Optimization of Standard Cell Based Detailed Placement for 16 nm FinFET Process

Yuelin Du and Martin D. F. Wong
Dept. of ECE, University of Illinois at Urbana-Champaign
Email: du6@illinois.edu, mdfwong@illinois.edu

***Abstract*—FinFET transistors have great advantages over traditional planar MOSFET transistors in high performance and low power applications. Major foundries are adopting the FinFET technology for CMOS semiconductor device fabrication in the 16 nm technology node and beyond. Edge device degradation is among the major challenges for the FinFET process. To avoid such degradation, dummy gates are needed on device edges, and the dummy gates have to be tied to power rails in order not to introduce unconnected parasitic transistors. This requires that each dummy gate must abut at least one source node after standard cell placement. If the drain nodes at two adjacent cell boundaries abut each other, additional source nodes must be inserted in between for dummy gate power tying, which costs more placement area. Usually there is some flexibility during detailed placement to horizontally flip the cells or switch the positions of adjacent cells, which has little impact on the global placement objectives, such as timing conditions and net congestion. This paper proposes a detailed placement optimization strategy for the standard cell based designs. By flipping a subset of cells in a standard cell row and switching pairs of adjacent cells, the number of drain to drain abutments between adjacent cell boundaries can be optimally minimized, which saves additional source node insertion and reduces the length of the standard cell row. In addition, the proposed graph model can be easily modified to consider more complicated design rules. The experimental results show that the optimization of $100k$ cells is completed within 0.1 second, verifying the efficiency of the proposed algorithm.**

I. Introduction

In the sub-20 nm technology nodes, fin based multiple-gate field-effect transistors (FinFET) show great advantages over traditional planar MOSFET transistors in high performance and low power applications [1], [2]. Unlike a planar MOSFET, the FinFET employs a vertical fin-like structure protruding from the substrate with the gate wrapping around the sides and top of the fin, thereby producing transistors with low leakage currents and fast switching performance. Major foundries are adopting FinFET technology for advanced node fabrication. Recently Taiwan Semiconductor Manufacturing Co. (TSMC) announced their plans for initial production of its 16 nm FinFET process around the end of 2013 [9].

Despite the excellent control of short channel effects [3], [11], FinFETs also suffer from various challenges, such as high parasitic capacitance, high parasitic resistance and edge device degradation [8]. Edge device degradation was already observed with planar process [5], and is even more severe with 3D fin structure. As illustrated in Fig. 1, the fin stress increased by dummy gate removal leads to defect formation [4], and such defects may induce high resistance or capacitance, which degrades the device performance. In contrast, as long as the dummy gates are in place, the fin stress becomes fairly uniform [10], indicating the necessity of keeping the dummy gates.

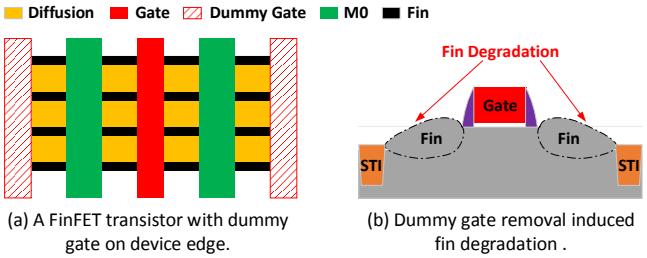

Fig. 1. The edge device degradation induced by dummy gate removal.

However, dummy gates introduce parasitic edge devices, which may potentially increase leakage power or even cause logic failures if not dealt with carefully. Figure 2 shows an example of two FinFET transistors abutting each other. The parasitic transistor introduced by the shared dummy gate and its schematic view are illustrated in Fig. 2(a) and Fig. 2(b) respectively. If the dummy gate inside the red circle is left unconnected, there will be large leakage between the drain node of the left transistor and the source node of the right transistor. In the worst case, the left drain is directly connected to the right source, resulting in logic failures. One straightforward solution to this is tying such dummy gates to power rails, i.e., the dummy gates of a PFET should be tied up to power supply and the dummy gates of an NFET should be tied down to ground.

