UT DA

Electromigration-aware Redundant Via Insertion

Jan. 21st, 2015

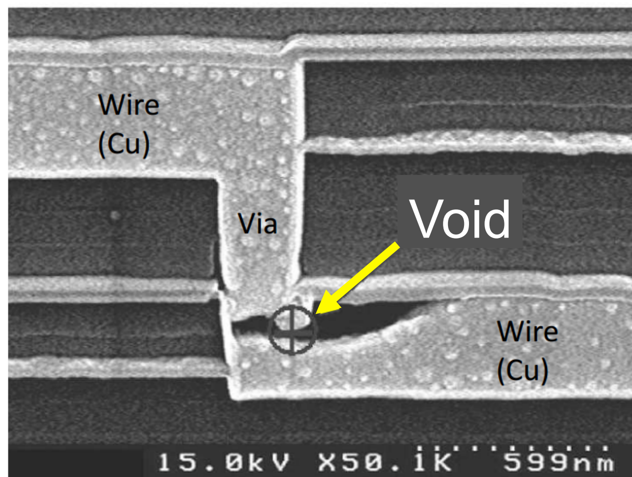
Jiwoo Pak, Bei Yu, David Z. Pan
The University of Texas at Austin

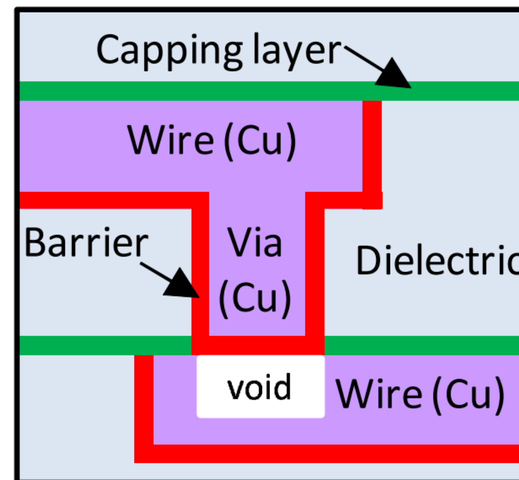
Outline

1. Introduction
2. EM modeling for redundant vias
3. EM-aware via insertion
4. Results
5. Summary

Electromigration (EM)

- Electromigration (EM) is getting severe in the modern ICs
  - EM: atomic diffusion due to high current density
  - Under the local via trench is one of the weakest point
  - EM is a function of current density, mechanical stress and temperature





Motivation: EM in Redundant Via

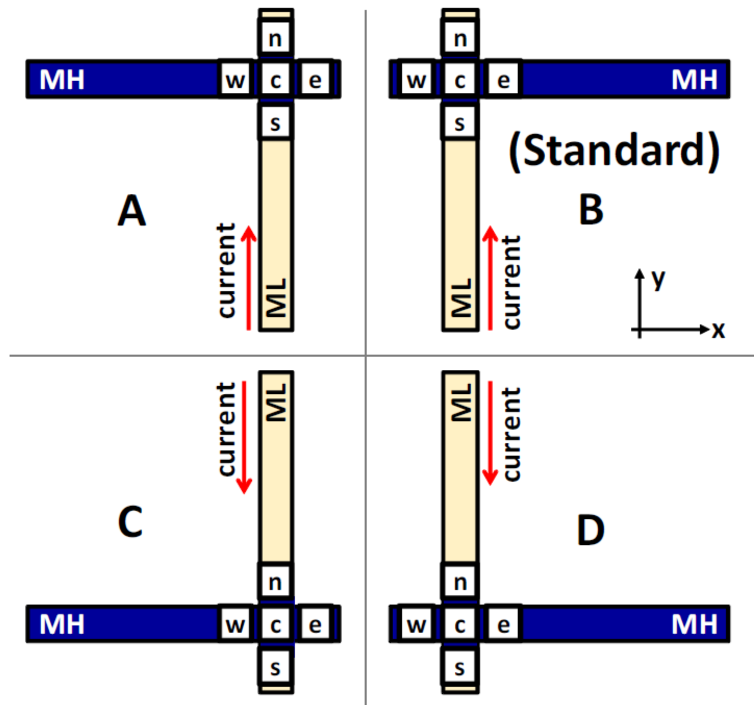
- Redundant via can increase yield in general
- However, maximizing total # vias may not be the best for EM, because current densities of nets can be different

"How can we qualify & optimize EM-related lifetime during post-layout via insertion?"

