j). The weight associated with such an edge is equal to the smallest η cost between v_i and v_j when placing v_i (on the left) and v_j (on the right) adjacently, i.e., $P(v_i, v_j)$. Note that η cost function is in general not monotonic with whitespace [4], and the smallest η can be obtained by our η lookup table.

Refer to Fig. 1 for the graph corresponding to a placement with three cells. Note that weights for edges $(v_{A,l}, v_{A,r})$, $(v_{B,l}, v_{B,r})$, $(v_{C,l}, v_{C,r})$ are all 0. As an example, we show how to compute the weight of edge $(v_{A,l}, v_{B,r})$. Since the weight of an edge refers to the η for the part between the two nodes, we have to first flip cell A and cell B to make the left side of cell A (corresponding to $v_{A,l}$) directly connect to the right side of cell B (corresponding to $v_{B,r}$). Denote the flipped C by C_f. Thus, the weight for $(v_{A,l}, v_{B,r})$ is equal to $\eta\left(P\left(C^l_{A,f}, C^r_{B,f}\right)\right)$.

To compute the optimal η-driven cell placement for this group of cells, it suffices to compute a path visiting each node exactly once such that the total edge weights along the path is minimized. This problem is the *Minimum Cost Hamiltonian Path Problem*. As it is an NP-complete problem [8], the following closest-point heuristic (similar to [8]) is used to compute the efficient approximation. Define *a partial Hamiltonian path* to be an incomplete Hamiltonian path. The algorithm begins with picking an arbitrary node in G. At each step, the node which is not yet included in the partial Hamiltonian path and is closest to any point along the partial Hamiltonian path is identified. Denote this node by u and suppose that it is closest to v along the partial path. We will insert

TABLE I
PERFORMANCE OF EACH ALGORITHM ON ISCAS'89 BENCHMARK CIRCUITS

Circuit		Cell Flipping			Single Row Optimization			Multiple Row Optimization		
Name	#Nodes	Wire Inc.	Variation Red.	CPU (s)	Wire Inc.	Variation Red.	CPU (s)	Wire Inc.	Variation Red.	CPU (s)
s838	317	0.04%	6.7%	0.1	0.06%	10.5%	0.1	0.31%	16.8%	0.2
s1238	439	0.15%	4.2%	0.2	0.18%	9.8%	0.3	0.34%	15.4%	0.5
s1423	511	0.04%	5.4%	0.2	0.20%	10.7%	0.3	0.45%	15.8%	0.5
s1488	429	0.06%	5.6%	0.2	0.38%	10.0%	0.4	0.61%	14.6%	0.5
s5378	1227	0.19%	6.5%	0.7	0.57%	8.3%	0.8	0.72%	12.3%	1.2
s9234	1162	0.18%	5.0%	0.6	0.29%	9.2%	0.9	0.50%	13.6%	1.3
s15850	3621	0.05%	5.9%	2.9	0.31%	11.2%	5.2	0.39%	17.2%	8.1
s35932	13460	0.17%	4.8%	24.3	0.48%	8.5%	58.0	0.73%	14.3%	103.1
s38417	8965	0.32%	5.3%	11.4	0.59%	9.5%	18.9	0.81%	15.7%	39.3
s38584	10463	0.07%	5.8%	14.8	0.11%	9.9%	22.5	0.32%	16.5%	43.5
Average	4059	0.13%	5.5%	5.5	0.32%	9.8%	10.7	0.52%	15.2%	19.8

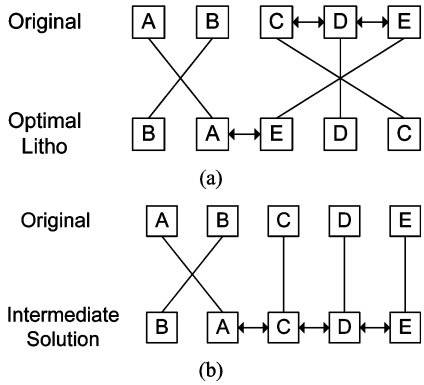

Fig. 2. Obtaining tradeoff between manufacturability cost and wirelength: (a) two initial solutions with best manufacturabiliy cost (Optimal Litho) and best wirelength (Original) and (b) an intermediate solution is obtained by exchanging cells with maximum link crossings (which are C, E in this case). The spacing with the same color refers to the same spacing.

u to the path just after v. Whenever a node is included, another node belonging to the same cell will also be included.