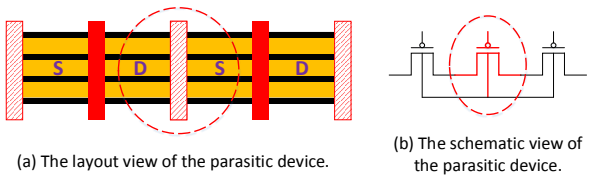

Fig. 2. A parasitic transistor is introduced by two FinFETs abutting each other.

In the 16 nm technology node circuit design, the local

This work was partially supported by the National Science Foundation under grant CCF-1017516 and a grant from the Semiconductor Research Corporation (SRC).
978-3-9815370-2-4/DATE14/©2014 EDAA

interconnect (LI) layer (or metal 0 layer) is used to connect active nodes (i.e., source and drain), and the direction of the LI patterns is perpendicular to the fins. Thanks to the LI layer, a dummy gate can be easily tied to power rails as long as it abuts a source node, as illustrated in Fig. 3(a). However, it is difficult to route a dummy gate to a non-adjacent source node due to limited cell level routing resources. As a result, during the standard cell placement, whenever two drain nodes are placed abutting each other, the dummy gates of the two individual cells cannot merge into one, and additional source nodes must be inserted to tie the dummy gates to power rails, as illustrated in Fig. 3(b).

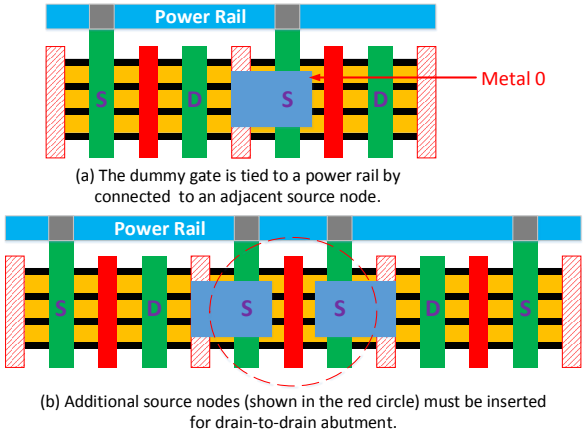

Fig. 3. A dummy gate must abut a source node in order to be tied to power rails.

In the standard cell based design, all cells in a standard cell library are of the same height and each cell is considered as a small block for higher level placement and routing, where only the input/output pins are visible to the placer and router. By performing global placement and legalization, all cells are packed into standard cell rows, each with thousands of cells. Based on the pin locations and an input netlist, the global placer tries to optimize certain performance objectives, such as timing, net congestion, etc. [7]. Since the detailed layout information (e.g., the active node types at cell boundaries) is usually hidden from the global placer, it is very challenging to consider the source/drain abutment constraint during the global placement. Usually detailed placement is performed after global placement and legalization are completed, where there is some flexibility flipping a cell horizontally or switching the positions between two adjacent cells, which has little impact on either timing status or net congestion. However, by properly flipping a subset of cells on each standard cell row and switching pairs of adjacent cells, the number of drain-to-drain (D2D) abutments can be minimized, which saves the area of additional source nodes inserted for the purpose of dummy gate power tying. Note that a D2D abutment exists between two adjacent cells if either the P-diffusions or the N-diffusions have a D2D abutment situation. Figure 4 shows a demo of the placement optimization. In Fig. 4(a), additional columns of source nodes are needed between both pairs of adjacent cells for dummy gate power tying. However, by horizontally flipping cell C and switching cell B and cell C, no D2D abutment exists any more, and consequently, no additional source nodes are needed for the optimized placement shown in Fig. 4(b). By this means, the total length of the standard cell row can be minimized.