- In this study,
  - Study EM of different combination of redundant via layouts
  - Suggest smart allocation of redundant vias considering EM

Transpose of Layouts for EM Analysis

- 8 wire position cases exist with 2 wires and 1 via
  - With two orthogonal wires in the adjacent routing layers
- EM analysis of any of them is equivalent to the EM analysis of the another case. ➔ Transposition!



Example of Transposition

- {w} in case A == {e} in case B
- {n} in case D == {s} in case B

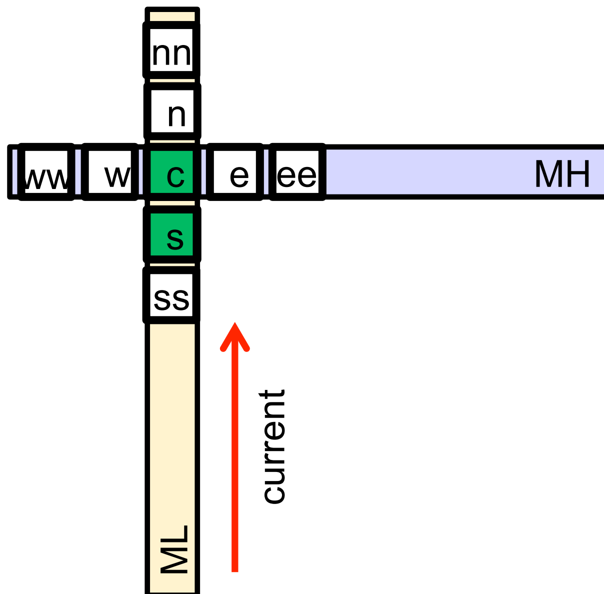
in terms of EM

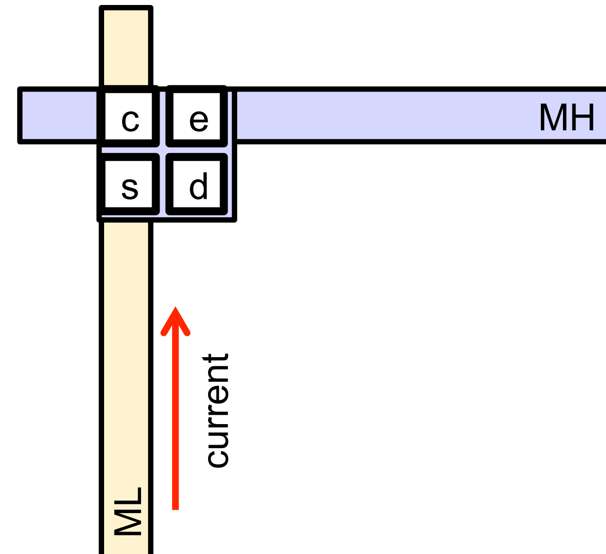
Redundant Via Layout Cases

- Redundant via layout cases in our study

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

- Example: *cs*[c] means center via [c] with *cs* via formation

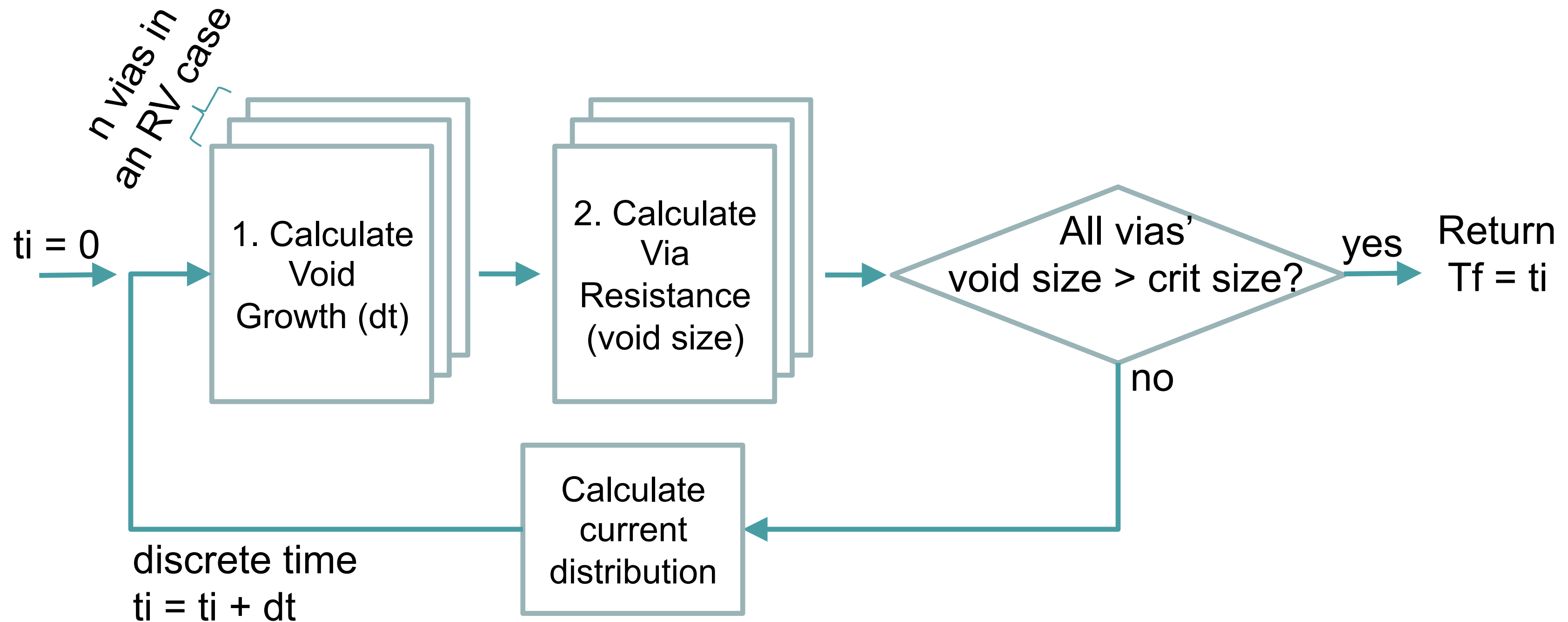




1. Introduction
2. EM modeling for redundant vias
