Electromigration-aware Redundant Via Insertion

Jiwoo Pak
ECE Department
University of Texas at Austin
Austin, TX, USA
jiwoo.pak@utexas.edu

Bei Yu
ECE Department
University of Texas at Austin
Austin, TX, USA
bei@cerc.utexas.edu

David Z. Pan
ECE Department
University of Texas at Austin
Austin, TX, USA
dpan@ece.utexas.edu

ABSTRACT

As the feature size shrinks, electromigration (EM) becomes a more critical reliability issue in IC design. EM around the via structures accounts for much of the reliability problems in ICs, and the insertion of redundant vias can mitigate the adverse effect of EM by reducing current density. In this paper, we model EM reliability of redundant via structures, considering current distribution with different via layouts. Based on our EM model, we choose redundant via layouts that can increase the EM-related lifetime by using integer linear programming (ILP). To overcome the runtime issue of ILP, we also propose speed-up techniques for our EM-aware redundant via insertion. Experimental results show that our scheme brings much more EM-robustness to circuits with the similar number of redundant vias, compared to the conventional redundant via insertion techniques.

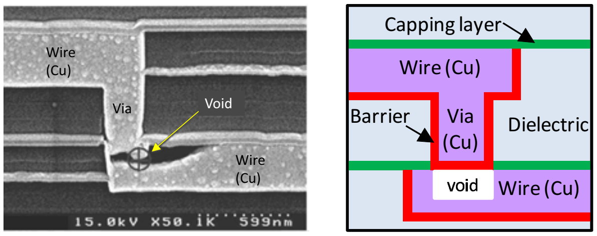
1. INTRODUCTION

Vias are critical structures of advanced IC design that help expanding the design space by connecting layers vertically. Despite its importance in circuit design, vias are often considered as one of the major sources of process and reliability issues that may degrade circuit performance, or may even fail the circuits [1].

One major source of via reliability issues is Electromigration (EM), diffusion of metal atoms induced by electron current. As IC technology advances, current density increases due to the reduction of cross-sectional via area, which negatively affects failure time. Reduction in the failure time from EM can be worsened even further by high temperatures and mechanical stress around the vias. With EM, the interface of the via and the metal wire is one of the weakest points to EM. We can see this phenomenon in Fig. 1(a) with an SEM image of the local via and wires [2]. To be more specific, Fig. 1(b) shows the schematic view of metal wires, local via structure and an EM-induced void with Cu dual-damascene process.

By inserting redundant vias, we can mitigate the adverse effect of electromigration, because it can reduce the current density of each via. Since via redundancy has been known to improve yield and reliability, much work has been done to maximize the insertion of redundant vias during post-layout optimization [3–6] or during the routing stage [7,8]. However, those studies mostly focused on the quantity of redundancy, not on the quality of redundancy. Though some work addressed via yield issue by considering line end extension and redundant vias altogether in the context of EM reliability [6], there has been little work that addresses the EM related lifetime as the main objective of via insertion.

In this paper, we propose an EM-aware redundant via insertion approach that can be applicable for the post-layout optimization. Our contributions are summarized as follows: (1) We model and analyze electromigration (EM) for various redundant-via structures. Our model includes a holistic failure model on how the early failure of one via can affect the lifetime of the remaining vias in a structure with multiple vias. (2) Based on the model, we find that current imbalance in redundant vias affects EM reliability of the whole structure. Also, unlike previous work that preferred on-track double-via layouts [4], our modeling results suggest that off-track layouts may benefit EM reliability. (3) We propose a via-insertion algorithm that chooses the best redundant via layout for the EM-prone nets, which can maximize the EM reliability compared to the conventional redundant via insertion. To the best of our knowledge, this study is the first one to focus on EM-aware redundant via insertion. (4) We present a set of speed-up techniques to achieve better tradeoff between runtime and performance during via-insertion.



(a) (b)

Figure 1: (a) SEM image of via with void due to EM [2], (b) Schematic view.

The rest of the paper is organized as follows. After preliminaries in Section 2, Section 3 models and analyzes EM with a detailed failure model for various multiple via structures. Section 4 explains our EM-aware redundant via insertion flow, and Section 5 suggests some speed-up techniques. Section 6 discusses our experimental results, followed by conclusions in Section 7.

