

Self-Aligned Double Patterning-Aware Detailed Routing with Double Via Insertion and Via Manufacturability Consideration

Yixiao Ding
Department of Electrical and
Computer Engineering
Iowa State University
Ames, IA, 50010, USA
yxding@iastate.edu

Chris Chu
Department of Electrical and
Computer Engineering
Iowa State University
Ames, IA, 50010, USA
yxding@iastate.edu

Wai-Kei Mak
Department of Computer
Science
National Tsing Hua University
Hsinchu, Taiwan 30013
wkmak@cs.nthu.edu.tw

ABSTRACT

In 10nm technology node, self-aligned double patterning (SADP) and triple patterning lithography (TPL) allow us to achieve minimum wiring pitch of around 45nm. While metal layers can be printed by SADP, via layer manufacturing requires TPL to maintain design rules. SADP-aware detailed routing is proposed to ensure decomposability of metal layer patterns. However, its routing solution does not automatically guarantee TPL decomposable via layers. Vias have an inherently low reliability and via failure causes a great yield loss. Double via insertion (DVI) is an effective means to increase yield by reducing via failures. With the restriction of SADP design rules and consideration of TPL decomposability for via layers, DVI becomes a more challenging problem. In this paper, we consider DVI and via layer TPL manufacturability simultaneously in SADP-aware detailed routing. The experimental results demonstrate our router can obtain 100% routability and TPL decomposable via layers with reduced dead via count.

1. INTRODUCTION

Triple patterning lithography (TPL) and self-aligned double patterning (SADP) allow us to achieve a minimum wiring pitch of around 45nm for the metal layers in 10nm technology node. In litho-etch-litho-etch-litho-etch (LELELE) type TPL, three masks are used in three exposure/etching processes. However, the practical resolution limit of LELELE is significantly larger than 1/3 that of the single exposure due to the misalignment errors between exposures. With the self-alignment property, SADP can achieve comparable wiring pitch as TPL with the use of only two masks (mandrel mask and cut/trim mask). In this paper, we focus on the problem of SADP-aware detailed routing.

In order to print adjacent vias in the 10nm technology node, we note that TPL is required. It is because the minimum width and minimum spacing constraints of the cut/trim

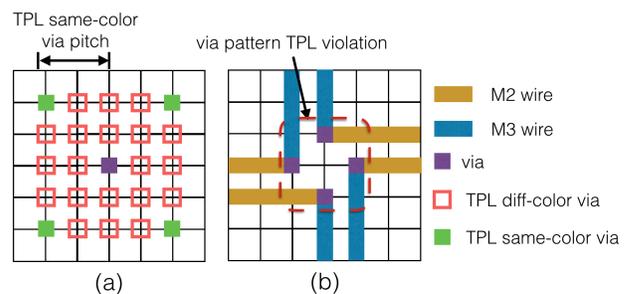


Figure 1: Via layer manufacturability by TPL. (a) TPL same-color via pitch. (b) SADP-aware routing with via layer TPL violation.

mask of SADP prohibit the printing of two tiny features so close to each other. In 10nm technology node, the same-color via pitch (i.e., the minimum center-to-center distance of a pair of vias that can be assigned to the same mask in TPL) is slightly larger than two wiring tracks as shown in Fig. 1(a) [1]. As a result, SADP-aware detailed routing does not automatically guarantee TPL decomposable via layers. Fig. 1(b) shows a SADP-aware detailed routing solution which produces a via pattern with TPL violation. Therefore, it is necessary for SADP-aware detailed routing to consider the TPL decomposability of via layers. Otherwise, the TPL violation will make the via layers not manufacturable.

Vias inherently have low reliability (e.g., due to stress related via voids) and via failures cause a great yield loss in chip manufacturing [2]. Double via insertion (DVI), which inserts a redundant via adjacent to a single via, is an effective means to increase yield and improve reliability. We call the single via that cannot have a redundant via without design rule violation a *dead via*. Post-routing stage DVI is limited by the inherent dead vias in detailed routing solution. To effectively reduce the dead via count, considering DVI during detailed routing stage is helpful. In Fig. 2(a), a redundant via is inserted for *via b* while *via a* is a dead via in the detailed routing solution. On the contrary, if the routing considers DVI as shown in Fig. 2(b), all vias can be protected by redundant vias. However, with the restriction of SADP design rules and consideration of TPL decomposability of via layers, DVI becomes an even more challenging problem. In this work, we address the SADP-aware detailed

