Minimum Implant Area-Aware Placement and Threshold Voltage Refinement

Wai-Kei Mak, *Member, IEEE*, Wan-Sin Kuo, Shi-Han Zhang, Seong-I Lei, and Chris Chu, *Fellow, IEEE*

***Abstract*—Threshold voltage assignment is a very effective technique to reduce leakage power consumption in modern integrated circuit (IC) design. As feature size continues to decrease, the layout constraints (called MinIA constraints) on the implant area, which determines the threshold voltage of a device, are becoming increasingly difficult to satisfy. It is necessary to take these constraints into consideration during the placement stage. In this paper, we propose to resolve the MinIA constraint violations of a given placement by performing simultaneous detailed placement and threshold voltage refinement. We first present an optimal and efficient mixed integer-linear programming (MILP)-based algorithm to handle intra-row MinIA constraints. We then extend the MILP-based algorithm to handle both inter-row and intra-row MinIA constraints. Experimental results demonstrate that our algorithms only perturb the original placement and threshold voltage assignment solutions minimally to eliminate all violations and are fast in practice.**

I. Introduction

Multiple threshold voltages (multi-V_t) are commonly employed in advanced power-aware high-performance chip design. For example, a chip design can make use of standard cells operating at standard V_t (SVT), low V_t (LVT) and ultra-low V_t (ULVT). Cells operating at lower threshold voltage are faster but more leaky. So, to attain high performance while controlling the leakage power, designers can use cells with lower V_t on critical timing paths and cells with higher V_t on non-critical paths.

There are physical constraints associated with the threshold voltage assignment. During fabrication, local V_t implant areas can be formed at different locations of a chip. The V_t of a placed cell is determined by the ion implantation of the implant area that it belongs to. Manufacturing restrictions for the V_t implant areas led to the minimum width and spacing design rules [1]–[3]. First, each V_t implant area must have a certain minimum width W. Second, two V_t implant areas of the same type must be separated by a certain minimum spacing S. We call these the intra-row minimum implant area (MinIA) constraints. The intra-row MinIA constraints are illustrated in Figure 1, which shows a standard cell row with five standard cells. Cell b is assigned a threshold voltage different from the rest of the cells. Cell b is not wide enough and hence it will cause a minimum implant width violation. Cells d and e belong to two separate V_t implant areas of the same type that are too close to each other. Hence they will cause a minimum implant spacing violation. Note that although the spacing between cells b and c are less than the required minimum implant spacing, they are of different threshold voltage and so will not cause any minimum spacing violation.

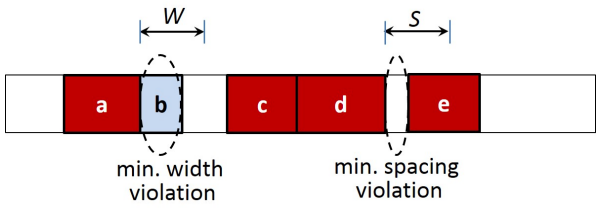

Fig. 1. Intra-row minimum implant width and spacing constraints.

There are many works on multiple threshold voltage assignment for leakage power minimization, e.g., [4]–[8]. However, those works did not consider the MinIA constraints. Consequently, the threshold voltage assignment may result in implant areas violating the MinIA design rules.

Filler cell insertion is a common way to fix intra-row MinIA violations as illustrated in Figure 2. For a narrow V_t-island, if there is sufficient whitespace adjacent to it, then a filler cell of the same implant type can be inserted into the whitespace to enlarge the V_t-island to satisfy the minimum width constraint as shown in Figure 2(a). Two V_t-islands of the same type separated by a small whitespace can be merged into one island by inserting a filler cell of the same implant type into the whitespace to eliminate the minimum spacing constraint violation as shown in Figure 2(b). Filler cell insertion is supported by commercial tools to fix the minimum implant area violation. For instance, Synopsys IC Compiler [9] offers a threshold-voltage-aware filler cell insertion flow which allows the user to define the V_t filler cell to be inserted into each whitespace according to the threshold voltages of the cells on the two sides of the whitespace. For example, the user can specify rules to always insert LVT filler cell in the whitespace between LVT and SVT cells, and always insert ULVT filler cell in the whitespace between ULVT and LVT cells or between ULVT and SVT cells.

In the past, intra-row MinIA constraints are not an important design consideration as intra-row MinIA violations are relatively easy to fix by filler cell insertion. However, as feature size keep diminishing, the critical features are printed by multiple patterning or other expensive processes while the less-critical implant areas are not. Thus the implant areas scale at a slower rate and hence become larger comparing to the feature size. As a result, intra-row MinIA constraints have become

This work was supported in part by the Ministry of Science and Technology under Grant MOST 104-2628-E-007-003-MY3.

Wai-Kei Mak, Wan-Sin Kuo, Shi-Han Zhang, and Seong-I Lei are with the Department of Computer Science, National Tsing Hua University, 101 Kuan Fu Rd. Sec. 2, Hsinchu, Taiwan 300 R.O.C.

Chris Chu is with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50010.

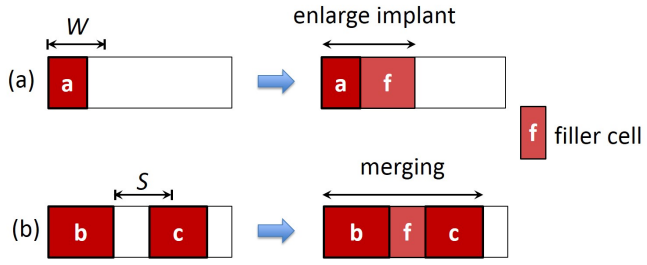

Fig. 2. Filler cell insertion to fix intra-row MinIA violations.

more difficult to satisfy after the 22nm technology node [1]. Filler cell insertion into existing whitespace alone is usually not enough to fix all violations as illustrated in Figure 3. In Figure 3(a), cell a would still violate the minimum implant width constraint after filler cell insertion. In Figure 3(b), cells c and e violates the minimum implant spacing constraint but the implant areas of cells c and e cannot be merged by filler cell insertion as cell d of a different implant type is between them.

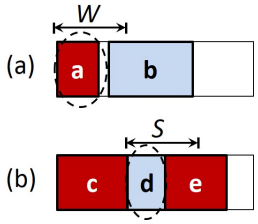

Fig. 3. Examples of intra-row MinIA violations that cannot be fixed by filler cell insertion.

Furthermore, at 10nm node and below it becomes necessary to consider new inter-row minimum implant width constraints [10] as well. Consider the layout in Figure 4 in which two cells b and d with the same V_t on adjacent rows form a narrow staircase. If the width of the abutting region is less than W, the placement is prohibited. With the emergence of both intra-row and inter-row MinIA constraints, it becomes unavoidable to take into account these new placement constraints in order to produce a legal layout.

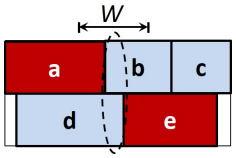

Fig. 4. Inter-row minimum implant width constraint.

In this paper, we consider a problem of refining standard cell placement and threshold voltage to address the manufacturability issue posed by the MinIA design rules for low power IC design in advanced process nodes. We have already pointed out that even if only intra-row MinIA rule violations are targeted, filler cell insertion alone is usually not enough to fix them all. We note that typical placement utilization in a chip is much less than 100% which gives much flexibility in moving the cells within each standard cell row. Shifting the cells by a short distance can resolve both intra-row and inter-row MinIA violations (directly or by facilitating filler cell insertion) with minimal impact on timing and routing congestion. In addition, MinIA rule violations can also be fixed by appropriately re-assigning the threshold voltages of some cells. Doing this may cause a little sacrifice on power but may reduce the cell displacement required. Hence, we propose to fix all MinIA violations by simultaneously performing detailed placement refinement and threshold voltage re-assignment.

