On Effective Usage of APR Tools for Display Driver IC Layout Generation

Kai-Liang Liang, Li-Yu Lin and Hung-Ming Chen
Institute of Electronics and SoC Center, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
Email: liang1034.ee08g@nctu.edu.tw; davidlin0857.ee04@g2.nctu.edu.tw; hmchen@nycu.edu.tw

***Abstract*—Current APR (automatic place-and-route) tools are in very mature status while realizing digital or mixed signal ICs, however some special purpose ICs are still in the struggle of obtaining better support from the vendors. This study focuses on the display driver IC (DDIC), which has an extreme aspect ratio and thus causing a severe congestion problem. We propose a series of treatments, trying to resolve the routing resource shortage in modern APR methodology. In the congestion identification method, we consider the crowding level of local global routing congestion to predict the location of the detailed routing violations (DRVs). We have utilized the customized techniques, including blockages application, cell inflation, and module adjustment, to achieve our goals. The experiments show that those practical techniques can help identify the congestion regions and effectively decrease the number of DRVs. Moreover, these methods can be integrated into the existing APR flow in the leading-edge design house, providing a rather comprehensive study for routability improvement to reduce the iterations on placement and detailed routing.**

I. Introduction

With the advance of very-large-scale integration design technology, routability is getting more critical in physical design. The quality of placement affects the routability so significantly that a bad placement increases the difficulty of routing and leads to more design rule violations (DRV) and more iterations of fixing. Detailed routing is the most complex and time-consuming process in the physical design flow. When there is any violation reported by design rule checking (DRC) after detailed routing, designers will need to rip up and reroute several times or even go back to the placement stage to modify placement to solve these violations. To reduce the number of iterations of detailed routing, it becomes necessary to identify the congested regions and improve the routability in the early stage of physical design.

Routability problem is especially challenging for the digital blocks of display driver IC (DDIC)[1] design, among all kinds of digital blocks. An illustration of DDIC's configuration is shown in Fig 1. Typically, DDIC is located in the side area of a display panel and sends driving signals and data to control the LCD[2] or OLED[3] panel. The development trend of display technology[4] is a larger screen and higher resolution, which require a larger and more complex design and memory to process the huge image data. While the width of DDIC is increasing with the growing size of the display panel, the height cannot scale up as much as the width to achieve a larger screen-to-body ratio. This results in a very extreme aspect ratio of DDIC and thus unbalance of horizontal and vertical routing resources in its APR region, which can cause severe horizontal congestion. Consequently, there is an urgent need to look for some methodologies that improve the routability of such designs.

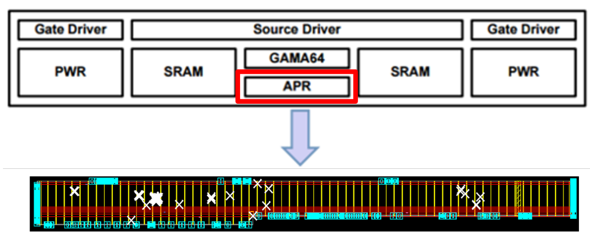


Fig. 1. Configuration of a Display Driver IC (DDIC). The portion for APR has even more extreme aspect ratio.

A. Previous Works

Routability detection has been widely discussed in recent years. Some researches proposed machine learning frameworks to predict congestion. Various prediction models and features were used to detect the occurrence of short violations. [5] utilized support-vector machine (SVM) model, and [6] employed neural network (NN). [7], [8] both applied multivariate adaptive regression splines (MARS); the former was to foresee DRVs; the latter aimed to make the global routing congestion map closer to detailed routing. [9] proposed a CNN-based framework to predict the DRV map and utilized an under-sampling to identify critical samples and separate less important ones for training, which not only speeds up the training process but also improves the prediction accuracy. [10] worked on extreme aspect ratio designs using CNN. In addition to the machine learning frameworks, other approaches were also proposed to estimate the congestion. [11] proposed a probabilistic congestion estimation technique, although the result was quite inaccurate. [12] was based on a wire density per net to estimate the behavior of a router. [13] presented an unroutable placement recognizer to affirm unroutable placements. Despite the good results on machine learning methodologies, we believe that it requires additional effort to construct the training model, prepare data sets and lots of training data, it may not fit the current design methodologies.