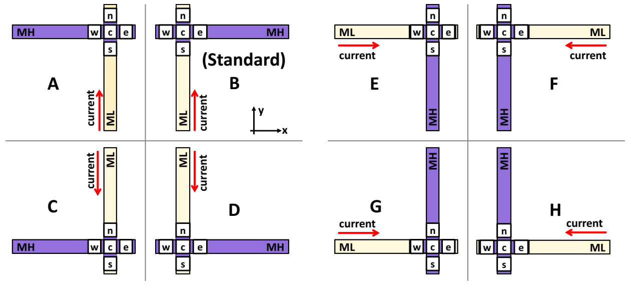
2. PRELIMINARIES

2.1 Basics of Electromigration

Electromigration (EM) refers to the mass transport and diffusion of metal atoms because of momentum transfer between conducting electrons and atoms [9]. Once atoms migrate with electrons, voids can be formed and grow at the point where the flux diverges, while atomic accumulation takes place at the other sides. Void formation and growth increase the resistance of the metal line, and may lead open circuits eventually [9].

In many modern circuits, the void generated by EM is one of the major reasons of interconnect failure [1, 10]. EM-induced failure time of an interconnect can be expressed by a two-phase model, with void nucleation time followed by void growth time [11]. However in the deep-submicron copper-based interconnects, nucleation time can be assumed much shorter than the void growth time, because it is almost impossible to have void-free adhesion of copper and barrier/liner material during the manufacturing process [12]. Therefore in this work, we assume that we already have pre-nucleated voids under the via trench, and consider EM failure time as the void growth time.

978-1-4799-7792-5/15/$31.00 ©2015 IEEE



(a) (b)

Figure 2: Wire position cases that can be generated from the two orthogonal wires.

2.2 Layout Cases with Redundant Vias

Fig. 2 shows eight possible wire position cases, which can be made from the two orthogonal wires in the adjacent routing layers. Here *MH* and *ML* stand for higher metal and lower metal wire, respectively. For the wire position cases, we assume 1) two wires are located in the adjacent metal layers, and 2) current flows from the lower metal to the higher metal (=downstream EM). Regarding the bi-directional current cases, previous studies [13, 14] showed that they can be analyzed using uni-directional current model for EM. Thus the uni-directional current is assumed in this paper.

Assuming *case B* as the *standard layout*, we note that the EM of a certain via position in any wire position case can be equivalent to that of another via position in the standard layout. For example, considering current direction, via position of *(w,n,e,s)* of *case A* can be considered as *(e,n,w,s)* of *case B* respectively when we calculate EM. We call this conversion as *transposition* in this work.

Next, we introduce the *unit structure* as a unit of evaluating the EM-failure in the following sections.

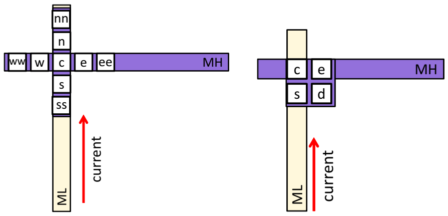
Definition 1 *A unit structure is the one with 1) two orthogonal wires with standard layout, and 2) a center via* c*.*

As all the wire position cases are transposable with each other, we will analyze EM for *case B* in the rest of this paper, which we call the *standard layout.*

As we add redundant vias on top of the unit structure, depending on the combination of redundant vias, we can have various layout cases. Fig. 3(a) shows possible redundant via positions for double/triple via cases from the unit structure. At the cross point of the wires in a unit structure, the original via *c* is located. Additional redundant vias occupy positions closer to *c* first to minimize blockage penalty. Thus {*s, e, n, w*} are the possible candidates for the first redundant via. If we consider *s* as the next redundant via, we may choose one among the closest candidates for each direction, {*ss, e, n, w*} positions. We call it a '*css*' layout case if we choose three vias *(c, s, ss)*, and call it '*cse*' for a *(c, s, e)* selection. This way, we can have ten possible layout cases for triple vias, *css, cse, csn, csw, cee, cen, cew, cnn, cnw* and *cww*. For double vias, *cs, ce, cn, cw* are four possible layout cases for our redundant via scheme. To indicate each via in an RV case, brackets are used in this paper; for instance, *cs[c]* means center via among *cs* formation. We note that stubs can be needed depending on the redundant via position. For example, in the *cs* structure, a higher metal (MH) stub should exist to connect with an *s* via. In the case of quadruple via cases, we consider only a single case, which consists of a 2 by 2 array, with a stub structure for both MH and ML, as shown in Fig. 3(b). Here *d* is the diagonal via from the center via *c*. Table 1 summarizes the single via and redundant via (RV) cases that we consider during redundant via insertion. We will use these RV cases for the rest of the paper.

#Via	1	2	3	4
RV case	c	cs, ce, cn, cw	css, cse, csn, csw, cee, cen, cew, cnn, cnw, cww	cesd

Table 1: Redundant via (RV) cases for EM analysis



(a) (b)

Figure 3: (a) Redundant via positions for double/triple via cases, (b) Quadruple via position.

