Supervia: Relieving Routing Congestion using Double-height Vias

Saptadeep Pal (saptadeep@ucla.edu)
Advisor: Prof. Puneet Gupta (puneetg@ucla.edu)
Master of Science Project Report
Department of Electrical Engineering, University of California Los Angeles
Collaborators: UCLA - Yasmine Badr, UCSD - Hyein Lee, Kwangsoo Han and Prof. Andrew B. Kahng

***Abstract*—With increase in transistor packing density and use of uni-directional metal routing, resources on local metal layers are increasingly limited. A major contributor to routing congestion is the minimum metal area (minArea) design rule, which has been steadily increasing over the past few technology nodes. For a net which crosses multiple metal layers (e.g., M2 to M4), polygons on intermediate layers (e.g., M3) i.e. via landing pads must satisfy the minArea rule; this creates unnecessary routing blockage, which can lead to area overhead.**

In this work, we investigated the benefits of introduction into the BEOL stack of a new "supervia" structure, namely, a double-height via spanning two metal layers without a landing pad on an intermediate metal layer. We study the benefit of supervia using (i) routing clip-based evaluation using an optimal ILP-based router (OptRouterSV) and (ii) chip-level evaluation using a commercial routing tool in conjunction with MILP-based supervia aware legalization. With the latter, if the legalization approach fails, the failures are localized to clips, which are then routed optimally using OptRouterSV. Our results suggest that when the P&R tool is allowed to generate via structures which optimizes for minArea in stacked vias, using supervia can save $\geq$2% of the chip area whereas in absence of this option, supervia can save as much as 20% of the chip area.

I. Introduction

Modern ICs are more complex and dense with every technology generation. With continued efforts toward reduction of feature size, use of multiple-patterning and other advanced lithography techniques has been on the rise [12][28]. Also, the metal deposition process has been evolving to support continuous technology scaling. As the process is getting more and more complex, foundries are enforcing stricter design rules on physical design to maximize yield. For example, back-end-of-line (BEOL) design rules in advanced nodes using 193nm lithography enforce unidirectional (1D) routing alongside strict requirements of minimum metal width, spacing, fill density, via-pitch, etc. Density scaling is increasingly limited not by transistors but by wires in local metal layers [1], [19], [20], [6]. Local metal layers are also some of the most challenging layers to pattern, and their resistance does not scale well (due to increased contribution of the non-scaling barrier layers) either. For our present work, three causes of increased local metal congestion are of particular interest. (1) The minimum metal area rule has not scaled (due to challenges in deposition and lithography processes) – e.g., going from the 65nm node to the 20nm node, this rule has worsened from 3x to 6x (i.e., multiples of minimum metal width).[1] (2) There is increased via blockage as more nets are routed on intermediate or global metal layers due to performance reasons, as well as due to enforcement of unidirectional routing. (3) Pin access challenges, particularly with emerging device architectures which scale the front-end well, increase pin density.

To mitigate local metal congestion, especially on Metal 1 (M1), middle-of-line (MOL) layers have been introduced below M1 in 20nm and below nodes [10][16]. The MOL layers are primarily used to connect fins and gate to contact connections within standard cells. These connections are highly resistive, and the manufacturing process is optimized only for short connections. Hence, while MOL innovations help in scaling standard cells, they do little to relieve congestion in router-accessible layers, and may in fact make matters worse by further reducing cell size and increasing pin density.

Local metal congestion has been addressed throughout the design flow, even up to such early stages as logic synthesis [3]. Via minimization[21], [4] has received substantial attention as well, both to minimize via-blockage and improve yield/reliability. A comprehensive via blockage model [2] has shown that via blockage on local interconnect layers can waste up to 50% of the wiring area. Several recent works have tried to incorporate via minimization explicitly in global routing [27], [11], [18]. Pin access improvement also has been discussed in physical design [25], [15] as well as standard cell design [23]. Though all the above approaches are helpful in improving routability, they fail to address one of the key critical issues driving congestion: the minArea rule. As noted above, this rule has unfortunately not scaled well with technology, and as a result can be as high as 6x (six times the minimum metal width), i.e., three times the minimum metal pitch. This results in unnecessary blockage primarily in local metal layers when no routing segment (including via landing pads) can be smaller than minArea. With this as motivation, in this work we propose a new *supervia* technology primitive aimed at addressing this challenge. A supervia is a double-height via which (unlike conventional stacked vias) does not require a landing pad in the intermediate layer. The contributions of this work are as follows.

- We propose the supervia structure and discuss its manufacturability as well as the layer where it can be most useful.
- We assess the potential routability improvement from supervias using an *optimal routing* framework.
- We develop a supervia-aware legalization flow which allows us to assess supervia benefits at chip-scale. We then analyze density benefits coming from supervias for a variety of designs and minArea rule values.

The remainder of this paper is organized as follows. Section II introduces the physical structure and manufacturing process of supervia. In Section III, we describe our two approaches that are used to evaluate supervia benefit: (i) an optimal MILP-based router *OptRouterSV* for clip-level evaluation, and (ii) an MILP-based legalizer used in conjunction with commercial place-and-route (P&R) tools for chip-level evaluation. Section IV gives details of our supervia-aware evaluation flow, comprising commercial CAD tools, chip-level legalizer and *OptRouterSV*, which allows use of supervias in a physical layout while maintaining a given minArea rule. Experimental setup and results are discussed in Section V-A. Section VI concludes our work.

II. Background: Min-Area Rules and SuperVias

We now review the origins of the minArea rule, and how supervias can potentially mitigate the density loss caused by this rule.

[1] In the following, we adopt the convention that an "Nx" minimum-area (minArea) rule requires metal area of Nx^2, where x is the minimum metal width.

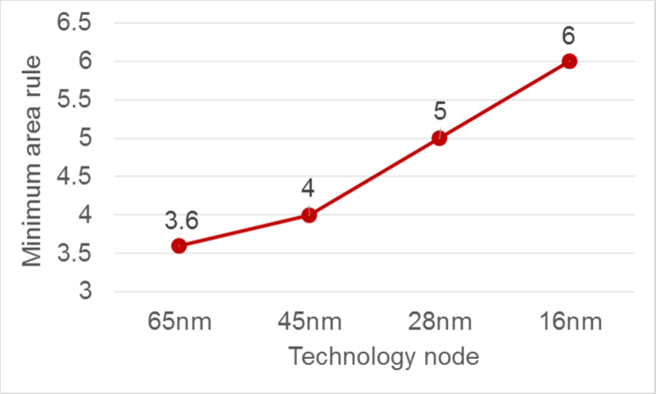


Fig. 1: Minimum metal area rule vs. technology nodes.

A. minArea Rule

Current technologies enforce a minArea requirement on every metal polygon. This rule has experienced an increasing trend over the past few technology nodes, as shown in Figure 1. Two main contributing factors to this trend are lithography and deposition. Local metal layers have started to extensively use multiple-patterning (MP), which decreases the metal pitch but does not help the minArea requirement which is still dictated by a single exposure. Further, copper metallization requires metal trenches to be lined with a barrier material (e.g., Ta, TaN, TiN, TiW) to prevent highly reactive copper atoms from leaching into silicon [26] [24]. However, forming a uniform thin barrier layer is a challenging deposition task, more so when the metal line trench and via openings are small. So, this constrains the minimum size of a trench opening in a dual damascene process.