For technology nodes in which only intra-row MinIA design rules are concerns, we propose a mixed integer linear programming (MILP)-based algorithm to fix all MinIA constraint violations while minimizing the total displacement of the cells from their original locations and the total power overhead. For technology nodes in which both intra-row and inter-row MinIA design rules need to be handled, we propose an algorithm by extending the MILP-based algorithm. Both of our algorithms guarantee to always fix all MinIA rule violations, and are efficient.

The rest of the paper is organized as follows. We review some related works in Section II. In Section III, we define our optimization problem formally. In Section IV, we present some useful observations. Then in Section V and Section VI, we introduce our mixed integer linear programming based algorithms to handle only intra-row rules and both inter-row/intra-row rules, respectively. The experimental results are reported in Section VII. Finally, we conclude the paper in Section VIII.

II. Previous Works

Tseng et al. [11] and Han et al. [10] tried to address the minimum implant area constraints and some other cell spacing constraints at the detailed placement stage. Tseng et al. [11] considered only intra-row MinIA constraints and proposed a cluster-based detailed placement algorithm which clusters cells with the same V_t first. On the other hand, both inter-row and intra-row MinIA constraints are considered in Han et al. [10] using an integer linear programming (ILP) approach. However, the ILP formulation in [10] utilizes a huge number of integer variables. For a layout with $|C|$ cells and $|S|$ placement sites, the number of integer variables is $O(|C| \cdot |S|)$ in its formulation. So, it has to partition a layout into a large number of small windows of cells and solve them independently in parallel (with 40 threads in [10]), and the process has to be repeated a number of times with a different set of windows each time in order to resolve violations among cells in different windows. Both [10] and [11] did not allow the threshold voltage of a cell to be changed.

On the other hand, post-placement refinement is a popular way to resolve different kinds of manufacturability issues (e.g., [12]–[14]). In particular, to fix intra-row MinIA constraint violations, Kahng and Lee [1] presented a placement refinement heuristic. The heuristic identifies all narrow cells that violate the minimum implant width constraint and tries to fix each by performing the following procedures in order: (1) filler

cell insertion into the existing whitespace, (2) V_t swapping[1], (3) moving neighboring cells to create new whitespace for filler cell insertion, and (4) downsizing neighboring cells to create new whitespace for filler cell insertion. Unfortunately, the iterative heuristic is not optimal and fixing one violation may create a new violation next to it. So, it is not guaranteed that all violations can be fixed at the end. And we note that since their approach only consider forming a larger implant area for a narrow cell with its immediate left and right whitespace/neighbors, the success rate will likely reduce for future technology nodes when average cell size decreases further. In comparison, we propose to apply whitespace re-distribution, filler cell insertion, and V_t re-assignment concurrently which do not have the shortcomings described above.

III. Problem Definition

In this section, we give the formulation of the minimum implant area-aware detailed placement and threshold voltage refinement problem. We assume that a circuit has been placed and power-optimized using conventional methods. So, we have an initial detailed placement of the circuit and the initial threshold voltage assignment of each cell, but there are intra-row and/or inter-row MinIA constraint violations. Our goal is to eliminate all MinIA constraint violations while preserving the quality of the initial solution. We make the following assumption as in [1]. The left or the right end of an implant area is never located within a cell. The *MinIA-aware Detailed Placement and V_t Refinement problem* is formally defined below.

Problem 1 (MinIA-aware Detailed Placement and V_t Refinement)**.** *Given an initial detailed placement of a netlist and an initial threshold voltage assignment of each standard cell, a minimum implant width W, a minimum implant spacing S between same implant type, and the allowable displacement range of each cell from its original location. The problem is to find simultaneously a V_t re-assignment, a legal placement[2], and a filler cell insertion solution such that the total cell displacement and leakage power overhead are as small as possible.*

In order to preserve timing, we restrict the displacement of each cell from its original location. In particular, we assume that there is an allowable displacement range for each cell depending on its timing criticality. A non-timing critical cell may have a large allowable displacement range while a highly timing critical cell may have a zero allowable displacement range. Since the circuit has already been power-optimized so that cells with sufficient timing slack have already been swapped to higher threshold voltage, the V_t of each cell is allowed to be reduced but not raised.

[1] V_t swapping in [1] is limited to changing the V_t of a narrow cell to the V_t of its left/right neighboring cells, or changing the V_t of its left/right neighboring cells to the narrow cell's V_t.

[2] In this paper, a legal placement means that all cells are placed in valid placement sites without cell overlap within the given placement region such that there is no MinIA design rule violation after filler cell insertion.

IV. Useful Observations

Before presenting our proposed approach, we first make some useful observations which we are going to take advantage of. First, the heuristic in [1] always inserts just a single type of filler cell into the whitespace between two regular cells. But we note that a better approach is to divide the whitespace into a left subspace and a right subspace (a subspace can be empty) such that each is inserted with its own desired filler cell type. For example, in Figure 5, inserting just a single filler cell type into the whitespace between cell a and b can fix either the intra-row MinIA width violation on the left or on the right of the whitespace as in Figure 5(b) and Figure 5(c), respectively. However, dividing the whitespace into two subspaces and insert two filler cell types as in Figure 5(d) will fix both intra-row MinIA width violations on the left and on the right at the same time. And it is easy to see that the left (right) subspace should always be inserted with filler cell type matching the implant type of the cell on its left (right). So, we propose a MILP model that will automatically determine the optimal division for each whitespace assuming that the left (right) subspace is always inserted with filler cell type matching the implant type of the cell on its left (right).

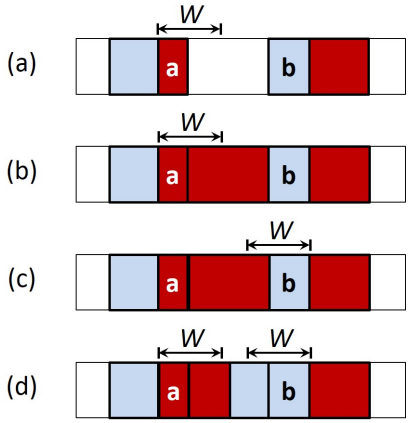

Fig. 5. A better approach of filler cell insertion.

Second, the minimum spacing S is no larger than the minimum width W in practice. So, if a cell row is made up of abutting implant areas all satisfying the minimum width constraint with no empty gap, then the minimum spacing constraint for the same implant type will be automatically satisfied. It is because two implant areas of the same type will have at least one implant area of a different type in between them, hence their separation will be at least $W(\geq S)$. As explained in the paragraph above, our proposed MILP model assumes that all whitespaces in a row will be inserted with filler cells leaving no empty gap in the row, thus it is sufficient to consider minimum implant width constraint only.

V. Handling Intra-row MinIA Design Rules

In this section, we present an algorithm targeting the technology nodes in which intra-row MinIA design rules are applicable but inter-row MinIA design rules are not. First, we

observe that we may solve the MinIA-aware Detailed Placement and V_t Refinement problem by solving a subproblem for each row independently. Then, by combining the optimal solution for each subproblem, we can get the optimal solution for the whole problem. Here, we propose an efficient MILP model for handling the subproblem.

