Layer Assignment for Yield Enhancement [1]

Zhan Chen and Israel Koren
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA 01003, USA

Abstract

In this paper, two algorithms for layer assignment with the goal of yield enhancement are proposed. In the first, vias in an existing layout are moved in order to decrease its sensitivity to defects. A greedy algorithm for achieving this objective is presented. In the second, we formulate the layer assignment problem as a network bipartitioning problem. By applying the primal-dual algorithm[1] (a variation of the Kernighan-Lin algorithm[2]), the objective of critical area minimization can be achieved. These two methods are applied to a set of benchmark circuits to demonstrate their effectiveness.

1 Introduction

Layout modification has been proven to be an effective approach for yield enhancement, and various layout synthesis algorithms [3, 4, 5, 6] have been developed to minimize the critical area of VLSI circuits. These proposed layout modification algorithms, however, do not take the number of vias into consideration.

Vias have a very important impact on yield. For example, for the defect density reported in [7], the fault probability of a single metal_1 to metal_2 contact (0.266ppm) is equivalent to the probability of a short-circuit type fault in a metal_2 wire segment (4.949/100m) of length 5.5 microns. The number of vias is determined in the stage of layer assignment, and the objective of layer assignment has traditionally been to minimize the total number of vias in the layout [8, 9, 10]. This has been motivated primarily by a need to minimize the manufacturing cost and maximize the reliability of IC circuits. However, the number of vias is not the only factor that affects yield. The length of the wire segments and the spacings among them need to be considered as well.

In this paper, we propose two layer assignment algorithms for yield enhancement, which consider the weighted critical area including the number of vias and the critical area for open-circuit and short-circuit faults, as the objective function. The paper is organized as follows: in Section 2, the two yield-enhancement layer assignment algorithms are presented. In Section 3, the results for several benchmark examples are presented and discussed. The conclusions are summarized in Section 4.

2 Algorithms

We consider two-layer routing and assume that placement of circuit components and routing of signal nets have already taken place. The objective in our layer assignment is to minimize

[1] This work was supported in part by NSF under contract MIP-9305912.

the weighted cost, defined as:

$$Cost = C_v \cdot N_v + C_{1o} \cdot A_{1o} + C_{1s} \cdot A_{1s} + C_{2o} \cdot A_{2o} + C_{2s} \cdot A_{2s} \quad (1)$$

where $C_v, C_{1o}, C_{1s}, C_{2o}, C_{2s}$ are the probabilities of via fault, first layer open-circuit fault, first layer short-circuit fault, second layer open-circuit fault and second layer short-circuit fault, respectively, N_v is the number of vias , and A_{1o}, A_{1s}, A_{2o}, and A_{2s} are the critical areas for first layer open-circuit fault, first layer short-circuit fault, second layer open-circuit fault and second layer short-circuit fault, respectively.

2.1 Algorithm 1: A Greedy Algorithm for Via-Moving

There are many algorithms available for two layer channel routing as well as general routing. In many cases, vias can be moved from their original positions to achieve a better yield, no matter what routing algorithm is adopted. Take a channel routing problem (Figure 1(a)) for example. In Figure 1(b), via_1 was moved to the left from its original position in Figure 1(a), and part of the critical area between net 1 and net 2 (shaded area in Figure 1(a)) has been eliminated by this move. In Figure 1(c), via_1 is further moved to the corner of the net. Via_1 and via_2 now overlap, and both of them can be eliminated. This results in a optimal solution for this layout, if yield is the primary goal.