As noted above, the minArea rule has added to the congestion challenge on local metal layers. Vias that traverse two layers will require a minArea landing pad on the middle (intervening) layer – for example, an M2-M4 via will need a minimum-area landing pad on M3 – which can cause excessive via induced blockage. A given landing pad must satisfy the minimum area rule, i.e. it must have a minimum length of m, where m is equal to minArea/minWidth. This leads to unnecessary via blockage on intermediate metal layers. Our experiments on a projected 7nm library and two small design blocks indicate that such intermediate layer blockage can be 15-20% of total via blockage on the M2 and M3 layers, and that more than 50% of signal net edges traverse more than two metal layers while routing. Further, resistance of local metal layers continues to rapidly increase (in part due to scattering at copper grain boundaries, as well as the highly resistive barrier layer which does not scale in thickness) [31]; this forces more nets to traverse up to intermediate and global metal layers, which in turn exacerbates the via blockage problem on lower metal layers.

Table I shows how the minArea rule affects achievable maximum *utilization* in a P&R block, for three small designs AES, MIPS and ARM CORTEX M0. We report the numbers from two different tools *Cadence Encounter 11.10* and *Cadence Innovus 16.10*. The utilization of a block is the fraction of block area occupied by the standard cells; we use utilization (equivalently, layout density) as an indicator of chip area in this paper.[2] The experimental setup used to obtain these utilization values is described in Section V-B below. As seen from the table, increasing the minArea rule has a negative impact on utilization. For sub-10nm technology, this minArea rule is expected to be $\geq$ 6x. For the testcases we have used, the resulting drop in achievable utilization could be $\geq$ 35% using *Cadence Encounter v11.10* while up to only 7% when we use the latest version of *Innovus* with the option to generate optimal via structures enabled. However with optimal via generation disabled, 7x minArea rule can lead to an overall area overhead of 19%. This is obviously of great concern, as such a loss of utilization would outstrip recent node-to-node layout density gains. Though *Encounter* results were showed that minimum area rule is a big roadblock in improving design density in advanced technology nodes, *Innovus* results diminished the utilization hit coming from a large minimum area rule. This is because of an improved router with ability to optimally generate via structures taking minimum area rule into consideration at every stage of P&R.

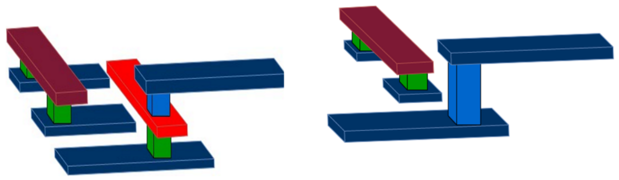
TABLE I: Maximum achievable utilization with different minArea rules (1x, 3x, 5x, 7x) and different numbers of metal layers.

Testcase	#Layers	Tool	Via Generate	MinArea rule			
				1x	3x	5x	7x
MIPS_4	4	*Enc_v11*	N	95%	90%	86%	<60%
	4	*Invs_v16*	N	97%	95%	92%	91%
	4	*Invs_v16*	Y	97%	97%	95%	94%
MIPS_5	5	*Enc_v11*	N	97%	94%	93%	<60%
	5	*Invs_v16*	N	97%	95%	93%	92%
	5	*Invs_v16*	Y	98%	98%	97%	95%
M0_5	5	*Enc_v11*	N	84%	75%	68%	<60%
	5	*Invs_v16*	N	88%	77%	71%	69%
	5	*Invs_v16*	Y	91%	89%	88%	84%
M0_6	6	*Enc_v11*	N	88%	82%	74%	<60%
	6	*Invs_v16*	N	94%	91%	89%	85%
	6	*Invs_v16*	Y	95%	95%	92%	91%
AES_5	5	*Enc_v11*	N	92%	85%	79%	<60%
	5	*Invs_v16*	N	97%	95%	92%	87%
	5	*Invs_v16*	Y	91%	88%	86%	84%
AES_6	6	*Enc_v11*	N	93%	86%	83%	<60%
	6	*Invs_v16*	N	97%	95%	95%	93%
	6	*Invs_v16*	Y	97%	95%	94%	93%

B. Supervia

Figure 2(a) shows a normal stacked via with landing pad on the intermediate metal layer. Our proposed *supervia* structure is presented in Figure 2(b). A supervia is a single double-height via fabricated at once. As seen from the figures, the landing pad on the intermediate layers blocks the track surrounding it and thus increases congestion. The advantage of having a supervia comes from the fact that it can be fabricated at once between metal layer N and metal layer N-2. Hence, the possibility of overlay or alignment error coming from fabricating stacked vias in two steps is no longer relevant, and this eliminates the need to have the extra metal polygon on the intermediate layer.

The supervia structure can be realized with deep-etch technologies, which have been developed and used for high-aspect ratio cut layers in various emerging technologies such as 3D-NAND flash [5], [22], [17].



((a)) Stacked via with landing pad ((b)) Supervia

Fig. 2: Non-supervia vs. supervia.

Since adding supervias to a via layer will incur cost (high aspect ratio etch and possibly an extra mask), minimizing the number

[2] Roughly speaking, chip area is inversely proportional to utilization or layout density.

of layers that permit supervias is important. Our experiments show that the utilization penalty from the minArea rule primarily comes from the local layers (M2 and M3). Table II shows that the greatest drop in utilization comes from applying the minArea rule on M2. (The table denotes testcases according to the naming convention Design_NumLayers, e.g., MIPS_4 is the MIPS testcase implemented with four metal layers.) In the following, we assume that supervias are used only to connect M1 to M3 (i.e., we focus on the potential impacts of enabling just one supervia layer). These experiments were done using *Encounter 11.10*. While using AES as the testcase with *Innovus 16.10* , we noticed the same trend of maximum area hit coming from M2 minArea rules.

TABLE II: Achievable utilization with given minArea rule. Case1: utilization with rule applied on all layers; Case2: rule not applied on M2; Case3: rule not applied on M2 and M3. Tool used is *Cadence Encounter v11.10*

Scenario	Testcase	1x	3x	5x
Case1	MIPS_4	95%	90%	86%
	M0_5	84%	75%	68%
	AES_5	92%	85%	79%
Case2	MIPS_4	95%	93%	91%
	M0_5	84%	82%	81%
	AES_5	92%	88%	88%
Case3	MIPS_4	95%	95%	94%
	M0_5	84%	82%	82%
	AES_5	92%	90%	90%

III. Supervia Aware Layout Legalization

In this section, we describe our supervia and minArea-aware legalizers. We propose two legalizers: (i) an optimal MILP-based router for clip-level evaluation and (ii) an MILP-based chip-level legalizer. The notations used in this section are summarized in Table III.

TABLE III: Notations.