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DAC '16, June 05-09, 2016, Austin, TX, USA

© 2016 ACM. ISBN 978-1-4503-4236-0/16/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2897937.2898088>

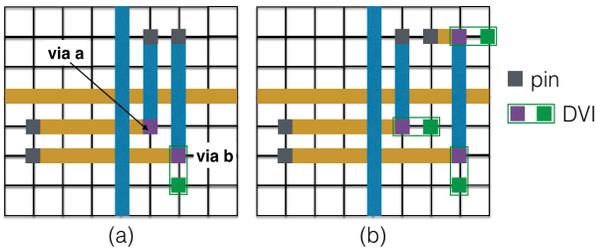


Figure 2: DVI in detailed routing. (a) Without DVI consideration. (b) With DVI consideration.

routing problem that simultaneously considers both DVI and via manufacturability.

There are several previous works on SADP-aware detailed routing [3, 4, 5, 6, 7, 8]. However, all these previous works ignore via layer manufacturability in 10nm technology node. DVI in post-routing stage is studied by [9, 10, 11, 12, 13, 14]. However, in this stage only slight layout modification is allowed, this methodology will restrict the feasibility of DVI. Thus, considering DVI during detailed routing stage is proposed in [15, 16, 17, 18]. [15] formulates the problem as a multi-constrained shortest path problem solved by a Lagrangian relaxation technique, which has a high time complexity. Meanwhile, the double via constraint for each net greatly restricts routability. [16, 17] considers DVI based on gridless routing model while grid based detailed routing is applied in our paper. Double patterning lithography (DPL)-aware detailed routing with redundant via insertion is considered in [18]. However, it is litho-etch-litho-etch type DPL targeting 32nm/22nm technology node. Moreover, it does not give the exact cost function used to consider redundant via insertion in detailed routing.

In this paper, we study SADP-aware detailed routing considering DVI and via layer TPL manufacturability. Our major contributions are summarized as follows:

- This is the first work to consider DVI in SADP-aware detailed routing.
- This is the first work to consider via layer manufacturability by TPL in 10nm technology node during detailed routing. The via layers in our routing solution is ensured to be TPL decomposable.
- This is the first work to consider TPL design rules when performing DVI in post-routing stage.
- The experimental results demonstrate the effectiveness and efficiency of our algorithm. Furthermore, the overhead of considering both DVI and via layer manufacturability during routing stage is minimal.

The rest of the paper is organized as follows. Section 2 presents our problem formulation and some preliminaries. The overall flow and details of our proposed solution are presented in Section 3. Section 4 shows our experimental results, and finally Section 5 concludes the paper.

2. PRELIMINARIES

2.1 Problem formulation

We assume that there is a preferred routing direction for each layer and the other direction perpendicular to the preferred routing direction is defined as non-preferred routing direction. We do not completely disable routing in the non-preferred routing direction. We refer to this routing be-

havior as restricted detailed routing. Given a placed netlist, a multi-layer routing grid, and a set of design rules, we perform restricted detailed routing to generate a legal routing solution. The objective is to minimize the total wire length, via count, and dead via count while achieving 100% routability. Moreover, in the final routing solution, the metal layer patterns should be compliant to SADP design rules, and via layers should be TPL decomposable.

2.2 Color pre-assignment approach

Considering SADP in detailed routing is a challenging problem due to the non-intuitive SADP layout decomposition. The idea of color pre-assignment is proposed in [3] to simplify the problem, and we adopt this approach for our SADP-aware detailed routing in this paper. Before detailed routing, the routing grid is pre-assigned with colors as shown in Fig. 3(a). The colored routing grid determine where mandrel and cut/trim mask patterns may form. In this way, the SADP layout decomposition is known at the moment when a net is routed. Hence, the occurrence of design rule violation is foreknown during detailed routing and can be easily minimized. [3] defines a preferred turn, a feasible turn, and a forbidden turn according to the SADP manufacturing cost, which are shown in Fig. 3(b). A forbidden turn is a L-shaped metal layer pattern not allowed in detailed routing due to design rule violation. Thus, the forbidden turn should be strictly avoided in order to guarantee a SADP decomposable layout.