We assume that the cells in row r are indexed from 1 to n_r from the left to the right. The k-th cell in row r is denoted by $C_{r,k}$. We assume that there are three levels of threshold voltages which are SVT, LVT, and ULVT, though our model can be easily extended to any number of threshold voltage levels. Below are the inputs to the MinIA-aware detailed placement and V_t refinement problem.

- n_r: number of cells in row r.
- L_r: length of row r.
- W: minimum implant width.
- $W_{r,k}$: width of cell $C_{r,k}$.
- $X_{r,k}$: original location of cell $C_{r,k}$'s left boundary.
- $\delta_{r,k}^{min}$, $\delta_{r,k}^{max}$: [$\delta_{r,k}^{min}, \delta_{r,k}^{max}$] is the range of allowed displacement for cell $C_{r,k}$ ($\delta_{r,k}^{min} \leq 0$ and $\delta_{r,k}^{max} \geq 0$).
- $\Delta_{r,k}^{S}, \Delta_{r,k}^{L}, \Delta_{r,k}^{UL}$: power penalties if cell $C_{r,k}$'s threshold voltage is re-assigned to SVT, LVT, ULVT, respectively.

The outputs of the MinIA-aware detailed placement and V_t refinement problem are as follows.

- $a_{r,k}^{S}, a_{r,k}^{L}, a_{r,k}^{UL}$: 1 if cell $C_{r,k}$'s threshold voltage is re-assigned to SVT, LVT, ULVT, respectively; 0 otherwise.
- $\delta_{r,k}$: final displacement of cell $C_{r,k}$ from its original location.
- $m_{r,k}$: an intermediate location between the final locations of cells $C_{r,k}$ and $C_{r,k+1}$ that divides the whitespace between cells $C_{r,k}$ and $C_{r,k+1}$ into two subspaces such that the left (right) subspace is inserted with filler cell type matching the implant type of cell $C_{r,k}$ ($C_{r,k+1}$).

Our MILP model for row r is given below.

$$\text{Min. } \alpha \sum_k \left(\Delta_{r,k}^{S} \cdot a_{r,k}^{S} + \Delta_{r,k}^{L} \cdot a_{r,k}^{L} + \Delta_{r,k}^{UL} \cdot a_{r,k}^{UL}\right) + \beta \sum_k \bar{\delta}_{r,k}$$

$$\text{s.t. } a_{r,k}^{S} + a_{r,k}^{L} + a_{r,k}^{UL} = 1 \quad \forall k \tag{1}$$

$$a_{r,k}^{\tau} = 0 \quad \forall k, \tau \text{ s.t. cell } C_{r,k} \text{ cannot be assigned threshold voltage } \tau \tag{2}$$

$$a_{r,k}^{S}, a_{r,k}^{L}, a_{r,k}^{UL} = 0 \text{ or } 1 \quad \forall k \tag{3}$$

$$\delta_{r,k}^{min} \leq \delta_{r,k} \leq \delta_{r,k}^{max} \quad \forall k \tag{4}$$

$$\bar{\delta}_{r,k} \geq \delta_{r,k} \quad \forall k \tag{5}$$

$$\bar{\delta}_{r,k} \geq -\delta_{r,k} \quad \forall k \tag{6}$$

$$X_{r,k} + \delta_{r,k} + W_{r,k} \leq m_{r,k} \leq X_{r,k+1} + \delta_{r,k+1} \quad \forall 1 \leq k \leq n_r - 1 \tag{7}$$

$$0 \leq X_{r,1} + \delta_{r,1} \tag{8}$$

$$X_{r,n_r} + \delta_{r,n_r} + W_{r,n_r} \leq L_r \tag{9}$$

$$d_{r,k} \leq 1 - a_{r,k+1}^{S} + 1 - a_{r,k}^{S} \quad \forall 1 \leq k \leq n_r - 1 \tag{10}$$

$$d_{r,k} \geq a_{r,k+1}^{S} - a_{r,k}^{S} \quad \forall 1 \leq k \leq n_r - 1 \tag{11}$$

$$d_{r,k} \leq 1 - a_{r,k+1}^{L} + 1 - a_{r,k}^{L} \quad \forall 1 \leq k \leq n_r - 1 \tag{12}$$

$$d_{r,k} \geq a_{r,k+1}^{L} - a_{r,k}^{L} \quad \forall 1 \leq k \leq n_r - 1 \tag{13}$$

$$d_{r,k} \leq 1 - a_{r,k+1}^{UL} + 1 - a_{r,k}^{UL} \quad \forall 1 \leq k \leq n_r - 1 \tag{14}$$

$$d_{r,k} \geq a_{r,k+1}^{UL} - a_{r,k}^{UL} \quad \forall 1 \leq k \leq n_r - 1 \tag{15}$$

$$d_{r,k} = 0 \text{ or } 1 \quad \forall 1 \leq k \leq n_r - 1 \tag{16}$$

$$m_{r,k+j} - m_{r,k-1} \geq (d_{r,k-1} + d_{r,k+j} - 1)W \quad \forall 1 \leq k \leq n_r \text{ and } \forall j \geq 0 \text{ s.t. } (k + j \leq n_r \text{ and } \sum_{i=0}^{j} W_{r,k+i} < W) \tag{17}$$

$$d_{r,0} = 1 \tag{18}$$

$$d_{r,n_r} = 1 \tag{19}$$

$$m_{r,0} = 0 \tag{20}$$

$$m_{r,n_r} = L_r \tag{21}$$

The objective of the MILP is to minimize a weighted sum of power overhead and total cell displacement. The power overhead is given by $\sum_k (\Delta_{r,k}^{S} \cdot a_{r,k}^{S} + \Delta_{r,k}^{L} \cdot a_{r,k}^{L} + \Delta_{r,k}^{UL} \cdot a_{r,k}^{UL})$ while the total displacement of all cells is given by $\sum_k \bar{\delta}_{r,k}$. α and β are user-defined weighting constants. Constraints (1)-(3) ensure that each cell is re-assigned to a legal threshold voltage. In particular, we use constraint (2) to forbid any cell to be re-assigned to a threshold voltage higher than its initial threshold voltage to avoid timing violation. Constraint (4) bounds the displacement for each cell based on the given allowed displacement range. As a negative displacement represents a displacement to the left and a positive displacement represents a displacement to the right, we use constraints (5)-(6) to get a tight lower bound for the magnitude of the displacement of cell $C_{r,k}$ denoted by $\bar{\delta}_{r,k}$. Moreover, since $\bar{\delta}_{r,k}$ appears in the minimization objective, it will force $\bar{\delta}_{r,k}$ to be exactly equal to the magnitude of the displacement of cell $C_{r,k}$. Constraint (7) ensures that the cells in the row will not overlap and the division point of the whitespace between any two adjacent cells must be legal, i.e., between the two cells' final locations. Constraints (8)-(9) ensure that the leftmost and the rightmost cells must lie within the given placement region.