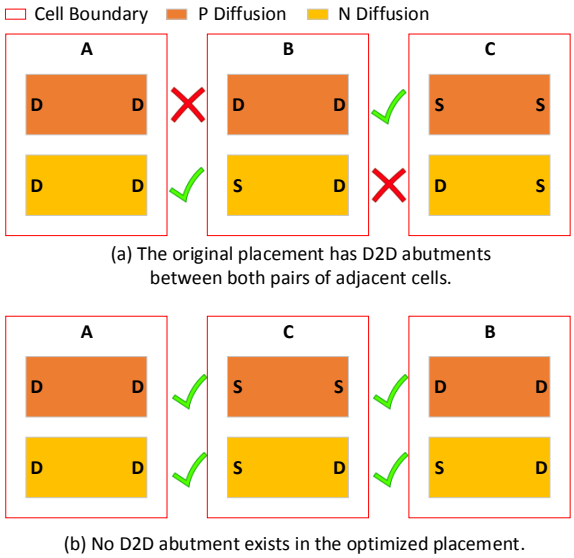

Fig. 4. D2D abutments are removed via placement optimization. Note that only the diffusion layers and the cell boundaries are displayed.

In this paper, we propose a detailed placement optimization algorithm to minimize the number of D2D abutments in a standard cell row, which saves unnecessary source nodes for dummy gate power tying and minimizes the placement area for the 16 nm FinFET technology. As far as we know, this is the first work on detailed placement optimization for the FinFET process. Our algorithm is able to handle cell flipping and cell switching simultaneously, and optimal solutions can always be obtained in $O(nlogn)$ time, where n denotes the number of cells in a standard cell row. The experimental results show that every test case is completed within 0.1 second, verifying the efficiency of our algorithm. In addition, the proposed graph model can be easily updated to minimize cell flippings/ switchings and consider more complicated design rules.

The rest of the paper is organized as follows. The D2D abutment minimization problem is defined in Section II. Section III solves the overall optimization problem by solving its subproblems and combining the subproblem solutions. Then the experimental results are reported in Section IV. Section V expands the proposed graph models and adapts it to other considerations. Finally, Section VI concludes the paper.

II. Problem Definition

In this section, we define the detailed placement optimization problem, where we only consider cell flipping and adjacent cell switching as feasible operations for detailed placement.

Definition 1: **D2D Abutment Minimization Problem**
Given a row of standard cells and the boundary node types (i.e., source or drain) of the diffusion regions (i.e., N-diffusion and P-diffusion) in each cell, horizontally flip a subset of cells and select pairs of adjacent cells to switch their positions in the row, such that the total number of D2D abutments between adjacent cells is minimized.

III. Problem Solution

In this section, we divide the D2D abutment minimization problem into two subproblems. In the first subproblem, only cell flipping is allowed, and in the second one, only adjacent cell switching is allowed. The graph models targeting each subproblem are introduced in Subsection III-A and Subsection III-B respectively. Then Subsection III-C integrates the two graph models into a complete one to solve the overall problem.

A. Cell Flipping Problem

Definition 2: **Cell Flipping Problem (CFP)**
Given a row of standard cells and the boundary node types (i.e., source or drain) of the diffusion regions (i.e., N-diffusion and P-diffusion) in each cell, horizontally flip a subset of cells, such that the total number of D2D abutments between adjacent cells is minimized.

In CFP, each cell has two candidate orientations in the horizontal direction. If we exhaustively enumerate all possible combinations, the time complexity will be $O(2^n)$, where n denotes the number of cells in the row. In a standard cell design, there may be thousands of cells in each standard cell row, and hence the exponentially increased runtime will be too slow to be accepted in practice. In fact, the orientation of each cell only impacts the abutment conditions with adjacent cells. As a result, we only need to consider the abutment combinations between each pair of adjacent cells. Based on the above analysis, we propose a graph model and solve the problem by performing the shortest path algorithm. An example of five consecutive cells, the corresponding CFP graph model and the optimization result are illustrated in Fig. 5.

The graph model is constructed as follows. For each cell c_i, two nodes are introduced in the graph, namely o_i and f_i, corresponding to the original orientation and flipped orientation of c_i respectively. For any pair of adjacent cells c_i and c_{i+1}, four directed edges are introduced connecting from o_i and f_i to o_{i+1} and f_{i+1}, each assigned with a cost value. When the orientations of two adjacent cells introduce a D2D abutment, the corresponding edge cost is 1. Otherwise the edge cost 0. For example, in Fig. 5(a), the original c_1 and the flipped c_2 introduce a D2D abutment, so in Fig. 5(b) the cost of the edge connecting from o_1 to f_2 is 1. Finally, an additional source node s is introduced connected to the o_1 and f_1, and an additional target node t is introduced connected from o_n and f_n. After the graph model is constructed, the shortest path from s to t automatically picks up the optimal orientations for every cell in the row. In Fig. 5(b), the shortest path is marked