3. EM-aware via insertion
4. Results
5. Summary

Flow of EM Modeling for Redundant Vias

- For each RV case, get a failure time Tf



  - Void growth can be calculated through EM equations
  - Using look up table from FEM simulation, via resistance can be calculated

Step 1: Calculation of Void Growth

- Vacancy flux equations for EM modeling

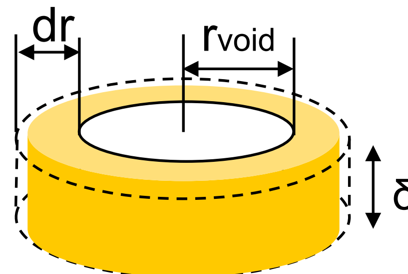
$$\vec{q}_v = \underbrace{\frac{Dc_v}{kT} Ze\rho \vec{j}}_{\text{Current density driven}} + \underbrace{\frac{Dc_v}{kT} Q \frac{\nabla T}{T}}_{\text{Temperature grad. driven}} - \underbrace{\frac{Dc_v}{kT} f\Omega\nabla\sigma}_{\text{Stress grad. driven}}$$

$$D = D_o \exp\left(\frac{-Ea}{kT}\right)$$

- Cylindrical void growth model

$$dV = \alpha f \Omega A q_v dt = 2\pi\delta\, r_{void} dr$$

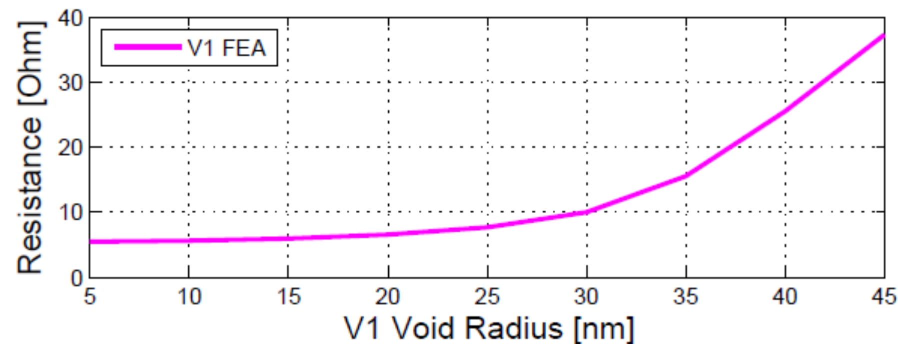
$$dr = \frac{\alpha f \Omega A q_v dt}{2\pi\delta\, r_{void}}$$