3. EM MODELING FOR REDUNDANT VIAS

3.1 EM Modeling Flow for Redundant Vias

In a multiple via case, if a current imbalance exists between vias, the growth of one via becomes faster than the other. The larger void size in a via increases resistance of the via, and it affects current distribution between the entire structure, creating a feedback loop. To simulate EM-related failure with resistance change, we model void growth time of redundant via structure considering the transient void growth effect on the current distribution.

```
foreach case;     /* redundant via position case */
do
    ti = 0;                          /* discrete time */
    while (1) do
        foreach Via i in case do
            voidSize[case][i]=getVoidGrowth(dt);
            resistance[case][i]=R-LUT(voidSize[case][i]);
            if voidSize[i] ¿ critSize then
                Tf[case][i] = ti;
            end
        end
        if all the vias in case failed then
            return Tf[case];
        end
        updateCurrent(resistance[case]);
        ti = ti + dt                  /* dt=time step */
    end
end
```

Algorithm 1: EM modeling flow for redundant vias

Algorithm 1 presents our EM modeling flow. Input of our modeling is the initial current density of the wire, before any growth of voids occurred. For the discrete time step, we calculate the void growth of each via for the current time step. Inspired by the previous studies [15, 16], we use a cylindrical void model under the via, and calculate the void radius *dr* that grows during the time step *dt*. Assuming vacancy flux during time *dt* generates void growth with volume *dV* [15, 16],

$$dV = \alpha f \Omega J_v dt = \alpha f \Omega \frac{DC}{kT} Ze\rho j \cdot dt = 2\pi r_{void} \delta dr \qquad (1)$$

where J_v is vacancy flux, δ is thickness of void and r_{void} is radius of the cylindrical void. Diffusivity of vacancies D is expressed as an Arrhenius equation with initial diffusivity D_0, which is an exponential function of temperature T and

Parameter	Description	Value
T	Temperature	$423K = 150\ ^\circ C$
k	Boltzmann const.	1.38×10^{-23}
α	Ratio of captured vacancies	1.0 [15]
f	Ratio of vacancy volume	0.4 [17]
Ω	Atomic volume	1.182×10^{-29} [17]
D_o	Initial diffusivity	0.0047
Ea	Activation Energy	$0.9eV = 1.44 \times 10^{-19}V$
Z^*	Effective charge const.	1.0 [15]
e	Electron charge	$1.6 \times 10^{-19}C$

Table 2: Parameter values for EM modeling

activation energy Ea as depicted in Eqn. (2).

$$D = D_0 \cdot exp\left(\frac{-Ea}{kT}\right) \tag{2}$$

The parameter values in our model are shown in Table 2. Our EM failure criteria is based on the size of the voids. If one of the vias, say via_i, has a larger than the critical void size (=via size), we report the current time step as the failure time of via_i. Still, if we have other vias alive, the unit structure is functionally working. Thus the ultimate failure time of the unit structure is the time when all the vias fail in it.

The next step is to update the current distribution to reflect the void growth in the current time step. Once the void under one via gets larger, the other via can experience more current crowding. To simulate current distribution for each time step, FEA simulation-based look-up tables and resistance network models are used, similar to the work in [16].

3.2 Current Distribution of Redundant Vias

Different layouts of redundant vias have different current distributions, therefore they affect failure time from EM. We use an FEA simulator to get the initial current distribution of each via. As an example, we show two triple via cases in Fig. 4. It displays the 75 percentile of current density of each via within different layout cases, *css* (on-track case) and *cnn* (off-track case), when the input current density of the lower wire is $1e10[A/m^2]$. In the figure, we can see that on-track cases show a similar current density between vias, while off-track cases show a more uneven distribution. Double via cases have a similar trend, off-track cases show more uneven current density. During the EM modeling that we discussed in Section 3.1, we use *effective resistance* of each of redundant vias to consider the current balancing effect from different layouts. As an example, if current density of triple vias are $a : b : c$ from our simulation, effective resistance of them can be the inverse value of it, $1/a : 1/b : 1/c$. On top of the effective resistance from current imbalance, we consider the resistance change due to the voids as we discussed in Section 3.1.