We introduce binary variable $d_{r,k}$ which indicates if the pair of adjacent cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages or not. Therefore, we want $d_{r,k}$ to be equal to 1 if and only if cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages. It is accomplished by constraints (10)-(16). If cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages, then the conjunction of constraints (11), (13) and (15) is equivalent to $d_{r,k} \geq 1$ while the conjunction of constraints (10), (12), and (14) is equivalent to $d_{r,k} \leq 1$, hence $d_{r,k}$ will be 1. On the other hand, if cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned the same threshold voltage, then the conjunction of constraints (11), (13) and (15) is equivalent to $d_{r,k} \geq 0$ while the conjunction of constraints (10), (12), and (14) is equivalent to $d_{r,k} \leq 0$, hence $d_{r,k}$ will be 0. If $\sum_{i=0}^{j} W_{k+i} < W$, then it is possible that cells $C_{r,k}$ to $C_{r,k+j}$ would form a V_t-island that violates the minimum width constraint. To ensure that it will never happen, we impose constraint (17) whenever $\sum_{i=0}^{j} W_{k+i} < W$. Note that cells $C_{r,k}$ to $C_{r,k+j}$ will form a V_t-island only if cell $C_{r,k}$'s V_t is different from that of cell $C_{r,k-1}$ (i.e., $d_{r,k-1} = 1$)

and cell $C_{r,k+j}$'s V_t is different from that of cell $C_{r,k+j+1}$ (i.e., $d_{r,k+j} = 1$). In that case, constraint (17) will become $m_{r,k+j} - m_{r,k-1} \geq W$ enforcing that such island is large enough. And constraint (17) is trivially satisfied unless both $d_{r,k-1,r,k}$ and $d_{r,k+j}$ are 1. Finally, constraints (18)-(21) take care of the boundary conditions.

We note that our MILP is guaranteed to be feasible since $a^{UL}_{r,k} = 1$ for all k, $\delta_{r,k} = 0$ for all k, and $m_{r,k} = X_{r,k+1}$ for $k = 1, \ldots, n_r - 1$ is always a feasible solution to it. That is, one way (not the best way) to resolve all intra-row MinIA constraint violations is to re-assign all cells' threshold voltage to ULVT and fill all whitespace with ULVT filler cells without moving any cell. Our approach always tries to find the best feasible solution to fix all intra-row MinIA constraint violations with the minimum power overhead and total cell displacement.

A. A Simpler and Faster MILP Model

In this subsection, we discuss a possible simplification to our proposed MILP. Instead of using binary variable $d_{r,k}$ which indicates if cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages or not, we may introduce binary variable $d'_{r,k}$ such that $d'_{r,k}$ must be 1 if cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages. And we may replace constraints (10)-(19) by constraints (22)-(28) below. If cells $C_{r,k}$ and $C_{r,k+1}$ are re-assigned distinct threshold voltages, then the conjunction of constraints (22), (23) and (24) is equivalent to $d'_{r,k} \geq 1$, hence $d'_{r,k}$ will be 1. Otherwise, the value of $d'_{r,k}$ can be freely set by the MILP solver.

$$d'_{r,k} \geq a^S_{r,k+1} - a^S_{r,k} \quad \forall 1 \leq k \leq n_r - 1 \tag{22}$$

$$d'_{r,k} \geq a^L_{r,k+1} - a^L_{r,k} \quad \forall 1 \leq k \leq n_r - 1 \tag{23}$$

$$d'_{r,k} \geq a^{UL}_{r,k+1} - a^{UL}_{r,k} \quad \forall 1 \leq k \leq n_r - 1 \tag{24}$$

$$d'_{r,k} = 0 \text{ or } 1 \quad \forall 1 \leq k \leq n_r - 1 \tag{25}$$

$$\begin{aligned} m_{r,k+j} - m_{r,k-1} \geq (d'_{r,k-1} + d'_{r,k+j} - 1)W \\ \forall 1 \leq k \leq n_r \text{ and} \\ \forall j \geq 0 \text{ s.t. } (k + j \leq n_r \text{ and} \\ \textstyle\sum_{i=0}^{j} W_{r,k+i} < W) \end{aligned} \tag{26}$$

$$d'_{r,0} = 1 \tag{27}$$

$$d'_{r,n_r} = 1 \tag{28}$$

As explained previously, suppose $\sum_{i=0}^{j} W_{k+i} < W$, then we need to make sure that $m_{r,k+j} - m_{r,k-1} \geq W$ only if cell $C_{r,k}$'s V_t is different from that of cell $C_{r,k-1}$ and cell $C_{r,k+j}$'s V_t is different from that of cell $C_{r,k+j+1}$. But we know that when cell $C_{r,k}$'s V_t is different from that of cell $C_{r,k-1}$ and cell $C_{r,k+j}$'s V_t is different from that of cell $C_{r,k+j+1}$, both $d'_{r,k}$ and $d'_{r,k+j}$ will be equal to 1, so constraint (26) will become $m_{r,k+j} - m_{r,k-1} \geq W$ as desired. Otherwise, the MILP solver is free to set the value of at least one of $d'_{r,k}$ or $d'_{r,k+j}$ to satisfy constraint (26) trivially.

B. Efficiency of Our Model

We note that in the detailed placement legalization method of [10], the intra-row MinIA design rules are enforced by introducing a 0-1 variable between every pair of adjacent placement sites in a row to indicate if the pair have the same V_t or not, and by defining for every W consecutive placement sites in a row a set of W constraints on these variables. On the other hand, we introduce a 0-1 variable $d_{r,k}$ between every pair of adjacent cells in a row, and define one constraint (17) for every chain of consecutive cells with total length less than W. Since the number of cells in a row is at least a few times less than the number of placement sites in a row, we can save a lot on the number of integer variables and constraints.

Conventionally, placement of cells should be left-aligned with placement sites, i.e., $X_{r,k} + \delta_{r,k}$ should be integral. In our model, we simply defined variables $m_{r,k}$ and $\delta_{r,k}$ as continuous variables and do not need to restrict them to be integer variables. This approach dramatically reduces the number of integer variables and hence significantly speeds up the algorithm. If V_t re-assignment is not considered, our model will even become a linear program without any integer variable. We show in the following that an optimal integral (i.e., legal) placement solutions can always be obtained.

Lemma 1. *An integral optimal solutions can always be obtained for our MILP formulation.*

Proof. For our MILP formulation, all the parameters in the constraints $(X_{r,k}, W_{r,k}, W, \delta^{min}_{r,k}, \delta^{max}_{r,k}, L_r)$ are integers. If we fix the binary variables in our MILP, all the constraints are of the form:

$$\begin{cases} A \leq v \leq B \\ C \leq v - v' \leq D \end{cases}$$

where v and v' are continuous variables, and A, B, C, and D are some integer constants. Consider the feasible region, which is a convex polytope formed by these constraints, the vertices of it must be integral. This implies that there exist integral optimal solutions for our MILP formulation. It is also known that an integral optimal solution will always be obtained using a simplex method based algorithm [15]. □

C. Dealing with Fixed Macros

In practice, there can be many fixed macros in the given placement that cannot be moved. Our approach can be easily adapted to handle that. We just need to scan the design once and divide each row of cells into sub-rows separated by the fixed macros. Then we can formulate an MILP for each sub-row with the left end point and the right end point of the sub-row as the boundaries when refining the placement of the cells in the sub-row. Hence, we form and solve an independent MILP for each sub-row.

VI. Handling Both Intra-row and Inter-row MinIA Design Rules

In this section, we present an algorithm targeting the technology nodes in which both intra-row and inter-row MinIA design rules are applicable. For such advanced technology nodes, it is also necessary to ensure that the V_t islands of the same type on adjacent rows will not form any narrow staircase with width less than W. We observe that a narrow staircase of the same V_t can be formed as in Figure 6(a) or

Figure 6(b). In Figure 6(a), the V_t of cells $C_{r,k}$ and $C_{r+1,k'+1}$ are identical and are different from cells $C_{r,k+1}$ and $C_{r+1,k'}$ while $W > m_{r,k} - m_{r+1,k'} > 0$. In Figure 6(b), the V_t of cells $C_{r,k+1}$ and $C_{r+1,k'}$ are identical and are different from cells $C_{r,k}$ and $C_{r+1,k'+1}$ while $W > m_{r+1,k'} - m_{r,k} > 0$. So, an optimal MILP formulation under both intra-row and inter-row MinIA design rules can be obtained by combining the MILPs for all the rows described in the previous section and adding inter-row constraints for the inter-row MinIA design rules.