For the routability improvement of placement, even though not directly targeting detailed routing congestion to perform placement, recent research proposed several measures to resolve the congestion problem. [14] adjusted cell density by iterative cell inflation to resolve congestion. [15] realized a fast global router as a built-in routing congestion estimator for placers without using maze routing. [16] dynamically adjusted the target density during the global placement stage. [17] computed a routing-difficulty score for every cell in the design library and applied detailed placement techniques called MILOR to disperse local congestion. [18] presented a routability optimizer *ROpt*, building a global routing instance to obtain global and local congestion information for guiding global re-placement to reduce both global and local routing congestion levels of a given placement. [19] optimized routability by net distribution-based cell inflation, inflation ratio adjustment and net replacement. The most important concept repeatedly mentioned in the prior arts on congestion improvement is to evacuate cells in routing critical regions. Although not able to control the behavior of commercial placer directly in design house flow, we can set up certain placement constraints to prevent high cell density and guide the placer for better routability.

B. Contributions

In this paper, we have utilized the customized techniques (not novel though), including blockages application, cell inflation, and module adjustment, for better usage of modern APR tools to resolve the serious congestion. The contributions of our work are summarized as follows:

- Accurately identify the DRV-risky congestion regions by considering the track usage ratio and the crowding level of overflowed gcells, which are usually used as unit in routability evaluations.
- Effectively reduce DRV number by relocating the placement location of a specific module, creating placement constraints in congestion regions with the proper size, density constraint, and cell inflation to lower cell density at congested regions to improve placement results and routability.

- Realize DRV-fixing flow automation with little runtime overhead (compared to usual in-house procedure) without really applying detailed routing and save significant manual effort and spending time on fixing DRVs.

The rest of the paper is organized as follows. Section II describes our problem. Section III shows the techniques we employ in this study. Section IV presents our empirical results and Section V concludes this work.

II. Problem Description

The primary target of this work is to minimize the DRV number after detailed routing. For this purpose, we dedicate to the placement refinement for routability improvement. When given placement and its global routing result, we are to identify congestion regions and generate placement constraints to guide the placer to perform a placement with better routability and fewer overflows, and finally reduce the DRV number. This problem can be divided into two steps described as follows:

- ***Congestion identification.*** First, we need to tell whether the congestion in global routing can be resolved at the detailed routing stage. So, given the global routing result at the placement stage that contains the demand and supply track numbers of every gcell, we need to consider the crowding level of overflowed gcells and identify the gcells where the DRVs will actually occur after detailed routing. It can be verified by comparing the distribution of identified gcells and DRVs.
- ***Placement constraint insertion.*** Given the identified gcells, the placement constraints are created to help resolve congestion around these gcells by lowering the cell density or avoiding congested modules under power stripes. We also need to find a proper size and cell density constraint for the blockages and keepout margins to achieve the best routability improvement and routing result with a minimized DRV number.

III. Methodology

We propose several placement refinement strategies for routability improvement based on the global routing results of a placement. First, we start from the analysis of the congestion map to identify the most congested gcells. Then, we generate placement constraints on the locations of these identified gcells and re-place. We will introduce different kinds of placement constraints such as placement blockage, cell inflation, and module bound. At last, we perform detailed routing to evaluate the routability of placement results and compare the congestion maps and DRV distribution.

A. Congestion Identification

Due to the gap between global routing and detailed routing in advanced nodes, the congestion obtained at the placement stage can be inconsistent with the DRV results after detailed routing. One reason is the complex design rules that make it harder to estimate routing resources accurately. Not every overflow in global routing leads to DRVs because the detailed router will solve DRVs by borrowing nearby empty tracks for de-touring wires. In other words, whether DRVs will emerge depends on a single Gcell and the whole track usage around it. Consequently, we can identify the most congested regions with the highest risk of DRVs by considering whole regional overflow information.

First, we dump all the gcells with overflow numbers larger than 0, using the congestion reporting command. These overflowed gcells are our input set G, and each of them contains its center coordinate x_g y_g, supply $supply_g$ and overflow track number $overflow_g$. We divide the overflow track number by the supply track number, which is the overflow ratio of a gcell. Then, for every overflowed gcell, we sum up the total overflow ratio of its neighboring gcells multiplied by a distance factor d_g within a local square window. The distance factor decreases from the center to the corners. It is calculated as follows (L_{win} is the side width of local window):

$$d_g = 1 - (|x_g - x_c| + |y_g - y_c|)/L_{win} \quad (1)$$

We define an overflow density D_g of a gcell as its total overflow ratio divided by the total number of gcells in a window to represent the crowding level of overflows near this gcell.

$$D_{g_i} = \left(\sum_{g_j \in B_{win}} d_{g_j} \left(\frac{overflow_{g_j}}{supply_{g_j}} \right) \right) / N_g \quad (2)$$

where B_{win} is the local window box, and N_g is the total number of gcells in a local window. The local window size should approach the de-tour wires range as close as possible.