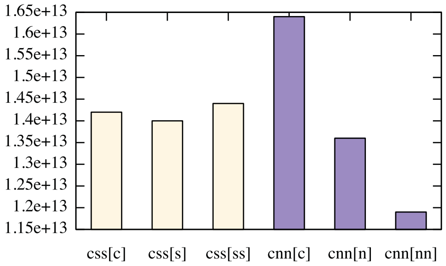
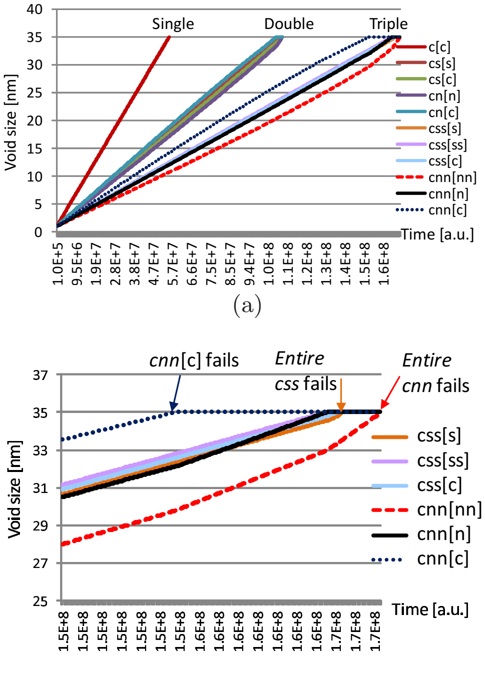


Figure 4: Current density of each via with different layouts, in $[A/m^2]$.

3.3 Effect of Current Imbalance in Void Growth

Fig. 5(a) shows void growth time of a few example RV cases from our model. As expected, the single via fails first, followed by double and triple layout cases from the figure.



(a)

(b)

Figure 5: (a) Void growth time of redundant vias up to three vias, (b) zoom-in shot for triple via cases.

Within each layout case, larger current imbalance of a case often correlates with larger reliability. The *cnn* case is an example from Fig. 5(b); one of its via *cnn[c]* fails earlier than the other vias in triple via layouts due to current imbalance. However it helps the off-track via *cnn[n]* remain relatively unaffected by EM. As a result, *cnn* can have a longer lifetime than the on-track layout *css* case. Although the difference between the lifetime of *css* and *cnn* may not be very significant, it still shows that on-track redundant vias do not have any advantages in terms of EM reliability at least. The double via case shows a similar trend: the off-track RV case *cn* shows an equal or larger lifetime than the one with on-track vias, *cs*.

3.4 EM Library for Layout Optimization

Since the evaluation of the lifetime should be done quickly and accurately during our optimization, we pre-generate the EM library to store failure time of each layout case by sweeping the current density of the wire. The failure time of each case is calculated by our EM estimation, as discussed in Algorithm 1. The inputs of the EM library are RV case and initial current density of the original via, and the output is failure time of the case. With the EM library, we can estimate the lifetime of the structure with known input current, with minimum runtime. Although we do not show the temperature difference in this paper, our EM library can be easily expandable to the temperature dimension as well, since our model considers temperature as shown in Eqn. (2).

4. EM-AWARE REDUNDANT VIA INSERTION

Fig. 6 illustrates our EM-aware redundant via insertion flow, which is a four-step process. (1) Redundant via candidate generation by selecting vias vulnerable to EM (Section 4.1); (2) Conflict graph construction (Section 4.2); (3) ILP based redundant via insertion (Section 4.3); (4) Maximize the number of the via after EM-optimization (Section 4.4). It shall be noted that independent component and articulation point are speed-up techniques, and will be discussed in

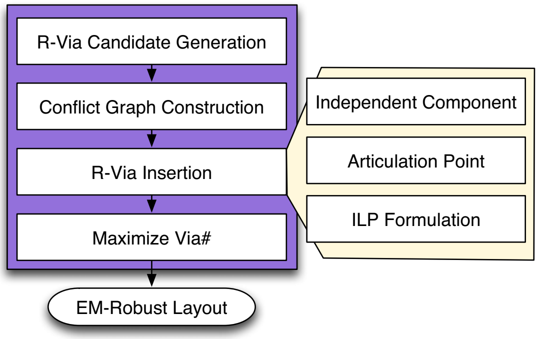


Figure 6: Overall flow of EM-aware via insertion.

Section 5.