Notation	Meaning
N	set of multi-pin nets
n_k	k^{th} multi-pin net
s_k	source of n_k
T_k	set of sinks of n_k
$t_{k,i}$	i^{th} sink of n_k
$G(V, A)$	routing graph
U	set of vertices (of the routing graph)
u_i	a vertex with the location (x_i, y_i, z_i)
A	set of directed arcs
$a_{i,j}$	a directed arc from u_i to u_j
$e^k_{i,j}$	0-1 indicator whether $a_{i,j}$ is used in the routing of n_k
$\beta^k_{i,j}$	cost for $a_{i,j}$ in the routing of n_k
$f^k_{i,j}$	flow variable for $a_{i,j}$ in the routing of n_k
$p^k_{i,c}$ ($q^k_{i,c}$)	c^{th} left (right) EOL extension option for via vertex u_i, net n_k
Γ	set of metal layer numbers
Π	set of via layer numbers
W	set of all horizontal wire segments
m^γ_w	minimum width on metal layer $\gamma \in \Gamma$
m^γ_s	minimum space on metal layer $\gamma \in \Gamma$
m^γ_l	minimum length on metal layer $\gamma \in \Gamma$
h^π_w	via width on via layer $\pi \in \Pi$
z_i	layer number of wire segment w_i
l_i (r_i)	left (right) variable of wire segment $w_i \in W$
l^{orig}_i (r^{orig}_i)	left (right) original location of wire segment $w_i \in W$
v_j	location variable for a via connected to $w_i \in W$
$b_{j,j'}$	0-1 flag indicating whether via j and j' are vertically aligned. 0 means vias j and j' are vertically aligned
$land(j, j')$	the wire segment which is only a landing pad for both vias j and j'
Q	Set of wire segments which are not landing pads
Δ^l_i (Δ^r_i)	left (right) perturbation to wire segment $w_i \in W$
S_i	Elastic/Slack variable representing a minArea violation (Non-negative)

A. Clip-level Legalizer (OptRouterSV)

In *OptRouter* [7], minArea rule constraints and supervias are not considered. We extend the ILP formulation in [7] to comprehend minArea rules and supervia. Note that a minArea rule can be converted to a minLength rule (m^γ_l) for each metal layer, by assuming 1D routing.

For a given three-dimensional routing resource, horizontal metal track x_i, vertical metal track y_i and metal layer z_i, we formulate our MILP optimization as follows.

Minimize: $$\sum_{n_k \in N} \sum_{a_{i,j} \in A} \beta^k_{i,j} \cdot e^k_{i,j}$$

Subject to:

$$\sum_{n_k \in N} (e^k_{i,j} + e^k_{j,i}) \leq 1 \quad a_{i,j}, a_{j,i} \in A \tag{1}$$

$$e^k_{i,j} \geq \frac{f^k_{i,j}}{|T_k|} \quad a_{i,j} \in A, n_k \in N \tag{2}$$

$$\sum_{u_j : a_{i,j} \in A} f^k_{i,j} - \sum_{u_j : a_{j,i} \in A} f^k_{j,i} = \begin{cases} |T_k| & \text{if } u_i = s_k, n_k \in N \\ -1 & \text{else if } u_i \in T_k, n_k \in N \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The objective is to minimize the weighted sum of total wirelength and the number of vias. Note that we can change the objective to a constant value to check feasibility with the constraints. Constraints (1), (2) and (3) enable multi-commodity flow for multi-pin-net routing. Constraint (1) ensures that each arc is used by only one net. Constraint (2) pertains to the binary variable $e^k_{i,j}$, which indicates whether there is a flow through $e_{i,j}$. Constraint (3) ensures source-sink connectivities (flow conservation).

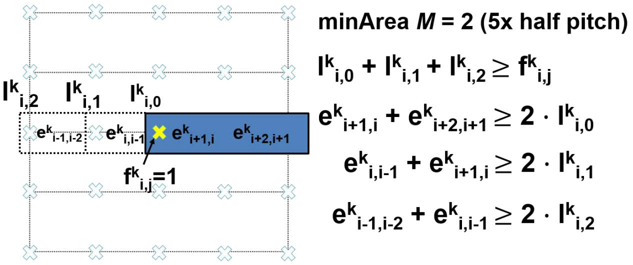

minArea M = 2 (5x half pitch)
$l^k_{i,0} + l^k_{i,1} + l^k_{i,2} \geq f^k_{i,j}$
$e^k_{i+1,i} + e^k_{i+2,i+1} \geq 2 \cdot l^k_{i,0}$
$e^k_{i,i-1} + e^k_{i+1,i} \geq 2 \cdot l^k_{i,1}$
$e^k_{i-1,i-2} + e^k_{i,i-1} \geq 2 \cdot l^k_{i,2}$
$l^k_{i,2}$
$l^k_{i,1}$
$l^k_{i,0}$
$e^k_{i-1,i-2}$
$e^k_{i,i-1}$
$e^k_{i+1,i}$
$e^k_{i+2,i+1}$
$f^k_{i,j}=1$

Fig. 3: minLength rule constraints with EOL extension variables.

End-of-line (EOL) extension and minArea rule constraints. As we can identify the locations of EOL from via locations in 1D routing if we do not consider EOL extension, we might be able to control minLength rule by using via spacing rules. However, this is not correct and realistic, and tight via spacing rules coming from a larger minLength rule will restrict routing solutions severely in sub-10nm technology nodes. Thus, we introduce EOL extension variables to allow wire segments to be extended to fix minLength violations.

The EOL variables represent the locations of EOL extension options for a wire segment. Each wire segment is extended according to the EOL location defined by the corresponding EOL variable selected by ILP. Note that EOL variables are created at via locations, where EOLs exist, since we assume 1D routing. Figure 3 shows an example of EOL variables, where a wire segment with a flow $f^k_{i,j}$ on a via, where $x_j = x_i$, $y_j = y_i$ and $z_j = z_i \pm 1$ (i.e., a via is placed at location (x_i,y_i) between layers z_j and z_i), and the flow continues to the right direction (i.e., the wire segment has a left EOL). In the example of Figure 3, the minLength $m^\gamma_l = 2$. For the left EOL, we introduce three EOL extension options $p^k_{i,0}$, $p^k_{i,1}$, $p^k_{i,2}$ which indicate EOL extensions toward left. We force only one of the options to be chosen among the three EOL options. Note that we do not need more than three options since $m_l^\gamma = 2$. For each EOL option, we force edges between the EOL location and the via location of the wire segment to be one to make the wire segment extended. To do so, the sum of corresponding e variables must meet the minLength m_l^γ. If $p^k_{i,0}$ is selected, $e^k_{i+1,i}$ and $e^k_{i+2,i+1}$ must be one.

A generalized formulation is given in Constraint (4).

$$\sum_{c\in C} p^k_{i,c} \geq f^k_{i,j} \qquad a_{i,j} \in A \tag{4}$$

$$\sum_{a_{i,j}\in A'} e^k_{i,j} \geq m_l^\gamma \cdot p^k_{i,c} \qquad c \in C \tag{5}$$

If a flow variable $f^k_{i,j}$ on a via is used (i.e., whenever there is a via at location (x_i, y_i), Constraint (4) forces that one of the left EOL extension variables must be selected. Given a minLength m^γ, if an EOL extension variable $p^k_{i,c}$ is selected, the sum of corresponding $e^k_{i,j} \in A'$ must be larger than or equal to the minLength (Constraint (5)). We can similarly treat right EOL extension variables $q^k_{i,c}$ and minLength rule constraints.

Constraints (4) and (5) handle only left EOLs of wire segments. To handle right EOLs of wire segments together, the constraints are rewritten as follows.

$$\sum_{c\in C}(p^k_{i,c} + q^k_{i,c}) \geq f^k_{i,j} \qquad a_{i,j} \in A \tag{6}$$

$$\sum_{a_{i,j}\in A'} e^k_{i,j} \geq m_l^\gamma \cdot p^k_{i,c} \qquad c \in C \tag{7}$$

$$\sum_{a_{i,j}\in A'} e^k_{i,j} \geq m_l^\gamma \cdot q^k_{i,c} \qquad c \in C \tag{8}$$

Supervia constraints. To enable supervia, minLength rule constraint should not be applied for the intermediate layer when two vertically aligned consecutive vias are used. In our formulation, we have additional constraints as follows.

$$\sum_{c\in C}(p^k_{i,c} + q^k_{i,c}) \geq f^k_{i,j} - f^k_{i,j'} \qquad a_{i,j} \in A \tag{9}$$

$f^k_{i,j}$ and $f^k_{i,j'}$ are the flows of two vertically aligned vias, where $x_i = x_j = x_{j'}$, $y_i = y_j = y_{j'}$, $z_j = z_i + 1$ and $z_{j'} = z_i - 1$.

B. Chip-level Legalizer

We propose a supervia-aware legalization method based on Mixed Integer Linear Programming (MILP). The input of the legalizer is a routed layout with minArea-rule violations and the objective is to minimize the minArea violations by applying supervias. There are two main differences between our legalizer and the classic migration/legalization algorithm [8]. First, our legalizer does not change the front-end-of-line (FEOL) of standard cells but only the BEOL, and in addition it keeps pin locations of standard cells and intra-cell routes unchanged. Second, we consider supervias in the MILP formulation to take the advantage of the supervias during the legalization. The migration is performed in the X-direction followed by the Y-direction. Two iterations of migration are performed (Experimentally no further iterations were needed). We explain our layout representation and MILP formulation in the following paragraphs.

MILP formulation.

In this section, we show the MILP formulation for migration in the X-direction. The MILP for the Y-direction is similar.

An example of our layout representation is shown in Figure 4, where the variables representing three metal segments and a via are shown. All the routes are assumed to be unidirectional. Each layout rectangle on the metal layers M2 and above is represented by its four edges. The variables l_i and r_i correspond to the left and right edges of rectangle i respectively, as shown in Figure 4.

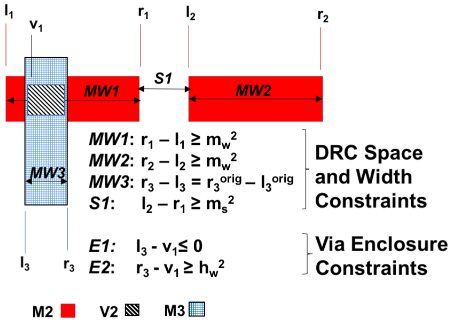


Fig. 4: Example of our layout representation and the MILP variables and constraints for the X-direction legalization. The generated DRC and via enclosure constraints are shown. m_w^2 is the minimum width value DRs for M2. m_s^2 is the minimum space allowed on M2. h_w^2 is the dimension of each via hole on via2 layer.

Since we assume all vias of each layer have the same dimensions, only the bottom left corner (thus left edge and bottom edge) are enough to represent the via.

$$\textbf{Minimize:} \quad minimize \sum_i (\Delta_i^l + \Delta_i^r) + \lambda * \sum_i S_i \tag{10}$$

Subject to:

$$\Delta_i^l \geq l_i - l_i^{orig} \qquad \forall w_i \in W \tag{11}$$

$$\Delta_i^l \geq l_i^{orig} - l_i \qquad \forall w_i \in W \tag{12}$$

$$\Delta_i^r \geq r_i - r_i^{orig} \qquad \forall w_i \in W \tag{13}$$

$$\Delta_i^r \geq r_i^{orig} - r_i \qquad \forall w_i \in W \tag{14}$$

$$l_i - r_{i'} \geq m_s^\gamma \qquad \forall (w_i, w_{i'}), \gamma \in \Gamma \tag{15}$$

$$r_i - l_i \geq m_w^\gamma \qquad \forall \gamma \in \Gamma \tag{16}$$

$$b_{j,j'} = 0 \implies r_i - l_i + S_i \geq m_l^\gamma \quad \forall w_i | w_i = land(v_j, v'_j) \tag{17}$$

$$r_i - l_i + S_i \geq m_l^\gamma \qquad \forall w_i \in Q \tag{18}$$

$$l_i \leq v_j \qquad \forall w_i \in W, v_j \in w_i \tag{19}$$

$$r_i \geq v_j + h_w^{z_i} \qquad \forall w_i \in W, v_j \in w_i \tag{20}$$

We use the cost function of the relaxed 1D minimum layout perturbation problem proposed in [8], which aims at minimizing the change to the layout while fixing the design rule violations. However, we only allow relaxation of the minArea constraints, using the elastic variables S_i. The details are described in the following paragraphs. The non-linear **cost function** in the X-direction is:

$$minimize \sum_i |l_i - l_i^{orig}| + |r_i - r_i^{orig}| + \lambda * \sum_i S_i$$

This cost function is linearized by adding a variable (Δ_i^r) for each edge (r_i), representing the absolute value of the perturbation done to the edge. Constraints (11), (12), (13) and (14) are added accordingly. Summation of the absolute variables are then minimized as shown in Equation (10).

The constraints are explained below.

Design rule constraints. The enforced design rules are minimum width and minimum space on metal layers.

The minimum space constraints are generated between every two opposite edges from different polygons, as shown in S1 constraint in Figure 4. Constraint (15) shows the minimum space constraint. Along the direction of the routing, the length of the segment must not decrease below the minimum width rule. The width of the segment is constrained to be equal to the original width. For example, in Figure 4, MW1, MW2 and MW3 represent the minimum width constraints, which are described in Constraint (16). Non-minimum width polygons are preserved like [8], but non-min-space distances are not preserved.

MinArea and supervia constraints. If a metal segment overlaps with exactly one via above it and one via underneath (hence the segment only exists to be a landing pad for both vias) and the two vias are perfectly aligned, then a supervia can be created. Otherwise, the metal segment has to obey minArea rule. These two cases are shown in Figure 2. This is created as a conditional Constraint (17), which is linearized using Indicator Constraint Transformation [29]. We only allow supervias to replace a stack of via1 and via2 shapes. The general minArea rule is enforced through Constraint (18). In both Constraints (17) and (18), an elastic variable S_i [30] is added to allow the violation of the minArea rule, with a penalty in the cost function.

Via enclosure constraints. The only multi-layer constraints are the via enclosure constraints which are Constraints (19) and (20). These force the via to remain enclosed within its top and bottom metal layers; i.e. all edges of the via lying within the two metal polygons. Accordingly the connectivity is preserved. An example is shown in Figure 4, where the via enclosure constraints are E1 and E2.

IV. Evaluation Flow

We now describe our evaluation flow to demonstrate area benefits of using supervia. Our evaluation flows are: (i) clip-level supervia evaluation flow based on *OptRouteSV*, and (ii) chip-level evaluation flow based on *Legalizer*.

A. Clip-level Evaluation Flow

It is challenging to quantify the benefits of using supervia due to (i) lack of support for double-height vias in commercial detailed routers, and (ii) the large turnaround time of the entire P&R flow. Thus, we use a routing clip-based evaluation framework (*OptRouterSV*), to study the benefit of supervia. Our clip-level evaluation framework provides *routing completion rates*, i.e., how much percentage of routing clips are routable with a particular configuration, for routing clips for the given minArea rule and via option.

For the clip-level evaluation, we use 10 × 10 tracks routing clips extracted from routed designs as input instances. The routing clips are selected based on violating points that cannot be solved by the chip-level legalization method. Figure 5 shows how we convert extracted routing clips to routing problems. We first map all routing and pin segments into a 3D routing map. We then remove internal routing segments, except internal pin segments for incoming and outgoing nets and routing segments at boundaries for feed-through nets. We then run *OptRouterSV* for each routing clip to see whether the clip is routable for a given minArea rule. In our experiments, we run 100 routing clips and check routing completion rate. The details of our experimental setup and results are described in Section V-A. We note that although the clip-level evaluation might not be a proxy for chip-level evaluation, it can provide statistics for multiple evaluation data points which can be used for a projection to chip-level evaluation. More specifically, the routing completion rates can be used to calculate a projected number of minArea violations.

B. Chip-level Evaluation Flow

Although the clip-level evaluation flow can project routability for a given minArea rule and via option, it is not sufficient to derive chip-level benefits in terms of power, performance and area. Thus, using the clip-level evaluation results (e.g., routing completion rates), along with our *Legalizer*, we propose a supervia chip-level evaluation flow.

For a given design, our chip-level evaluation flow offers the maximum achievable utilization for minArea rules and the two via options (supervia and non-supervia). Our chip-level evaluation flow implicitly relies on several "precepts", as follows.

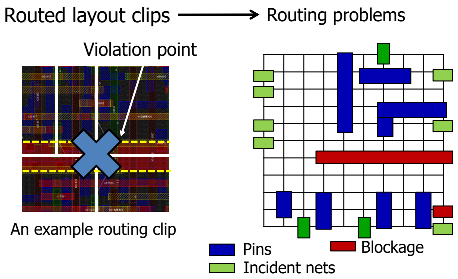


Fig. 5: An example of a routing clip. We convert the extracted routing clips to routing problems by removing internal routing segments except for internal pins and routing segments at the boundaries.

- **Precept 1:** If a design layout that is implemented with a particular initial utilization is DRC-clean for a given minArea rule and a via option, we regard the utilization as an achievable utilization for the given minArea rule and via option.
- **Precept 2:** We regard the difference between achievable utilizations with and without supervias for a particular minArea rule as the area "benefit" of supervias for this minArea rule.
- **Precept 3:** An upper bound of supervia benefits can be obtained from designs implemented with $minArea = 1x$.
- **Precept 4:** Our rule checker acts as a proxy for DRC-cleanliness of a given layout with respect to a given minArea rule and via option.

Precept 1 is used in our initial study on maximum achievable utilizations for different minArea rules. **Precept 2** describes how we evaluate the area benefits of supervia. We study the difference between achievable utilizations in the cases with and without use of supervia since the utilization can be a proxy of the die area. We note that utilization is the "initial" standard cell row utilizations that are used in the floorplan stage. As standard cell areas are from the input gate-level netlists, die areas can be calculated such that it meets the input utilizations. Thus, higher utilization results in smaller die area.

In Precept 3, we note that a layout implemented with $minArea = 1x$ can show the case where supervias are maximally used, relative to $minArea > 1x$. Here, the stacked vias with no routing segments connected to the intermediate metal polygon would resemble supervia. The reason we use layouts implemented with $minArea = 1x$ is that, if we give $minArea > 1x$ rules to P&R tools, the tools will try to legalize every routing segment in terms of the given minArea rules without being aware of supervia; this can prevent supervia from being used. The layouts implemented with $minArea = 1x$ can be either legal or illegal when legalized and checked with respect to $minArea > 1x$ rules.

Precept 4 is explained in Subsection IV-C.

C. Supervia-Aware MinArea Rule Legalizer

As the commercial P&R tool we use in our experiment did not support supervia, we develop our own supervia-aware minArea rule checker using *MentorGraphics Calibre*. We start with a DRC clean $minArea = 1x$ design and legalize the design such that minArea rule is enforced on all metal polygons except the intermediate metal segment of stacked via routes by using our supervia-aware legalization flow and clip-based evaluation results. The supervia-aware legalizer takes routed layouts and minArea rules, via options as inputs and indicates whether the input layout is legal with respect to the input rule and option. The flow of our rule checker is illustrated in Figure 6.

In the flow, the routed layout is given as input to the supervia-aware legalizer, which enforces the given minArea rule and via option. If the legalization is not successful, the legalizer outputs the list of violations which it could not fix. We note that some of the violations from the legalizer can be handled by rip-up-and-reroute using *OptRouterSV*. Thus, the number of violations from the legalizer can further decrease. However, due to runtime limitation of *OptRouterSV*, it is challenging to evaluate routability of thousands of routing clips for each design. Thus, we use the pre-characterized routing completion rates α derived from the clip-based evaluation results to project the number of "real" violations after rip-up-and-reroute using *OptRouterSV*. The projected number of violations is calculated by $inital\ \#violations \cdot (1 - \alpha)$.

D. Overall Flow

With our rule checker and supervia-aware legalizer, we iteratively run P&R with different utilizations until the numbers of minimum violations become less than a threshold. The overall flow is shown in Figure 6. We first run P&R with $minArea = 1x$. We then run our rule checker with the output layouts implemented with $minArea = 1x$, for given minArea rules (e.g., $minArea = 5x, 7x$) and via options (e.g., with and without supervia). If the projected number of violations is less than a threshold (δ), which we assume such numbers of violations are fixable by designers, we regard the layout as a "legalizable" layout with respect to the given minArea rules and via options. Thus, we record the utilizations of the input layouts as achievable utilizations. We repeat the entire flow until we get a legal layout and report the maximum achievable utilization for each minArea rule and via option. We report the minimum of the utilizations from Encounter P&R run and Legalizer for non-supervia case.

P&R flow. The details of P&R flow that we perform are described as follows.

- Floor-planning with an input utilization and placement.
- Perform power signal routing.
- Trial routing is then done.
- PreCTS optimization is conducted after RC extraction.
- Perform CTS
- After post CTS optimization, global and detailed routing are performed.

Denoising. Validation of "achievable" maximum utilization is very difficult since (i) we have a lot of SP&R knobs (e.g., timing/power constraints, optimization effort levels, etc.), (ii) many data points are required to conclude the maximum achievable number and (iii) there exists tool noise [32] [33]. In our experiment, we use 1% as the step of input utilizations. To eliminate tool noise, we validate each utilization point by executing P&R run five times, perturbing the utilization by a small amount (specifically, +0.1, +0.05, +0.0, -0.05, -0.1). If any one of the perturbed-utilizations runs is feasible (i.e., DRC-clean routing), we regard the corresponding utilization point as achievable.

V. Experimental Setup and Results

In this section, we present our experimental setup and results. We experiment on the utilization used to route the design and the number of violations that exist after each stage of the flow. These experiments show the chip-scale benefit of using supervia against conventional non-supervia case.

