

A Novel Approach for Yield Optimization

Vazgen S. Karapetyan

Ponte Solutions Inc.,
Russian-Armenian (Slavonic) State University,
e-mail vazgen.karapetyan@pontesolutions.com

Abstract

In this paper a novel approach for yield optimization is presented. While there are different types of defects or other issues affecting the yield, only the "extra conductive material" type (potentially causing shorts circuits) is studied. The approach tends to improve the yield of the IC layout by reducing the area which is sensitive to random defects, the so-called critical area.

Introduction

The VLSI design automation technology has a very significant progress throughout its whole history and especially during the last decade. It is capable of handling very large IC designs with enormous complexity and number of topologies. However even the most advanced technology in EDA does not allow to follow the huge and even increasing set of objectives. The fact is that the capability of the modern physical implementation tools is still very limited - they have to solve a very difficult problem of placement and routing. This task is becoming even more difficult with the growing number of design rules and their complexity. As a result place&route tools are not able to efficiently follow all the constraints, satisfy the modern Design-for-Manufacturability(DFM) rules and Design-For-Yield(DFY) objectives for better yield and follow signal quality improvement recommendations. So how should these issues be addressed? The solution is in post P&R layout optimization. Many different optimization objectives can be defined for such an optimization step - critical area minimization, via etching and planarity improvement for yield optimization, reduction of necessary OPC amounts for printability, crosstalk/RC minimization for signal integrity and so on.

In this paper the general problem of post P&R layout optimization is formulated. Based on that, the yield enhancement flow is constructed and the high level description of its algorithm is described.

A New Generic Layout Optimization Flow

An Abstract Class of Problems

Let us define the following class of abstract mathematical problems:

Definition: Let $V = \{v_1, v_2, \dots, v_n\}$ be a vector of real continuous variables, let us also use $A = \{a_1, a_2, \dots, a_n\}$ and $B = \{b_1, b_2, \dots, b_n\}$ vectors of bounding parameters such that $A \succeq V \succeq B$. Further in the paper these variables will be called decision variables. Please note, that by setting the same value for a_i and b_i one can fix the value of the variable v_i .

Definition: Let $CNS : \{i, j\} \xrightarrow{i, j \in \{1..n\}} \mathbb{R}$ be a table function that corresponds a real number to a pair of indices. We will refer to the function CNS by a term "constraint function" ¹.

Definition: Let $COST : \{v_i, v_j\} \xrightarrow{i, j \in \{1..n\}} \mathbb{R}$ be another function that corresponds a real number to a pair of variables. We will refer to this function by a term "cost function"

Definition: we can now define the objective function

$$F = \sum_{i, j \in \{1..n\}} COST(v_i, v_j)$$

which is the arithmetical sum of all the cost function defined on the variable set V .

Definition: Now let us define $P = \{V, A, B, CNS, F\}$ as an optimization problem of the variable set V , the constraint function defined on the pair of indices CNS , and the objective function F , so that

$$P = \begin{cases} \text{minimize: } F \\ \text{subject to: } \forall i, j \in \{1..n\} \implies (v_j - v_i) \geq CNS(i, j) \\ A \succeq V \succeq B \end{cases}$$

Definition: The collection of values $S = \{s_1, s_2, \dots, s_n\}$ for the decision variables is called the optimal solution or solution, if the objective function F reaches its optimal value while all constraints are satisfied.

Definition: Let $P = \{P_i\{V_i, A_i, B_i, CNS_i, F_i\}\}$ be the class of all possible problems with all arbitrary cost functions and objective functions satisfying to the definitions given above.

It can be easily shown that P is the subclass of more general class of linearly constrained optimization problems. One of its special subclasses is the case when the objective function is at most quadratic. In that case such a problem is called Quadratic Programming (QP) problem. An even more special case of great importance is where the objective function and the constraints are entirely linear; this is called Linear Programming (LP) problem.

Generic Optimization Model

The outline of the abstract layout optimization model proposed in this paper is given by the figure 1. It is a system of the following three functional components: *ProblemGenerator*, *Solver*, *SolutionApplicator*.

