Mitigating Layout Dependent Effect-induced Timing Risk in Multi-Row-Height Detailed Placement

Li-Chen Wang and Shao-Yun Fang
National Taiwan University of Science and Technology, Taipei, Taiwan

***Abstract*—With the development of advanced process technology, the electrical characteristic variation of MOSFET transistors has been seriously influenced by layout dependent effect (LDEs). Due to these LDEs, two cells of specific cell types may suffer from timing degradation when they are adjacently and closely placed with specific orientations. To mitigate the timing risk of critical paths and thus optimize the performance of a target design, this work proposes a dynamic programming (DP)-based method for multi-row-height detailed placement with cell flipping and cell shifting. Experimental results shows the efficiency and effectiveness of the proposed DP-based approach.**

I. Introduction

In advanced process nodes using either the CMOS technology or the FinFET technology, the electrical characteristic variation of MOSFET transistors has been more and more seriously influenced by layout dependent effects (LDEs) [1], [2]. In digital designs, since standard cells are placed into rows, the length of oxide diffusion (LOD) and the oxide-to-oxide spacing effect (OSE) become two major LDEs that can cause timing degradation for problematic cell abutments. LOD and OSE determine the degree of stress that is caused by neighboring shallow trench isolations (STIs) and imposes on each transistor [8], [9], and the stress seriously varies carrier mobility.

Performing LDE-aware timing analysis can be done after cell placement [27], where the surrounding environments of the cells on critical paths can be extracted. However, the analysis is based on time-consuming simulation and is inefficient to refine placement for timing optimization. To systematically guide LDE-aware cell placement, the timing degradation due to LDEs for a cell due to an adjacent cell can be considered by regarding the adjacent cell as the attacker cell and regarding the target cell as the victim cell whose timing will be degraded if no enough spacing presents between them. Figure 1 shows an example of timing degradation due to the risky abutment of two cells, the lookup table lists the cell speed ratio of the victim cell C2 when C2 are adjacent to C1. As shown in the table, the speed of C2 can be reduced by 20% if its left cell boundary is abutted to the right cell boundary of C1. In order to mitigate the risky abutment between C1 and C2, two operations may be applied: cell shifting and cell flipping. As shown in Figure 1, with cell shifting, at least five placement sites is required between C1 and C2 to guarantee no timing degradation will be caused on C2. On the other hand, by flipping C2 such that the right cell boundary of C2 is adjacent to C1 instead, only three placement sites will be required to completely get rid of the timing risk.

In advanced process nodes, using multi-row-height standard cells become a popular choice for large-scale high-performance designs to satisfy different design requirements. The detailed placement problem for the designs with multi-row-height cells is much more complicated and difficult than that for the designs with single-row-height cells. This paper proposed the first work on LDE-induced timing optimization during multi-row-height detailed placement. To practically optimize timing and simultaneously consider a full-chip placement, a dynamic programming-based algorithm is then proposed, which utilizes cell shifting and cell flipping to mitigate the timing degradation of critical paths caused by risky cell abutments.

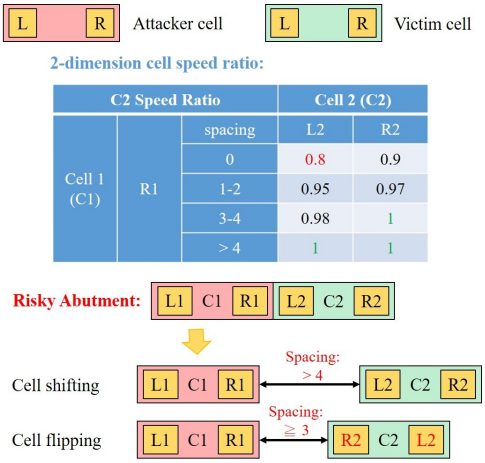

Fig. 1. Timing degradation due to the risky abutment of two cells caused by LDEs.