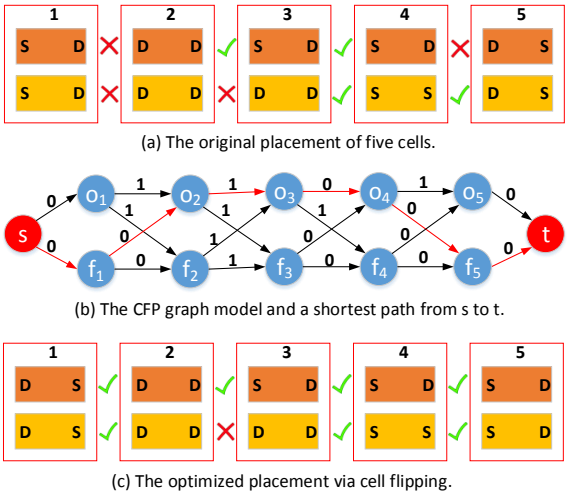

Fig. 5. Solving the CFP problem by constructing CFP graph model and applying the shortest path algorithm on it.

in red, and nodes f_1, o_2, o_3, o_4 and f_5 are picked up by the path. Correspondingly, the optimal orientations for c_1, c_2, c_3, c_4 and c_5 are 'flipped', 'original', 'original', 'original' and 'flipped' respectively, as illustrated in Fig. 5(c). In addition, the total cost of the shortest path shown in Fig. 5(b) is 1, and consequently, there is only 1 D2D abutment in the optimal solution, as marked by the red cross in Fig. 5(c).

B. Cell Switching Problem

Definition 3: **Cell Switching Problem (CSP)**
Given a row of standard cells and the boundary node types (i.e., source or drain) of the diffusion regions (i.e., N-diffusion and P-diffusion) in each cell, select pairs of adjacent cells to switch their positions in the row, such that the total number of D2D abutments between adjacent cells is minimized.

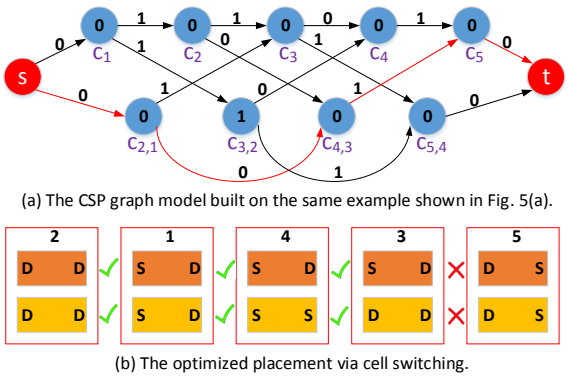

Fig. 6. Solving the CSP problem by constructing CSP graph model and applying the shortest path algorithm on it.

Fig. 6(a) demonstrates the CSP graph model for the same example shown in Fig. 5(a), where both nodes and edges are assigned with cost values. The graph model is constructed as follows. In the top row, a zero cost node is introduced

for each standard cell, denoted by c_1 to c_5 in Fig. 6(a). Next, since each cell is allowed to switch position with its adjacent cells, one additional node is introduced for each pair of switched cells, denoted by $c_{2,1}$, $c_{3,2}$, $c_{4,3}$ and $c_{5,4}$ in Fig. 6(a). Node $c_{i+1,i}$ denotes that the positions of cells c_i and c_{i+1} are switched during detailed placement. If such a switching introduces a D2D abutment between c_i and c_{i+1}, node $c_{i+1,i}$ will be assigned with cost 1. Otherwise it has 0 cost. For example, switching c_2 and c_3 introduces a D2D abutment between them, so the cost of node $c_{3,2}$ is 1. The edges and their cost assignments are defined as follows.

