LUTPLACE: AN IMPROVED LOOKUP TABLE-BASED PLACEMENT FOR ROUTABILITY

Yihang Qiu[1,2], Yan Xing[1], Shuting Cai[1], Xingquan Li[2] , Xiaoming Xiong[1]*
[1]School of Integrated Circuits, Guangdong University of Technology, Guangzhou 510006, China
[2] Peng Cheng Laboratory, Shenzhen 518000, China
*Corresponding Author's Email: yanxing@gdut.edu.cn

2023 China Semiconductor Technology International Conference (CSTIC) | 979-8-3503-1100-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSTIC58779.2023.10219242

ABSTRACT

Congestion estimation is critical in routability-driven placement since it guides subsequent congestion optimization. This paper explores a Rectangular Uniform wire DensitY (RUDY) based algorithm for routability estimation that achieves a better balance between efficiency and accuracy compared to invoking a global router. We build a lookup table based on each net's pin count and aspect ratio to refine RUDY's wirelength computation and enhance RUDY's accuracy in congestion estimation. The experimental results on the DAC 2012 benchmarks show that the lookup table helps improve scaled HPWL by an average of 3.1% and routing congestion by an average of 2.3%.

INTRODUCTION

Standard cell placement is a significant challenge in modern very large-scale integration (VLSI) designs, especially in terms of routability. Routability-driven placement can be divided into two phases: congestion estimation and congestion optimization. Congestion optimization is based on the congestion map generated by congestion estimation, so the accuracy and efficiency of congestion estimation are essential. Modern placers[1][2] often call a global router to get the congestion map, but this method relies on the router's performance and is more time-consuming. In contrast, Rectangular Uniform wire DensitY (RUDY)[3] has a better compromise between accuracy and efficiency, so it is often used to estimate congestion.

However, RUDY's wirelength estimation is modeled as the half-perimeter wirelength (HPWL), while signal nets are routed as Steiner trees, which are ideally modeled as Rectilinear Steiner Minimum Trees (RSMT). There is a gap in accuracy between these two models when a net has more than three pins.

There are two methods for estimating RSMT. One is a construction method, such as FLUTE[4], which is accurate but slow. The other is a lookup table (LUT) method, by weighting HPWL according to a lookup table of scaling factors. The LUT parameters include pin count (PC) [5], and the aspect ratio (AR) of the net's bounding box[6]. Compared with the construction method, the LUT method can provide a better trade-off between accuracy and efficiency, reducing the design cycle required for multiple iterations in VLSI design.

Our work makes the following contributions.

1. We are the first to apply LUT, whose parameters are the pin count and aspect ratio of each net, to improve RUDY's wirelength computation. We then use the improved RUDY to guide congestion optimization.

2. We explore the effect of RSMT cost on routability optimization by LUTs. We find that different LUTs significantly influence routability, especially Total Overflow (TOF) and Maximal Overflow (MOF), highlighting the importance of accurate LUTs.

3. We conducted experiments on the DAC2012 benchmarks[7], and the results show that the LUT helped improve scaled HPWL (sHPWL) by 3.1% and routing congestion by 2.3% on average.

OVERVIEW OF OUR METHODOLOGY

Our Framework

The framework is shown in Figure 1. We implement our idea on DREAMPlace[8], using an improved RUDY (LUT-RUDY) model to generate a route utilization map and using cell inflation for routability optimization. After global placement and legalization, we call NTUplace4h[9] to finish detailed placement.

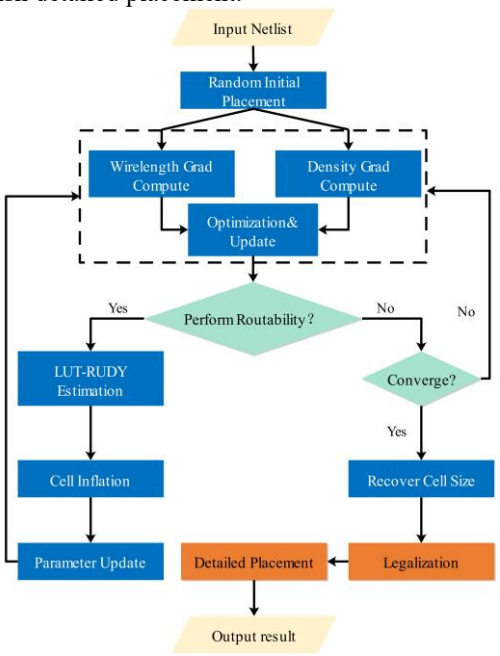

Figure 1: The framework of our proposed routability-driven placement algorithm.