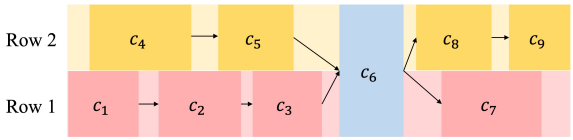

Fig. 2. The cell dependency graph of a two-row example.

II. The Dynamic Programming-based Approach

Figure 2 shows an example of two cell rows with one double-row-height cell. Similar to a few existing dynamic programming-based algorithms for single-row-height detailed placement, the optimization process is done from the left to the right of the chip. The cost of a single-row-height cell is computed by the accumulated cost of its left adjacent cell from the first cell in the row plus the cost induced by itself. For each multi-row-height cell, its cost needs to counts all the accumulated costs of its left adjacent cells plus its own cost. The directed edges in Figure 2 indicate the dependencies of cost computation. For example, the cost of the cell c_6 should be computed after the costs of c_3 and c_5 have been known.

Since cell shifting has to follow the maximum displacement constraints, each cell only has a fixed number of choices in detailed placement. Suppose the maximum allowable displacement amount D is set to 1; that is, each cell can only right or left shift with 1 placement site. Then, each cell can only have $2(2D+1) = 6$ choices. Figure 3 shows a graph illustrating the placement combinations of Cells c_2, c_3, c_5, c_6, c_7, and c_8 in Figure 2. For every possible choice of each cell c_i, the dynamic programming algorithm computes a lower bound of the accumulated cost before and including c_i. For example, v_3^0 represents that c_3 is not flipped and is left shifted by 1 site. The optimal cost of v_3^0 is found by separately placing v_3^0 with v_2^0–v_2^5 and finding the placement combination of c_2 and c_3 at v_3^0 that results in the minimum cost. For c_6, since there are two left adjacent cells c_3 and c_5, the optimal cost of v_6^0 is found by composing the optimal

978-3-9819263-7-8/DATE23/© 2023 EDAA

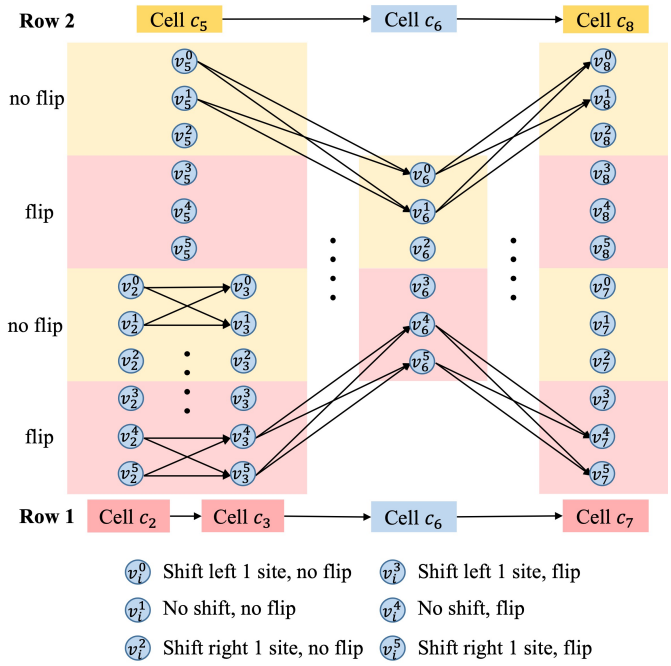

Fig. 3. Cost computation for partial cells in Figure 2

.

combination of c_3 and c_6 at v_6^0 and the optimal combination of c_5 and c_6 at v_6^0. The cost considered in the proposed dynamic programming-based approach is composed of three elements: the cell displacement, the cell abutment cost, and whether cell overlapping occurs when a certain cell placement configuration is considered.