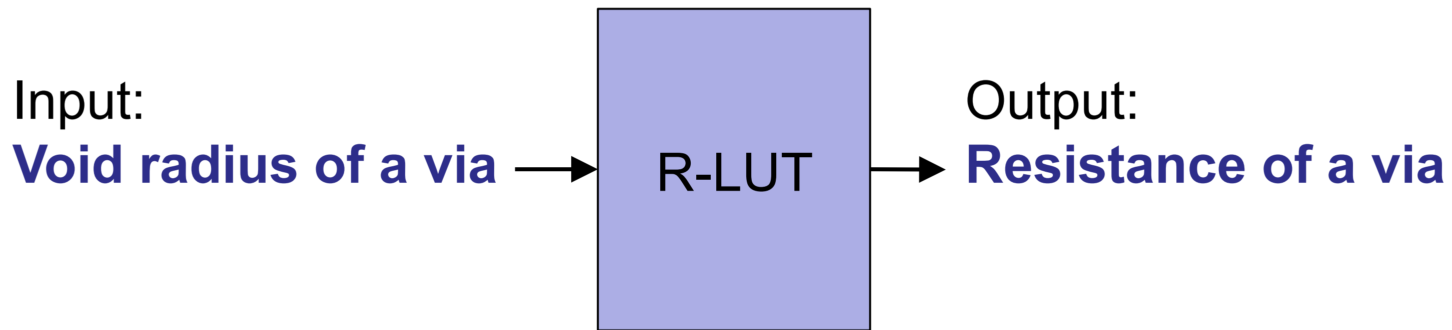


$\vec{q}_v$: Total vacancy flux
D : Diffusivity of vacancy
c_v : Vacancy concentration
$\vec{j}$: Current density vector
σ : Hydrostatic stress
T : Temperature

Step 2: Calculation of Resistance

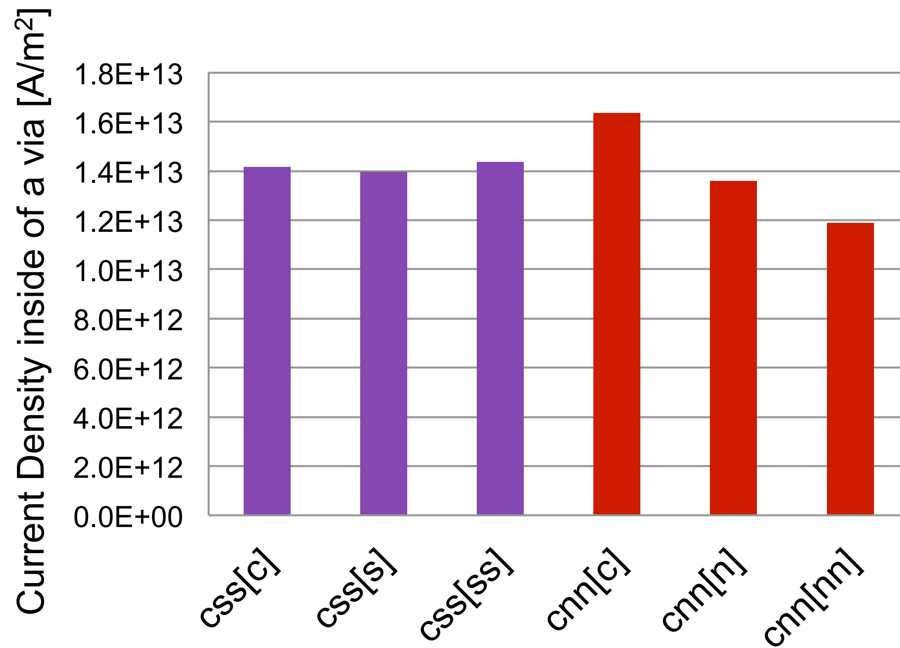
- Generate look-up tables (LUTs) with FEA simulation
  - Input: radius of void
  - output: resistance of the structure