ProblemGenerator. This module is responsible for translation of the given layout optimization problem to a problem $P = \{V, A, B, CNS, F\}$ by modeling each of its members. This is done by:

- Constructing the vector of variables V and defining its correspondence to the topologies of the IC layout. Besides, parameter sets A and B should be defined for putting constraints on lower and upper bounds of variables.

¹As you can see the constraint function that is defined here is a binary function of only two variables. At the moment this is sufficient for most of the problems that can be potentially modeled. In general however, this definition can be changed so that CNS can be defined on an arbitrary number of variables

- Generating all the constraints in *CNS* corresponding to the design rules, timing constraints, special constraints defined by the user, etc.
- Defining the objective function F appropriate to that particular optimization criteria.

Solver. This unit does the job of finding such a feasible solution which satisfies to the set of constraint functions and for which the objective function reaches its minimum value.

SolutionApplicator. Once the optimal solution is found, the solution applicator does the reverse job by translating it back to the layout.

THE UNIFIED LAYOUT OPTIMIZATION MODEL

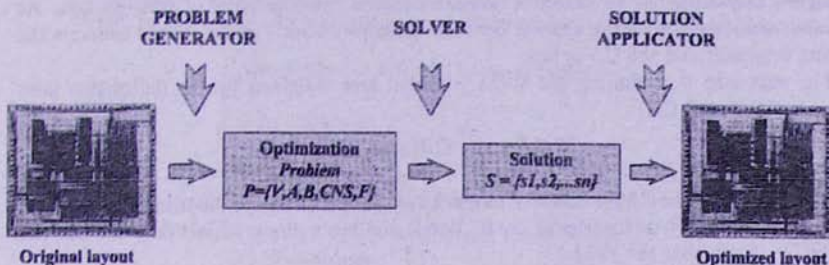


Figure 1: The optimization model

Each one of these subsystems should be defined depending on the nature of the given particular optimization problem.

The Defect Limited Yield Optimization Problem

Here we will construct the so called Defect Limited Yield (DLY) Optimization problem (DLYO). To do this, we will need a quick overview on the problem of DLY in modern IC manufacturing process. It is known that one of the most significant contributors to the overall yield loss is the existence of so-called random defects. Defects are defined as any physical anomaly that causes a circuit to fail and can be roughly divided into two types: extra conductive material that can potentially cause short connections between two topologies of different nets, and missing material potentially leading to open nets. Obviously not all random defects result circuit failure. Whether or not a defect will cause open or short circuit depends on design specific factors such as its density, etc, as well as defect size and location.

The DLYO problem described here is aimed to improve the design tolerance to short type defects, so let's focus on that type only.

A random defect will only cause a circuit failure if it lands in a location where it can produce an electrical short between two wires. The total area of all these locations is called the critical area (CA) for that specific defect.

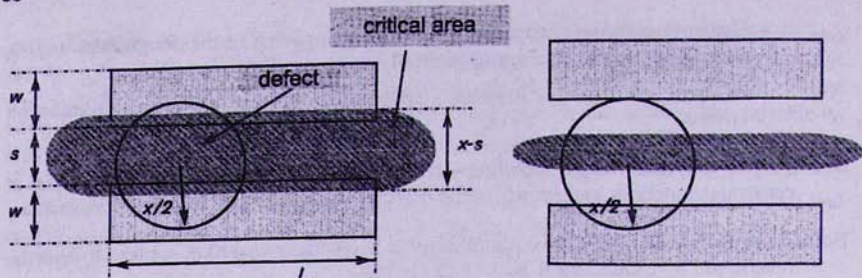
Figure2: The CA for defect size x

Figure shows the CA for defect of radius x between two topologies of different nets. As you can see it depends on the spacing between those two wires - in the second example the spacing is greater and the CA is less.