4.1 Redundant Via Candidate Generation

The first step of EM evaluation of redundant vias is to transpose the wire positions to the *standard layout* case, as discussed in Section 2.2. After that, for each unit structure, we estimate the failure time of each redundant via (RV) case. In our work, $Tf_{(id,case)}$ stands for the failure time of RV case *case* of the unit structure id *id*. Evaluation of EM-failure time is done by utilizing pre-generated look-up table that we discussed in Section 3.4, based in EM model described in Section 3. We evaluate the *EM safeness (EMS)* based on the failure time, as in Eqn. (3).

$$EMS_{(id,case)} = \begin{cases} MaxCost, & \text{if } Tf_{(id,case)} \geq Tf_{th} \\ MinCost, & \text{otherwise} \end{cases} \tag{3}$$

Our goal of EM-aware redundant via is to maximize the number of unit structures that are EM-robust. Thus, if an RV case has already achieved long enough failure time than the threshold, additional vias are not needed in terms of EM; it may be better to reserve the routing area for more redundant vias of EM-dangerous nets. Therefore, if the lifetime is longer than the target, the RV case is likely to be chosen. $MaxCost$ and $MinCost$ are determined empirically, we use 3 and 1, respectively.

4.2 Conflict Graph Construction

After evaluating EM failure time of all the possible RV layout cases, we construct the conflict graphs. A conflict graph is an undirected graph with a single set of vertices **V**, and two sets of edges, **IE** and **EE**, which contain the internal edges and external edges, respectively. For each unit structure and each of its possible RV layout cases, we generate a vertex $(id, case) \in V$. Here id and $case$ refer to the unit structure id and one of its possible RV layout cases, respectively. In our experiments, we can have up to 16 vertices for a unit structure, according to the RV cases as we discussed in Table 1. An edge is introduced if and only if the two corresponding vertices are conflict with each other. In other words, if two vertices are connected by an edge, we can choose only one of them as the final solution. If both cases are with the same id, this edge belongs to internal edges IE. Otherwise, two vertices are from the different unit structures, and this edge belongs to external edges EE. It shall be noted that for one unit structure, any two cases would be connected by one internal edge.

Fig. 7 illustrates a conflict graph example with two unit structures i and j. Each unit structure has five cases, and there are totally four external edges. During conflict graph construction, to save search space, we use *bins* to limit the neighbor search space. Bins are big grids in the x-y plane, and we check conflicts in the current and neighboring bins only while checking the conflicts.

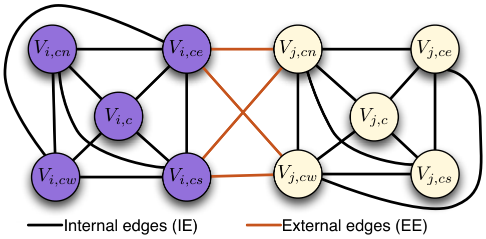


Figure 7: Example of conflict graph.

In our algorithm, a single via case c is assigned as one of the RV cases. In case of a vertex having an external edge with neighboring single via case, the vertex should not be selected. Therefore, during conflict graph construction, we prune edges and vertices that have external conflicts with single via case. Table 3 shows an example of pruning process to prevent zero-via cases. In the example, $(3, cs)$ should not be selected because it has an external edge with neighboring single via, $(1, c)$ case. So we remove $(3, cs)$ from the V. In the set of edges, all the edges connecting to $(3, cs)$ are now obsolete, thus they need to be removed. In this example, an edge $((2, c), (2, cn))$ is not pruned, because it is an internal edge originated from the same unit structure id.

Table 3: Example of conflict edge pruning

Before pruning		After pruning	
Vertices	Edges	Vertices	Edges
(1,c)	((1,c),(3,cs))	(1,c)	–
(3,cs)	((2,cn),(4,cww))	–	((2,cn),(4,cww))
(2,cn)	((1,c),(1,cs))	(2,cn)	((1,c),(1,cs))
(4,cww)	((1,c),(1,cn))	(4,cww)	((1,c),(1,cn))
(1,cs)	((5,cne),(3,cs))	(1,cs)	–
(1,cn)	((3,c),(3,cs))	(1,cn)	–
(5,cne)	((2,c),(2,cn))	(5,cne)	((2,c),(2,cn))
(3,c)		(3,c)	
(2,cn)		(2,cn)	

4.3 ILP Formulation

The EM-aware redundant via insertion is formulated as an integer linear programming (ILP):

$$\max \sum_{(id,case)\in V} EMS_{(id,case)} \cdot R_{(\text{id},\text{case})} \tag{4}$$

$$\text{s.t.} R_{(id,case)} + R_{(id,case')} = 1, \forall((id, case), (id, case')) \in IE$$

$$R_{(id,case)} + R_{(id',case')} = 1, \forall((id, case), (id', case')) \in EE$$

where the objective is to maximize the number of EM-safe unit cases. We represent the possible redundant via insertions using a binary variable $R_{(id,case)}$, and id and $case$ represent unit structure id and redundant via position case, respectively. $R_{(id,case)} = 1$ indicates that vertex $(id, case)$ is selected for the design. The constraints are based on both the internal edges and the external edges in the conflict graph.

4.4 Maximizing Vias After EM-Optimization

From the design for manufacturing (DFM) perspective, it is always desirable to have more redundant vias, to increase yield. Thus [5] focused on maximizing the total number of redundant vias. Because we do not *over-insert* RVs for the safe enough unit structures, we may achieve that goal with smaller number of redundant vias than the conventional DFM-aware RV insertion. However, if we put smaller number of redundant vias than the DFM-aware RV insertion, that may be inferior for yield, although it can be good enough for EM.

Thus, to increase yield, which is our second goal, we suggest a post via insertion technique to maximize the number of vias, after the initial EM-aware via insertion of Section 4.3. The basic idea is to insert as many as possible additional vias on top of the EM-aware optimized RV solution. By this way, we can achieve both EM-awareness and yield improvement.

5. SPEED-UP TECHNIQUES

For practical design, solving ILP may suffer from runtime overhead problem. In this section we present several techniques to speed-up the expensive ILP. The main idea is that instead of solving the whole conflict graph, we can divide the whole graph into a set of components. Then each component can be solved independently.

5.1 Simplified Conflict Graph

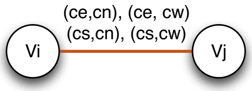


Figure 8: Simplified conflict graph.

To provide more flexibilities to achieve further speed-up, we introduce a simplified conflict graph model. For the initial conflict graph in Fig. 7, the corresponding simplified conflict graph is illustrated in Fig. 8. It shall be noted in the simplified graph, to connect two via units, some hyper edges would be introduced. The motivation of the simplified conflict graph is twofold: 1) In the simplified conflict graph only external conflict edges are maintained, thus the graph size can be significantly reduced. For example, instead of 24 edges in initial graph in Fig. 7, only one edge is left in simplified conflict graph. 2) Due to the simplification, more articulation points can be identified. The discussion regarding the articulation point detection appears in Section 5.3. Based on a simplified conflict graph, we introduce two techniques to further divide and simplify the graph.

5.2 Independent Component Computation

Our first technique is called *independent component computation*, similar to that of the work [5]. In a conflict graph of real design, we observe many isolated subgraphs. By breaking down the whole conflict graph into several independent components, we partition the initial graph into several small components or subgraphs. Since no edge exists between any two components, the redundant via insertion problem can be solved independently for each component, and the final solution can be obtained by taking the union of sub-solutions.

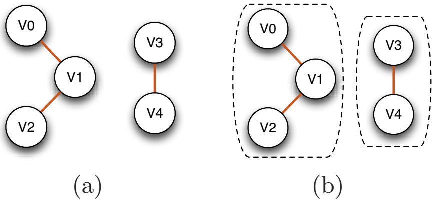


Figure 9: Independent component computation.

Fig. 9 illustrates one example of independent computation computation. Given the conflict graph, we can detect two independent components, and can apply ILP formulation to solve them separately. All the components can be calculated in linear time $O(|V|+|E|)$, where $|V|$ and $|E|$ are the vertex number and the edge number, respectively.

5.3 Articulation Point Computation

Our second technique is called *articulation point computation*. In the conflict graph, a vertex is an articulation point if and only if removing it (and edges through it) disconnects the graph into two or several components. For example, as illustrated in Fig. 10 (a), in the conflict graph vertices $v0$ and $v3$ are articulation points, since removing either of them would divide the whole conflict graph into two components. In our simplified conflict graph, each vertex represents one via unit. For each articulation point, we apply a process to check whether one redundant via case can be pre-selected. If yes, then the vertex would be temporally removed from conflict graph, and the conflict graph can be divided. To the best of our knowledge, this study is the first one to introduce articulation point computation to redundant via insertion; this method can be only applied to the redundant vias with different costs (e.g. failure time) because we need to pre-select the via case based on that.

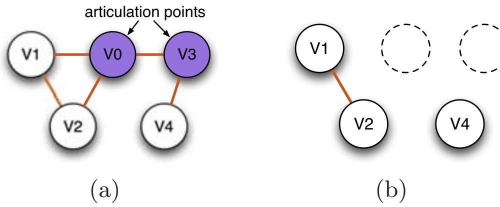


Figure 10: Articulation point computation.

To find all articulation points in a conflict graph, we apply a depth-first search (DFS) algorithm presented by Tarjan [18]. Note that all articulation points can be detected during only one DFS in linear time $O(|V|+|E|)$, where $|V|$ and $|E|$ are respectively the number of vertices and the number of edges in a conflict graph.

6. EXPERIMENTAL RESULTS

We implement our algorithm in Python and C++ languages. All the experiments are performed on a 2.93GHz Intel Quad Core Linux Machine. For benchmark generation, we synthesize OpenSPARC T1 designs based on Nangate 45nm standard cell library [19]. We choose GUROBI [20] as the ILP solver. As we mentioned in Section 2, we use the unit structure to count the EM-violations. Temperature is assumed to be $150^{\circ}C$ and the initial current density of the wire in each unit structure is randomly chosen between $1e8$ and $3e10[A/m^2]$. We set the temperature higher than the most of the normal conditions to accelerate the EM failure, similar to that of oven test of fabricated chips. Thus the number of failed units in our results may be larger than that of normal temperature.

Table 4 shows our experiment results that compare our proposed method (EM-RV) against the conventional RV insertion. EM-RV(S) represents EM-aware RV with speed-up techniques, which will be explained later. We limit the maximum number of total vias of a unit structure to four in our experiments, for both conventional RV mode and EM-RV modes. In the conventional RV mode, we put as many as possible, similar to the approach in work [5]. Here the target is to maximize the total number of vias. In EM-RV, we follow the steps described in Section 4.1 to 4.4, mainly targeting to improve EM lifetime. In this way, we can compare conventional redundant via insertion to our EM-aware one.

In the fourth column, we show that our proposed EM-aware redundant via insertion can reduce the number of failed unit. If the failure time of the unit structure is smaller than threshold, 2e8 seconds in our case, it indicates that the structure is EM-failed. Throughout the designs, EM-RV shows less failed units than the baseline. For example, with *alu* design, EM-RV shows 24.9% reduction in terms of failed units. Since we insert redundant vias to the EM-prone nets to satisfy the lifetime criteria, we can achieve better EM reliability with

Table 4: EM-aware via insertion results.

Design	Mode	# Unit	# Failed Unit	# Via	runtime
alu	RV	5661	710 (-)	21346	2.7s
	EM-RV	5661	533 (-24.9 %)	21023	2.5s
	EM-RV(S)	5661	535 (-24.6 %)	21026	0.6s
byp	RV	24383	3331 (-)	90166	18.3s
	EM-RV	24383	2298 (-31.0 %)	88221	14.7s
	EM-RV(S)	24383	2318 (-30.4 %)	88221	3.5s
div	RV	11635	1411 (-)	43807	6.1s
	EM-RV	11635	1016 (-28.0 %)	43107	5.5s
	EM-RV(S)	11635	1022 (-27.6 %)	43128	1.2s
ecc	RV	4068	488 (-)	15470	1.7s
	EM-RV	4068	375 (-23.2 %)	15283	1.5s
	EM-RV(S)	4068	376 (-23.0 %)	15286	0.3s
efc	RV	3130	356 (-)	11977	1.2s
	EM-RV	3130	278 (-21.9 %)	11845	1.1s
	EM-RV(S)	3130	278 (-21.9 %)	11856	0.2s
ffu	RV	5078	601 (-)	19347	2.1s
	EM-RV	5078	473 (-21.3 %)	19128	1.9s
	EM-RV(S)	5078	474 (-21.1 %)	19121	0.5s
mul	RV	44085	5594 (-)	165913	28.2s
	EM-RV	44085	4124 (-26.3 %)	163303	21.1s
	EM-RV(S)	44085	4142 (-26.0 %)	163429	4.7s

smaller number of total vias (21023 vs. 21346). On the other hand, in case of conventional RV mode, because the goal is to maximize the total number of redundant vias, EM-prone nets may not be selected to insert more redundant vias, and EM-safe nets can have more unnecessary vias in terms of EM.

The runtime of ILP can be a problem if the problem size is huge. Thus we apply speed-up techniques as we discussed in Section 5, for EM-RV(S) mode. With *articulation point computation* described in Section 5.3, we divide a conflict graph into subgraphs. As a result, runtime is reduced significantly, for example, from 2.5s to 0.6s in case alu. The penalty of smaller runtime is loosing the optimality in terms of EM. However, because our method favors EM-resistant RV cases during resizing graphs, the amount of EM degradation between EM-RV and EM-RV(S) is very small. For example, in terms of number of failed units of *alu*, there are 533 units in EM-RV mode and 535 in EM-RV(S) mode.

7. CONCLUSION

With advanced manufacturing technology, reliability becomes one of the most important issues that requires careful planning. In this work, we suggest a design methodology to enhance EM-robustness.

Our modeling of EM with diverse layouts of redundant vias shows that the current imbalance in different vias of a unit structure affects the reliability. We suggest that off-track via layouts could be preferable in terms of EM, deviating from the previously conventional view of redundant via insertion.