Next, we have to find the critical point of overflow density to induce DRVs, that is, the threshold overflow density D_{th}. For this reason, we have tried different combinations of local window sizes and threshold overflow densities for congestion identification in our test cases. After comparing the congestion map at placement and DRV distribution at the routing stage, we set the threshold overflow density to 0.02 to correlate the identified gcells and DRVs distribution for the best.

```
Input: G_ov: Overflowed gcells
Output: G_c: Congested gcells
L_win: Side width of local window;
D_th: Threshold overflow density;
G_c = Φ;
for each gcell g_i in G do
    x_ll = x_gi - L_win/2;
    y_ll = y_gi - L_win/2;
    x_ur = x_gi + L_win/2;
    y_ur = y_gi + L_win/2;
    B_win(x_ll, y_ll, x_ur, y_ur): Boundary box of local window;
    O_gi=0;
    for each gcell g_j in B_win do
        d_gj = 1-(|x_gi-x_gj|+|y_gi-y_gj|)/L_win;
        O_gi=O_gi+d_gj*(overflow_gj/supply_gj);
    end
    D_gi=O_gi/L_win^2;
    if D_gi > D_th then
        add g_i to G_c;
    end
end
return G_c;
```

Algorithm 1: Congestion Identification

Fig. 2 demonstrates the calculation of overflow density. The congestion map is shown in Fig.2(a), and we use a five gcell(5g) wide window to sum up. The distance factor decreases from 1 to 0 and from the origin gcell to the corners of the window. Finally, we obtain the overflow density for every gcell, which represents the severity of congestion. We set the threshold overflow density as 0.035 in this example and therefore identify the most congested three gcells in the middle while excluding the upper left gcell with high overflows but less crowding level. Algorithm 1 is applied to identify the congestion status, and adopted in the following fixup techniques.

B. Blockage Stripe

In DDIC design, which is often in flat shapes with a very extreme aspect ratio, the number of horizontal and vertical metal tracks is extremely unbalanced: the wires are seriously congested in the horizontal direction. Usually, Metal 1, the bottom metal layer which is also horizontal though, can be only used for pin access because cell

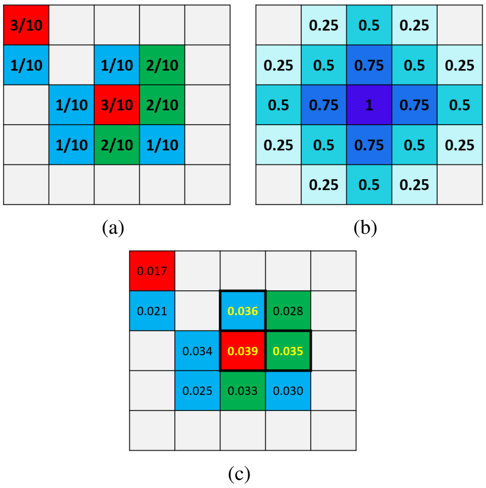
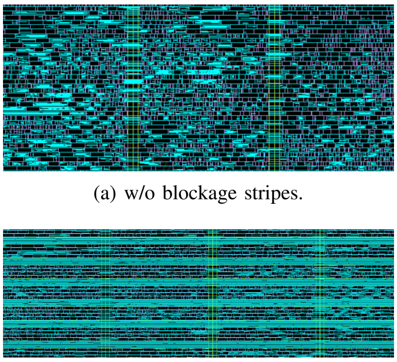


Fig. 2. Example of congestion identification with a 5g window. (a) overflow/supply tracks; (b) distance factors in a 5g window; (c) overflow density of each gcell.

pins have occupied many M1 tracks. To make efficient use of Metal 1 as an additional routing resource, we create hard placement blockage stripes to keep cells out and their M1 pins can use complete Metal 1 tracks for long nets routing. Figure 3 shows that Metal 1 usage increases dramatically with blockage stripes.



(b) w/ blockage stripes.

Fig. 3. Metal 1 usage w/ and w/o blockage stripes. M1 usage increases dramatically with blockage stripes.

The height of blockage stripes is in units of row height, and it affects the efficiency of Metal 1 routing. Blockage stripe with larger height can reserve more M1 tracks, but at the same time, the number of vertical connections across the blockage also increases, which may harm the routability instead. The routing results of different height blockage stripes are shown in Section IV-A.