- Each node $c_i (1 \leq i \leq n-1)$ is connected to its adjacent node c_{i+1} by a directed edge. If a D2D abutment exists between c_i and c_{i+1}, the edge cost is 1. Otherwise the edge cost is 0.
- Each node $c_i (1 \leq i \leq n-2)$ is connected to node $c_{i+2,i+1}$ by a directed edge. If a D2D abutment exists between c_i and c_{i+2}, the edge cost is 1. Otherwise the edge cost is 0.
- Each node $c_{i+1,i} (1 \leq i \leq n-2)$ is connected to node c_{i+2} by a directed edge. If a D2D abutment exists between c_i and c_{i+2}, the edge cost 1. Otherwise the edge cost is 0.
- Each node $c_{i+1,i} (1 \leq i \leq n-3)$ is connected to node $c_{i+3,i+2}$ by a directed edge. If a D2D abutment exists between c_i and c_{i+3}, the edge cost 1. Otherwise the edge cost is 0.

Finally, an additional source node s is introduced connected to c_1 and $c_{2,1}$, and an additional target node t is introduced connected from c_n and $c_{n,n-1}$. Both s and t and the edges connecting them have 0 cost. Similarly as in CFP, along the shortest path from s to t, the subscripts of the selected nodes provide the optimal sequence of the cells. For example, in Fig. 6(a), the path in red is the shortest path between s and t. Correspondingly, the optimal cell sequence is $\{c_2, c_1, c_4, c_3, c_5\}$. The optimized placement result is illustrated in Fig. 6(b). Again, the number of D2D abutments in Fig. 6(b) is 1, which equals to the total cost of the shortest path.

C. Overall Problem Solution

In this subsection, the overall graph model for the D2D abutment minimization problem is constructed by integrating the CFP graph model and the CSP graph model. Fig. 7 demonstrates the overall graph model construction for the example shown in Fig. 5(a).

In the first two rows of Fig. 7(a), node o_i denotes the original orientation of cell c_i, and node f_i denotes the flipped orientation of c_i. Each node in the first two rows has 0 cost. Then in the following four rows, each node denotes a pair of switched cells with certain orientations. Node $o_{i+1}o_i$ denotes that both c_{i+1} and c_i are in the original orientation; node $o_{i+1}f_i$ denotes that only c_i is flipped; node $f_{i+1}o_i$ denotes that only c_{i+1} is flipped; node $f_{i+1}f_i$ denotes that both cells are flipped. The node cost assignments for the last four rows are similar as in CSP. Whenever a D2D abutment is introduced between switched cells, the corresponding node cost is 1. Otherwise the node cost is 0. The first two rows in Fig. 7 can be considered as split from the first row in Fig. 6, the last four rows in Fig. 7 as split from the second row in Fig. 6. Then for a node splitting, each edge connecting from/to it is also split into multiple ones, which composes the edge set in Fig. 7. Similar as in CFP and CSP, whenever two nodes connected by an edge introduce a D2D abutment, the corresponding edge cost is 1. Otherwise the edge cost is 0. Again, an additional source node s and an additional target node t are introduced in the overall graph model, and the shortest path between them provides the optimal sequence and orientations of all cells. In this example, the shortest path and the corresponding optimal placement solution is shown in Fig 7(c). By flipping c_1 and c_5 and switching c_3 and c_4, no D2D abutment exists in the optimal solution.

D. Timing Analysis

Let n denote the number of cells in a standard cell row. Then the number of nodes in the overall graph model is $6n-4$, which is linear in n. Similarly, the number of edges in the overall graph model is also linear in n. In the implementation, the graph model is constructed in linear time, and the shortest path algorithm is implemented using Fibonacci heaps [6]. Therefore, the entire time complexity of our algorithm is $O(nlogn)$.

IV. Experimental Results

We implement our algorithm in C++ on a Unix machine with 1.7GHz CPU and 4GB RAM. Then we design a standard cell library with 42 cells for the 16 nm FinFET process. The benchmarks are generated by randomly placing the standard cells in rows. We show the benefits of the proposed algorithm by comparing lengths of the standard cell rows before and after placement optimization. The experimental results are displayed in Table I.