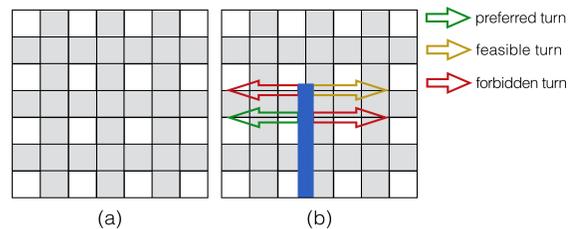


Figure 3: Color pre-assignment approach. (a) Colored routing grid prepared for SADP-aware detailed routing. (b) Restriction on the L-shape routing pattern

2.3 DVI feasibility

The DVI process is to add a redundant via adjacent to a single via without design rule violation. In Fig. 4(a), given a single via v , four adjacent via locations a , b , c , and d are candidate locations to insert a redundant via for v . So, there are four *DVI candidates* for via v . For simplicity, we call them DVICs. To connect to the inserted redundant via, metal wire may extend from the single via location to a DVIC location as shown in Fig. 4(b). In Fig. 4(b), a forbidden turn occurs on M3 when connecting to DVIC d , which makes the metal layer pattern not manufacturable. Thus, this DVIC is not feasible. For the same reason, DVIC c is also not feasible due to the occurrence of a forbidden turn on M2. Meanwhile, DVIC b is not feasible since the location is occupied by a metal layer pattern from another routed net. Thus, only DVIC a in Fig. 4(a) is feasible.

The DVIC feasibility of each single via is determined both by where the single via locates and how the metal wires connected by the single via are routed. Given a single via from a routed net, it is easy to find all its feasible DVICs. Due to page limit, we do not enumerate all the cases here.

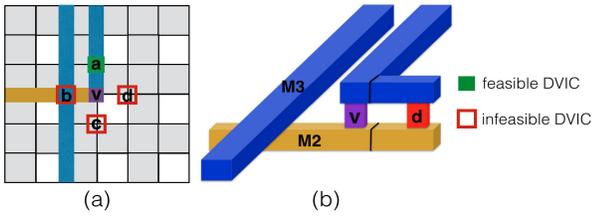


Figure 4: Double via insertion. (a) Each single via has four DVICs. (b) The DVIC d is infeasible.

2.4 Forbidden via pattern

The TPL decomposition of via layers can be transformed to a 3-coloring problem on the decomposition graph [19]. However, maintaining a decomposition graph during detailed routing is expensive in terms of runtime and memory usage, and the 3-coloring problem is NP-Complete. Instead, we propose to examine all subregions of size 3x3 in the routing grid and extract the via pattern within it. Then, determining if the via pattern is 3-colorable can be done in $O(1)$ time. If the via patterns in all 3x3 subregions in the routing grid are 3-colorable, then the decomposition graph is highly likely to be 3-colorable. We define a *forbidden via pattern* as a via pattern within a 3x3 subregion which is not 3-colorable. For simplicity, we refer to it as FVP. All FVPs can be categorized by via count and how they are distributed within 3x3 subregion as follows:

1. Via patterns with 6 or more vias are all FVPs.
2. For via patterns with via count equal to 5, unless 4 of 5 the vias are on four corners of the 3x3 subregion, they are FVPs. Fig. 5(a) shows a non-FVP via pattern with 5 vias and Fig. 5(b) shows an FVP via pattern with 5 vias.
3. For via patterns with via count equal to 4, unless 2 of 4 the vias are on diagonally opposite corners of the 3x3 subregion, they are FVPs. Fig. 5(c) shows a non-FVP via pattern with 4 vias and Fig. 5(d) shows an FVP via pattern with 4 vias.
4. Via patterns with 3 or fewer vias are not FVPs.

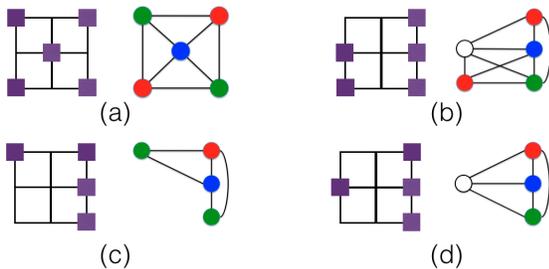


Figure 5: Via patterns in 3x3 subregion. (a) A via pattern with 5 vias which is not an FVP. (b) An FVP with 5 vias, in which one via is uncolorable. (c) A via pattern with 4 vias which is not an FVP. (d) An FVP with 4 vias, in which one via is uncolorable.

3. PROPOSED SOLUTION

3.1 Overall flow

The overall flow is shown in Fig. 6. The inputs are a placed netlist, a multi-layer routing grid, and a set of design

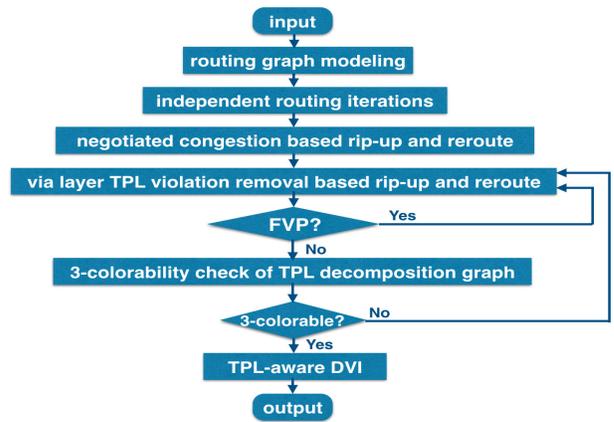


Figure 6: Overall flow.

rules. The routing graph modeling, independent routing iteration, and negotiated congestion based rip-up and reroute are explained with details in [3]. Different from the previous work, we incorporate a cost assignment scheme to consider DVI and via layer TPL decomposability during routing of each single net. The major advantage of this approach is the overhead of considering DVI and via layer TPL decomposability is minimized. Thus, the SADP-aware detailed routing can keep high performance as before. Then, we propose a via layer TPL violation removal based rip-up and reroute to eliminate all FVPs. After that, we will do a fast 3-colorability check of TPL decomposition graph. If not 3-colorable, the rip-up and reroute is called to fix any remaining coloring conflicts. If yes, DVI considering via layer TPL decomposability is performed in post-routing stage. The final output is SADP-friendly detailed routing solution which via layers are guaranteed to be TPL decomposable.

3.2 Single net routing considering DVI and via layer TPL

Similar to the graph model in [3], we view each grid segment and via as a vertex. An edge exists between two vertices if they are directly connected in the routing grid. A cost is associated with each edge to indicate the expense of routing from vertex in one end to the vertex on the other end. To consider DVI and via layer TPL decomposability during routing stage, potential dead vias and via patterns with TPL violations should be penalized during single net routing. Thus, a cost assignment scheme is designed to incorporate additional cost to the graph model after routing of each net. Fig. 7 helps explain how the cost assignment scheme works, and Algorithm 1 is the pseudocode describing how additional costs are added to the routing graph G .

During sequential routing, the routing impact of a new net on the DVI feasibility of routed nets and itself are different. Suppose via_u of a routed net_i has two feasible DVICs a and b as shown in Fig. 7(a), net_j is the new net to be routed. For the routed net_i , as long as net_j is routed across via location a or b , the feasible DVIC count of via_u is reduced. Thus, the cost assignment scheme penalizes the routing net_j across via at location a or b . The penalty cost can be computed by $DBC/(\# \text{ of feasible DVICs of } via_u)$, where DBC stands for DVIC block cost, which is a constant. In this way, routing resource used by DVI for the via with smaller feasible DVIC count is assigned with higher cost, and vias from routed nets

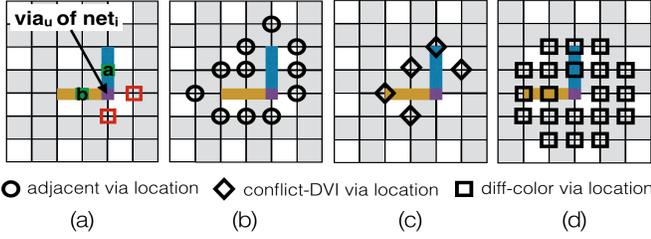


Figure 7: An example to illustrate how cost assignment algorithm works.

can be prevented from becoming dead vias.

For the net_j being routed, if it is routed with a via_v at any via location adjacent to routed nets that we called adjacent via location as shown in Fig. 7(b), it will probably reduce the DVI feasibility of via_v . Thus, the cost assignment scheme penalizes the routing net_j using vias at adjacent via locations. The penalty cost is referred as adjacent via cost (AVC), which is a constant. Again for the net_j being routed, if it is routed with a via_v at any conflict-DVI via location shown in Fig. 7(c), one DVIC of via_v is in conflict with one feasible DVIC of via_u , since they share the same via location. Meanwhile, vias with fewer feasible DVICs will have higher chance to be in conflict during DVI process. Thus, the cost assignment scheme penalizes the routing net_j using vias at conflict-DVI via locations. The penalty cost can be computed by $CDC/(\# \text{ of feasible DVICs of } via_u)$, where CDC stands for conflict-DVI cost, which is a constant. In summary, the cost assignment scheme applies addition costs to penalize routing which will reduce DVI feasibility of both routed nets and the new net to be routed, thus fewer dead vias will occur.

To avoid TPL violation of via patterns, the cost assignment scheme applies TPL cost (TPLC) to penalize routing net_j using vias at locations less than the TPL same-color via pitch from via_u of routed net_i which we call diff-color via location in Fig. 7(d). The TPLC is a constant.

3.3 Via layer TPL violation removal based rip-up and reroute

TPL decomposability of the via layers should be a hard constraint for detailed routing solution. However, the cost assignment scheme only discourages the occurrence of via patterns with TPL violation. Thus, we propose the via layer TPL violation removal based rip-up and reroute presented in Algorithm 2. Similar to the negotiated congestion based rip-up and reroute in [3], we also apply a base cost (BC), a usage cost (UC), and a history cost (HC). As mentioned before, maintaining a decomposition graph during rip-up and reroute iterations is expensive in terms of runtime and memory usage. Moreover, the problem of deciding if a decomposition graph is 3-colorable is NP-Complete. Thus, we target to remove TPL violation from via layers by eliminating all FVPs through rip-up and reroute iterations. The major advantage of this approach is finding all the FVPs in the layout is $O(p \times q)$ and updating FVPs after a rip-up and reroute is $O(n)$, where n is via count in the routed net, and $p \times q$ is the size of routing grid.

A couple of techniques are applied in order to obtain a faster convergence of rip-up and reroute. Firstly, before rip-up and reroute starts, some via locations are blocked in line 2 of Algorithm 2 to prevent reroute from generating new FVPs. The blocked via locations are updated in lines 7 and 10 after

```

for each  $via_u$  of routed  $net_i$  do
  for each  $via_k$  at feasible DVIC location of  $via_u$  do
    find  $vertex_k$  in  $G$  represents  $via_k$ ;
    for each  $vertex_m$  adjacent to  $vertex_k$  in  $G$  do
      for each edge  $e$  incident to  $vertex_m$  do
        cost( $e$ ) +=  $\frac{DBC}{(\# \text{ of feasible DVICs of } via_u)}$ ;
      end
    end
  end
  for each  $via_k$  at conflict-DVI via location of  $via_u$  do
    find  $vertex_k$  in  $G$  represents  $via_k$ ;
    for each edge  $e$  incident to  $vertex_k$  do
      cost( $e$ ) +=  $\frac{CDC}{(\# \text{ of feasible DVICs of } via_u)}$ ;
    end
  end
  for each  $via_k$  at diff-color via location of  $via_u$  do
    find  $vertex_k$  in  $G$  represents  $via_k$ ;
    for each edge  $e$  incident to  $vertex_k$  do
      cost( $e$ ) += TPLC;
    end
  end
end
for each  $via_k$  at adjacent via location of  $net_i$  do
  find  $vertex_k$  in  $G$  represents  $via_k$ ;
  for each edge  $e$  incident to  $vertex_k$  do
    cost( $e$ ) += AVC;
  end
end

```

Algorithm 1: Cost assignment scheme.

a rip-up and reroute. Fig. 8 shows several examples of how via location should be blocked. Given a 3×3 subregion, for each unused via location, if an FVP will form after inserting a via at that location, then it should be blocked. Even with blocked via locations, the route could possibly generate a new FVP if multiple via locations are used within 3×3 subregion. In this case, the HC of all the edges incident to each via in the newly generated FVP are increased as shown in line 15. In this way, the vias in FVPs grow more expensive to use and FVPs can be potentially removed.

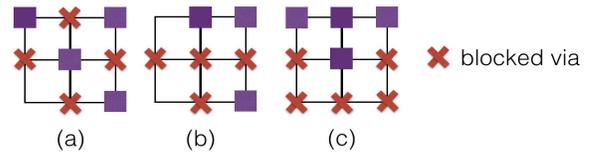


Figure 8: Examples of blocked via locations in 3×3 subregion.

3.4 3-colorability check of TPL decomposition graph

As discussed in previous subsection, the via layer TPL violation removal based rip-up and reroute is targeted to eliminate all the FVPs in the routing solution. However, even if all FVPs are successfully eliminated, there is a small chance that the decomposition graph is still not TPL decomposable. Thus, we do a fast 3-colorability check of TPL decomposition graph. A greedy based Welsh-Powell algorithm [20] is applied here. If it is 3-colorable, TPL pre-coloring exists. If not, rip-up and reroute is re-called to fix the coloring conflict. We note that for all experiments in Section 4,

```

1 initialize priority queue (PQ) and enqueue all FVPs;
2 block via locations that will potentially generate FVP;
3 while PQ is not empty do
4   violation = Dequeue(PQ);
5   choose rip-up net N to resolve violation;
6   update UC, DBC, AVC, CDC, TPLC after removing N;
7   update blocked via locations after removing N;
8   reroute N;
9   update UC, DBC, AVC, CDC, TPLC after rerouting N;
10  update block via location after rerouting N;
11  if reroute causes congestion then
12    update HC for congested routing resource;
13    PQ.enqueue(congestion);
14  else if reroute causes FVP then
15    update HC for vias in that FVP;
16    PQ.enqueue(FVP);
17 end

```

Algorithm 2: Via Layer TPL violation removal based rip-up and reroute.

the elimination of all FVPs in the via layer TPL violation removal based rip-up and reroute can already make the via layers 3-colorable.

3.5 TPL-aware DVI

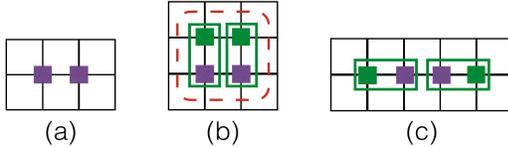


Figure 9: TPL-aware DVI. (a) Two adjacent single vias. (b) Without TPL awareness, TPL violation occurs with inserted redundant vias. (c) With TPL awareness, TPL violation can be avoided after DVI.

Our detailed routing solution is optimized for DVI, and via layers are TPL decomposable. However, the DVI in the post-routing stage will generate a large number of redundant vias, which will potentially cause TPL violation. As shown in Fig. 9, if DVI is performed without considering via layer TPL decomposability, a new TPL violation may occur. Thus, TPL-aware DVI is proposed to ensure TPL decomposable via layers even after DVI. In order to fairly evaluate DVI of our detailed routing solution, we formulate the TPL-aware DVI as an ILP and use an ILP solver to solve it optimally. Solving the ILP may not be efficient when the problem size is big. Alternatively, we propose a fast heuristic which can generate solution with similar quality as ILP. However, we will not discuss here, since it is not the focus of this paper. The ILP formulation is explained as follows:

For each feasible $DVIC_j$, a binary variable $DVIC_j$ indicates whether it is used. $DVIC_j = 1$ if $DVIC_j$ is used. For each via_i , four binary variables $rVia_i, gVia_i, bVia_i, uVia_i$ indicate its color. $rVia_i/gVia_i/bVia_i/uVia_i=1$ if via_i is red/green/blue/uncolorable. For each feasible $DVIC_j$, three binary variables $rDVIC_j, gDVIC_j, bDVIC_j$ indicate its color if it is used. α is a parameter to indicate the relative cost of a TPL violation to a dead via. It should be a big constant since TPL violation should be totally avoided. min_c is the same-color via pitch in Fig. 1(a). B is a large

constant.

$$\text{Objective : maximize } \sum_{j=1}^n DVIC_j - \alpha \times \sum_{i=1}^m uVia_i$$

where $n = \#$ of feasible $DVIC$ s and $m = \#$ of vias

Constraints :

(1) For each via_i ,

$$\sum_j DVIC_j \leq 1 \quad j \in \text{feasible } DVIC \text{ of } via_i$$

(2) If $DVIC_j$ and $DVIC_{j'}$ are in conflict,

$$DVIC_j + DVIC_{j'} \leq 1$$

(3) For each via_i ,

$$rVia_i + gVia_i + bVia_i + uVia_i = 1$$

(4) For each $DVIC_j$,

$$rDVIC_j + gDVIC_j + bDVIC_j - B \times (DVIC_j - 1) \geq 1$$

$$rDVIC_j + gDVIC_j + bDVIC_j + B \times (DVIC_j - 1) \leq 1$$

(5) If via_i and $via_{i'}$ are within min_c

$$rVia_i + rVia_{i'} \leq 1$$

$$gVia_i + gVia_{i'} \leq 1$$

$$bVia_i + bVia_{i'} \leq 1$$

(6) If via_i and $DVIC_j$ are within min_c

$$rVia_i + rDVIC_j + B \times (DVIC_j - 1) \leq 1$$

$$gVia_i + gDVIC_j + B \times (DVIC_j - 1) \leq 1$$

$$bVia_i + bDVIC_j + B \times (DVIC_j - 1) \leq 1$$

(7) If $DVIC_j$ and $DVIC_{j'}$ are within min_c

$$rDVIC_j + rDVIC_{j'} + B \times (DVIC_j + DVIC_{j'} - 2) \leq 1$$

$$gDVIC_j + gDVIC_{j'} + B \times (DVIC_j + DVIC_{j'} - 2) \leq 1$$

$$bDVIC_j + bDVIC_{j'} + B \times (DVIC_j + DVIC_{j'} - 2) \leq 1$$

(8) $uVia_i, rVia_i, gVia_i, bVia_i, uVia_i \in \{0, 1\}$

$$DVIC_j, rDVIC_j, gDVIC_j, bDVIC_j \in \{0, 1\}$$

4. EXPERIMENTAL RESULTS

We implemented our proposed algorithm in C++ programming language. We run all the experiments on a machine with a 2.4 GHz Intel Core i5 CPU and 8 GB memory. Gurobi 6.5 is called to solve the ILPs. Benchmarks from [8] are used to generate experimental results. Each circuit contains three routing layers M1, M2, and M3. M1 is not allowed for routing, and the preferred routing direction for M2 and M3 are horizontal and vertical respectively. Other benchmarks statistics, including the number of nets and design sizes, are listed in [8]. As shown in Table 1, we did four sets of experiments. The baseline is basic SADP-aware routing, and the other three are SADP-aware routing considering DVI, SADP-aware routing considering via manufacturability by TPL, and SADP-aware routing considering both DVI and via manufacturability by TPL.

As shown in Table 1, ‘‘CPU’’ is the detailed routing runtime, ‘‘#DV’’ denotes dead via count and ‘‘#UV’’ denotes the number of uncolorable vias. Both ‘‘#DV’’ and ‘‘#UV’’ are obtained from TPL-aware DVI ILP solution. We do not list the ILP runtime (on average about 1h) in the table, since it is not the optimization target in this paper. In practice, if the ILP solving time is forbidden, a heuristic based TPL-aware DVI can be used alternatively. It is on

Table 1

Ckt	SADP-aware routing					Consider DVI					Consider via layer TPL					Consider DVI & via layer TPL				
	WL	#Vias	CPU(s)	#DV	#UV	WL	#Vias	CPU(s)	#DV	#UV	WL	#Vias	CPU(s)	#DV	#UV	WL	#Vias	CPU(s)	#DV	#UV
ecc	35423	4969	15.7	291	24	35453	4957	17.3	168	17	35712	4947	20.5	175	0	35724	4966	19.6	146	0
efc	45856	7707	30.2	880	104	45984	7760	35.6	657	64	46704	7769	44.2	515	0	46604	7809	45.5	477	0
ctl	56902	9132	31.5	663	63	57046	9158	32.8	457	42	57504	9142	38.5	400	0	57642	9209	43.1	336	0
alu	56986	10053	38.6	1227	113	57418	10173	32.8	871	66	58239	10112	43.0	738	0	58289	10249	50.2	655	0
dvi	120267	20153	86.2	2302	272	121098	20374	109.3	1783	163	122494	20243	135.5	1437	0	122810	20399	147.3	1325	0
top	379114	70185	261.1	9068	1327	381538	70835	282.0	6687	901	389109	70834	382.5	6104	0	389998	71588	386.1	5271	0
Ave.	115758	20367	77.2	2405	317	116423	20543	84.9	1771	209	118294	20510	110.7	1562	0	118511	20703	115.3	1368	0
Nor.	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.10	0.74	0.66	1.02	1.01	1.43	0.65	0	1.02	1.02	1.49	0.57	0

average more than 600x faster than the ILP, and can offer similar solution quality according to our experiments. Note that the routability for all benchmarks in four sets of experiments are 100%, thus we do not list it due to the limited table size. As shown in first two sets of experiments in Table 1, without considering via TPL manufacturability, there are numerous uncolorable vias in TPL-aware DVI ILP solution, which means via patterns with TPL violations exist in SADP-aware detailed routing solution. Compared with the baseline, SADP-aware detailed routing considering DVI can reduce dead via count by 26%. Meanwhile, SADP-aware detailed routing considering via TPL manufacturability can ensure via layers are TPL decomposable in the final routing solution. Note that the consideration of via TPL manufacturability indirectly helps to reduce dead via count since vias are more spread out. Finally, the SADP-aware detailed routing with both DVI and via layer manufacturability by TPL consideration can simultaneously reduce dead via count by 43% and ensure the via layers are TPL decomposable. The overhead is only 2% increase on wirelength and via count respectively. The 49% increase in runtime comes from more rip-up and reroute iterations.

5. CONCLUSION

In this paper, we investigated SADP-aware detailed routing targeted 10nm technology node. We considered both DVI and via layer manufacturability by TPL in detailed routing. To the best of our knowledge, it is the first work to address DVI and via layer manufacturability in SADP-aware detailed routing. The proposed algorithm can reduce dead via count greatly and ensure via layers are TPL decomposable simultaneously. For the future work, we want to find a faster solution for the TPL-aware DVI problem.

Acknowledgement

This work was supported in part by the Ministry of Science and Technology under grant MOST 104-2628-E-007-003-MY3.

6. REFERENCES

- [1] L. Liebmann, V. Gerousis, and P. Gutwin et al. Demonstrating production quality multiple exposure patterning aware routing for the 10nm node. In *Proc. of SPIE*, page 905309, March 2014.
- [2] P. Gupta and E. Papadopoulou. Yield analysis and optimization. In *The Handbook of Algorithms for VLSI Physical Design Automation*. CRC Press, June 2010.
- [3] Y. Ding, C. Chu, and W. Mak. Detailed routing for spacer-is-metal type self-aligned double/quadruple patterning lithography. In *Proc. of DAC*, June 2015.
- [4] Seong-I Lei, Chris Chu, and Wai-Kei Mak. Double patterning-aware detailed routing with mask usage balancing. In *Proc. of ISQED*, March 2014.
- [5] Y. Du, Q. Ma, and H. Song et al. Spacer-is-dielectric-compliant detailed routing for self-aligned double patterning lithography. In *Proc. of DAC*, June 2013.
- [6] L. Liu, S. Fang, and Y. Chang. Overlay-aware detailed routing for self-aligned double patterning lithography using the cut process. In *Proc. of DAC*, June 2014.
- [7] Jih-Rong Gao and David Z. Pan. Flexible self-aligned double patterning aware detailed routing with prescribed layout planning. In *Proc. of ISPD*, March 2012.
- [8] X. Xu, B. Yu, and J. Gao et al. PARR: Pin access planning and regular routing for self-aligned double patterning. In *Proc. of DAC*, June 2015.
- [9] Kuang-Yao Lee and Ting-Chi Wang. Post-routing redundant via insertion for yield/reliability improvement. In *Proc. of ASPDAC*, January 2006.
- [10] K. Lee, C. Koh, and T. Wang et al. Optimal post-routing redundant via insertion. In *Proc. of ISPD*, April 2008.
- [11] K. Lee, S. Lin, and T. Wang. Redundant via insertion with wire bending. In *Proc. of ISPD*, March 2009.
- [12] C. Lei, P. Chiang, and Y. Lee. Post-routing redundant via insertion with wire spreading capability. In *Proc. of ASPDAC*, January 2009.
- [13] K. Lee, S. Lin, and T. Wang. Post-routing redundant via insertion and line end extension with via density consideration. In *Proc. of ISPD*, March 2009.
- [14] S. Lin, K. Lee, and T. Wang et al. Simultaneous redundant via insertion and line end extension for yield optimization. In *Proc. of ASPDAC*, January 2011.
- [15] G. Xu, L. Huang, and D. Pan et al. Redundant-via enhanced maze routing for yield improvement. In *Proc. of ASPDAC*, January 2005.
- [16] C. Lin, Y. Lin, and G. Su et al. Dead via minimization by simultaneous routing and redundant via insertion. In *Proc. of ASPDAC*, January 2010.
- [17] H. Chen, M. Chiang, and Y. Wen et al. Full-chip routing considering double-via insertion. In *Proc. of DAC*, July 2006.
- [18] K. Yuan, K. Lu, and D. Pan. Double patterning lithography friendly detailed routing with redundant via consideration. In *Proc. of DAC*, July 2009.
- [19] B. Yu, K. Yuan, and B. Zhang et al. Layout decomposition for triple patterning lithography. In *Proc. of ICCAD*, November 2011.
- [20] D. J. A. Welsh and M. B. Powell. An upper bound for the chromatic number of a graph and its application to timetabling problems. *The Computer Journal*, 10:85–86, 1967.