C. Module Adjustment/Bounding Module Movability

If most of the overflows from the congestion map are on the power stripes and the cells under the power stripes have high internal connectivity, module adjustment/bounding module movability can be applied to reduce DRVs on power stripes.

First, we learn the design hierarchy chart from the physical design tool and sort the design hierarchies in decreasing order by their cell numbers. From the top of the chart, if any module has a high $\# \ of \ nets$ / $\# \ of \ pins$ ratio and is placed under multiple power stripe metal layers at the same time, we will identify this module as high internal connectivity module and try to adjust its location.

We create a placement boundary for this module before the initial placement to restrain its movability during placement. In the new location, the power stripe only occupies one metal layer, and therefore we can avoid the DRVs caused by the extra power stripe metal layer. Also, after the module adjustment, we can observe that the original congested power stripe region has fewer DRVs than those without module adjustment. This is because the placer now places modules with lower internal connectivity in the original congested region. The size of the bound is defined as

$$T * \kappa * \frac{a_m}{a}, \ 1 < \kappa \leq 1.5, \tag{3}$$

where T indicates die size, a indicates total area of cells and a_m indicates total area of cells of a given module. We set $\kappa = 1.3$ for the experiments in Section IV-B.

D. Cell Inflation

We implement cell inflation in the congested area for a given placement and iteratively place cells to refine the placement result, we treat it as an incremental placement. By reducing the cell density in the congested areas, a given placement is easier to be routed.

In the first step, we have to decide the utilization after multiple iterations of cell inflation. The higher target utilization indicates that the cells inside congested areas are placed more sparsely, thus creating more free space for routing in the congested area. However, setting target utilization too high may create new congested areas outside the original ones since there is less room for cells outside the congested areas to be placed.

We define the target utilization for cell inflation as follows:

$$u_s + (1-u_s) * \frac{\alpha}{(1-\beta) + \beta * aspect \ ratio}, \ 0 < \alpha \leq 1, \ 0 < \beta \leq 0.1, \tag{4}$$

where u_s indicates the original utilization. If the aspect ratio increases, the target utilization should be lowered in the formula so that cells will not be placed too sparsely and the number of long nets is reduced. The inflated area of each iteration, which is derived from target utilization, equals to

$$A * \frac{u_t - u_s}{iter_{max} - 1}, \tag{5}$$

where u_t indicates the target utilization, A is the total area of design, and $iter_{max}$ is the number of iterations. For each iteration, we increase the same amount of utilization by cell inflation in the congested region. Since the placement runtime for different designs may vary, we allow more iterations for designs with less runtime.

After identifying the congestion location (Section III-B), we have to decide the keepout margin for cells inside these regions for cell inflation. According to prior arts, there are different strategies of cell inflation to get the better placement result. [14] inflates cells based on their pin numbers and the number of times in congested regions. [20] inflates cells based on the amount of overflow in each gcell. We compare two cell inflation methods in our work. In the first method, the keepout margin is the same for every cell inside congested regions. In the second method, the keepout margin is proportional to the number of pins in a cell, which is defined as:

$$a * p_{cell} / p_{total}, \tag{6}$$

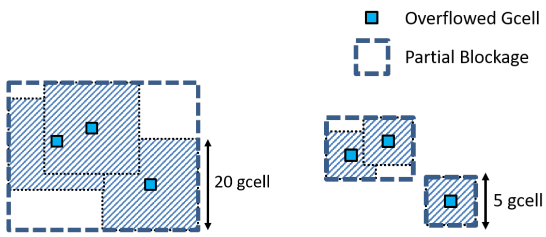
where p_{cell} is the number of pins in a cell, p_{total} is the total number of pins of all congested regions combined, and a is the die size. Since we run multiple iterations for cell inflation flow, if a cell has been included in the congested region for more iterations, the cell will have a larger keepout margin at the end of the flow.

The total amount of keepout margin is partitioned evenly for each iteration, and we allow multiple placement and cell inflation iterations. However, there is an issue about the utilization after placement in each iteration. The placement command in the physical design tool includes steps like buffer tree adjustment. These steps may cause a mismatch between the utilization after placement and target utilization. Therefore, we adjust the total cell inflation area after each iteration to reduce the mismatch of utilization. Finally, we pick the best placement result among these placements based on the lowest sum of overflow, shown in Section IV-C.

E. Partial Blockage

Lowering cell density is a common and efficient way to solve congestion. In commercial APR tools, users can create blockages with specific cell density constraints in specified regions to prevent congestion when re-placing the design.