B. Manufacturability-Wirelength Tradeoff

When wirelength is not considered, optimal η-driven placement can be computed as in Section IV-B. This subsection deals with turning an unconstrained solution into a wirelength constrained solution. Let us illustrate the approach using a simple example. Suppose that the original cell placement and optimal η-driven cell placement are as shown in Fig. 2. Our algorithm first identifies an ordered set of cell pairs for location exchange to turn the optimal η-driven placement (best η solution) into the original cell placement (best wire solution). For this, an iterative procedure is used. At each iteration, a pair of cells are identified for location exchange. A cell is first linked to its target location (which is the location in best wire placement) as in Fig. 2. A cell which is not at its target location and is with maximum link crossings is then identified. In case of a tie, an arbitrary cell in tie is identified. Another cell to be exchanged with it is the one at the target location of the identified cell. For example, in Fig. 2, cell E is first identified and it is to exchange location with cell C. The ordered set of exchange cell pairs is $\{EC, BA\}$.

Cell location exchange is performed to the ordered set one by one, where spacings between cells are inherited from the ones before location exchange with the following exception. For placement patterns of cell pairs which can be found in the best wire placement, spacings are set according to it. For example, after E, C are exchanged in Fig. 2, spacing between CD and DE will be set as in original placement and spacing between AC is inherited, as that of AE in optimal η solution. Since optimizations are performed within a group, we need to guarantee that placement of the group will not overlap with its neighboring groups after optimizations. Thus, location exchange terminates when wirelength increase ratio of the placement falls below α and the placement does not overlap with its neighboring groups.

C. Extension to Multiple Row Optimization

When a row is optimized, its neighboring rows might need to be accordingly modified to achieve an overall good design. This is due to the fact that a net often spans a few neighboring rows, and wirelength could be reduced with adjusting several rows simultaneously. As a consequence, some previously "infeasible" placements (i.e., the one violating wirelength constraint) which provide large amount of η reduction may become feasible. This may eventually leads to a solution with more η reduction. The algorithm for single row optimization can be readily extended to handle multiple rows. In multiple row optimization, m neighboring rows will be grouped to form a *row group* and optimizations are performed within each row group. For each row within a row group, cells are grouped as in single row optimization, and approaches in Sections IV-B and IV-C can be directly applied.

V. EXPERIMENTAL RESULTS

All algorithms are implemented in C++ and are tested on a Pentium IV computer with a 2.8 GHz CPU. In performing lithography-driven postprocessing optimizations, the maximum tolerable total wirelength increase ratio α is set to 1% as an example. We first perform experiments on ISCAS'89 benchmark circuits. Logical synthesis and technology mapping (using Berkeley SIS) with a cell library for 65-nm technology, which consists of 22 cells, are performed to the circuits. Our lookup table for manufacturability cost is built as described in Section II. Initial placements are computed using FastPlace [9]. Since it does not handle spacings between cells, we scale each cell to create spacings. CD variation reduction is defined as $1 - (\text{CD variation after optimization}/\text{CD variation before optimization})$.

The results are summarized in Table I. For Cell Flipping, on average about 5% CD variation reduction is obtained with 0.13% additional wire. It runs fastest among all algorithms, which makes sense as the smallest amount of effort is needed there. For Single Row Optimization, on average 9.8% CD variation reduction is obtained, which improves the results by Cell Flipping. The amount of additional wire is still small. On average, only 0.32% more wire is needed. For Multiple Row Optimization, on average 15.2% CD variation reduction is obtained, which gives the best results among all three algorithms. At the same time, it needs more wire. Multiple Row Optimization runs slowest among all three algorithms.