TABLE I
Experimental Results

♯ Cells	Operations	Org. Len. (mm)	Opt. Len. (mm)	Saved Len. (mm)	Runtime (ms)
10k	flip only	9.70	9.53	0.17	2
	flip&switch	9.70	9.24	0.46	7
20k	flip only	19.43	19.07	0.36	8
	flip&switch	19.43	18.50	0.93	20
40k	flip only	38.67	37.96	0.71	12
	flip&switch	38.67	36.81	1.86	28
60k	flip only	58.28	57.21	1.07	14
	flip&switch	58.28	55.51	2.77	44
80k	flip only	78.02	76.58	1.44	20
	flip&switch	78.02	74.27	3.75	59
100k	flip only	96.88	95.12	1.76	23
	flip&switch	96.88	92.24	4.64	75

The first column of table I shows the number of cells in a standard cell row for each test case. The feasible placement

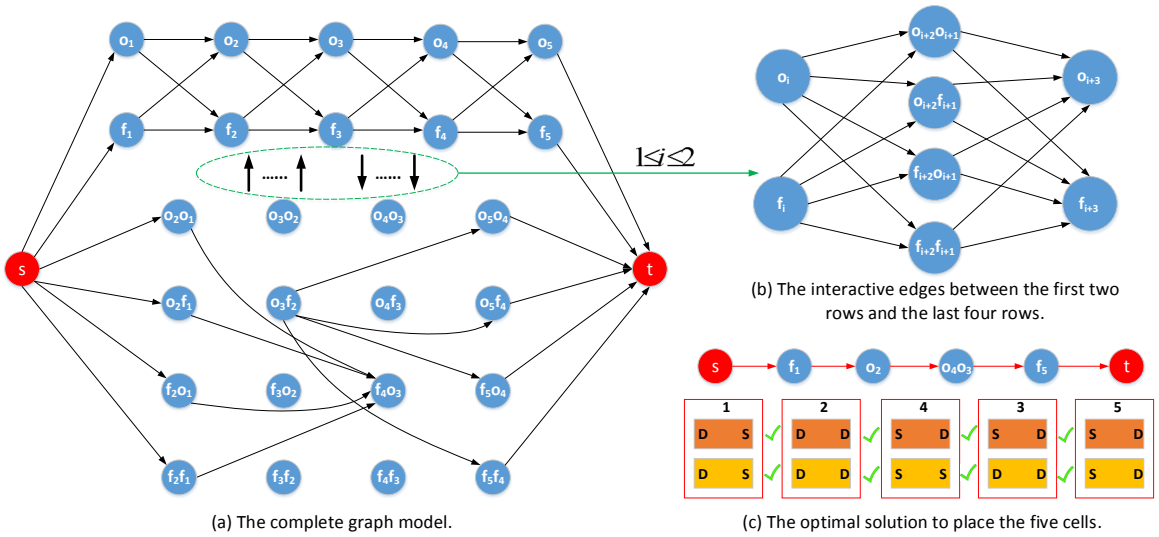

Fig. 7. An overall graph model built on the placement of five standard cells. Note that only a subset of edges in the last four rows are shown in (a), and the interactive edges between the first two rows and the last four rows are illustrated in (b). Cost values are not displayed.

operations are shown in the second column. The following three columns illustrate the original cell row length, the length after placement optimization and the saved length by the optimization respectively. Finally, the last column shows the runtime of the optimization algorithm.

As illustrated in the second column, for each test case, we compare two types of optimizations with different feasible placement operations. In the first one, only cell flipping is allowed, and in the second one, adjacent cell switching is allowed as well. The comparison of the two sets of experimental results shows that adjacent cell switching makes a great contribution to area saving. At least twice the area can be saved by allowing cell switching than allowing cell flipping only. Totally around 5% of the chip area can be saved by the proposed detailed placement optimization strategy. The last column shows that every test case is completed within 0.1 second, verifying the efficiency of our algorithm.

V. Problem Expanding and Discussions

Sometimes the proposed detailed placement optimization strategy may introduce other problems such as net congestions and timing variations due to too many cell flipping and switching operations. Designers may be willing to pay certain area cost in order to resolve those net congestions and timing issues. In other words, to reduce the impact on the global placement result, the number of cell flipping and switching operations should be minimized during the detailed placement optimization. On the other hand, for the 16 nm FinFET process, more complicated design rules may need to be taken into consideration in practice. In this section, we demonstrate that the proposed graph model can be easily modified and adapted to an expanded placement optimization problem.