After completing the above cost computation process for a full-chip design, the exact detailed placement choice selected by each cell, can be obtained by tracing back the optimal costs and left adjacent neighbors. Determining the final detailed placement choice for each single-row-height cell is the same as those done in existing dynamic programming-based single-row-height detailed placement works. For each multi-row-height cell c_m, more complicated scenarios need to be considered. Since the optimal costs of the right adjacent cells may not correspond to the same choice of c_m, the best choice for c_m that leads to the minimum cost between c_m and all its right adjacent cells is selected. Note that during the choice selection process for each multi-row-height cell c_m, since the specific choices of c_m that are the base of the optimal costs of its right adjacent cells may be different, the proposed dynamic programming-based detailed placement method cannot guarantee to derive optimal solutions. However, the dynamic programming-based approach is efficient and can also resolve most of risky cell abutments.

III. Experimental Results

We implement the proposed method with C++ programming language, and all experiments are run on a 1200 MHz Linux workstation with 70GB Memory. The test circuits from OpenCores [23] are adopted for the experiments. Each initial netlist is synthesized with the ASAP 7 nm standard cell library [24] and Design Compiler [25], and Innovus [26] is adopted to produce the initial detailed placement. The number of randomly generated risky cell pairs is about half of the number of total cell abutment combinations.

We compare our work with the approach of the sole previous work on timing-aware multi-row-height detailed placement [19]. According to the experimental results, the ratio of reduced total abutment cost after optimization achieved by our approach is 91% on average, and no obvious difference is found among different circuit sizes and among different numbers of critical paths. Compared with [19], the proposed dynamic programming-based approach can further reduce 9% cell abutment risk and 69% cell displacement on average with reasonable runtime overhead. The demonstrated results show that the proposed dynamic programming-based multi-row-height detailed placement approach is effective in fixing most cell abutment risks with small and acceptable cell displacements and pretty high efficiency.

IV. Conclusions

To mitigate the increasingly serious LDE-induced timing degradation problem, this work proposes an optimal ILP formulation and a dynamic programming-based multi-row-height detailed placement approach that is able to consider all the cells in a design at a time. With cell flipping and shifting, the total abutment cost of the cells on critical paths can be efficiently. Future work will focus on proposing more sophisticated algorithms that can derive near-optimal solutions with similar complexity to the dynamic programming-based method proposed in this paper.

References

[1] Y.-Z. Gu, H.-S. Lu, X.-Q. Zhang, M. Lin, X.-W. Zou, and W. Wong, "A study of LDE on stdcell device performance in advance FinFET technology," ICSICT, 2018.

[2] C. Ndiaye, V. Huard, R. Bertholon, M. Rafik, X. Federspiel, and A. Bravaix, "Layout dependent effect: impact on device performance and reliability in recent CMOS nodes," IIRW, 2017.

[3] X. Dong and L. Zhang, "EA-based LDE-aware fast analog layout retargeting," TVLSI, vol. 27, no. 4, pp. 854–863, 2019.

[4] R. Martins, N. Lourenço, R. Póvoa, and N. Horta, "On the exploration of design tradeoffs in analog IC placement with layout-dependent effects," SMACD, 2019.

[5] H.-C. Ou, K.-H. Tseng, J.-Y. Liu, I-P. Wu, and Y.-W. Chang, "Layout dependent-effects-aware analytical analog placement," TCAD, vol. 35, no. 8, pp. 1243–1254, 2016.

[6] T. Liao and L. Zhang, "An LDE-aware gm/ID-based hybrid sizing method for analog integrated circuits," TCAD, vol. 40, no. 8, pp. 1511-1524, 2020.

[7] Y. Zhang, B. Liu, B. Yang, J. Li, and S. Nakatake, "CMOS op-amp circuit synthesis with geometric programming models for layout-dependent effects," ISQED, 2012.

[8] J. V. Faricelli, "Layout-dependent proximity effects in deep nanoscale CMOS," IEEE Custom Integrated Circuits Conference, 2010.