TABLE II
PERFORMANCE OF EACH ALGORITHM ON ISPD'04 BENCHMARK CIRCUITS

	Cell Flipping			Single Row Optimization			Multiple Row Optimization		
Circuit	Wire Inc.	Variation Red.	CPU (s)	Wire Inc.	Variation Red.	CPU (s)	Wire Inc.	Variation Red.	CPU (s)
IBM01	0.02%	5.3%	15.7	0.30%	12.8%	32.8	0.38%	17.8%	55.7
IBM02	0.10%	5.0%	33.5	0.17%	10.9%	81.7	0.32%	18.5%	139.2
IBM03	0.18%	6.1%	51.0	0.35%	11.2%	127.8	0.39%	14.8%	205.7
IBM04	0.12%	5.9%	65.1	0.42%	10.7%	190.5	0.48%	17.3%	274.5
IBM05	0.05%	7.1%	75.5	0.12%	13.4%	215.7	0.15%	15.6%	350.9
IBM06	0.08%	4.8%	98.3	0.25%	9.5%	298.8	0.37%	13.1%	507.8
IBM07	0.10%	5.4%	182.1	0.16%	11.4%	502.2	0.22%	18.2%	983.5
IBM08	0.13%	7.2%	279.8	0.22%	12.1%	557.6	0.46%	16.9%	1378.4
IBM09	0.14%	5.2%	383.0	0.20%	9.9%	832.0	0.50%	12.7%	1518.6
IBM10	0.11%	6.3%	489.2	0.17%	12.3%	1013.5	0.21%	17.1%	1591.2
IBM11	0.20%	6.5%	675.3	0.38%	11.6%	1295.8	0.54%	15.4%	1765.7
IBM12	0.12%	5.6%	820.2	0.27%	10.8%	1775.7	0.31%	12.9%	2031.8
IBM13	0.17%	6.1%	1455.7	0.22%	9.5%	1938.9	0.45%	13.5%	2482.0
IBM14	0.09%	5.8%	2082.7	0.29%	10.5%	3395.1	0.35%	15.0%	4322.3
IBM15	0.11%	4.5%	2305.6	0.37%	9.2%	3724.5	0.49%	12.3%	4506.1
IBM16	0.07%	6.7%	3385.1	0.15%	9.7%	3851.8	0.32%	15.8%	4815.9
IBM17	0.03%	6.5%	3932.0	0.14%	11.0%	4829.1	0.20%	16.7%	6137.5
IBM18	0.05%	5.7%	2970.5	0.12%	11.5%	4521.9	0.18%	15.2%	4880.7
Average	0.10%	5.9%	1072.2	0.24%	11.0%	1570.7	0.35%	15.5%	2108.2

TABLE III
CELL FLIPPING USING PRUNING TECHNIQUE IN [5]

Circuit	Wire Inc.	Variation Red.	CPU (s)
s838	0.03%	4.1%	0.1
s1238	0.10%	3.2%	0.2
s1423	0.04%	5.2%	0.2
s1488	0.07%	5.0%	0.2
s5378	0.15%	5.6%	0.6
s9234	0.12%	3.9%	0.5
s15850	0.05%	5.7%	2.7
s35932	0.22%	3.7%	21.8
s38417	0.30%	4.8%	10.9
s38584	0.05%	5.5%	13.7
Average	0.11%	4.7%	5.1

Since our pruning technique considers the impact to the not-yet-processed cells, it leads to better manufacturability cost than [5], which is clearly demonstrated in Table III.

We next perform experiments on ISPD'04 benchmark circuits [9].[1] They are first placed using FastPlace [9] and then optimized for manufacturability cost. The results are summarized in Table II. Since the original circuits for ISPD'04 benchmark are not known to us, their gate types are randomly assigned. It is clear from Table II that the ratios of CD variation reduction and wirelength increase are similar to those of ISCAS'85 benchmark circuits.

VI. CONCLUSION

Traditionally, design and manufacturing process are separate. This trend should be turned so as to make RETs easy and less expensive to apply. In this paper, cell flipping, single row and multiple row optimization techniques are proposed for manufacturability-driven cell placement perturbation. Our experimental results demonstrate that $> 15\%$ reduction in CD variation can be obtained in post-OPC stage by the new approaches with only $< 1\%$ additional wire.

[1][Online]. Available: http://www.public.iastate.edu/~nataraj/ISPD04_Bench.html

REFERENCES

[1] A.-K. Wong, *Resolution Enhancement Techniques in Optical Lithography*. Bellingham, WA: SPIE Press, 2001.

[2] L.-D. Huang and D. Wong, "Optical proximity correction (opc)-friendly maze routing," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2004, pp. 186–191.

[3] J. Mitra, P. Yu, and D. Pan, "Radar: Ret-aware detailed routing using fast lithography simulations," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2005, pp. 369–372.

[4] P. Gupta, A. Kahng, and C.-H. Park, "Detailed placement for improved depth of focus and CD control," in *Proc. Asia South Pac. Des. Autom. Conf.*, 2005, pp. 343–348.

[5] S. Hu and J. Hu, "Pattern sensitive placement for manufacturability," in *Proc. Int. Symp. Phys. Des.*, 2007, pp. 27–34.

[6] D. M. Pawlowski, L. Deng, and M. D.-F. Wong, "Boundary-based cellwise opc for standard-cell layouts," in *Proc. SPIE, Des. for Manufacturability through Des.-Process Integr.*, 2007, vol. 6521, pp. 65211O–65211O.

[7] O. Coudert, "Gate sizing for constrained delay/power/area optimization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 4, pp. 465–472, Dec. 1997.

[8] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Boston, MA: MIT Press, 2001.

[9] N. Viswanathan and C. C.-N. Chu, "Fastplace: Efficient analytical placement using cell shifting, iterative local refinement and a hybrid net model," in *Proc. Int. Symp. Phys. Des.*, 2004, pp. 26–33.