After identifying those most congested gcells, we need to lower the cell density in these gcells and their neighbor regions because extra routing resource is necessary for de-tour nets when DRVs occur in the congestion region. So the blockages have to cover these gcells with some additional margin, illustrated in Fig.4. To specify the size and location of partial blockages, we extend every identified gcell to a bigger square with a side width of 5~30 times of gcell (5~30 g) width. Then, we merge all overlapping squares into several polygons. The boundary boxes of these polygons are the shapes of the partial blockages. Finally, the cell density within the blockage area is obtained by the cell density reporting command, and the cell density constraint is set 5%~10% lower than the original one, detailed results are in Section IV-D.



(a) (b)

Fig. 4. Illustration of partial blockage generation. (a)20 gcell wide extension; (b)5 gcell wide extension.

F. Combining Treatments to Improve DRVs

During the placement refinement, it usually takes many iterations to produce the best/acceptable result regarding routability because every time the placement constraints are created, new but minor congestion raises beside the original congestion regions due to the cells being pushed outwards. To overcome this problem, we combine these two placement refinement techniques for more effective and efficient routability improvement. After congestion identification, the partial blockages are generated, and the cells inside these blockages are inflated at the same time to prevent them from dense grouping during re-placement.

Since the congestion in different locations inside one design may exist varied causes, it requires to apply different measures for corresponding congestion types. We cope with each type of congestion one by one in a specific order to achieve the best DRV reduction. First, blockage stripes are created for severe horizontal routing resources shortage. Then, module adjustment/bounding module movability will be utilized if there is any congestion emerging on power stripes. After that, congestion identification spots the most congested regions where placement refinement actions, partial blockage generation, and cell inflation will be performed to resolve local congestion, results are shown in Section IV-E.

IV. Experimental Result

The congestion identification and partial blockage generation are implemented in TCL script language. All the APR operations are accomplished in a commercial APR tool, including placement, CTS, global routing, and detailed routing.

TABLE I
INDUSTRIAL DDIC DESIGNS USED IN OUR EXPERIMENT.

	# Cells	# Nets	Die size (μm^2)	# Gcells	Tech node	# DRVs
Case 1	23345	25088	8899.44×319.5	561752	150nm	14784
Case 2	420846	396096	3897×957.6	18947520	55nm	9132
Case 3	1437053	1473664	7372.54×890.88	54796560	40nm	1211
Case 4	88787	88258	5500×555	982905	150nm	1674

Table I lists the statistics of 4 industrial DDIC designs as our test cases. The heights of cases are shrunk 5% to further reduce the routing resource and produce a more critical routing situation, compared to already verified designs. Case 1 is a small design but has an extremely high aspect ratio up to 27.84, which leads to serious congestion in the middle where all wires from both sides and I/O ports meet together. Case 2 has lots of macros on the left, leaving a narrow channel for their outgoing nets. Thus in these places, congestion emerges. Case 3 is the biggest design of the four, while it is the least congested design and has the fewest DRVs. Case 4 contains DRVs caused by multiple power stripe layers and is only used to evaluate the module adjustment method, which we mentioned in Section III-C.

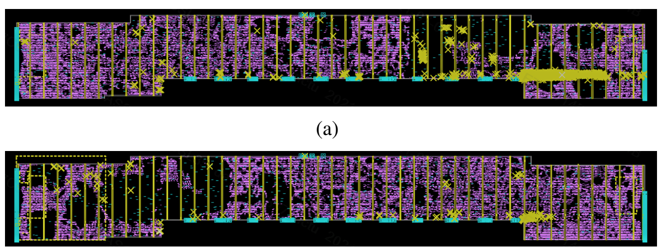
A. Blockage Stripe Result

TABLE II
BLOCKAGE STRIPE RESULTS IN CASE 1.

Height (# row ht)	#Stripes	Spacing (# row ht)	Utilization	# DRVs
1	12	2	61.00%	11109
2	6	4	59.41%	6041
3	4	6	59.08%	7185
4	3	8	61.81%	7670
5	3	8	66.02%	9473
6	2	9	60.10%	9675
12	1	-	60.07%	69098

The blockage stripes are utilized in case 1, the smallest but the most routing-critical design. The routability of this design is so bad that the routing stage takes over 24 hours and still cannot be completed. We tried different heights of blockage stripes and uniformly distributed them in the vertical direction. The number of stripes decreases with the increasing height to keep the total area of blockage stripes and prevent high utilization. Table II shows the DRV numbers of different blockage stripe pattern. Thinner and denser blockages stripes result in better routability except for the single-site-height stripes, which may be too thin and not able to contain enough Metal 1 tracks. Although more blockage stripes can preserve more routing tracks, the vertical crossing connections also increase, which may cause the lack of vertical routing resources.