[9] J.-Y. Xue, Y.-D Deng, Z.-C Ye,H.-R Wang, L. Yang, Z.-P Yu, "A framework for layout-dependent STI stress analysis and stress-aware circuit optimization," TVLSI, vol.12, no. 3, pp. 498 - 511, 2012.

[10] A. B. Kahng, P. Sharma, R. O. Topaloglu, "Exploiting STI stress for performance," ICCAD, 2007.

[11] A. B. Kahng, P. Sharma, R. O. Topaloglu, "Chip optimization through STI-stress-aware placement perturbations and fill insertion," TCAD, vol.27, no. 7, pp. 1241-1251, 2008.

[12] A. Chakraborty, S. X. Shi, D. Z. Pan, "Stress aware layout optimization leveraging active area dependent mobility enhancement," TCAD, vol.29, no. 10, pp. 1533-1545, 2010.

[13] J. Li, B. Yang, X.-C. Hu, Q. Dong, S. Nakatake, "STI stress aware placement optimization based on geometric programming," GLSVLSI, 2009.

[14] J. Li, B. Yang, Q. Dong, S. Nakatake, "Post-placement STI well width adjusting by geometric programming for device mobility enhancement in critical path," IEEE, 2010.

[15] Y.-W , Tseng, Y.-W. Chang, "Mixed-cell-height placement considering drain-to-drain abutment," ICCAD, 2018.

[16] J.-L. Chen, Z.-r Zhu, L.-k. Guo,Y.-W. Tseng,Y.-W. Chang, "Mixed-cell-height placement with drain-to-drain abutment and region constraints," TCAD, vol.41 , no. 4, pp. 1103-1115, 2022.

[17] Y.-B. Lin, B, Yu, X,-Q. Xu,J.-R. Gao,N. Viswanathan, W.-H. Liu, Z. Li, C. J. Alpert, D. Z. Pan, "MrDP: multiple-row detailed placement of heterogeneous-sized cells for advanced nodes," TCAD, vol. 37, no. 6, pp. 1237-1250, 2018.

[18] J.-L. Chen, Z.-R. Zhu; Q.-H. Liu; Y.-M. Zhang; W.-X. Zhu, Y.-W. Chang, "Hamiltonian path based mixed-cell-height legalization for neighbor diffusion effect mitigation," DAC, 2020.

[19] C.-L. Hsu, S. Guo, Y. Lin, X. Xu, M. Li, R. Wang, R. Huang, and D. -Z. Pan, "Layout-dependent aging mitigation for critical path timing," ASPDAC, 2018.

[20] C. Han, K. Han, A. B. Kahng, H. Lee, L. Wang, B. Xu, "Optimal multi-row detailed placement for yield and model-hardware correlation improvements in sub-10nm VLSI," ICCAD, 2017.

[21] C. Han, A. B. Kahng, L. Wang, B. Xu, "Enhanced optimal multi-row detailed placement for neighbor diffusion effect mitigation in sub-10 nm VLSI," TCAD, vol. 38, no. 9, pp. 1703 - 1716, 2019.

[22] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/

[23] OpenCores. https://opencores.org/

[24] ASAP 7nm library. https://asap.asu.edu/

[25] Synopsys Design Compiler. https://www.synopsys.com/

[26] Cadence Innovus Implementation System. https://www.cadence.com/zh_TW/home/tools/digital-design-and-signoff/soc-implementation-and-floorplanning/innovus-implementation-system.html/

[27] P. Hurat, R. O. Topaloglu, R. Nachman, P. Pathak, J. Condella, S. Madhavan, L. Capodieci, "Timing variability analysis for layout-dependent-effects in 28nm custom and standard cell-based designs," Proc. SPIE 7974, Design for Manufacturability through Design-Process Integration V, 797412 (4 April 2011); doi: 10.1117/12.882508, https://doi.org/10.1117/12.882508