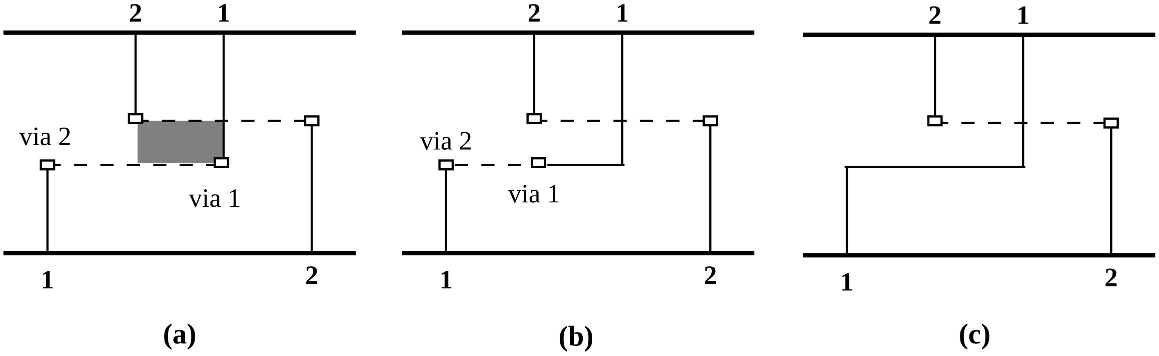


Figure 1: Via-moving for yield enhancement.

An efficient greedy algorithm for via moving has been developed for yield-enhancement. Before presenting the algorithm, we introduce some definitions used in the description of our algorithm.

In two-layer grid-based routing, two nets are said to be *neighbors* if these two nets are overlapping and placed in adjacent grid lines. In gridless routing, however, neighbors are defined as two overlapping nets separated by less than twice the minimum distance required by the design rules. If two nets are not neighbors, they are called *disjoint*. In Figure 2, for example, net_2 and net_3 are neighbors, but net_1 and net_3 are disjoint. Vias in two neighboring nets are also called neighbors (e.g., via_2 and via_3), and those in two disjoint nets are called disjoint (e.g., via_1 and via_2). Two vias are said to *block* each other, if these two vias are both close enough to the same crossing point and further movements of the two vias in particular directions (*blocked directions*) are impossible. In Figure 2(a), via_2 and via_3 block each other, since we cannot move via_2 further left or via_3 further up. The directions *left* for via_2 and *up* for via_3 are called blocked directions. The blocked via movements can be *released* if both of the two blocked vias are moved in their blocked directions simultaneously. In Figure 2, via_2 and via_3 can be moved to the left and up at the same time to release the blocked via movements (Figure 2(b)).

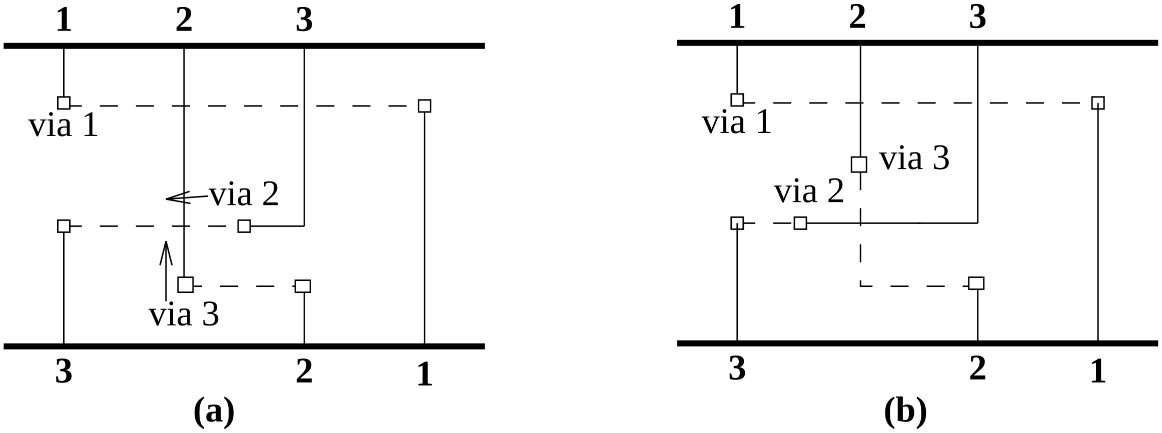


Figure 2: Via-moving block (a) and release (b).

Our greedy algorithm can be described as follows:

1. For each via i do
   Find the optimal position (a position that maximizes the gain in the weighted critical area) for this via, and record the gain associated with this optimal move.
2. Sort the vias according to their gain values. Choose five vias with the largest positive gain as candidates.
3. From the five candidates, select a via which has no neighbors and has the largest gain. If no such via exists (i.e., every candidate is a neighbor of at least one of the other candidates), randomly select one via out of the five candidates. The probability of a via to be selected is made proportional to the value of the gain associated with the optimal move of that via.
4. Move the selected via to its optimal position. Update the optimal moves and the corresponding gain values for the other vias in the same net and for the vias in the neighboring nets.
5. Check the gain list. If there are positive gains, go to 2; otherwise randomly select one blocked via pair and release them. Continue the process until a via-moving with positive gain is found, and go to 2. Stop if no positive gain can be achieved even after all blocked via pairs have been released.