B. Module Adjustment/Bounding Module Movability Result



(a)

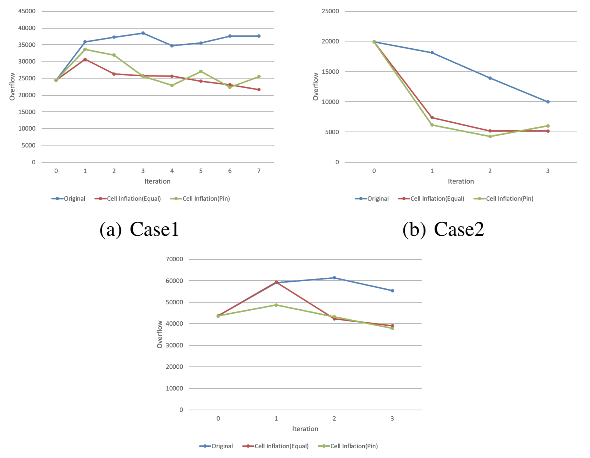
(b)

Fig. 5. The DRV distribution in case 4 (a) w/o module adjustment and (b) w/ module adjustment. We have bounded the key module movement so that the original place (right side of the region) has much less DRVs.

We can see that from Fig. 5 the module we choose to move is now located on the left side of the die. Both the total overflow and the number of DRVs are reduced from the benefit of module adjustment, the overflow has dropped from 20550 to 9894, while the number of DRVs is reduced from 1674 to 236. This is because that the new location of the module only has one power stripe layer instead of two. Furthermore, the module we moved has only a few DRVs increase in the new location of the design.

C. Cell Inflation Result

This subsection examines the placement results with cell inflation against the placement results without cell inflation. Fig. 6 is the overflow change with respect to the number of iterations. The blue line is the result of placer re-run without cell inflation flow, the red line is the result of cell inflation flow with an equal keepout margin, and the green line is the result of cell inflation flow with keepout margin proportional to the pin numbers of each cell. We observe that the placement with the least overflow often emerges at close to the end of iterations. Comparing these three curves in all cases shows that both cell inflation flow with equal keepout margin and keepout margin proportional to the pin numbers can achieve lower overflow than placer re-run without cell inflation flow.



(a) Case1 (b) Case2

(c) Case3

Fig. 6. The overflow change with respect to the iterations regarding cell inflation of DDIC.

After some iterations, we select the placements with the least overflow from these methods and perform detail routing. Table III is the number of DRVs for all cases by the aforementioned methods.

TABLE III
THE NUMBER OF DRVS AFTER CELL INFLATION.

	No inflation	Equal margin	Margin based on #pins
Case 1	14784	17329	10777
Case 2	3956	4077	1985
Case 3	1298	15413	380

The cell inflation flow with keepout margin proportional to the pin numbers achieves the lowest number of DRVs in all three cases. Unfortunately, although overflow decreases, the cell inflation with equal keepout margin does not generate significant improvement on DRV reduction in all three cases, especially in case 3, where the number of DRVs was increasing by a large margin. The reason is that although cell density decreases in the congested regions, the maximum pin density may not decrease in a design, and thus, some congestion still exists after re-placing. Therefore, reducing pin density rather than cell density may be a better approach for DRV reduction during cell inflation flow.

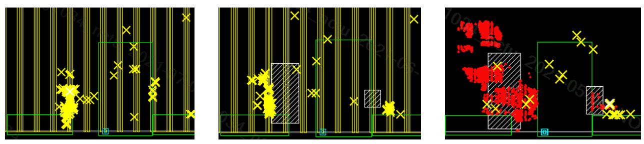
D. Partial Blockage Result

Table IV shows the P&R results of partial blockage insertion with different combinations of blockage sizes (5~30 times of gcell width) and cell density constraints (5%~10% lower than the original cell density). The overflow and DRV number can be reduced most of the time in all cases. However, the DRV number does not necessarily decrease with respect to the overflow number, especially in case 1, which needs a more precise arrangement of blockages for its critical routing resource. The partial blockage insertion can resolve most of the congestion and DRVs in it. But it also raises the cell density next to it and sometimes leads to additional congestion and DRVs that are minor than the original, what we call a congestion shift. When two blockages are too close to each other, a significant congestion shift may occur between the blockages like case 1 and case 2.