The next step is in finding the WCA - critical area weighted by the defect size (also known as "average" CA)

$$WCA = \int_0^{\infty} CA(x) * D(x) dx$$

where $CA(x)$ is the CA for defect of radius x and $D(x)$ is the defect distribution function². WCA is one of the key attributes of the IC layout and has a direct impact on yield: the less the WCA - the higher the yield.

The DLYO problem defined below improves the yield by optimally choosing spacings between the routing topologies. It is a so-called one-directional solution of the problem - it supposes to decrease the critical area by modifying the layout in either vertical direction, or horizontal. Depending on this we will differ vertical and horizontal DLYO steps, and the whole optimization flow will be a sequence of such one-directional steps. It is not proven that the sequence of such one-dimensional iterations is converging to the overall optimum value. Anyway it can be easily shown that after each iteration the WCA value is decreasing. Hence the number of iterations can be determined based on the runtime versus optimization results trade-off.

To construct the optimization problem, we will need few more definitions -

Definition: Let's denote by $T = \{t_1, t_2, \dots, t_n\}$ the set of all routing topologies in IC layout.

Definition: Let's define the function $loc: \{t_i\} \rightarrow (R, R)$ as the location function for the topology t_i ³.

Definition: Let's use the function $spacing: \{(t_i, t_j)\} \rightarrow R$ for the minimum allowed spacing between the corresponding topologies of the layout. The function is defined if the topologies t_i and t_j are adjacent⁴.

²For more details and terminology on yield modeling, critical area and random defects please refer to [3] and [5]

³We will assume that the coordinates are real numbers. Without loosing in generality, we can also assume that the coordinates of lower left endpoint of the bounding box of a topology are used as a corresponding location.

⁴Since this is just a high level description, the intuitive understanding of the term "adjacency" is enough.

Definition: Another function - $span: \{(t_i, t_j)\} \rightarrow R$ will be used for quantifying the visibility segment of the two topologies t_i and t_j ⁵. Again, it is defined for two adjacent items and is zero otherwise.

Definition: Finally let's denote by the function $D: (R) \rightarrow R$ the defect density distribution function.

Now we are ready to formulate the DLYO problem $P_{YO} = \{V_{YO}, A_{YO}, B_{YO}, CNS_{YO}, F_{YO}\}$ where:

- $V_{YO} = \{v_i, \forall v_i \text{ corresponds to } loc(t_i).y \text{ if the direction is vertical, or to } loc(t_i).x \text{ if the direction is horizontal}\}$
- A_{YO} and B_{YO} = can be defined as follows: for each non-fixed topology t_i , a_i and b_i can be defined as bottommost and topmost coordinates of the chip bounding box accordingly if the optimization direction is vertical, and leftmost, rightmost - in case of horizontal optimization direction; for each fixed topology t_i , parameters a_i and b_i should be set equal to its existing location (or the value of the corresponding variable).
- $CNS_{YO}(i, j) = \begin{cases} spacing(t_i, t_j) & \text{if } t_i \text{ and } t_j \text{ are adjacent,} \\ -\infty & \text{otherwise} \end{cases}$
- $COST_{YO}(v_i, v_j) = \begin{cases} span(t_i, t_j) \int_{v_j-v_i}^{\infty} (x - (v_j - v_i)) D(x) dx & \text{if } t_i \text{ and } t_j \text{ are adjacent,} \\ 0 & \text{otherwise} \end{cases}$
and
 $F_{YO} = \sum_{i,j \in \{1..n\}} COST_{YO}(v_i, v_j)$

Now let's see what are the functional components of this optimization model:

- **The ProblemGenerator** is creating the vector of variables, one for each layout topology, setting its lower and upper bounds, etc. Then it extracts all the constraints between the topologies based on design rules, timing information, etc. Finally it calculates and keeps the visibility information between the layout items.
- **The Solver** module does the most difficult job of solving the optimization problem.
- **The SolutionApplicator** assigns the values of the optimal solution to the corresponding items in the layout. In other words it moves the topologies so that their locations equal to the appropriate values.