Congestion Estimation Model

The RUDY of net n is shown in Equation (1). The $WireArea_n$ is calculated by HPWL times the wire width. The $BBoxArea_n$ is the area of the minimum enclosing rectangle of net n. Figure 2 depicts the elements for computing the RUDY of a net.

$$RUDY_n = \frac{WireArea_n}{BBoxArea_n} \quad (1)$$

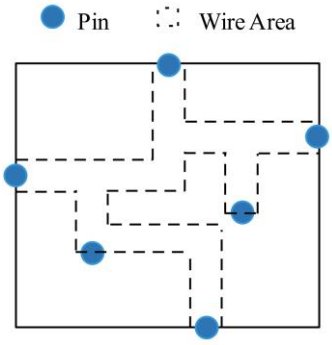

Figure 2: Calculation of Rectangular Uniform wire DensitY(RUDY) for a net.

Cheng[5] estimated the RSMT cost as a correction factor to the HPWL, and the factor is a function of pin count. Andrew further added each net's aspect ratio to improve RSMT cost accuracy and provided a new LUT in [6]. The new LUT is shown in Table I. Table I shows the average RSMT values for pointsets that have bounding box half-perimeter equal to one. By utilizing this new LUT, we can refine RUDY's wirelength computation and enhance RUDY's accuracy in congestion estimation. Finally, Equation (1) can be replaced with Equation (2).

$$LUTRUDY_n = LUT(PC, AR) * \frac{WireArea_n}{BBoxArea_n} \quad (2)$$

Additionally, we need to calculate the ratio of the overlapping area between each net and the routing grid, and use Equation (3) to obtain the routing utilization map. The routing demand of an edge e on the routing utilization map is denoted as D(e) and can be obtained using LUTRUDY. The routing capacity of an edge e is denoted as C(e) and is typically defined in process technology files.

$$\frac{D(e)}{C(e)} = \sum_{i=1}^{m} LUTRUDY_i * \frac{Overlap(i)}{C(e)} \quad (3)$$

TABLE I. WIRELENGTH LOOKUP TABLE USING PIN COUNT(PC) AND ASPECT RATIO(AR)

PC \ AR	1	2	4	10
4	1.06	1.05	1.03	1.01
5	1.13	1.11	1.07	1.03
6	1.19	1.16	1.11	1.05
8	1.32	1.27	1.18	1.08
10	1.42	1.36	1.25	1.12
15	1.66	1.59	1.41	1.21
20	1.87	1.78	1.57	1.29
30	2.22	2.10	1.84	1.45

Cell Inflation

The congestion map can be obtained according to Equation (3). Then all cells within congested grids are inflated using Equation (4). The cell inflation ratio is limited to a maximum of 2.5 and y_{super} is also set as 2.5. Besides, we prevent the total incremental area from being too large by dynamically adjusting the inflation rate. If the total incremental area exceeds the predefined maximum value, we reduce the inflation rate of each cell by a small value to decrease the size of inflation and recheck the total area. We repeat the above process until the total incremental area becomes smaller than the predefined maximum value. Figure 3 illustrates the process of cell inflation.

$$\frac{D'(e)}{C'(e)} = \max\left(\left(\frac{D(e)}{C(e)}\right)^{ysuper}, 2.5\right) \quad (4)$$

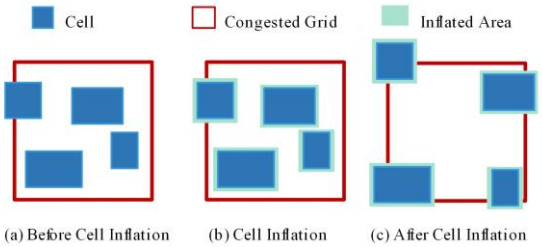

Figure 3: The process of cell inflation.

EXPERIMENTAL RESULTS

Our algorithm was implemented in C++ and Python. The experiments were performed on Intel(R) Xeon(R) Platinum 8380 CPU @ 2.30GHz, Nvidia A100 40GB PCIe GPU under Linux workstation. The benchmarks are from DAC2012 contest. Since superblue2 and superblue11 cannot get legal results with DREAMPlace, only the remaining eight cases will be considered.

We used four quality metrics and runtime to evaluate our placement results. These quality metrics are defined as shown in Equation (5-9). $ACE(x)$ is the average peak congestion defined in [7]. We used the evaluation script provided by DAC2012 contest to get all these values of quality metrics. And the metrics of runtime are recorded after finishing global placement.

$$RC = max(100, PWC) \quad (5)$$

$$PWC = \frac{\sum(K_x * ACE(x))}{\sum K_x}, x = 0.5,1,2,5 \quad (6)$$

$$sHPWL = HPWL * (1 + 0.03 * (RC - 100)) \quad (7)$$

$$TOF = \sum Overflow(e) \quad (8)$$

$$MOF = max(Overflow(e)) \quad (9)$$

Our results are shown in Table II, and the results show that the LUT helps improve sHPWL by 3.1% and routing congestion by 2.3% on average. We also find that different LUTs significantly influence routability, especially TOF and MOF. Figure 4 shows the MOF map of superblue6. We can clearly find that our method is effective.

TABLE II. OUR EXPERIMENTAL RESULTS ON DAC2012 BENCHMARKS.

	Without LUT					With LUT				
Case	***sHPWL***	***RC***	***TOF***	***MOF***	***Time***	***sHPWL***	***RC***	***TOF***	***MOF***	***Time***
SB3	30.75	100.58	9922	10	463	32.14	100.41	6930	10	676
SB6	32.55	101.12	7852	28	486	32.48	100.55	4590	8	519
SB7	37.95	100.78	11318	20	497	38.55	100.76	12926	20	532
SB9	21.73	101.13	12672	12	322	21.77	100.57	6748	10	349
SB12	70.72	164.69	977304	92	523	64.95	156.78	855472	60	514
SB14	23.84	104.17	22482	16	295	22.62	101.90	13732	14	310
SB16	25.94	101.23	11958	20	309	26.37	100.75	9160	20	453
SB19	18.94	112.69	48304	34	200	15.47	103.82	17726	18	231
Normal	100%	100%	100%	100%	100%	96.9%	97.7%	84.2%	69.0%	115.8%

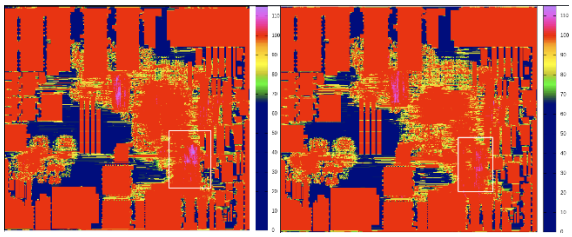

Figure 4: The maximal overflow map of superblue6. The left is the result without using lookup table, and the right is the result with using lookup table.

CONCLUSION

In this paper, we explore a RUDY-based algorithm for routability estimation, which better compromises efficiency and accuracy than invoking a global router. We build a lookup table based on each net's pin count and aspect ratio to improve RUDY's accuracy. The experimental results on the DAC 2012 benchmarks show that the lookup table helps improve scaled HPWL by 3.1% and routing congestion by 2.3% on average. In the future, we will continue to explore more accurate LUTs, and one of the methods is machine learning.

ACKNOWLEDGEMENTS

This project is supported in part by Key-Area Research and Development Program of Guangdong Province(NO. 2022B0701180001), the Major Key Project of PCL (PCL2021A08), the Ministry of Education's Cooperative Education Project (No. 202102172025), and the Guangdong-Hong Kong-Macao Joint Innovation Field Project (No. 2021A0505080006).

REFERENCES

[1] C.-K. Cheng, A. B. Kahng, I. Kang, and L. Wang, "RePlAce: Advancing Solution Quality and Routability Validation in Global Placement," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 38, no. 9, pp. 1717–1730, Sep. 2019, doi: 10.1109/TCAD.2018.2859220.

[2] X. He *et al.*, "Ripple 2.0: high quality routability-driven placement via global router integration," in *Proceedings of the 50th Annual Design Automation Conference on - DAC '13*, Austin, Texas: ACM Press, 2013, p. 1. doi: 10.1145/2463209.2488922.

[3] P. Spindler and F. M. Johannes, "Fast and Accurate Routing Demand Estimation for Efficient Routability-driven Placement," in *2007 Design, Automation & Test in Europe Conference & Exhibition*, Nice, France: IEEE, Apr. 2007, pp. 1–6. doi: 10.1109/DATE.2007.364463.

[4] C. Chu and Y.-C. Wong, "FLUTE: Fast Lookup Table Based Rectilinear Steiner Minimal Tree Algorithm for VLSI Design," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 1, pp. 70–83, Jan. 2008, doi: 10.1109/TCAD.2007.907068.

[5] C. E. Cheng, "Risa: Accurate And Efficient Placement Routability Modeling," in *IEEE/ACM International Conference on Computer-Aided Design*, Nov. 1994, pp. 690–695. doi: 10.1109/ICCAD.1994.629897.

[6] A. E. Caldwell, A. B. Kahng, S. Mantik, I. L. Markov, and A. Zelikovsky, "On wirelength estimations for row-based placement," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 9, pp. 1265–1278, Sep. 1999, doi: 10.1109/43.784119.

[7] N. Viswanathan, C. Alpert, C. Sze, Z. Li, and Y. Wei, "The DAC 2012 routability-driven placement contest and benchmark suite," in *Proceedings of the 49th Annual Design Automation Conference on - DAC '12*, San Francisco, California: ACM Press, 2012, p. 774. doi: 10.1145/2228360.2228500.

[8] Y. Lin *et al.*, "DREAMPlace: Deep Learning Toolkit-Enabled GPU Acceleration for Modern VLSI Placement," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 40, no. 4, pp. 748–761, Apr. 2021, doi: 10.1109/TCAD.2020.3003843.

[9] M.-K. Hsu *et al.*, "NTUplace4h: A Novel Routability-Driven Placement Algorithm for Hierarchical Mixed-Size Circuit Designs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 12, pp. 1914–1927, Dec. 2014, doi: 10.1109/TCAD.2014.2360453.