The solutions of the partial blockage insertion that make the largest amount of reduction of DRV number differ from the three cases that are 5g, 5% for case 1, 30g, 7% for case 2, and 30g, 5% for case 3. Even 2% of cell density constraint difference with the same blockage shape can greatly affect the congestion shift. It is not efficient to eliminate DRVs by fine adjustment of blockage size and cell density constraint. Therefore we tend to incrementally insert partial blockages on the congestion shift then perform placement again and so on until no identifiable congestion. The results of iterative, incremental blockage insertion in Table V show that we can achieve similar DRV reduction as the best partial blockage solutions within three iterations at most.

E. Results of Combining Treatments to Improve DRVs

Table VI shows the number of DRVs after partial blockage insertion, cell inflation, and the mixed method, respectively. The mixed method further improves the routing result of case 1 and case 3 because the cell keepout margins prevent cells from gathering too close after being placed outside partial blockages during re-placement. As shown in Fig. 7, the red cells which are originally located in congestion regions are inflated and separated around the congestion regions after re-placement. However, in case 2, the mixed method cannot reduce DRVs as much as partial blockage and cell inflation. We observe that during re-placing, the inflated cells are pushed far away from the original congested regions, and the cells with no keepout margin are placed inside these regions. Thus, the reduction of DRVs does not meet the expectation because of these cells.



(a) original/no fix (b) partial (c) mixed

Fig. 7. Comparison of DRV distribution. (a) Original DRV distribution; (b) Partial blockage; (c) Mixed method.

V. CONCLUSIONS

Physical synthesis has been a great challenge in modern chip design, not to mention extreme aspect ratio DDICs. In this work, we hope to study the techniques in prior arts and modify to fit the current design house flow. With congestion identification, the techniques including blockage stripe, bounding module movability, cell inflation and partial blockage insertion are experimented for DRV reduction, thus presenting the effectiveness on DDIC layout generation. Our work can be a good-to-have add-on methodology, providing clear and accurate DRV fixing guidance for physical designers.

TABLE IV

PLACEMENT AND ROUTING RESULTS AFTER PARTIAL BLOCKAGE INSERTION WITH DIFFERENT BLOCKAGE DIMENSIONS AND CELL DENSITY CONSTRAINTS, WHERE g INDICATES HOW MANY GCELLS WIDTH FOR A BLOCKAGE AND % INDICATES HOW MUCH PERCENTAGE THE CELL DENSITY CONSTRAINT IS LOWER THAN THE INITIAL CELL DENSITY.

	Original		5g 10%		10g 10%		15g 10%		20g 10%		30g 10%	
	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #
Case1	19597	14784	3674	1779	4890	1652	4862	1482	4698	2498	-	-
Case2	14147	9132	11995	08084	7148	3841	8086	5049	7138	4325	9927	5982
Case3	39117	1211	38803	403	39993	1027	38610	1576	40040	890	38094	616
	Original		5g 7%		10g 7%		15g 7%		20g 7%		30g 7%	
	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #
Case1	19597	14784	4220	1735	3725	1987	4420	2338	5772	2410	-	-
Case2	14147	9132	8535	04717	6292	5351	7870	4778	6962	4288	3183	1565
Case3	39117	1211	39710	802	37814	779	37714	804	38424	553	37401	377
	Original		5g 5%		10g 5%		15g 5%		20g 5%		30g 5%	
	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #
Case1	19597	14784	3233	1037	5657	2002	4875	2154	3734	2044	-	-
Case2	14147	9132	22291	11714	8144	4284	9846	5960	8536	5346	8355	5063
Case3	39117	1211	38081	867	37613	1238	38497	1331	38404	1221	36232	374

TABLE V

PLACEMENT AND ROUTING RESULTS AFTER ITERATIVE INCREMENTAL PARTIAL BLOCKAGE INSERTION.

	Original		1st Blockage Insertion		2nd Blockage Insertion		3rd Blockage Insertion	
	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #	Overflow #	DRV #
Case1	19597	14784	4112	1617	4908	1986	2943	1286
Case2	14147	9132	9927	5982	2141	1328	-	-
Case3	39117	1211	39993	1027	38346	437	-	-

TABLE VI

THE NUMBER OF DRVS OF MIXED METHOD IN SECTION III-F

	Original	Partial blockage	Cell inflation	Mixed method
Case 1	14784	1037	10777	912
Case 2	9132	1565	1985	3421
Case 3	1211	374	380	299

REFERENCES

[1] WIKIPEDIA, "Wikipedia – Display driver," https://en.wikipedia.org/wiki/Display_driver.