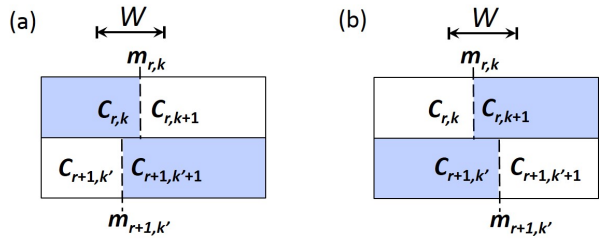

Fig. 6. Formation of narrow staircases.

We describe the additional inter-row constraints below. First, we note that if the final values of $m_{r,k}$ and $m_{r+1,k'}$ are at least W apart, then a narrow staircase like the ones in Figure 6 can never be formed. Now, since the feasible range for variable $m_{r,k}$ is between $X_{r,k} + \delta_{r,k}^{min} + W_{r,k}$ and $X_{r,k+1} + \delta_{r,k+1}^{max}$, we can figure out in advance if the final values of $m_{r,k}$ and $m_{r+1,k'}$ $(k = 1, 2, \ldots, n_r; k' = 1, 2, \ldots, n_{r+1})$ will always be at least W apart. In this way, we can filter out a lot of unnecessary inter-row constraints.

We propose to check the feasible ranges for $m_{r,k}$ and $m_{r+1,k'}$, and consider three different cases. Case 1 is that the feasible range for $m_{r,k}$ and that for $m_{r+1,k'}$ do not overlap and are at least W apart. In this case, we do not need to introduce inter-row constraints between $m_{r,k}$ and $m_{r+1,k'}$. Case 2 is that the feasible range for $m_{r,k}$ does not overlap with that for $m_{r+1,k'}$ but the two ranges are less than W apart. Case 3 is that the feasible range for $m_{r,k}$ overlaps with that for $m_{r+1,k'}$. Next, we describe the inter-row constraints for cases 2 and 3.

For case 2, the feasible range for $m_{r,k}$ does not overlap with that for $m_{r+1,k'}$ but the two ranges are less than W apart. We can distinguish two subcases depending on whether the final value of $m_{r,k}$ will always be greater than or always be smaller than that of $m_{r+1,k'}$. If the final value of $m_{r,k}$ will always be greater than that of $m_{r+1,k'}$, we add constraint (29) below to prevent the scenario shown in Figure 6(a). It is easy to see that unless $a_{r,k}^{\tau} = 1$, $a_{r+1,k'+1}^{\tau} = 1$, $a_{r,k+1}^{\tau} = 0$, and $a_{r+1,k'}^{\tau} = 0$, the right hand side of constraint (29) will be negative and constraint (29) will be trivially satisfied. When $a_{r,k}^{\tau} = 1$, $a_{r+1,k'+1}^{\tau} = 1$, $a_{r,k+1}^{\tau} = 0$, and $a_{r+1,k'}^{\tau} = 0$, which means the V_t of cells $C_{r,k}$ and $C_{r+1,k'+1}$ are identical but are different from cells $C_{r,k+1}$ and $C_{r+1,k'}$, constraint (29) becomes $m_{r,k} - m_{r+1,k'} \geq W$. As a result, the scenario shown in Figure 6(a) cannot occur.

$$m_{r,k} - m_{r+1,k'} \geq (a_{r,k}^{\tau} + a_{r+1,k'+1}^{\tau} - a_{r,k+1}^{\tau} - a_{r+1,k'}^{\tau} - 1)W \quad \text{for } \tau = S/L/UL \tag{29}$$

Similarly, if the final value of $m_{r,k}$ will always be smaller than that of $m_{r+1,k'}$, we add constraint (30) below to ensure that the scenario shown in Figure 6(b) cannot occur.

$$m_{r+1,k'} - m_{r,k} \geq (a_{r,k+1}^{\tau} + a_{r+1,k'}^{\tau} - a_{r,k}^{\tau} - a_{r+1,k'+1}^{\tau} - 1)W \quad \text{for } \tau = S/L/UL \tag{30}$$

For case 3, the feasible range for $m_{r,k}$ overlaps with that for $m_{r+1,k'}$. In this case, the final value of $m_{r,k}$ can be greater than, less than, or equal to that of $m_{r+1,k'}$. If the final values of $m_{r,k}$ and $m_{r+1,k'}$ are equal, neither scenario shown in Figure 6 can occur. But we have to handle the other two possibilities. We introduce binary variable $g_{r,k,r+1,k'}$ such that $g_{r,k,r+1,k'}$ must be 1 if the final value of $m_{r,k}$ is greater than that of $m_{r+1,k'}$. Similarly, we introduce binary variable $l_{r,k,r+1,k'}$ such that $l_{r,k,r+1,k'}$ must be 1 if the final value of $m_{r,k}$ is less than that of $m_{r+1,k'}$. We add the inter-row constraints (31)-(36) below, where M is a large constant.

$$g_{r,k,r+1,k'} = 0 \text{ or } 1 \tag{31}$$

$$m_{r,k} - m_{r+1,k'} \leq M \cdot g_{r,k,r+1,k'} \tag{32}$$

$$m_{r,k} - m_{r+1,k'} \geq (a_{r,k}^{\tau} + a_{r+1,k'+1}^{\tau} - a_{r,k+1}^{\tau} - a_{r+1,k'}^{\tau} - 1)W - M(1 - g_{r,k,r+1,k'}) \quad \text{for } \tau = S/L/UL \tag{33}$$

$$l_{r,k,r+1,k'} = 0 \text{ or } 1 \tag{34}$$

$$m_{r+1,k'} - m_{r,k} \leq M \cdot l_{r,k,r+1,k'} \tag{35}$$

$$m_{r+1,k'} - m_{r,k} \geq (a_{r,k+1}^{\tau} + a_{r+1,k'}^{\tau} - a_{r,k}^{\tau} - a_{r+1,k'+1}^{\tau} - 1)W - M(1 - l_{r,k,r+1,k'}) \quad \text{for } \tau = S/L/UL \tag{36}$$

Due to constraint (32), binary variable $g_{r,k,r+1,k'}$ must be 1 if $m_{r,k}$ is greater than $m_{r+1,k'}$, otherwise it can be set freely by the MILP solver. When $m_{r,k}$ is greater than $m_{r+1,k'}$, constraint (33) will ensure that the scenario shown in Figure 6(a) cannot occur. Otherwise, constraint (33) is trivially satisfied by setting $g_{r,k,r+1,k'}$ to 0. The other subcase where $m_{r,k}$ is less than $m_{r+1,k'}$ is symmetrical, and is captured by constraints (34)-(36) to prevent the scenario shown in Figure 6(b).

A. Speedup Techniques

Unfortunately, modelling the entire layout using a single MILP as described above could still be computationally expensive except for small layouts. So, we use two techniques to speed up the solution process.

Firstly, instead of handling the whole layout at once, we divide the whole layout horizontally into multiple strips. Each strip is a smaller layout consisting of several rows. We formulate a MILP for each strip. To ensure that there is no inter-row MinIA rule violation between two neighboring strips, we process the strips in the order from the topmost strip to the bottom one, and when we formulate the MILP for a strip,

we take into account the the last row in the strip above the current strip and treat it as fixed.