A. Experimental Setup

In our experiments, we use two designs (AES and MIPS) from OpenCores [37] and an ARM Cortex M0 design (M0). We synthesize the RTL netlists of the testcases using *Cadence RTL Compiler 11.20* [35] and then perform P&R with an abstracted 7nm library from an industrial IP provider using *Cadence Encounter 11.10* [34] and *Cadence Innovus 16.10*. The testcase information is summarized in Table IV. The naming convention follows '{design name}_{#metal layer}'. **#Inst** column shows the total number of instances, **#Nets** column shows the total number of nets; fourth to eighth columns show the number of vias, e.g., V12, V23, V34, V45 and V56. Each testcase is implemented with $minArea = 1x$, with the maximum achievable utilization. The utilization numbers are reported in Table I.

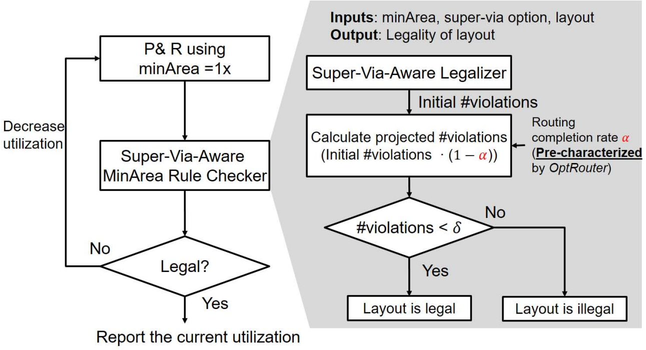

Inputs: minArea, super-via option, layout
Output: Legality of layout
P& R using minArea =1x
Decrease utilization
Super-Via-Aware MinArea Rule Checker
No
Legal?
Yes
Report the current utilization
Super-Via-Aware Legalizer
Initial #violations
Calculate projected #violations (Initial #violations · (1 − α))
Routing completion rate α (Pre-characterized by OptRouter)
#violations < δ
No
Yes
Layout is legal
Layout is illegal

Fig. 6: Chip-level evaluation flow for supervias with supervia-aware minArea rule checker.

TABLE IV: Testcase.

Testcase	Inst	Nets	V1	V2	V3	V4	V5
MIPS_4	11330	7942	29320	41637	13672	0	0
MIPS_5	11345	7942	29395	38502	14421	3737	0
M0_4	79059	9709	34579	50429	13672	0	0
M0_5	15209	9826	34573	46427	14421	5784	0
M0_6	14555	9769	34398	44921	14090	5748	2864
AES_5	20366	13867	48480	65017	20512	9302	0
AES_6	18410	13858	48362	61437	20203	9430	4533

OptRouterSV and *Legalizer* are written in C++ with *OpenAccess 2.2.4* API to support LEF/DEF and extract the polygon shape information from the routed layout. We use *CPLEX 12.6.3* [36] as our MILP solver. Geometric operations in the Legalizer are performed using the Boost library.

B. Clip-level routability assessment

We study routing completion rate for different minArea rules, along with supervia (SV) option and non-supervia (non-SV) options across different designs. The results are shown in Table V. We run *OptRouterSV* for 100 routing clips for each testcase. 100% means that 100 clips among the total 100 clips have legal routing with respect to a particular minArea rule and via option.

For $minArea = 3x$, all designs show close to 100% routing completion rate for both SV and non-SV cases. For $minArea = 5x$, we observe 77%∼99% completion rates. We note that M0_4 is a special case; the utilization is exceptionally low (i.e., 65%) as shown in Table I. Due to the low utilization, standard cells are placed sparsely and thus routability and pin accessiblity are improved, which also helps to legalize routing segments for the minArea rules. We observe that as the number of used metal layers increases, the completion rates decreases, except for M0_4,5,6, $minArea = 7x$, SV case.

From the results, one can see that there is no need for supervias in the $minArea = 3x$ case though need for routers to handle minimum area rule efficiently is exhibited. On the other hand there is a dramatic improvement in routability when $5x$ or $7x$ rules are used. Clip completion rates improve almost by a factor of two in the $minArea = 7x$ case showing the promise of supervias.

C. Chip-level area assessment

First we discuss the experimental results obtained using *Cadence Encounter v11.10*. We study the potential chip-scale utilization

TABLE V: Routing completion rate results.

	3x		5x		7x	
Testcase	non-SV	SV	non-SV	SV	non-SV	SV
MIPS_4	100%	100%	87%	97%	57%	88%
MIPS_5	100%	100%	87%	96%	50%	77%
AES_5	100%	100%	85%	95%	29%	50%
AES_6	100%	100%	77%	90%	34%	71%
M0_4	100%	100%	99%	100%	71%	87%
M0_5	99%	99%	87%	99%	39%	70%
M0_6	99%	99%	46%	79%	40%	74%

benefit of using supervia across different designs and different utilization points. We use 5 layers for AES and CORTEX M0 while 4 layers for MIPS. Using additional metal layers makes routing too relaxed and likely over-provisioned while one less routing layer makes utilization unreasonably low (see Table I). Table VI shows the number of routing violations after legalization and the projected number of violations assuming OptRouterSV fixed all the post-legalization problems. The substantial routability advantage of using Supervias is again obvious from the large difference between unfixed violations in most cases (especially $minArea = 5x, 7x$). To interpret these results as an area improvement, we call the layout "clean" if the number of projected violations is less than a threshold (i.e., manageably small that a human designer could fix them). The projected utilization improvement is shown in Table VII. We can make the following striking observations.

- Supervias are *not* useful if the minArea rule can be kept at $3x$ or smaller.
- For $minArea = 5x$, the density improvements range from 0% to 16%.
- For $minArea = 7x$, the improvement can be dramatic: as high as 20% points or more. Further without using supervias, routing may not be feasible at all for designs at any reasonable utilization. Such a low utilization would take away any incentive to scale down technology at all. For example in case of MIPS utilization drops from 96% to 50% (threshold = 500 violations) when going from $5x$ to $7x$ rule. This is a full technology node worth of benefit lost. Supervias allow us to recover that benefit back completely to get the same 96% utilization.

Now we discuss the experiments performed using *Cadence Innovus* (with optimal via generation option enabled) and supervia-aware legalizer. It is noteworthy that the maximum area benefit coming from supervia in this case is going to be much smaller than the results obtained using *Encounter*, since the maximum utilization/area hit due to minArea rule is at maximum 7% for CORTEX_M0 while doing P&R using *Innovus*.

In figure 7 and figure 8, total number of DRC violations vs utilization is shown for the following cases:

- Innovus run with 5x minArea rule
- Legalizer run with supervia option
- Legalizer run with without supervia option

As seen from these figures, there is large gap between legalizer no supervia and supervia case which indicates that supervia helps in chip-level routing. However, a fair comparison would be to compare legalization for 5x minArea rule with supervia option (green curve) against Innovus 5x (red curve) run. It is observed that the benefits of supervia is marginal ($\sim$2%), this is because the number of DRC violations with supervia using the legalizer i.e. the green curve is lower than that of supervia oblivious routing using Innovus' red curve for only a few higher utilization points, after which the red curve gets better than the green curve. The key factors resulting in this trend are the following:

- Innovus minArea aware via generation and router

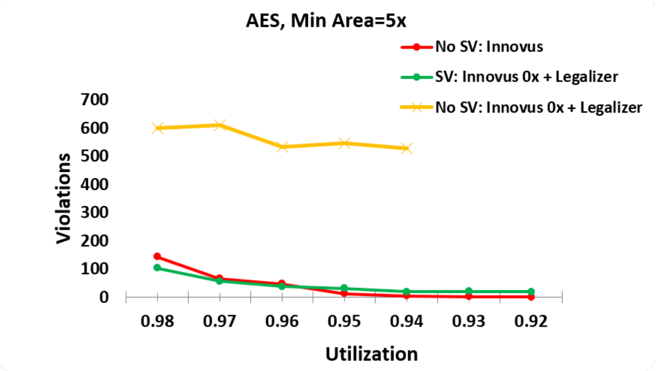


Fig. 7: Chip Level Evaluation for AES where the "minArea = 1x" design from Innovus is legalized for 5x minArea rule

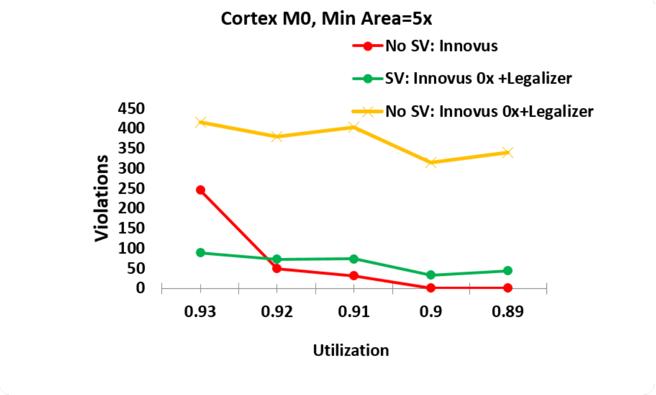


Fig. 8: Chip Level Evaluation for CORTEX_M0 where the "minArea = 1x "design from Innovus is legalized for 5x minArea rule

- Legalization is not the optimal tool to show the benefits of supervia due its numerous limitations.

Thus, we can conclude from these results that a minArea aware PR can recover most of the area hit coming from aggressive minArea rules in advanced technology nodes. However, from the small design testcases we used, we can observe that about $\sim$2% density benefit can be achieved using supervia for CORTEX_M0.

VI. Conclusions

In this work, we study the impact of minArea rule on achievable utilization and show how the minArea rules can result in area penalty for a design. To compensate the area penalty, we propose the use of the supervia. The benefit of supervia is evaluated by (i) routing clips using an optimal ILP-based router (*OptRouterSV*) and (ii) chip-level evaluation using a supervia-aware legalizer on commercial router results. Our results indicate that though the supervia technology primitive is almost indispensable in absence of minArea aware routing and via generation when minArea rules are large (5x or greater), most of the area benefit can be recovered by a minArea aware P&R. Additionally, using supervia provides up to $\sim$2% extra area benefit which might not be motivating enough for supervia adoption.

Thus, in conclusion supervia isn't promising enough for digital logic routing. However, supervia can be an interesting option to consider for density scaling in the following few applications/ use-cases among possibly many:

- STT-RAM based memory cell design where the memory cells are placed in the BEOL stack and supervias can be used to

TABLE VI: Results of supervia-aware chip-level evaluation flow. VioL denotes the number of violations after legalization. α denotes the routing completion rates from Table V. VioP is the projected number of violations assuming all post-legalization violations are fixed by OptRouterSV. The maximum utilization for every design in the table corresponds to the maximum achievable utilization in the $minArea = 1x$ case. The lowest utilization we report the results for corresponds to the case where VioP < 200 for the non-supervia case; or utilization drops below an unreasonable 50% or the commercial router is able to route the design without supervias.

minArea	Testcase	supervia				Non-supervia			
		Util	#VioL	α	#VioP	Util	#VioL	α	#VioP
3x	AES_5	92%	0	1	0	92%	2074	1	0
5x	AES_5	92%	122	0.95	7	92%	2827	0.85	425
5x	AES_5	87%	123	0.95	7	87%	2714	0.85	408
5x	AES_5	83%	117	0.95	6	83%	2514	0.85	378
7x	AES_5	92%	1725	0.50	862.5	92%	5814	0.29	4128
7x	AES_5	87%	1582	0.50	791	87%	5329	0.29	3784
7x	AES_5	83%	1328	0.50	664	83%	5179	0.29	3678
3x	MIPS_4	96%	0	1	0	96%	1360	1	0
5x	MIPS_4	95%	61	0.96	3	95%	1719	0.87	224
5x	MIPS_4	91%	60	0.96	3	91%	1663	0.87	217
5x	MIPS_4	88%	64	0.96	3	88%	1649	0.87	215
5x	MIPS_4	83%	52	0.96	3	83%	1527	0.87	199
7x	MIPS_4	95%	856	0.77	197	95%	3223	0.50	1612
7x	MIPS_4	90%	729	0.77	168	90%	3076	0.50	1538
7x	MIPS_4	86%	758	0.77	175	86%	3014	0.50	1507
7x	MIPS_4	70%	427	0.77	99	70%	2576	0.50	1288
7x	MIPS_4	50%	410	0.77	95	50%	2162	0.50	1081
3x	M0_5	84%	0	1	0	84%	911	1	0
5x	M0_5	84%	55	0.99	1	84%	2102	0.87	274
5x	M0_5	78%	66	0.99	1	78%	1946	0.87	256
5x	M0_5	73%	55	0.99	1	73%	1743	0.87	227
5x	M0_5	60%	52	0.99	1	60%	1486	0.87	194
7x	M0_5	84%	963	0.71	278	84%	3906	0.41	2305
7x	M0_5	78%	887	0.71	258	78%	3672	0.41	2167
7x	M0_5	73%	782	0.71	227	73%	3280	0.41	1936
7x	M0_5	60%	642	0.71	186.6	60%	2766	0.41	187
7x	M0_5	50%	623	0.71	181.6	50%	2673	0.41	1578

TABLE VII: Projected Utilization Improvements with supervia. Threshold1 = 200 violations; Threshold2 = 500 violations

Testcase	minArea	Supervia		Non-supervia	
		Threshold1	Threshold2	Threshold1	Threshold2
AES_5	3x	92%	92%	92%	92%
AES_5	5x	92%	92%	79%*	79%*
AES_5	7x	<50%	70%	<50%	<50%
MIPS_4	3x	96%	96%	96%	96%
MIPS_4	5x	96%	96%	83%	96%
MIPS_4	7x	96%	96%	<50%	<50%
M0_5	3x	84%	84%	84%	84%
M0_5	5x	84%	84%	68%*	84%
M0_5	7x	60%	84%	<50%	<50%

provide access connection directly from the FEOL contact layer to the logic cells [38].

- Since supervias are double-height vias realized by digging through multiple layers of SiO_2 at once, the resistive barrier layer on top of the bottom via can now be eliminated. This might find interest in on-chip power-distribution network design, since via resistance is becoming a major issue in the advanced nodes where the via dimensions are shrinking resulting in increased via resistance.

References

[1] C. Bachelu and M. Lefebvre, "A Study of The Use of Local Interconnect in Cmos Leaf Cell Design", *IEEE TVLSI* 1(4) (1993), pp. 566-571.

[2] Q. Chen, J. A. Davis, P. Zarkesh-Ha and J. D. Meindl, "A Compact Physical Via Blockage Model", *IEEE TVLSI* 8(6) (2000), pp. 689-692.

[3] M. Clarke, D. Hammerschlag, M. Rardon and A. Sood, "Eliminating Routing Congestion Issues with Logic Synthesis", *Whitepaper*, Cadence Design Systems,

[4] J. S. Deogun and B. B. Bhattacharya, "Via Minimization in VLSI Routing with Movable Terminals", *IEEE TCAD* 8(8) (1989), pp. 917-920.

[5] B. A. Ganji and B. Y. Majlis, "Deep Trenches in Silicon Structure using Drie Method with Aluminum as An Etching Mask", *IEEE International Conference on Semiconductor Electronics*, 2006, pp. 41-47.

[6] R. S. Ghaida et al, "Comprehensive Die-Level Assessment of Design Rules and Layouts", *Proc. ASP-DAC*, 2014, pp. 61-66.

[7] K. Han, A. B. Kahng and H. Lee, "Evaluation of BEOL Design Rule Impacts Using An Optimal ILP-Based Detailed Router", *Proc. DAC*, 2015, pp. 1-6.

[8] F.-L. Heng, Z. Chen and G. E. Tellez, "A VLSI Artwork Legalization Technique Based On A New Criterion of Minimum Layout Perturbation", *Proc. ISPD*, 1997, pp. 116-121.

[9] F. Inoue et al, "Novel Seed Layer Formation Using Direct Electroless Copper Deposition on ALD-Ru Layer for High Aspect Ratio TSV", *IEEE International Interconnect Technology Conference*, 2012, pp. 1-3.

[10] T. Kauerauf et al, "Reliability of MOL Local Interconnects", *Proc. IRPS*, 2013, pp. 2F.5.1-2F.5.5.

[11] T. H. Lee and T. C. Wang, "Congestion-Constrained Layer Assignment for Via Minimization in Global Routing", *IEEE TCAD* 27(9) (2008), pp. 1643-1656.

[12] Y. Lin, B. Yu, B. Xu and D. Z. Pan, "Triple Patterning Aware Detailed Placement Toward Zero Cross-Row Middle-of-Line Conflict", *Proc. ICCAD*, 2015, pp. 396-403.

[13] J. Mao et al, "Scaling of Copper Seed Layer Thickness Using Plasma-Enhanced ALD and Optimized Precursors", *IEEE Transactions on Semiconductor Manufacturing* 26(1) (2013), pp. 17-22.

[14] D. Y. Moon et al, "Copper Seed Layer Using Atomic Layer Deposition for Cu Interconnect", *Proc. INEC*, 2010, pp. 450-451.

[15] T. Nieberg, "Gridless Pin Access in Detailed Routing", *Proc. DAC*, 2011, pp. 170-175.

[16] M. Rashed et al, "Innovations in Special Constructs for Standard Cell Libraries in Sub 28nm Technologies", *Proc. IEDM*, 2013, pp. 9.7.1-9.7.4.

[17] Z. Sanae, S. A. M. Poudineh, M. Mehran and S. Mohajerzadeh, "High-Aspect-Ratio Deep Si Etching of Micro/Nano Scale Features with SF6/H2/O2 Plasma, In a Low Plasma Density Reactive Ion Etching System", *Nanotechnology: Electronics, Devices, Fabrication, MEMS, Fluidics and Computational*, 2 (2011), pp. 325-328.

[18] D. Shi, E. Tashjian and A. Davoodi, "Dynamic Planning of Local Congestion from Varying-Size Vias for Global Routing Layer Assignment", *Proc. ASP-DAC*, 2016, pp. 372-377.

[19] T. Taghavi, F. Dabiri, A. Nahapetian and M. Sarrafzadeh, “Tutorial on Congestion Prediction”, *Proc. SLIP*, 2007, pp. 15-24.

[20] T. Taghavi et al, “New Placement Prediction and Mitigation Techniques for Local Routing Congestion”, *Proc. ICCAD*, 2010, pp. 621-624.

[21] X. M. Xiong and E. S. Kuh, “A Unified Approach to The Via Minimization Problem”, *IEEE TCAS* 36(2) (1989), pp. 190-204.

[22] Q. Xu et al, “Enhanced Etch Process for TSV & Deep Silicon Etch”, *Proc. ASMC*, 2015, pp. 426-428.

[23] X. Xu et al, “Self-Aligned Double Patterning Aware Pin Access and Standard Cell Layout Co-Optimization”, *IEEE TCAD* 34(5) (2015), pp. 699-712.

[24] S.-Q. Wang, “Barriers Against Copper Diffusion into Silicon and Drift Through Silicon Dioxide”, *MRS Bulletin* 19(8) (1994), pp. 30-40.

[25] X. Xu et al, “Parr: Pin Access Planning and Regular Routing for Self-Aligned Double Patterning”, *Proc. DAC*, 2015, pp. 1-6.

[26] S. X. Zhang, S. W. R. Lee, L. T. Weng and S. So, “Characterization of Copper-To-Silicon Diffusion for the Application of 3D Packaging with Through Silicon Vias ”, *Proc. International Conference on Electronic Packaging Technology*, 2005, pp. 51-56.

[27] Y. Xu, Y. Zhang and C. Chu. “FastRoute 4.0: Global Router with Efficient Via Minimization”, *Proc. ASP-DAC*, 2009, pp. 576-581.

[28] B. Yu et al. “Layout Decomposition for Triple Patterning Lithography”, *Proc. ICCAD*, 2011, pp. 1-8.

[29] A. Agarwal, S. Bhat, Sooraj A. Gray and I. E. Grossmann, “Automating mathematical program transformations”, *Practical Aspects of Declarative Languages*, 2010, Springer, pp. 134-148.

[30] J. W. Chinneck, “Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods”, Springer Science & Business Media, 2007(118).

[31] N. Nagaraj, and T. Bonifield, A. Singh, R. Griesmer, and P. Balsara, “Interconnect Modeling for Copper/Low-K Technologies”, *Proc. Intl. Conf. on VLSI Design*, 2004, pp. 425-427.

[32] A. B. Kahng and S. Mantik, “Measurement of Inherent Noise in EDA Tools”, *Proc. International Symposium on Quality in Electronic Design*, March 2002, pp. 206-211.

[33] K. Jeong and A. B. Kahng, “Methodology From Chaos in IC Implementation”, *Proc. International Symposium on Quality Electronic Design*, 2010, pp. 885-892.

[34] Cadence Encounter Digital Implementation System, http://www.cadence.com/products/di/edi_system/pages/default.aspx

[35] Cadence RTL Compiler, http://www.cadence.com/products/ld/rtl_compiler/pages/default.aspx

[36] IBM ILOG CPLEX. www.ilog.com/products/cplex/

[37] OpenCores. http://opencores.org/

[38] Appeltans, Raf and Weckx, Pieter and Raghavan, Praveen and Kim, Ryoung-Han and Kar, Gouri Sankar and Furnmont, Arnaud and Van der Perre, Liesbet and Dehaene, Wim, “The effect of patterning options on embedded memory cells in logic technologies at iN10 and iN7”, *Proc. SPIE*, 2017, pp. 101480G-101480G-13.