[2] H. Kawamoto, "The history of liquid-crystal displays," *Proceedings of the IEEE*, vol. 90, no. 4, pp. 460–500, 2002.

[3] S. Kunić and Z. Šego, "Oled technology and displays," in *Proceedings ELMAR-2012*, 2012, pp. 31–35.

[4] YJ Kim and Paul Kim, "The role, and future, of the display driver integrated circuit in OLED displays," https://www.epdtonthenet.net/article/157207/The-role--and-future--of-the-display\-driver-integrated-circuit-in-OLED-displays.aspx.

[5] L. Chen, C. Huang, Y. Chang, and H. Chen, "A learning-based methodology for routability prediction in placement," in *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 2018, pp. 1–4.

[6] A. F. Tabrizi, N. K. Darav, L. Rakai, I. Bustany, A. Kennings, and L. Behjat, "Eh?Predictor: A Deep Learning Framework to Identify Detailed Routing Short Violations From a Placed Netlist," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 39, no. 6, pp. 1177–1190, 2020.

[7] Q. Zhou, X. Wang, Z. Qi, Z. Chen, Q. Zhou, and Y. Cai, "An accurate detailed routing routability prediction model in placement," in *2015 6th Asia Symposium on Quality Electronic Design (ASQED)*, 2015, pp. 119–122.

[8] Z. Qi, Y. Cai, and Q. Zhou, "Accurate prediction of detailed routing congestion using supervised data learning," in *2014 IEEE 32nd International Conference on Computer Design (ICCD)*, 2014, pp. 97–103.

[9] W.-T. Hung, J.-Y. Huang, Y.-C. Chou, C.-H. Tsai, and M. Chao, "Transforming global routing report into drc violation map with convolutional neural network," in *Proceedings of the 2020 International Symposium on Physical Design*, 2020, p. 57–64.

[10] S. Koh, Y. Jung, D. Hyun, and Y. Shin, "Routability Optimization for Extreme Aspect Ratio Design Using Convolutional Neural Network," *IEEE International Symposium on Circuits and Systems*, 2021.

[11] J. Westra, C. Bartels, and P. Groeneveld, "Probabilistic Congestion Prediction," in *Proceedings of the 2004 International Symposium on Physical Design*, 2004, p. 204–209.

[12] P. Spindler and F. M. Johannes, "Fast and accurate routing demand estimation for efficient routability-driven placement," in *2007 Design, Automation Test in Europe Exhibition (DATE)*, 2007, pp. 1–6.

[13] W.-H. Liu, T.-K. Chien, and T.-C. Wang, "Region-based and panel-based algorithms for unroutable placement recognition," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, vol. 34, no. 4, pp. 502–514, 2015.

[14] J. A. Roy, N. Viswanathan, G.-J. Nam, C. J. Alpert, and I. L. Markov, "Crisp: Congestion reduction by iterated spreading during placement," in *2009 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2009, pp. 357–362.

[15] W. Liu, Y. Li, and C. Koh, "A fast maze-free routing congestion estimator with hybrid unilateral monotonic routing," in *2012 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2012, pp. 713–719.

[16] M. Kim, J. Hu, D. Lee, and I. L. Markov, "A SimPLR method for routability-driven placement," in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2011, pp. 67–73.

[17] T. Taghavi, Z. Li, C. Alpert, G.-J. Nam, A. Huber, and S. Ramji, "New placement prediction and mitigation techniques for local routing congestion," in *2010 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2010, pp. 621–624.

[18] W.-H. Liu, C.-K. Koh, and Y.-L. Li, "Optimization of placement solutions for routability," in *2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2013, pp. 1–9.

[19] D. Li, C. Li, L. Tao, Y. Zhuang, and G. Chen, "Net-distribution-based routability optimization in global placement," in *2020 IEEE 15th International Conference on Solid-State Integrated Circuit Technology (ICSICT)*, 2020, pp. 1–3.

[20] C.-K. Wang, C.-C. Huang, S. S.-Y. Liu, C.-Y. Chin, S.-T. Hu, W.-C. Wu, and H.-M. Chen, "Closing the gap between global and detailed placement: Techniques for improving routability," in *Proceedings of the 2015 Symposium on International Symposium on Physical Design*, 2015, p. 149–156.