It is empirically determined, that the behavior of the defect density distribution function of the defect size x in the range $[x_0, x_M]$ is similar to the monotone decreasing function $\frac{1}{x^p}$, where x_0 is the minimum feature size in the chosen process and x_M is the maximum observed defect size. Parameter p is typically defined in the range (2.5,3) and is usually determined empirically.

⁵Again, the term visibility will not be defined here, it is not in the scope of this paper. Intuitive understanding is ok.

Assuming the given DLYO approach works for small-size defects, we will admit that $v_j - v_i$ is close to x_0 ⁶. Hence we can modify the formula of the *COST* function in the following way

$$\begin{aligned} \text{COST}_{YO}(v_i, v_j) &= \text{span}(t_i, t_j) \int_{v_j-v_i}^{x_M} (x - (v_j - v_i)) D(x) dx = \\ &\text{span}(t_i, t_j) \int_{v_j-v_i}^{x_M} x \cdot D(x) dx + \text{span}(t_i, t_j)(v_j - v_i) \int_{v_j-v_i}^{x_M} D(x) dx \end{aligned}$$

Based on the definition of the defect density distribution function and having $(v_j - v_i) \rightarrow x_0$, we can write

$$\int_{v_j-v_i}^{x_M} D(x) dx \approx \int_{x_0}^{x_M} D(x) dx = 1$$

and do the following estimation

$$\int_{v_j-v_i}^{x_M} x \cdot D(x) dx \approx \int_{x_0}^{x_M} x \cdot D(x) dx = C$$

As a result we will have the following rough approximation

$$\text{COST}_{YO}(v_i, v_j) \approx \text{span}(t_i, t_j)(C - v_j + v_i)$$

Obviously there can be given more precise ways of approximations of the *COST* function, a good overview on such approximation is presented in [1].

Given that, the cost function in the problem described above will become a linear decreasing function of the spacing. This leads to a very important result: by doing a proper approximation, the abovementioned problem can be reduced to a well known Linear Programming problem. By reducing the general problem to an LP, a number of advanced simplex, barrier or interior-point solving techniques can be used. In similar way, by defining a quadratic approximation to that function, the problem can be reduced to Quadratic Programming problem and so on...

References

- [1] R. Baldick, A. Kahng, A. Kennings, and I. Markov. "Efficient Optimization by Modifying the Objective Function". *IEEE Transactions on Circuits and Systems*, Vol 48, No 8:947-957, 2001.
- [2] C. Bamji and R. Varadarajan. "Leaf Cell and Hierarchical Compaction Techniques". Kluwer Academic Publishers, 1997.
- [3] A. V. Ferris-Prabhu. "Introduction to Semiconductor Device Yield Modeling". Artech House Publishers, 1992.
- [4] I. Koren. "Defect Tolerance in VLSI Circuits: Techniques and Yield Analysis". *Proceedings of the IEEE*, Vol 86, No 9:1817-1836, September, 1998.

⁶This assumption makes sense since the density of the routing topologies is usually high, as a result they are located at distance equal to min spacing. Moreover, because of design rules and timing constraints, they are not allowed to go too far from their original location...

- [5] Ch. Weber, V. Sankaran, K.W. Tobin, and G. Scher. "Quantifying the Value of Ownership of Yield Analysis Technologies". *IEEE Transactions on Semiconductor Manufacturing*, Vol 15, No 4:411-419, November, 1998.

Պիտանի արտադրանքի օպտիմալացման մի նոր մոտեցման մասին

Վ. Կարապետյան

Ամփոփում

Աշխատանքում ներկայացված է ԳՄԻՍ արտադրական խոտանի նվազեցման մի մոտեցման նկարագրություն: Դիտարկվում են հատուկ տիպի պատահական դեֆեկտները և դրանցով պայմանավորված "կարճ միացում" տեսակի խափանումները: Ուսումնասիրվում է դրանց պատճառը և առաջարկվում է ԳՄԻՍ տեղադրությունը փոխելու միջոցով պիտանի արտադրանքի չափի մեծացման մի մոտեցում: