Stitch-avoiding Detailed Routing for Multiple E-Beam Lithography

Kritanta Saha*[1], Pritha Banerjee†[2] and Susmita Sur-Kolay*[3]
*Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India
†Department of Computer Science and Engineering, University of Calcutta, Kolkata, India
[1]kritanta09@gmail.com, [2]banerjee.pritha74@gmail.com, [3]ssk@isical.ac.in

2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC) | 978-1-6654-9005-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/VLSI-SoC54400.2022.9939588

***Abstract*—Next-Generation Lithography techniques such as Electron Beam Lithography (EBL), Multiple E-Beam Lithography (MEBL), and Extreme Ultraviolet Lithography (EUVL), overcome the limitations of 193 nm immersion lithography. In MEBL, the layout is split into vertical stripes, with the stripe boundaries termed stitch-lines, and thousands or even millions of electron beams (e-beams) are used in parallel for good throughput. Patterns in different stripes are written either by distinct e-beams or in different passes. Hence, patterns cut by stitch-lines suffer from overlay error and thereby severe pattern distortions, particularly Via, Vertical Routing, and Short Polygon violations near the stitch-lines. In this paper, we propose a Stitch-avoiding Detailed Router that reduces these violations. It is integrated with (i) a traditional global router, and (ii) a Stitch-avoiding Global Router. Experimental results for these two routing flows are compared with those of a baseline flow comprising a traditional global router followed by a traditional detailed router. Significant reductions in stitch-line violations are obtained, especially for the second routing flow, compared to the baseline flow.**

***Index Terms*—Multiple e-beam lithography (MEBL), detailed routing, stitch-line, via violation, short polygon**

I. Introduction

The 193 nm immersion lithography reached its limits as it uses a higher wavelength of light for layout printing than the feature size of modern ICs. The *sub-wavelength lithography gap* is the difference between the printing wavelength and the feature size which leads to pattern distortion. [1], [2] unable to cope with this problem that leads to the development of Next-Generation Lithography (NGL). Electron Beam Lithography (EBL) [3], Multiple E-Beam Lithography (MEBL) [4] etc. are some of the well known NGLs. MEBL overcomes the low throughput of EBL [3] by using thousands or even millions of electron beams (e-beams) in parallel to write the patterns on the wafer. In MEBL, the layout is divided into vertical stripes, and the stripe boundaries are called *stitch-lines* as shown in Fig. 1(a). Patterns in different vertical stripes are printed by different e-beams or in different printing passes. However, two adjacent e-beams deflect each other around the stitch-lines, which produces overlay error [4] and causes pattern distortions and electrical violations. These errors may lead to poor circuit performance or even fabrication failure.

Fig. 1(c) shows stitch-lines causing distortions. The *stitch-unfriendly region* [4] is the area within the distance ϵ on both side of a stitch-line. A horizontal segment cutting a stitch-line and landing on a stitch-unfriendly region with a via is called *short polygon* [4]. In order to reduce distortion, Fig. 1(d) shows that we need to minimize the following violations i.e., (1) *Via violation*: vias cannot be cut by stitch-lines, (2) *Vertical routing violation*: vertical routing is not allowed on stitch-lines, and (3) *Short polygon*: vias should not produce short polygons. Vias and vertical routing are hard constraints as these causes severe distortion whereas a short polygon is a soft constraint [4] as it may not always cause severe distortion.

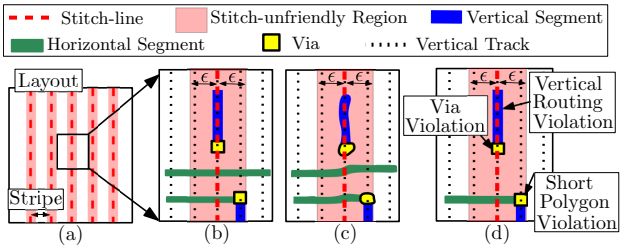

Fig. 1. (a) A layout divided into stripes by stitch-lines; (b) patterns cut by stitch-lines; (c) pattern distortion due to overlay error; (d) violations due to stitch-line [6]

The DP based placement method [5] refines local congestion but can only reduce the violations due to fixed pins. The method in [4] used a traditional global router but a new congestion constraint, and by avoiding the violations during track assignment and detailed routing reduced short polygons but increases via violations. The perturbation-based approach [7] reduced stitch-line violations in the post-routing phase but had a large number of vertical routing violations, and is not suitable for congested circuits. The method proposed in [8] introduced smart boundaries but needs complex e-beam control. A stitch-avoiding global router was proposed in [6] that reduces stitch-line violations. For further reduction of violations, stitch-line constraints must be considered in the detailed routing phase. Hence, a detailed router for MEBL should aim to minimize via and short polygon violations with no vertical routing violations.

In this paper, we propose a Stitch-avoiding Detailed Router for MEBL. Our method takes a 2D global routing solution and performs stitch-avoiding detailed routing to minimize the stitch-line violations. We test our proposed router on the Faraday [9] benchmark suite. The results show a significant reduction in the violations. The rest of the paper is organized as follows: Section II gives the preliminaries, formulation, net classification, and overall framework. Section III and IV describes our proposed method. Section V has the experimental results, and we conclude in Section VI.

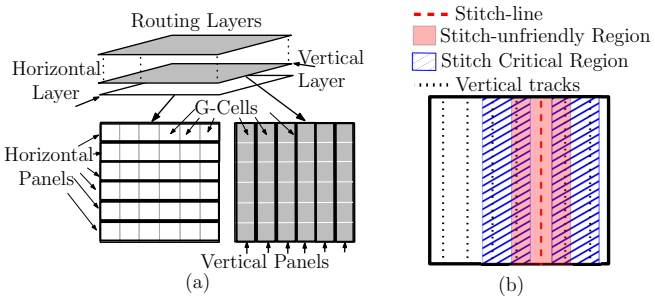

Fig. 2. (a) G-cell, HPanels and VPanels; (b) Stitch Critical Region

II. Stitch-avoiding Detailed Routing

In this section, we present the preliminaries, problem formulation, and framework of our proposed method.

A. Preliminaries

Definition II.1. In a grid graph model, the layout is divided into cells called *grid cells (G-cell)* denoted by c_i (Fig. 2(a)).

Definition II.2. A *horizontal panel (HPanel)* Π_m is the set of all G-cells of m^{th} row of a horizontal routing layer (HLayer). Fig. 2(a) shows the HPanels.

Definition II.3. A *vertical panel (VPanel)* Π_m is the set of all G-cells of m^{th} column of a vertical routing layer (VLayer). Fig. 2(a) shows the VPanels.

Definition II.4. *Stitch Critical Region* contains the tracks inside and adjacent to a stitch-unfriendly region. (Fig. 2(b)).

B. Problem Formulation

The Stitch-avoiding detailed routing problem in MEBL is formulated as follows: Given, a set of nets $N = \{N_1, N_2, ..., N_n\}$, a set of routing layers $L = \{L_1, L_2, ..., L_k\}$, locations of stitch-lines $S = \{S_1, S_2, ..., S_l\}$, and a global routing solution for each net. Perform stitch-avoiding detailed routing to minimize via violations and short polygon violations with no vertical routing violations.

C. Net Classification

Depending on the span of each net over multiple G-cells, the nets are classified as follows:

- Single-Cell Confined Net (SCN): a net which has all its segments within a single G-cell.
- Multi-Cell Confined Net (MCN): a net have segments that span over multiple G-cells. These types of nets can further be classified into two categories as follows.
  - Multi-Cell Single Panel Confined Net (MCSPN): a net with all segments are on a single VPanel or Hpanel.
  - Multi-Cell Multi Panel Confined Net (MCMPN): a net with all segments are on multiple VPanels or HPanels.

D. Stitch-avoiding Detailed Routing Framework

Our proposed Stitch-avoiding detailed router directly takes the 2D global routing solution without performing the intermediate layer assignment phase. During detailed routing, we carry out the layer assignment. Our strategy is inspired by a traditional detailed router called RegularRoute [10]. The approach in [10] routed nets with a minimum number of bends and tried to assign a segment on a single track as much as the capacity permits. RegularR [10] is an efficient detailed router for conventional lithography. It is not designed for MEBL and cannot handle the stitch-lines. However, the segment assignment technique can be beneficial in MEBL under the situation of reduced vertical capacity due to stitch-lines. Fig. 3 shows the entire flow of our proposed Stitch-avoiding detailed routing technique. We block the vertical tracks (VTrack) that coincide with the stitch-lines. SCNs are generally less flexible because they have smaller wirelength than others. Hence, we route SCNs first as shown in Fig. 3. Since the stitch-lines are vertical, we decompose the MCNs into horizontal segments (HSegments) and vertical segments (VSegments). Unlike RegularRoute, we routed the VSegments and HSegments differently in the order specified in Fig. 3. Starting from the bottom-most layer, we assign segments to tracks. We promote the terminals of unrouted segments to the upper layer. This continues till the topmost layer. Finally, routing bends may be reduced by local rip-up and reroute. It may be noted that fixed pins can be moved from stitch-lines only during placement and not during detailed routing.

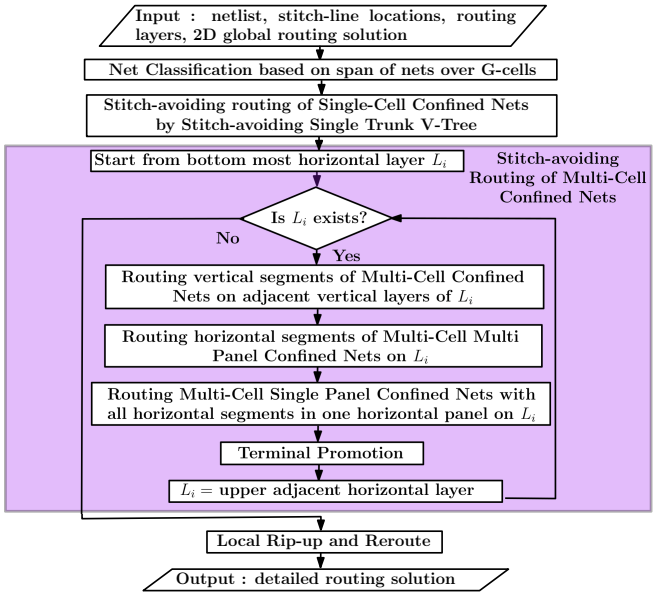

Fig. 3. Flow of Stitch-avoiding Detailed Routing

III. Stitch-avoiding Detailed Routing of SCNs

As the vertical routing capacity is reduced, we need to use fewer VTracks to route SCNs. Hence, we use *Single Trunk V-tree* as it has only one VSegment called V-trunk. For non-stitch holding cells, we assign the V-trunk to VTrack having a median x-coordinate of all the pins of a SCN in the lowest VLayer. The V-trunk spans from the minimum to maximum y-coordinate of the pins. We connect the HSegments with the V-trunk in the lowest HLayer. If multiple SCNs are present in a single G-cell, we use a shifting mechanism for overlap-free assignment of V-truck of all nets. Usually, we have a small number of SCNs and a limited number of tracks inside a G-cell. Hence, shifting is not expensive. However, choosing V-trunk is challenging in the presence of stitch-lines.

For example, let metal-1 and 3 be HLayers and metal-2 be VLayer. Fig. 4 shows a SCN. We first assign the v-trunk on metal-2 and then assign all the HSegments on metal-1.

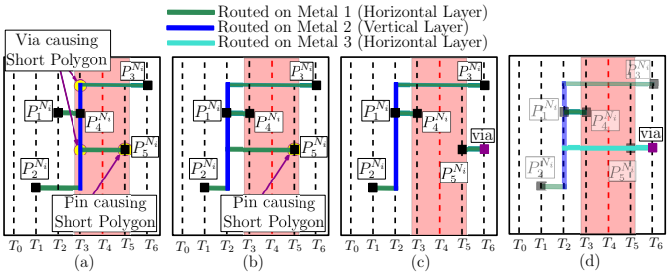

Fig. 4. (a) Single trunk V-tree with V-trunk at median x-coordinate of the pins causing short polygons; (b) Local shifting of V-trunk reduces some short polygons but at pin $P_5^{N_i}$ short polygon exists; (c) Introduction of a new HSegment and a via to eliminate short polygon at $P_5^{N_i}$; (d) Horizontal connection from via to the V-trunk is made at upper HLayer

However, the tree produces three short polygons, as shown in Fig. 4(a). To remove two short polygons, we shift the V-trunk to the adjacent track as shown in Fig. 4(b). However, to remove the remaining short polygon at pin $P_5^{N_i}$, we use a new via outside the stitch-unfriendly region and perform the HSegment connection between $P_5^{N_i}$ and the via on metal-1 as shown in Fig. 4(c). To connect the via with the vertical trunk, we use metal-3 shown in Fig. 4(d).

IV. Stitch-avoiding Detailed Routing of MCNs

We first decomposed MCNs into HSegments and VSegments and routed them independently. Here, HSegments and VSegments are routed in HLayers and VLayers, respectively.

A. Routing VSegments of MCNs

Algorithm 1 shows the routing procedure of VSegments of MCNs. Starting from the left-most VPanel in the current VLayer, the VSegments are assigned to tracks in a panel-by-panel manner. Algorithm 1 contains three major operations:

1) Routing in *non-stitch critical region* and identification of delayed segments list which is a set of VSegments need to be routed inside the *stitch critical region*.
2) Segment decomposition and routing of delayed segments in the *stitch critical region*.
3) Partial VSegment routing in the *non-stitch critical region*.

Let us briefly discuss these operations in the follows.

1) Routing in the Non-Stitch Critical Region and Identification of Delayed Segments list: For each VPanel in the current VLayer, the VSegment assignment problem is formulated as finding a *Maximum Weighted Independent Set* (MWIS) on a *Segment Conflict Graph* (SCG) similar to [10]. Fig 5 shows the construction of SCG for a VPanel. We assign weights to vertices of SCG. Solving MWIS on general graphs is NP-Complete, however, we use a heuristic method proposed in [10] to solve MWIS in polynomial time on SCG. We also calculate the benefit of each vertex using the vertices weights and consider the highest benefit vertex for routing.

We have however defined a new weight function for VSegment assignment which incorporates stitch-line constraints, instead of using the one in [10], as the VSegment assignments are likely to generate stitch-line violations in MEBL. The role of these constraints is to select the VSegments that should be preferably routed outside the *Stitch Critical Region*. In this

Algorithm 1: Routing of VSegments of MCNs

```
for each VPanel Π_i in VLayer L_i do
    Set Delayed_Segment_List=φ;
    if Π_i contains stitch-line then
        Find set Track_SCR of VTracks present in Stitch
          Critical Region of Π_i;
    end
    Find set VSEG of VSegments of Π_i;
    Construct Segment Conflict Graph (SCG) from VSEG;
    while SCG contains Vertices do
        Find MWIS on SCG to select the best segment track pair
          (Ψ, T);
        if T ∈ Track_SCR then
            Insert (Ψ, T) in Delayed_Segment_List and reserve
              vertical and horizontal routing resource;
        else
            Assign segment Ψ to track T;
            if Corresponding HSegment of S is unroted then
                Determine Via Position;
            end
            Maze Routing for VSegment-to-pin Connection;
        end
        Update SCG ;
    end
    Routing of segments in Delayed_Segment_List ;
    Partial VSegment Assignment;
end
```

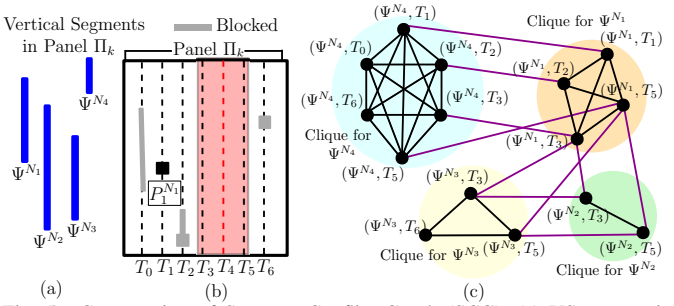

Fig. 5. Construction of Segment Conflict Graph (SCG); (a) VSegments in the Panel Π_k; (b) Present state of Π_k; (c) Corresponding SCG. Black edges represents conflict between different choices of same segment and Magenta edges represents conflict between choices of different segments

crucial phase, we also identify the list of Vsegments that have to be routed inside the *Stitch Critical Region*.

As mentioned above, we consider the highest benefit vertex for routing. Now, one of the following two scenarios can arrive

- Scenario I - The highest benefit vertex corresponds to a track inside the *stitch critical region*: Here, we do not assign the segment to that track. We called the segment *delayed segment*, and we put the segment in the $Delayed\_Segment\_List$ for routing in the latter stage.
- Scenario II - The highest benefit vertex corresponds to a track outside the critical region: we assign the corresponding VSegment to that track. Then we perform maze routing in the lower Hlayer of the present VLayer to connect Vsegment endpoints with the Pin/vias.

Next, we update SCG by removing the highest benefit vertex and all the vertices and edges that are adjacent to this vertex from SCG. Again, the weights are calculated for all the vertices of current SCG, and we again solve MWIS. We continue this process until no vertices are present in SCG. After that, we perform routing of delayed segments present in the set $Delayed\_Segment\_List$. These segments are critical as these segments are routed inside the *Stitch Critical Region*.

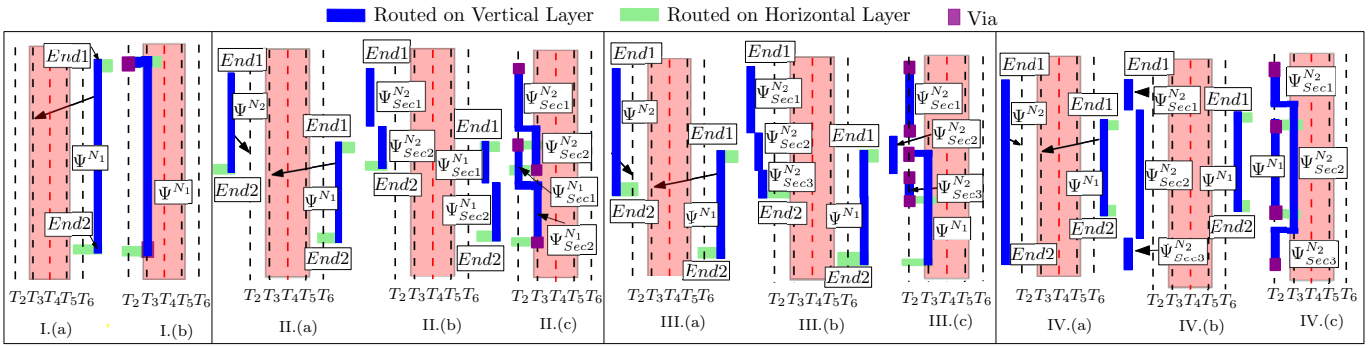

Fig. 6. Segment decomposition and routing of Delayed Segments for various cases — I.(a) Ψ^{N_1} on T_3 creates a short polygon at $End1$ of Ψ^{N_1}; I.(b) Segment routed on T_3 with a new via to remove the short polygon; II.(a) Assigning segments on the chosen VTracks creates a short polygon at $End1$ of Ψ^{N_1}; II.(b) Segment Decomposition for II.(a); II.(c) Decomposed Segments routed on the tracks with new vias to remove short polygon. III.(a) Assigning the segments on the chosen VTracks creates a short polygon at $End1$ of Ψ^{N_1}; III.(b) Segment Decomposition for III.(a); III.(c) Decomposed segments routed on the tracks with a new via eliminates short polygon violation. IV.(a) Assigning the segments on the chosen VTracks creates short polygons at $End1$ of Ψ^{N_1} and $End2$ of Ψ^{N_1}; IV.(b) Segment Decomposition for IV.(a); IV.(c) Decomposed segments routed on the tracks with a new via eliminates short polygon violation.

Now, we define our proposed weight function $W(\Psi^{N_i}, T_j)$ that assigns weights to each vertices (Ψ^{N_i}, T_j) of SCG. $W(\Psi^{N_i}, T_j)$ specifies the priority of assigning a segment to a track. A higher weight signifies a higher priority for routing. For VPanel, the weight function is defined in Eq. 1 which has stitch-line constraints to avoid stitch-line violations.

$$W(\Psi^{N_i}, T_j) = f(\Psi^{N_i}, T_j) \cdot L(\Psi^{N_i}) + \alpha_1 \cdot \Omega(\Psi^{N_i}) - \chi(\Psi^{N_i}, T_j) - \xi(\Psi^{N_i}, T_j) \quad (1)$$

where $L(\Psi^{N_i})$: length of a vertical segment of Ψ^{N_i}, i.e., the product of the number of G-cells Ψ^{N_i} spans within the VPanel and the number of HTracks present in a G-cell, Ω: VSegment density, $\chi(\Psi^{N_i})$: cost of VSegment-to-pin connection and $\xi(\Psi^{N_i}, T_j)$: stitch penalty.

The function $f(\Psi^{N_i}, T_j)$ assigns a high weight to a vertex of SCG that has chosen a track inside a *stitch critical region* and has $L(\Psi^{N_i}) \geq avgSegL$, the average segment length of that panel. We consider this as a critical case and set $f(\Psi^{N_i}, T_j) = (1 + r)^{\sigma}$, and 1 otherwise where σ is a user-specified constant parameter and $r = L(\Psi^{N_i})/avgSegL$. We consider this because if longer VSegments are routed inside *stitch critical regions* then number of VSegments inside it becomes less hence, the vias required for the horizontal connection becomes less inside *stitch critical regions*, and longer segments are more flexible.

$\Omega(\Psi^{N_i})$ is the average number of VSegments that crosses the boundaries of those G-cells that the segment Ψ^{N_i} passes through. $\Omega(\Psi^{N_i}) = \frac{1}{|B|}\sum_{b \in B} \lambda_b$ where B: set of G-cell boundaries that Ψ^{N_i} passes through and λ_b: number of segments crossing the boundary line segment b. If Ψ^{N_i} passes through a congested G-cell then $\Omega(\Psi^{N_i})$ is large which in turn increases $W(\Psi^{N_i}, T_j)$. A higher value of $W(\Psi^{N_i}, T_j)$ signifies higher priority of the segment Ψ^{N_i} to be routed earlier than those in less congested regions.

If Ψ^{N_i} is assigned to T_j, then the cost for VSegment-to-pin connection is denoted by $\chi(\Psi^{N_i}, T_j)$. If the HSegments of Ψ^{N_i} are already routed then we employ Maze Router to perform the segment-to-segment connection within a single G-cell, else to find a path between the pin and Ψ^{N_i}. If no path exists, we discard the vertex (Ψ^{N_i}, T_j) and its adjacent edges from SCG. If a path exists, we find its cost using Eq.2.

$$\chi(\Psi^{N_i}, T_j) = w_1 \cdot \eta_1 + w_2 \cdot \eta_2 + w_3 \cdot \eta_3 \quad (2)$$

where η_1 (η_2): number of detailed routing grid edges the path crosses in the preferred (non-preferred) routing direction, η_3: number of vias needed in the path, and w_1 ,w_2, w_3 are the corresponding weights. Note that $w_1 < w_2$ to reduce using the non-preferred routing direction as much as possible, and the value of w_3 is the highest as vias are more critical.

The stitch penalty $\xi(\Psi^{N_i}, T_j)$ of assigning Ψ^{N_i} to T_j is defined in Eq. 3

$$\xi(\Psi^{N_i}, T_j) = w_4 \cdot \eta_{SP} + w_5 \cdot \eta_{VV} \quad (3)$$

where η_{SP}: number of short polygons generated in the VSegment-to-pin connection and η_{VV}: number of via violations generated in the VSegment-to-pin connection. For the weights w_4 and w_5, $w_4 < w_5$ because a via violation cause more distortion than a short polygon violation.

2) Segment Decomposition and Routing of Delayed Segments: The $Delayed\_Segment\_List$ contains pairs of VSegments and tracks that need to be routed inside *stitch critical region*. Here, we perform routing of these segments. To reduce short polygons, we decompose these segments and rearrange them in the tracks based on several cases. From $Delayed\_Segment\_List$, let us take a segment Ψ^{N_1} that has track choice T_3 and also consider Ψ^{N_2} belongs to $Delayed\_Segment\_List$. Now, we want to assign Ψ^{N_1} to T_3. During the assignment of Ψ^{N_1} to T_3 several cases occur if we want to avoid stitch-line violations. Fig 6 shows four different solution strategies for solving all possible cases.

3) Partial VSegment Assignment: During the calculation of the weight W, track choices for a VSegment get rejected if maze routing for VSegment-to-pin connection is unable to find any path. It reduces the number of choices of tracks for each VSegment. Moreover, if all the track choices of a VSegment get rejected due to no path for VSegment-to-pin connection, then the VSegment has no chance for routing in that layer. In this phase, we perform the partial assignment of those VSegments. We use the same technique for solving MWIS but with a modified weight function W'. Only the VSegment is assigned to a VTrack, and its HSegments are routed during

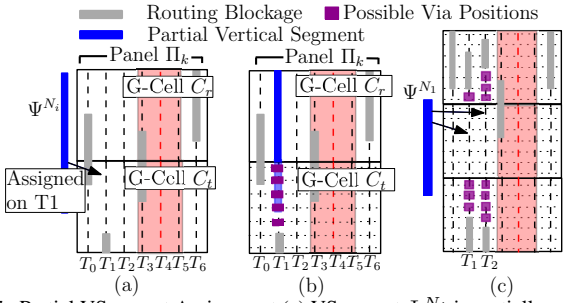

Fig. 7. Partial VSegment Assignment (a) VSegment Ψ^{N_i} is partially assigned on T_1; (b) Possible via positions for HSegment connection; (c) Possible Via positions for both ends of Ψ^{N_1} if assigned to T_1 or T_2

the routing of HSegments of MCMPNs in a latter phase. Hence, W' does not include the cost of a VSegment-to-pin connection as the position of the via is unknown in this step. Fig. 7(b) shows possible via positions for a partially routed VSegment Ψ^{N_i} assigned on track T_1. The x-coordinate of all possible via positions are same as that of the VTrack where it may be assigned. The y-coordinates of the vias are determined during the routing of HSegments of MCMPNs. We do not consider the tracks inside the *stitch-unfriendly region* for partial VSegment assignment. Hence, these vias cannot generate a short polygon. W' is defined in Eq.4 that includes the number of possibles via positions for horizontal connection.

$$W'(\Psi^{N_i}, T_j) = L(\Psi^{N_i}) + \alpha_1 \cdot \Omega(\Psi^{N_i}) - \xi(\Psi^{N_i}, T_j) + \alpha_2 \cdot \upsilon(\Psi^{N_i}, T_j) \quad (4)$$

where, $\upsilon(\Psi^{N_i}, T_j)$ is the minimum of number of possible via positions available for both segment ends. For example, Fig. 7(c) shows if Ψ^{N_1} is assigned to T_1 then $\upsilon(\Psi^{N_1}, T_1) = min(1, 3) = 1$ and if it is assigned to T_2 then $\upsilon(\Psi^{N_1}, T_2) = min(3, 4) = 3$. A higher value of υ indicates that a larger number of tracks are available for HSegment connection.

B. Routing HSegments of MCMPNs

The HSegments of MCMPNs are routed using Algo. 2 in a panel-by-panel for each HLayer and consists of two phases.

- *First Phase:* The HSegment assignment problem is formulated as a MWIS problem on SCG.
- *Second Phase:* The unrouted HSegments from first phase whose corresponding VSegments are already routed, are routed by a maze router for segment-to-pin connection.

While constructing SCG and solving MWIS are similar to that for VSegment assignment of MCNs, here we perform both full and partial assignment of HSegment together (6-10 lines of Algo.2). The weight is given in Eq. 5.

No HSegment is delayed as the highest benefit vertex cannot correspond to a track inside the *Stitch Critical Region*. If (Ψ, T) is the highest benefit vertex then we assign Ψ to T. If VSegment of Ψ is already routed then the intersection point of HSegment and VSegment gives the via position. However, if the VSegment is not yet routed then we left the HSegment as partial assignment without determining via position.

$$W''(\Psi^{N_i}, T_j) = L(\Psi^{N_i}) + \alpha_1 \cdot \Omega(\Psi^{N_i}) - \chi(\Psi^{N_i}, T_j) - \xi(\Psi^{N_i}, T_j) + \alpha_2 \cdot \upsilon(\Psi^{N_i}, T_j) \quad (5)$$

Algorithm 2: Routing of HSegments of MCMPNs

```
1  for each HPanel Π_i in HLayer L_i do
2      Find set HSEG of HSegments present in Π_i;
3      Construct SCG from HSEG;
4      while SCG contains Vertices do
5          Find MWIS on SCG to select the best segment track pair
             (Ψ, T);
6          if Corresponding VSegment of Ψ is already Routed then
7              Assign HSegment Ψ to T and set Via position;
8          else
9              Assign HSegment Ψ to T as a partial assignment
                 without determining Via position;
10         end
11         Maze Routing for HSegment-pin connection;
12         Update SCG ;
13     end
14     for each unrouted HSegment whose corresponding VSegment is
         already routed do
15         Maze Routing for VSegment-to-pin Connection;
16     end
17 end
```

C. Routing MCSPNs with all HSegments in one HPanel

Consider routing of MCSPNs whose all the HSegments are confined within a single HPanel. In this case, the VSegments of these MCSPNs are confined within a single G-cell, hence, they are not routed separately. We use *Multi-Layer Maze Routing* to route the HSegments and VSegments of these MCSPNs. For routing on a HLayer L_i, we additionally use adjacent VLayers and HLayers as patch layers to detour the routes in case of routing congestion in L_i. Here, multi-layer maze router is not very costly due to the following factors

- Usually, a small number of MCSPNs are present in a circuit.
- The exploration area for the maze router is restricted only to the HPanel where it resides.
- As the routing is done layer-by-layer starting from the bottom-most layer, a large number of MCSPNs are routed in the lower layers. Hence, the number of unrouted nets promoted to the upper layers decreases rapidly.
- The tracks in the bottom patch layers are mostly blocked by the previously routed nets. Hence, the exploration area of the maze router is small in the adjacent bottom patch layer.

Due to stitch-lines, we impose restrictions on the maze router. To avoid short polygon, the maze router cannot change layers when it is inside a *stitch-unfriendly region*. As the VTracks on the stitch-lines are already blocked, the maze router cannot generate any via and vertical routing violation. However, horizontal patch layers are used only when the maze router faces congestion in the current HLayer because it required additional vias and the vias or pins of the segment also need to be promoted to the upper layers if the maze router moves to any upper patch layer.

TABLE I
FARADAY BENCHMARK CIRCUITS OF ICCAD04 [9]

Circuits	#Layers	Size(μm^2)	#Nets	#Pins
DMA	6	408.4 × 408.4	13256	73982
DSP1	6	706 × 706	28447	144872
DSP2	6	642.8 × 642.8	28431	144703
RISC1	6	1003.6 × 1003.6	34034	196677
RISC2	6	959.6 × 959.6	34034	196670

TABLE II
EXPERIMENTAL RESULT SHOWING THE EFFECTIVENESS OF OUR PROPOSED STITCH-AVOIDING DETAILED ROUTER

	Baseline Routing Framework: BoxRoute [12] and RegularRoute [10]					Routing Framework I: BoxRouter [12] and our proposed Stitch-avoiding Detailed Router					Routing Framework II: Stitch-avoiding Global Router [6] and our proposed Stitch-avoiding Detailed Router				
Circuits	$\#VV$	$\#SHP$	$\#VRV$	%Rout.	CPU(S)	$\#VV$	$\#SHP$	$\#VRV$	%Rout.	CPU(S)	$\#VV$	$\#SHP$	$\#VRV$	%Rout.	CPU(S)
DMA	3589	1626	0	95.94	421	3546	600	0	96.87	351	2153	847	0	97.26	353
DSP1	7469	3496	0	96.56	1419	5568	1374	0	96.08	1156	3862	1867	0	97.47	1139
DSP2	7410	3462	0	96.69	1296	5462	1603	0	95.81	1003	3633	2022	0	97.89	1005
RISC1	10799	5246	0	95.92	3016	9536	2315	0	96.61	2704	4990	3078	0	96.79	2690
RISC2	10934	5565	0	96.09	2944	9586	2312	0	97.31	2660	5244	3054	0	97.37	2606
Comp.	1.00	1.00	-	1.00	1.00	0.8382	0.423	-	1.0031	0.8656	0.4946	0.5604	-	1.0115	0.8567

V. EXPERIMENTAL RESULTS

Our proposed Stitch-avoiding detailed router is implemented in JAVA on a Linux system with 2.40 GHz Intel(R) Xeon(R) CPU and 32 GB RAM. We perform testing on the Faraday benchmark of ICCAD'04 Mixed-size Placement Benchmarks [9], [11]. Table I shows the details of Faraday's benchmark. We perform placement by NTUplace3 [13]. The distance between two stitch-lines is taken as $15\times$ the routing pitch. The adjacent tracks of stitch-lines fall into a stitch-unfriendly region.

We construct two routing frameworks (a) *Routing Framework I:* a traditional global router *BoxRouter* [12] + our proposed Stitch-avoiding detailed router; (b) *Routing Framework II:* Stitch-avoiding global router [6] + our proposed Stitch-avoiding detailed router. For comparison, we also form a *Baseline Routing Framework* with *BoxRouter* [12] + *RegularRoute* [10], respectively. Due to the unavailability of *BoxRouter* and *RegularRoute* for academic research, we have implemented these routers. In the baseline detailed router, we block vertical routing on the stitch-lines to avoid vertical routing violations. We set $\alpha_1 = 2$, $\alpha_2 = 3$, $\sigma = 2$, $w_1 = 1$, $w_2 = 3w_1$, $w_3 = 2w_2$, $w_4 = 2w_3$ and $w_5 = 5w_3$.

Table II shows the comparison of all three Routing Frameworks, where, $\#VV$ denotes the number of via violations, $\#SHP$ the number of short polygons, $\#VRV$ the number of vertical routing violations, %Rout. the routability in percentage. Table II shows that Routing Framework I achieved 16.18% and 57.71% reduction in via violations and short polygon violations respectively, whereas Routing Framework II achieved 50.54% and 43.96% reduction in via violations and short polygon violations respectively compared with the *Baseline Routing Framework*. As the placement tools used to place the circuits in the Faraday benchmark suite were not disclosed in [4] and [7], it was not possible to compare with their results because placement affects the stitch-line violations. Routability and CPU time for both the Routing Frameworks I and II are better than that for the *Baseline Routing Framework*.

VI. CONCLUSION

This paper presents a Stitch-avoiding detailed router that handles stitch-line violations in MEBL. For the stitch-line violations, nets are classified into different categories and through multiple steps, it generates the solution with fewer stitch-line violations. Experimental results show a significant reduction for both *Routing Framework I* and *Routing Framework II* as compared with *Baseline Routing Framework*. In particular, *Routing Framework II* outperforms the other two as it comprises a Stitch-avoiding global router and our proposed Stitch-avoiding detailed router. Routing frameworks with other approaches may be set up in future for more evaluation. However, via violations due to fixed pins cannot be eliminated during routing. These have to be tackled during the placement phase, which is the next direction of work in future.

REFERENCES

[1] B. Yu, K. Yuan, D. Ding, and D. Z. Pan, "Layout decomposition for triple patterning lithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 3, pp. 433–446, 2015.

[2] S.-L. Huang, C.-W. Lin, and Y.-W. Chang,"Efficient provably good opc modeling and its applications to interconnect optimization," *2010 IEEE International Conference on Computer Design, IEEE*, pp. 336–341, 2010.

[3] B. J. Lin, "Future of multiple-e-beam direct-write systems," *Journal of Micro/Nanolithography, MEMS, and MOEMS*, vol. 11, no. 3, p. 033011, 2012.

[4] I.-J. Liu, S.-Y. Fang, and Y.-W. Chang, "Stitch-aware routing for multiple e-beam lithography," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 3, pp. 471–482, 2015.

[5] Y. Lin, B.Yu, Y. Zou, Z.Li, C.J. Alpent and D. Z. Pan. "Stitch aware detailed placement for multiple e-beam lithography." Integration 58 (2017): 47-54.

[6] K. Saha, S. Paul, P. Banerjee, and S. Sur-Kolay, "Stitch-avoiding Global Routing for Multiple E-Beam Lithography," *to appear in Proc. 35th International Conference on VLSI Design*, 2022. (https://bit.ly/3bibQEr).

[7] S. Paul, P. Banerjee, and S. Sur-Kolay, "Post-layout perturbation towards stitch friendly layout for multiple e-beam lithography," in Proceedings 2017 *IEEE International Conference on Computer Design* (ICCD). IEEE, 2017, pp. 411–414.

[8] C.-H.Hsu, and S.-Y. Fang. "Stitch-aware routing considering smart boundary for multiple e-beam lithography." In 2020 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), pp. 1-4. IEEE, 2020.

[9] "Faraday benchmarks," ICCAD 2004. (https://bit.ly/3OD7RzS)

[10] Y. Zhang and C. Chu, "RegularRoute: An Efficient Detailed Router Applying Regular Routing Patterns," *IEEE Transactions on Very Large Scale Integration* (VLSI) systems, vol. 21, no. 9, pp. 1655–1668, 2012.

[11] S. Adya, S. Chaturvedi, J. Roy, D. Papa, and I. Markov, "Unification of partitioning, floorplanning and placement," in *Proceedings of International Conference on Computer-Aided Design* (ICCAD), 2004, pp. 550–557.

[12] M. Cho and D. Z. Pan, "BoxRouter: A new global router based on box expansion and progressive ILP," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 12, pp. 2130–2143, 2007.

[13] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W.Chang, "NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1228–1240, 2008.