Secondly, we only add inter-row constraints on an as-needed basis. We will start with zero inter-row constraint, only if there are inter-row MinIA rule violations detected, we will introduce inter-row constraints between the corresponding variables $m_{r,k}$ and $m_{r+1,k'}$ according to whether it belongs to case 2 or case 3 above. For example, suppose we find a narrow staircase of the same V_t like the one in Figure 6(a). Furthermore, the feasible range for $m_{r,k}$ overlaps with that for $m_{r+1,k'}$, then it is case 3. And since it is a narrow staircase of the type shown in Figure 6(a), we will add constraints (31)-(33) to get rid of it. And in particular, we may instantiate constraint (33) with τ being the current V_t of cells $C_{r,k}$ and $C_{r+1,k'}$. This may require a few iterations before a solution with no inter-row MinIA rule violation is obtained. The process is summarized in Figure 7. Note that due to this technique, the number of 0-1 integer variables in our MILP is only $O(|C|)$ where $|C|$ is the number of cells in a strip.

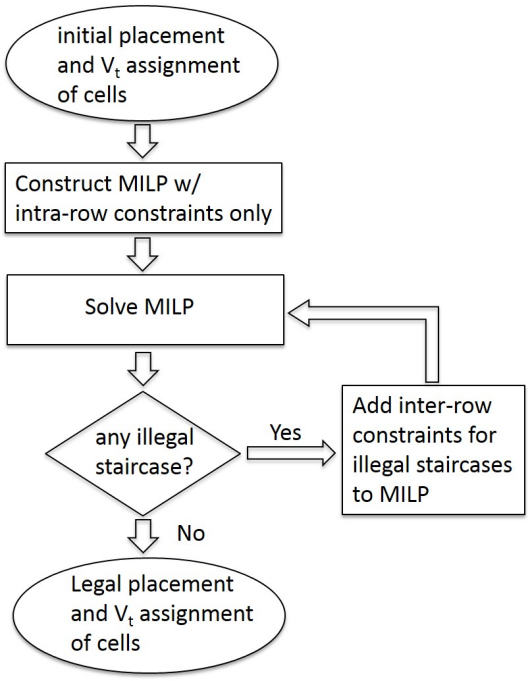

Fig. 7. The proposed technique to add inter-row constraints to our MILP on an as-needed basis.

VII. Experimental Results

We adopt the circuits from the ICCAD 2012 placement contest benchmark suite for our experiments. We ran a routability-driven placer [16] to obtain the initial placement for each circuit. Some key characteristics of the benchmarks are listed in Table I. Percentage of narrow cells denotes the proportion of cells with width less than the minimum implant width which is seven placement sites in our experiments. Besides, the ratio of cells assigned to threshold voltage SVT/LVT/ULVT initially is roughly 7:2:1. Cells assigned to SVT, LVT, and ULVT initially are assumed to be non-timing-critical, more timing-critical, and most timing- critical, respectively. We set the allowable displacement for a cell according to its timing criticality. The allowable displacement for a cell varies from zero to ten placement sites. In each benchmark, there are a number of fixed nodes (unmovable macros and unmovable cells) given. In our experiments, we assume that the unmovable macros have unknown V_t while the V_t of unmovable cells are generated in the same manner as the movable cells. Since the benchmarks did not come with any power information, we make the following assumption when computing the power penalty. The power penalty of changing the threshold voltage of a cell is proportional to the width of the cell. If the threshold voltage of a cell with width W_c is changed from SVT to LVT, or LVT to ULVT, or SVT to ULVT, then the incurred power penalty is $2W_c$, or $3W_c$, or $5W_c$.

TABLE I
Benchmarks Characteristics.

	#Cells	Utilization (%)	% of narrow cells
superblue1	817K	69	65
superblue3	887K	73	43
superblue4	561K	70	51
superblue5	750K	77	45
superblue7	1.33M	76	33
superblue10	1.14M	70	62
superblue16	680K	69	58
superblue18	467K	67	38

The number of intra-row and inter-row MinIA constraint violations for the initial placement of each benchmark are shown in Table II. Unless otherwise stated, the simpler MILP model in Section V.A was used to obtain the refinement results reported in this section. All experiments were conducted on a 3.3GHz Linux machine with 128GB memory. Gurobi [17] was used to solve the MILPs.

TABLE II
Number of intra-row and inter-row violations in the initial placements.

	# intra-row vio.	# inter-row vio.
superblue1	251597	86600
superblue3	208303	32246
superblue4	145748	40459
superblue5	209445	27020
superblue7	252163	160217
superblue10	409542	81473
superblue16	171005	18245
superblue18	100791	32647

In our first set of experiments, we assume that only intra-row MinIA design rules are applicable. As a baseline for comparison, we found the minimum power overhead to fix all intra-row MinIA constraint violations by optimal threshold voltage re-assignment and filler cell insertion without any placement perturbation. The baseline results were obtained by imposing a zero allowable displacement range for all cells in our MILP. Note that it is equivalent to setting $\alpha = 1$ and $\beta = \infty$. The results are shown in the left part of Table III. It can be seen that none of the circuits can satisfy the intra-row MinIA design rules by applying filler cell insertion alone even with the intelligent division of each whitespace for filler cell insertion presented in Section IV. Moreover, we have to

reduce the threshold voltage for as much as 17% of the cells to fix all intra-row MinIA constraint violations for superblue1 which has the highest proportion of narrow cells.

Then we tried different settings for the weighting constants α and β in the objective function of our MILP. Firstly, we tried $\alpha = 1$ and $\beta = 0$, which means we do not try to minimize the total cell displacement as long as each cell is moved within its allowable range. The results are shown in the middle part of Table III. The average and maximum displacement per cell in terms of number of placement sites are reported. For reference, the minimum cell width in all designs is two placement sites. It can be seen that cell movement, or equivalently, whitespace re-distribution can facilitate filler cell insertion to fix intra-row MinIA constraint violation, so the amount of cells that has to change threshold voltage is cut down largely compared to the baseline. As a result, the power penalty associated with threshold voltage reduction is also reduced by 77% on average compared to the baseline. Secondly, we adjusted the values of α and β to minimize the power overhead and total cell displacement at the same time. The results are shown in the right part of Table III. It can be seen that cell displacement can be effectively reduced by over 15 times on average while maintaining virtually the same power overhead reduction.

For comparison, we also implemented a simple algorithm similar to the one proposed in [1] to insert filler cells, change threshold voltages of cells, and move cells heuristically to fix intra-row MinIA constraint violations. The results are reported in Table IV. We found that due to the limited local view of the heuristic, it failed to resolve all violations at the end for all benchmarks.

TABLE IV
RESULTS BY A HEURISTIC SIMILAR TO [1] FOR FIXING INTRA-ROW MINIA CONSTRAINT VIOLATIONS.

	Intra-row vio. left	V_t(S/L/UL) (%)	Avg. disp.	Max. disp.	CPU time (s)
superblue1	27660	57.8 / 24.4 / 17.8	0.08	4	12
superblue3	22597	61.5 / 23.1 / 15.3	0.06	5	10
superblue4	15385	59.8 / 23.7 / 16.5	0.06	5	5
superblue5	28759	62.9 / 23.5 / 13.6	0.08	5	8
superblue7	16864	62.6 / 22.9 / 14.5	0.03	5	28
superblue10	60437	60.0 / 23.8 / 16.2	0.12	5	26
superblue16	16422	58.3 / 24.4 / 17.3	0.07	5	10
superblue18	10056	62.5 / 23.0 / 14.5	0.05	5	7
average	24772	60.7 / 23.6 / 15.7	0.07	4.9	13

In our second set of experiments, we consider both intra-row MinIA and inter-row MinIA design rules. The extended MILP-based approach was applied such that every benchmark was divided into strips of ten rows each. For ease of comparison, we have duplicated the results under intra-row MinIA rules only from the rightmost five columns of Table III into Table V. In order to address inter-row MinIA design rules, both the power penalty and cell displacement are increased. The runtime is increased a few times because it usually takes a few iterations of MILP to resolve all inter-row MinIA rule violations.