A. Minimal Cell Flippings and Switchings

In the overall graph model shown in Fig. 7, there may be multiple shortest paths between s and t with the same cost. However, one path may have fewer cell flippings and switchings than another. As we have mentioned previously, cell flipping and switching may impact circuit performance and introduce net congestions. Thus, the shortest path with the minimal cell flippings and switchings is preferred to others. On the other hand, sometimes too many cells have to be flipped or switched in order to save very little area. Designers may not want to make such sacrifice and prefer to pay the little area cost instead. In order to balance the number of cell flippings/switchings and the area saving, we update our graph model by introducing more cost terms: c_d, c_f and c_s, which denote the cost of a D2D abutment, a cell flipping and a cell switching respectively. The three cost values capture the relative importance among the cell operations and area saving. In the original graph model shown in Fig. 7, whenever a node or edge introduces a D2D abutment, the corresponding cost is 1. To update the graph model, we first replace each 1 value with c_d. Next, if a node has one flipped cell (e.g., node f_i, $f_{i+1}o_i$ and $o_{i+1}f_i$), the node cost is increased by c_f. If a node has two flipped cells (e.g., node $f_{i+1}f_i$), the node cost is increased by $2 \times c_f$. Finally, each node in the last four rows has its cost increased by c_s since it denotes a cell switching. On the updated graph model, the shortest path from s to t provides a balanced solution with customized c_d, c_f and c_s values.

B. Other Design Rule Considerations

In practice, the design rules of standard cell abutments for the 16 nm FinFET process can be much more complicated than merely active node type (e.g., D2D) considerations. For

example, usually there are certain minimum width requirements for 'U-shape' and 'stair-shape' jogs on the diffusion layers, as illustrated in Fig. 8(a) and Fig. 8(b) respectively. When designing a standard cell, such rules may not apply if its diffusion region does not have those jogs. However, when abutting two standard cells during detailed placement, the 'U-shape' and 'stair-shape' jogs are very likely to show up if adjacent diffusion regions have different widths. Whenever the minimum width rules are violated due to such cell abutment, a dummy diffusion region has to be inserted in between, as illustrated in Fig. 8(c), where w_j is less than w_u. In this situation, the dummy gates on device edges can be tied to power rails through the dummy diffusion.

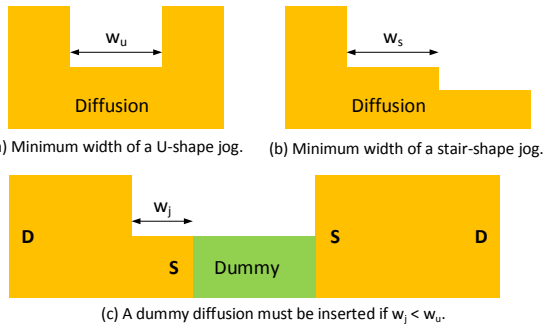

Fig. 8. The design rules for minimum jog widths.

Such dummy diffusions also result in area cost. To capture this in our graph model, we take the length of the dummy diffusion as a cost term, namely l_d. Depending on the shape of the diffusion regions and the minimum jog width requirements, the value of l_d may vary among different cell abutments. At the same time, the length of the inserted source nodes for a D2D abutment is denoted by l_s. Then an updated CFP graph model is illustrated in Fig. 9.

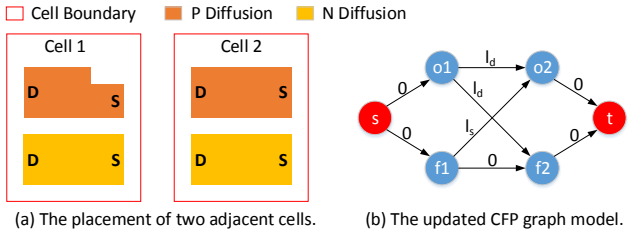

Fig. 9. The updated CFP graph model built on two adjacent cells.