We also propose a post-layout optimization method that can improve the EM reliability of redundant vias by exploiting the diverse design space of them, such as the via placement or the number of vias. Because the suggested method inserts the right amount of redundant vias for the EM-risky nets first to satisfy the target failure time, we can achieve much better EM reliability than the conventional redundant via insertion technique. To reduce runtime of solving ILP, we investigate a set of speed-up techniques as well.

Our results suggest that even with the similar number of total redundant vias as the conventional method, we can achieve better EM reliability by smart allocation of vias. Because EM problems are getting severe with smaller interconnect dimensions, we expect that the EM-aware via insertion would have more importance in the future.

Acknowledgment

This work is supported in part by National Science Foundation (NSF) and Semiconductor Research Corporation (SRC).

8. REFERENCES

[1] B. Li, T. D. Sullivan, T. C. Lee, and D. Badami, "Reliability challenges for copper interconnects," *Microelectronics Reliability*, vol. 44, no. 3, pp. 365–380, 2004.

[2] T. Wang, T. Hsieh, M.-T. Wang, D.-S. Su, C.-H. Chang, Y.L.Wang, and J. Y. min Lee, "Stress Migration and Electromigration Improvement for Copper Dual Damascene Interconnection," *Journal of The Electrochemical Society*, 2005.

[3] G. A. Allan, "Targeted layout modifications for semiconductor yield/reliability enhancement," *IEEE Trans. on Semiconductor Manufacturing*, vol. 17, no. 4, pp. 573–581, 2004.

[4] K.-Y. Lee and T.-C. Wang, "Post-routing redundant via insertion for yield/reliability improvement," in *ASPDAC*. IEEE, 2006, pp. 303–308.

[5] K.-Y. Lee, C.-K. Koh, T.-C. Wang, and K.-Y. Chao, "Fast and optimal redundant via insertion," *TCAD*, vol. 27, no. 12, pp. 2197–2208, 2008.

[6] S.-T. Lin, K.-Y. Lee, T.-C. Wang, C.-K. Koh, and K.-Y. Chao, "Simultaneous redundant via insertion and line end extension for yield optimization," in *ASPDAC*. IEEE, 2011, pp. 633–638.

[7] G. Xu, L.-D. Huang, D. Z. Pan, and M. D. Wong, "Redundant-via enhanced maze routing for yield improvement," in *ASPDAC*. ACM, 2005, pp. 1148–1151.

[8] H.-Y. Chen, M.-F. Chiang, Y.-W. Chang, L. Chen, and B. Han, "Full-chip routing considering double-via insertion," *TCAD*, vol. 27, no. 5, pp. 844–857, 2008.

[9] R. de Orio, H. Ceric, and S. S., "Physically based models of electromigration: From Black's equation to modern TCAD models," *Microelectronic Reliab.*, vol. 9, 2010.

[10] A. S. Oates and M. H. Lin, "Electromigration failure distributions of Cu/low-k dual-damascene vias: Impact of the critical current density and a new reliability extrapolation methodology," *IEEE Trans. Device. Mater. Reliab*, vol. 9, no. 2, pp. 244–254, 2009.

[11] J. R. Lloyd, "Electromigration Failure," *Journal of Applied Physics*, 1991.

[12] T. Gupta, "Copper Interconnect Technology," *Springer*, 2009.

[13] N. Nagaraj, C. Frank, H. Haldun, and Y. Duane, "A Practical Approach to Static Signal Electromigration Analysis," in *DAC*, 1998.

[14] D. T. Blaauw, C. Oh, V. Zolotov, and A. Dasqupta, "Static Electromigration Analysis for On-Chip Signal Interconnects," *TCAD*, vol. 22, 2003.

[15] R. de Orio, H. Ceric, and S. S., "Electromigration failure in a copper dual-damascene structure with a through silicon via," in *Microelectronic Reliab.* Elsevier, 2012.

[16] J. Pak, S. K. Lim, and D. Z. Pan, "Electromigration study for multi-scale power/ground vias in tsv-based 3d ics," in *ICCAD*, 2013.

[17] M. Sarychev, Y. Zhitnikov, C.-L. Borucki, L. andLiu, and T. Makhviladze, "General model for mechanical stress evolution during electromigration," in *Journal of Applied Physics*. AIP, 1999, pp. 3068–3075.

[18] R. E. Tarjan, "A note on finding the bridges of a graph," *Information Processing Letters*, vol. 2, pp. 160–161, 1974.

[19] "NanGate FreePDK45 Generic Open Cell Library," www.si2.org/openeda.si2.org/projects/nangatelib.

[20] "GUROBI," http://www.gurobi.com/html/academic.html.