In Table VI, we show the speedup of adopting the simplified MILP model proposed in Section V.A over the non-simplified MILP model by repeating the experiment above using the non-simplified MILP model. The runtime is about 2.79 times faster on average with the simplification.

TABLE VI
RUNTIME REDUCTION WITH THE PROPOSED MILP SIMPLIFICATION.

	Non-simplified MILP	Simplified MILP
	CPU time (s)	CPU time (s)
superblue1	7481	1789
superblue3	5322	1290
superblue4	1456	722
superblue5	1258	739
superblue7	5069	1780
superblue10	3559	1506
superblue16	4651	1347
superblue18	782	491
ratio	2.79	1

In addition, we also implemented an alternative MILP formulation for detailed placement and V_t refinement using the constraint modeling method for intra-row and inter-row MinIA rules presented in [10]. We compared the MILP file sizes and runtimes of the alternative MILP formulation against ours for a single strip consisting of the first ten rows of each benchmark[3] in Table VII. We set a timeout limit of 5000s for a single strip. It can be seen that our MILP file sizes are two to three orders of magnitude smaller than the reference MILP formulation. The time it took to solve our MILP was also up to thousands of times faster. In fact, timeout occurred for most benchmarks with the alternative MILP formulation. For the sub-problem from superblue1, our machine ran out of memory when solving the alternative MILP formulation. We verified that the two MILP formulations gave the same objective function values when timeout did not occur.

VIII. CONCLUSIONS

MinIA constraints have become more difficult to satisfy as average cell size continues to decrease in advanced processes. We considered a MinIA-aware detailed placement and threshold voltage refinement problem for low power chip design utilizing multiple threshold voltages. An optimal row-based algorithm based on mixed-integer linear programming was proposed when only intra-row MinIA constraints are of concern. We also extended it to handle both intra-row and inter-row MinIA constraints. Cell movement, intelligent whitespace division for filler cell insertion, and threshold voltage re-assignment are considered simultaneously in our algorithms. Experimental results showed that with limited total cell displacement, the power overhead for fixing all MinIA rule violations can be reduced significantly.

REFERENCES

[1] A.B. Kahng and H. Lee. Minimum implant area-aware gate sizing and placement. In *Proc. ACM Great Lakes Symposium on VLSI*, pages 57–62, 2014.

[2] A.B. Kahng. Lithography-induced limits to scaling of design quality. In *Proceedings of SPIE Advanced Lithography*, pages 905302–905302. International Society for Optics and Photonics, 2014.

[3] We note that in [10], MILP was applied to solve detailed placement sub-problems each involving a much smaller region.

TABLE III
POWER PENALTY AND DISPLACEMENT FOR FIXING ALL INTRA-ROW MINIA CONSTRAINT VIOLATIONS BY OUR APPROACH UNDER DIFFERENT SETTINGS.

	Placement perturbation disallowed ($\alpha = 1, \beta = \infty$)					Placement perturbation not optimized ($\alpha = 1, \beta = 0$)					Balanced placement perturbation ($\alpha = 1, \beta = 0.1$)				
	V_t(S/L/UL) (%)	Power penalty	Avg. disp.	Max. disp.	CPU time (s)	V_t(S/L/UL) (%)	Power penalty	Avg. disp.	Max. disp.	CPU time (s)	V_t(S/L/UL) (%)	Power penalty	Avg. disp.	Max. disp.	CPU time (s)
superblue1	53.2 / 28.2 / 18.6	1	0	0	149	65.0 / 22.7 / 12.3	0.22	2.19	10	130	65.0 / 22.7 / 12.3	0.22	0.19	9	154
superblue3	59.0 / 25.5 / 15.5	1	0	0	129	66.7 / 21.8 / 11.5	0.22	2.33	10	106	66.7 / 21.7 / 11.5	0.22	0.13	8	126
superblue4	56.7 / 26.8 / 16.5	1	0	0	96	65.6 / 22.4 / 12.0	0.26	2.07	10	79	65.6 / 22.4 / 12.0	0.26	0.15	9	88
superblue5	59.5 / 25.0 / 15.5	1	0	0	129	66.9 / 21.7 / 11.4	0.22	2.16	10	105	66.9 / 21.7 / 11.4	0.22	0.12	8	103
superblue7	60.5 / 24.8 / 14.7	1	0	0	196	66.6 / 21.8 / 11.6	0.27	2.06	10	179	66.6 / 21.8 / 11.6	0.27	0.12	9	184
superblue10	58.3 / 25.7 / 16.0	1	0	0	199	67.1 / 21.6 / 11.3	0.17	2.59	10	148	67.1 / 21.7 / 11.2	0.17	0.16	9	178
superblue16	53.8 / 28.0 / 18.2	1	0	0	127	64.9 / 22.7 / 12.4	0.24	2.38	10	112	64.9 / 22.7 / 12.4	0.24	0.20	9	118
superblue18	61.0 / 24.5 / 14.5	1	0	0	76	67.2 / 21.5 / 11.3	0.23	2.46	10	72	67.2 / 21.5 / 11.3	0.23	0.12	8	70
average	57.7 / 26.1 / 16.2	1	0	0	138	66.3 / 22.0 / 11.7	0.23	2.28	10	117	66.3 / 22.0 / 11.7	0.23	0.15	8.6	128

TABLE V
ADDRESSING INTRA-ROW AND INTER-ROW MINIA DESIGN RULES BY OUR APPROACH.

	Satisfy intra-row MinIA rules only ($\alpha = 1, \beta = 0.1$)							Satisfy inter- and intra-row MinIA rules ($\alpha = 1, \beta = 0.1$)						
	Intra-row violations	Inter-row violations	V_t(S/L/UL) (%)	Power penalty	Avg. disp.	Max. disp.	CPU time (s)	Intra-row violations	Inter-row violations	V_t(S/L/UL) (%)	Power penalty	Avg. disp.	Max. disp.	CPU time (s)
superblue1	0	49965	65.0 / 22.7 / 12.3	0.22	0.19	9	154	0	0	63.4 / 23.9 / 12.7	0.30	0.24	10	1789
superblue3	0	49947	66.7 / 21.7 / 11.5	0.22	0.13	8	126	0	0	65.5 / 22.7 / 11.8	0.32	0.19	10	1290
superblue4	0	29668	65.6 / 22.4 / 12.0	0.26	0.15	9	88	0	0	64.1 / 23.5 / 12.4	0.37	0.20	10	722
superblue5	0	35398	66.9 / 21.7 / 11.4	0.22	0.12	8	103	0	0	65.8 / 22.5 / 11.7	0.30	0.17	10	739
superblue7	0	71813	66.6 / 21.8 / 11.6	0.27	0.12	9	184	0	0	65.1 / 22.9 / 12.0	0.42	0.19	10	1780
superblue10	0	60356	67.1 / 21.7 / 11.2	0.17	0.16	9	178	0	0	66.2 / 22.3 / 11.5	0.24	0.21	10	1506
superblue16	0	40582	64.9 / 22.7 / 12.4	0.24	0.20	9	118	0	0	63.1 / 24.0 / 12.9	0.33	0.26	10	1347
superblue18	0	21880	67.2 / 21.5 / 11.3	0.23	0.12	8	70	0	0	66.2 / 22.2 / 11.6	0.34	0.17	10	491
average	0	44951	66.3 / 22.0 / 11.7	0.23	0.15	8.6	128	0	0	64.9 / 23.0 / 12.1	0.33	0.20	10	1208

TABLE VII
MILP FILE SIZE AND RUNTIME COMPARISONS FOR A SINGLE SUBPROBLEM CONSISTING OF TEN PLACEMENT ROWS.