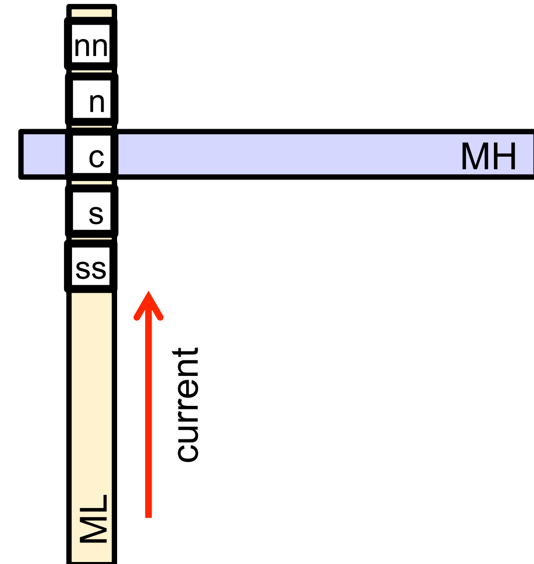




Current Density of 'Off-track' and 'On-track' vias

- Current density of each via with different layouts

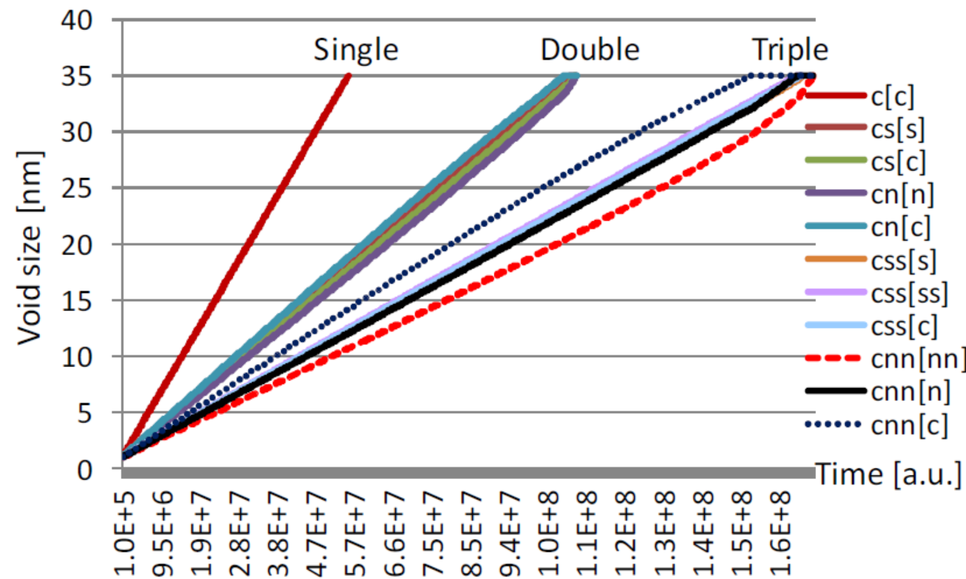


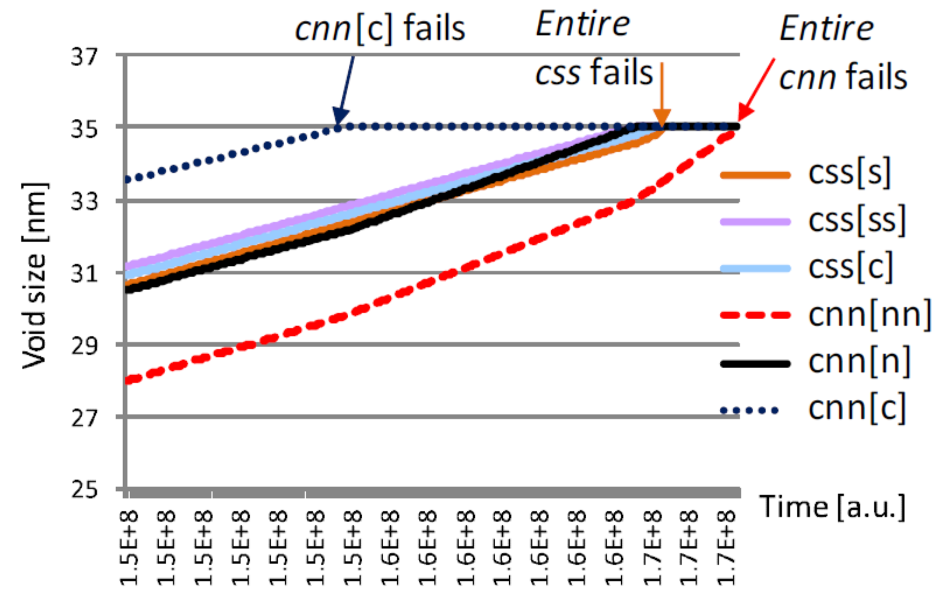


- On-track 'css' case shows more balanced current densities between vias