As illustrated in Figure 9(a), even though the original orientations of cell 1 and cell 2 have source to drain abutments in both the P-diffusion and the N-diffusion, the 'U-shaped' jog introduced by the cell abutment violates the minimum jog rule. To resolve the violation, dummy diffusion has to be inserted in between, and consequently, cost l_d is assigned to the edge connecting from o_1 to o_2. Similarly, cost l_d is assigned to the edge connecting o_1 and f_2 as well. On the other hand, abutting f_1 and o_2 introduces a D2D abutment, and hence cost l_s is assigned to the edge connecting them. On the updated graph model, the shortest path from s to t provides the optimal placement solution considering both the D2D abutment penalty and the penalty of minimum jog width rule violations. Similarly, other design rules involving area penalty may also be formulated in the proposed graph model as additional cost terms.

VI. Conclusion

This paper proposes a standard cell based detailed placement optimization strategy for the 16 nm FinFET process. By flipping a subset of cells in a standard cell row and switching pairs of adjacent cells, the number of D2D abutments between adjacent cell boundaries is optimally minimized, which saves additional source node insertion and minimizes the placement area. The benefits and the efficiency of the proposed algorithm are verified by the experimental results. In the end, we also discussed the flexibility of updating the proposed graph model to minimize the cell flipping/switching operations and expanding it to consider practically more complicated design rules for the 16 nm FinFET process.

References

[1] K Ahmed and K Schuegraf. Transistor wars. *Spectrum, IEEE*, 48(11):50–66, 2011.

[2] C Auth, C Allen, A Blattner, D Bergstrom, M Brazier, M Bost, M Buehler, V Chikarmane, T Ghani, T Glassman, et al. A 22nm high performance and low-power cmos technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density mim capacitors. In *VLSI Technology (VLSIT), 2012 Symposium on*, pages 131–132. IEEE, 2012.

[3] S Bangsaruntip, GM Cohen, A Majumdar, Y Zhang, SU Engelmann, N Fuller, LM Gignac, S Mittal, JS Newbury, M Guillorn, et al. High performance and highly uniform gate-all-around silicon nanowire mosfets with wire size dependent scaling. In *Electron Devices Meeting (IEDM), 2009 IEEE International*, pages 1–4. IEEE, 2009.

[4] Munkang Choi, Victor Moroz, Lee Smith, and Oleg Penzin. 14 nm finfet stress engineering with epitaxial sige source/drain. In *Silicon-Germanium Technology and Device Meeting (ISTDM), 2012 International*, pages 1–2. IEEE, 2012.

[5] KF Dombrowski, A Fischer, B Dietrich, I De Wolf, H Bender, S Pochet, V Simons, R Rooyackers, G Badenes, C Stuer, et al. Determination of stress in shallow trench isolation for deep submicron mos devices by uv raman spectroscopy. In *Electron Devices Meeting, 1999. IEDM'99. Technical Digest. International*, pages 357–360. IEEE, 1999.

[6] Michael L Fredman and Robert Endre Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596–615, 1987.

[7] Chang Woo Kang and Sahun Hong. Timing-driven, congestion minimization, and low power placement for standard cell layouts.

[8] Rick Merritt. Collaborate to innovate finfet design ecosystem challenges and solutions. In *Electronic Design Process Symposium (EDPS)*, pages 4–4, 2013.

[9] Rick Merritt. EETimes: TSMC starts FinFETs in 2013, tries EUV at 10 nm. http://www.eetimes.com/electronics-news/4411693, 2013. [Online; accessed 4-July-2013].

[10] Victor Moroz. FinFET structure design and variability analysis enabled by TCAD. http://www.edn.com/design/eda-design/4398011/2/FinFET-structure-design-and-variability-analysis-enabled-by-TCAD-, 2012. [Online; accessed 4-July-2013].

[11] Kyoung Hwan Yeo, Sung Dae Suk, Ming Li, Yun-young Yeoh, Keun Hwi Cho, Ki-Ha Hong, SeongKyu Yun, Mong Sup Lee, Nammyun Cho, Kwanheum Lee, et al. Gate-all-around (gaa) twin silicon nanowire mosfet (tsnwfet) with 15 nm length gate and 4 nm radius nanowires. In *Electron Devices Meeting, 2006. IEDM'06. International*, pages 1–4. IEEE, 2006.