	Model after [1] ($\alpha = 1, \beta = 0.1$)		Our model ($\alpha = 1, \beta = 0.1$)	
	ILP size (MB)	CPU time (s)	ILP size (MB)	CPU time (s)
superblue1	9800	out of mem.	6.8	12.1
superblue3	1420	>5000	1.5	2.5
superblue4	1560	>5000	2.0	4.1
superblue5	2880	>5000	2.2	1.8
superblue7	4930	>5000	3.3	2.7
superblue10	4100	>5000	3.1	6.0
superblue16	60	61	0.3	1.2
superblue18	120	598	0.3	0.3
average	3109	-	2.44	3.8

[3] A.B. Kahng. New game, new goal posts: A recent history of timing closure. In *Proc. Design Automation Conference*, 2015.

[4] Vijay Sundararajan and Keshab K Parhi. Low power synthesis of dual threshold voltage CMOS VLSI circuits. In *Proc. International Symposium on Low Power Electronics and Design*, pages 139–144. ACM, 1999.

[5] Pankaj Pant, Rabindra K Roy, and A Chattejee. Dual-threshold voltage assignment with transistor sizing for low power CMOS circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 9(2):390–394, 2001.

[6] Mahesh Ketkar and Sachin S Sapatnekar. Standby power optimization via transistor sizing and dual threshold voltage assignment. In *Proc. International Conference on Computer Aided Design*, pages 375–378. IEEE, 2002.

[7] Yifang Liu and Jiang Hu. A new algorithm for simultaneous gate sizing and threshold voltage assignment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 29(2):223–234, 2010.

[8] G. Flach, T. Reimann, and G. Posser. Effective method for simultaneous gate sizing and v th assignment using lagrangian relaxation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(4):546–557, 2014.

[9] Synopsys IC Compiler user guide. http://www.synopsys.com/Tools/Implementation/PhysicalImplementation/Pages/ICCompiler.aspx.

[10] K. Han, A.B. Kahng, and H. Lee. Scalable detailed placement legalization for complex sub-14nm constraints. In *Proc. International Conference on Computer Aided Design*, pages 867–873, 2015.

[11] K.-H. Tseng, Y.-W. Chang, and C.C.C. Liu. Minimum-implant-area-aware detailed placement with spacing constraints. In *Proc. of Design Automation conference*, 2016.

[12] P. Gupta, A.B. Kahng, and C.H. Park. Detailed placement for improved depth of focus and CD constrol. In *Proc. of Asia and South Pacific Design Automation Conference*, pages 343–348, 2005.

[13] H. Tian, Y. Du, H. Zhang, Z. Xiao, and D.F. Wong. Triple patterning aware detailed placement with constrained pattern assignment. In *Proc. of International Conference on Computer Aided Design*, pages 116–123, 2014.

[14] T. Lin and C. Chu. TPL-aware displacement-driven detailed placement refinement with coloring constraints. In *Proc. of International Symposium on Phyiscal Design*, pages 75–80, 2015.

[15] G.B. Dantzig. *Linear Programming and Extensions*. Princeton University, Princeton, 1963.

[16] Tao Lin and Chris Chu. POLAR 2.0: An effective routability-driven placer. In *Proc. of Design Automation Conference*. IEEE, 2014.

[17] Gurobi. http://www.gurobi.com.

PLACE PHOTO HERE

Wai-Kei Mak Wai-Kei Mak received the B.S. degree from the University of Hong Kong, Hong Kong, in 1993, and the M.S. and Ph.D. degrees from the University of Texas at Austin, U.S., in 1995 and 1998, respectively, all in computer science.

Dr. Mak is a Professor with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. He was an Assistant Professor with the Department of Computer Science and Engineering, University of South Florida, U.S., from 1999 to 2003. His current research interests include VLSI physical design automation, and CAD for field-programmable technologies.

Dr. Mak is a recipient of the IEEE/ACM Asia and South Pacific Design Automation Conference 2014 Best Paper Award for his work in e-beam lithography throughput optimization. His lab won the first place at the FPT 2008 Logic Block Clustering Contest, the third place at the IEEE CEDA PATMOS 2011 Timing Analysis Contest, and the second place at the TAU 2013 Variation-Aware Timing Analysis Contest. He has served on the Program and/or the Organizing Committee of Asia South Pacific Design Automation Conference, the International Conference on Field Programmable Logic and Applications, and the International Conference on Field-Programmable Technology (FPT). He was the Technical Program Chair of FPT in 2006 and was the General Chair of the same conference in 2008. Since 2009, he has been a Steering Committee Member of the International Conference on Field-Programmable Technology.

PLACE PHOTO HERE

Wan-Sin Kuo Wan-Sin Kuo received the B.S. degree in information management from National Central University, Taoyuan, Taiwan, in 2015, and she is currently pursuing the M.S. degree in National Tsing Hua University, Hsinchu, Taiwan.

Her current research interests include very large scale integrated physical design automation and electronic design automation application on FPGA routing optimization.

PLACE PHOTO HERE

Shi-Han Zhang Shi-Han Zhang received the B.S. degree in information management from National Central University, Taoyuan, Taiwan, in 2015, and he is currently pursuing the M.S. degree in National Tsing Hua University, Hsinchu, Taiwan.

His current research interests include very large scale integrated physical design automation and electronic design automation application on placement.

PLACE PHOTO HERE

Seong-I Lei Seong-I Lei received the B.S., M.S. and Ph.D. degrees in computer science from National Tsing Hua University, Hsinchu, Taiwan, in 2006, 2008 and 2015, respectively.

He is currently a Software R&D Engineer with AnaGlobe Technology Company, Hsinchu, Taiwan. His current research interests include analog routing and layout decomposition for multiple patterning lithography.

PLACE PHOTO HERE

Chris Chu Chris Chu received the B.S. degree in computer science from the University of Hong Kong, Hong Kong, in 1993. He received the M.S. degree and the Ph.D. degree in computer science from the University of Texas at Austin in 1994 and 1999, respectively.

Dr. Chu is a Professor in the Electrical and Computer Engineering Department at Iowa State University. His area of expertises include CAD of VLSI physical design, and design and analysis of algorithms.

Dr. Chu is currently an associate editor for IEEE TCAD. He has served on the technical program committees of several major conferences including DAC, ICCAD, ISPD, ISCAS, DATE, ASP-DAC, and SLIP.

Dr. Chu received the IEEE TCAD best paper award at 1999 for his work in performance-driven interconnect optimization. He received another IEEE TCAD best paper award at 2010 for his work in routing tree construction. He received the ISPD best paper award at 2004 for his work in efficient placement algorithm. He received another ISPD best paper award at 2012 for his work in floorplan block shaping algorithm. He received the ASPDAC best paper award at 2014 for his work in stencil design for electron-beam lithography. He received the Bert Kay Best Dissertation Award for 1998-1999 from the Department of Computer Sciences in the University of Texas at Austin. He is a Fellow of IEEE.