In step 3, we associate a selection probability with each candidate when there is no via without neighbors. This allows vias with high gains (but not the highest) to have a chance to move. Our experiments show that this approach can usually lead to a better solution than the pure greedy approach, in which the via with the highest gain is always selected to be moved.

2.2 Algorithm 2: A Network Bipartitioning Algorithm

Following the definitions in [10, 11], a *potential via* is a place on a wire segment which can accommodate a via without violating the design rules. The number and location of potential vias allowed in the layout affect the quality of the layer assignment: the more potential vias are allowed in the routing, the better the result of layer assignment. On the other hand, having too many potential vias makes the optimization problem unnecessarily complex. Based on the results obtained from our greedy algorithm, we find that the following points are good candidates for potential vias (refer to Figure 3):

1. wire corners;
2. points on a wire segment crossing another wire segment, on both sides of the crossing point;
3. points on a wire segment where its neighboring nets start or end.

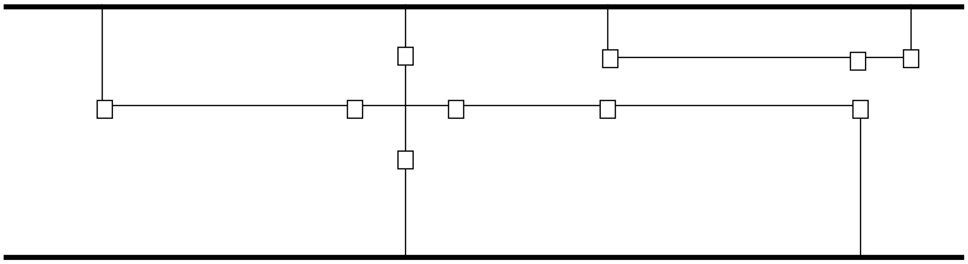


Figure 3: Selection of potential vias.

A *cluster*, denoted by s_i, is defined as a maximal set of mutually crossing wire segments [10, 11]. All wire segments in the routing can be divided into clusters, as shown in Figure 4. Furthermore, clusters can be separated into two *classes* K_1 and K_2 with class K_1 containing those clusters in which horizontal (vertical) wire segments are placed on layer I (II) and class K_2 containing those clusters in which horizontal (vertical) wire segments are placed on layer II (I).

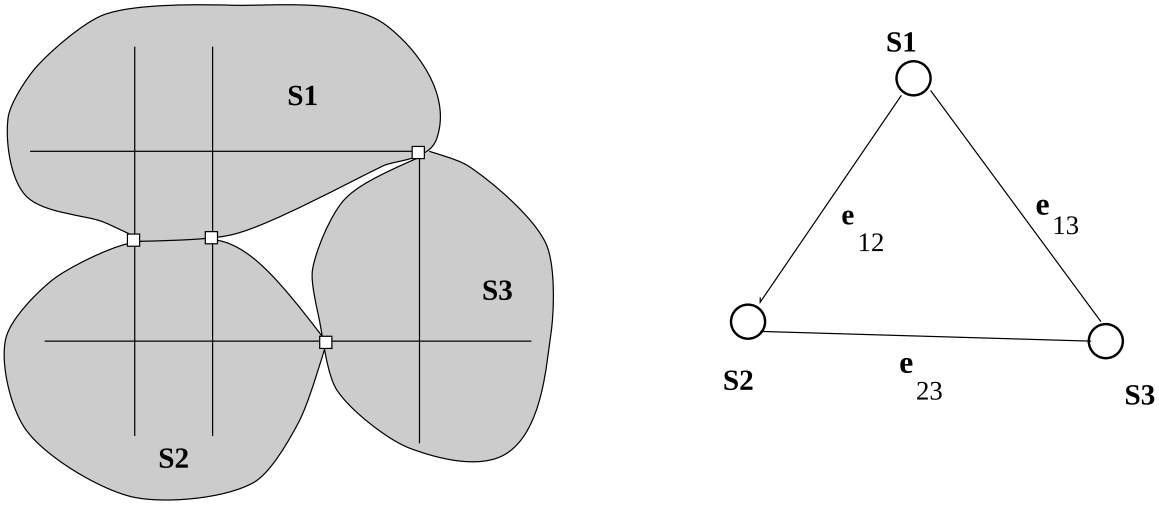


Figure 4: Clusters in layer assignment.

Let the relations among the clusters be represented by a graph $G = (S, E)$, where S is a set of vertices representing clusters, and E is a set of edges representing the relation between any two vertices, s_i and s_j, in the graph G. There is an edge e_{ij} between s_i and s_j if and only if there is at least one potential via between these two clusters or these two clusters contain at least one pair of neighboring wire segments as defined in Section 2.1.

The critical area of a cluster consists of two parts. One is the critical area inside the cluster; and the other is the critical area between itself and its neighboring clusters.

It is assumed that the costs of an open-circuit fault for both layers are the same, and the costs of short-circuit fault are also the same, i.e., $C_{1o} = C_{2o}$, and $C_{1s} = C_{2s}$. Under this assumption, the critical area inside the cluster will remain the same, no matter which class (K_1 or K_2) the cluster is assigned to. The critical area between clusters s_i and s_j can have two possible values, denoted by w^s_{ij} and w^d_{ij}, where w^s_{ij} is the intercluster critical area when clusters s_i and s_j are assigned to the same class, and w^d_{ij} is the critical area when the two clusters are assigned to different classes.

To each edge e_{ij}, we assign a weight w_{ij} equal to:

$$w_{ij} = w^d_{ij} - w^s_{ij} \tag{2}$$

This weight represents the cost of moving two clusters which were in the same class to different classes and it can be either negative or positive. A negative value means that the critical area between these two clusters will decrease if they are placed in different classes; a positive value has the opposite meaning.

The layer assignment problem can thus be formulated as a network bipartitioning problem of assigning each cluster to one of the two classes to obtain a minimum cut between these two classes, i.e.,

$$\text{Min Cost} = \sum_{e_{i,j} \in cut} w_{ij} \tag{3}$$

Unfortunately, this graph partitioning problem is *NP-complete* [12], and a heuristic algorithm is needed for its solution. It is reported in [13] that the *primal-dual algorithm* [1], which is a variation of the Kernighan-Lin Algorithm [2], is a better choice than the Fiduccia-Matheyses algorithm [14]. We have therefore employed the first algorithm to find a suboptimal solution to the network bipartitioning problem. The details of the algorithm can be found in [1].

3 Experimental Results

To test the effectiveness of the presented algorithms, two-layer layouts have been generated for a set of channel routing benchmarks [15] as well as two industrial general routing examples. In the original channel routing layouts, all horizontal wire segments are assigned to the metal_1 layer and the vertical wire segments are assigned to the metal_2 layer, while the two industrial examples are generated using IBM gridless router [16]. The costs for the different types of defects used in the examples are: $C_v = 15$, $C_{1o} = C_{2o} = 1$, $C_{1s} = C_{2s} = 5$

[7]. To simplify the calculations in the channel routing examples, we use the length of the overlap between wire segments in two adjacent rows or columns to represent the critical area for the short-circuit type faults. This simplification is based on the observation that the diameter x of a defect has a density function $f(x)$ that decreases as $1/x^3$ [17], and therefore, the error introduced by ignoring the critical area between non-adjacent wire segments is small. Since in channel routing all wire segments have the same width, we can use the length of the wire segments to represent the critical area for open-circuit type faults. In the two industrial examples, the distance between two adjacent wire segments can be any value greater than the minimum distance d required by the design rules. To facilitate computation, we define a unit critical area as two unit-length wire segments separated by the distance d. Due to the same reason as in channel routing, we ignore those adjacent wire segments which are separated by a distance greater than $2d$. For segments separated by a distance smaller than $2d$, we get their critical area by scaling their overlap length by the density function $f(x) = 1/x^3$. The same rule is applied to calculate the open-circuit critical area.

The results for these examples are shown in Table 1.

Examples	Original Layout	Algorithm 1		Algorithm 2	
	Crit. Area	Crit. Area	% Reduc.	Crit. Area	% Reduc.
ex1 [15]	2734	2411	11.8	2390	12.6
ex3a [15]	4748	4301	9.4	4165	12.2
ex3b [15]	6763	6208	8.2	6067	10.3
ex3c [15]	7833	6955	11.2	6919	11.7
Diff. Ex. [15]	25663	23173	9.7	23002	10.4
IBM ex1	3228.4	2948.9	8.7	2922.4	9.5
IBM ex2	17729.1	15902.8	10.3	15831.1	10.7
Average			9.9		11.1

Table 1: Results of the two layer assignment algorithms on benchmark examples.

The results show that by applying these two methods, the critical area can be reduced by about 10%, and Algorithm 2 seems to provide a better result than Algorithm 1.

Figure 5 shows the layouts of ex1 in [15] before and after using these yield-enhancement layer assignment techniques.

4 Conclusions

In this paper, we proposed two algorithms for yield-enhancement through layer assignment. The first is a via-moving greedy algorithm which can be used as a postprocessor for layer re-assignment of VLSI layouts. The second algorithm, a network bipartitioning algorithm, can be used for initial layout assignment. The critical area can be reduced by about 10% by applying these two algorithms to the channel routing as well as general routing. It is found that the second algorithm achieves a better result than the first one, possibly due to the greedy nature of the first algorithm which may cause it to reach a local, rather than a global, optimal solution.

The drawback of the proposed techniques is that they can only be used in two-layer routing. The yield enhancement layer assignment algorithms for three-layer and other multi-layer routing require further study.

Acknowledgment

The authors wish to thank Luen Heng of IBM T.J. Watson Research Center for his help in providing the two routing examples *IBM ex1* and *IBM ex2*.

References

[1] C. W. Yeh, C. K. Cheng and T. T. Y. Lin, "A General Purpose Multiple Way Partitioning Algorithm," *Proc. 28th ACM/IEEE Design Automation Conference,* pp.421-426, 1991.

[2] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning graphs," *Bell System Technical Journal,* Vol. 49, No. 2, pp. 291-307, Feb. 1970.

[3] G. A. Allan, A. J. Walton and R. J. Jolwill, "A Yield Improvement Technique for IC Layout Using Local Design Rules," *IEEE Trans. Computer-Aided Design,* Vol. 11, No. 11, pp.1355-1362, Nov. 1992.

[4] V. K. R. Chiluvuri and I. Koren, "New Routing and Compaction Strategies for Yield Enhancement," *Proc. IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems*, pp. 325-334, November 1992.

[5] V. K. R. Chiluvuri, I. Koren and J. L. Burns, "The Effect of Wire Length Minimization on Yield," *IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems,* pp. 97-105, Oct. 1994.

[6] S. Y. Kuo, "YOR: A Yield-Optimizing Routing Algorithm by Minimizing Critical Areas and Vias," *IEEE Trans. Computer-Aided Design,* Vol. 12, No.9, Sept. 1993.

[7] R. S. Collica et al., "A Yield Enhancement Methodology for Custom VLSI Manufacturing," *Digital Technical Journal*, 4(2), pp. 83-99, Spring 1992.

[8] D. A. Joy and M. J. Ciesielski, "Layer Assignment for Printed Circuit Boards and Integrated Circuits," *Proceedings of the IEEE,* Vol. 80, No. 2, pp. 311-331, Feb. 1992.

[9] C. P. Hsu, "Minimum-Via Topological Routing," *IEEE Trans. Computer-Aided Design,* Vol. 2, No. 4, pp.235-246, Oct. 1983.

[10] R. W. Chen, Y. Kajitani and S. P. Chan, "A Graph-Theoretic Via Minimization Algorithm for Two-Layer Printed Circuit Boards," *IEEE Trans. Circuits and Systems,* Vol. 30, No. 5, May 1983.

[11] M. J. Ciesielski, "Layer Assignment for VLSI Interconnect Delay Minimization," *IEEE Trans. Computer-Aided Design,* Vol.8, No.6, pp. 702-707, June 1989.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability: A guide to the Theory of NP-Completeness,* W. H. Freeman, 1979.

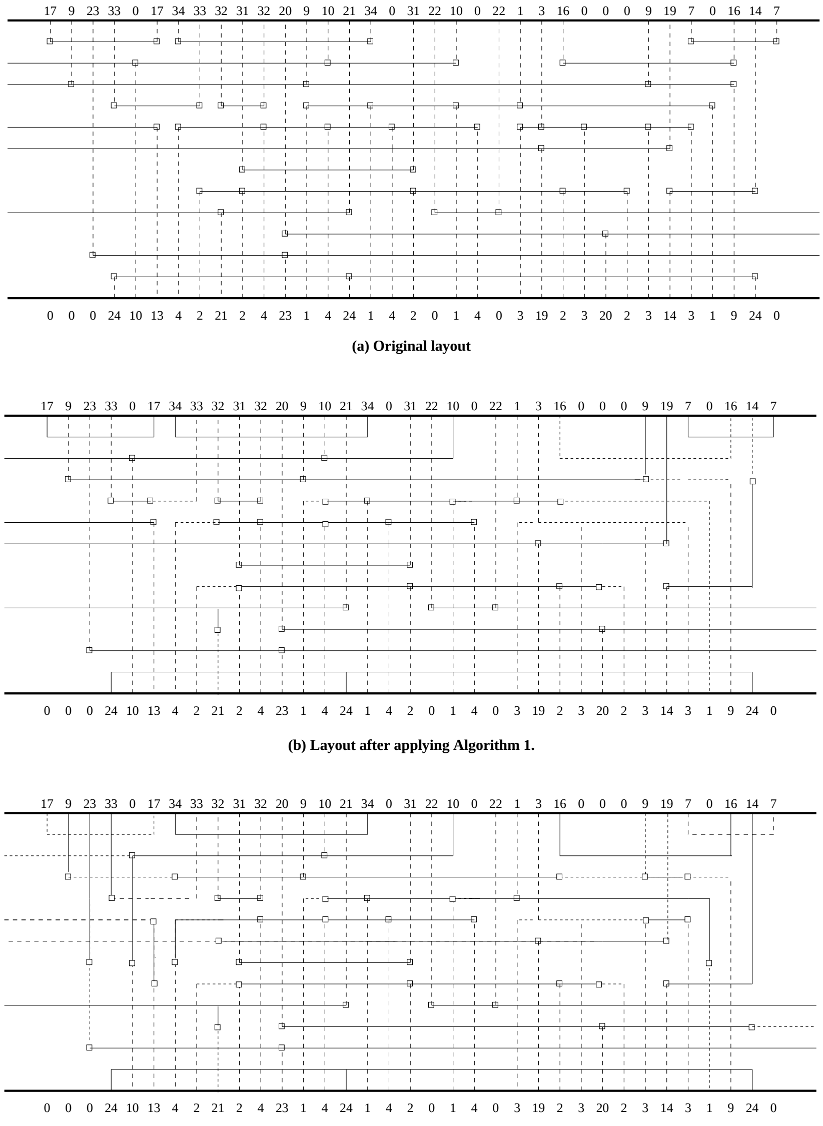
[13] C. W. Yeh, C. K. Cheng and T. T. Y. Lin, "Optimization by Iterative Improvement: An Experimental Evaluation on Two-Way Partitioning," *IEEE Trans. Computer-Aided Design,* Vol. 14, No. 2, Feb. 1993.

[14] C. M. Fiduccia and R. M. Mattheyses, “A Linear Time Heuristic for Improving Network Partitions,” *Proc. 19th ACM/IEEE Design Automation Conference,* pp. 175-181, 1982

[15] T. Yoshimura and E.S. Kuh, “Efficient Algorithms for Channel Routing,” *IEEE Trans. Computer-Aided Design,* Vol. 1, No. 1, pp. 25-35, Jan. 1982.

[16] IBM ABG User’s Manual, Internal Document, IBM Corporation, New York.

[17] I. Koren and A. D. Singh, “Fault Tolerance in VLSI Circuits,” *Computer* , *Special Issue on Fault-Tolerant Systems,* Vol. 23, No. 7, pp. 73-83, July 1990.



(a) Original layout

(b) Layout after applying Algorithm 1.

*

(c) Layout after applyng Algorithm 2.

Figure 5: Layout before and after applying yield enhancement layer assigment techniques.