Void Growth Time of Redundant Vias

- Off-track ‘cnn’ formation shows discrepancy in Tf
- Off-track vias can live longer than the on-track vias



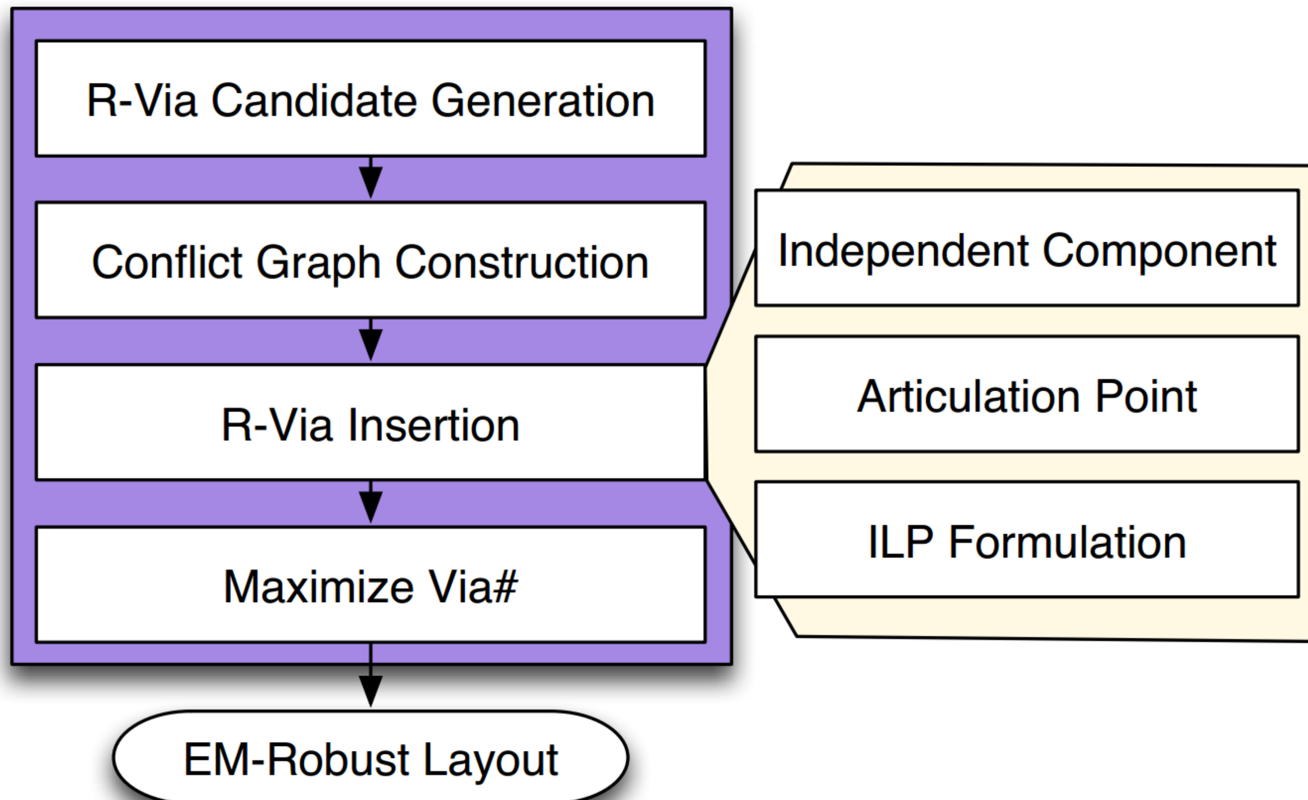


Outline

1. Introduction
2. EM modeling for redundant vias
3. EM-aware via insertion
4. Results
5. Summary

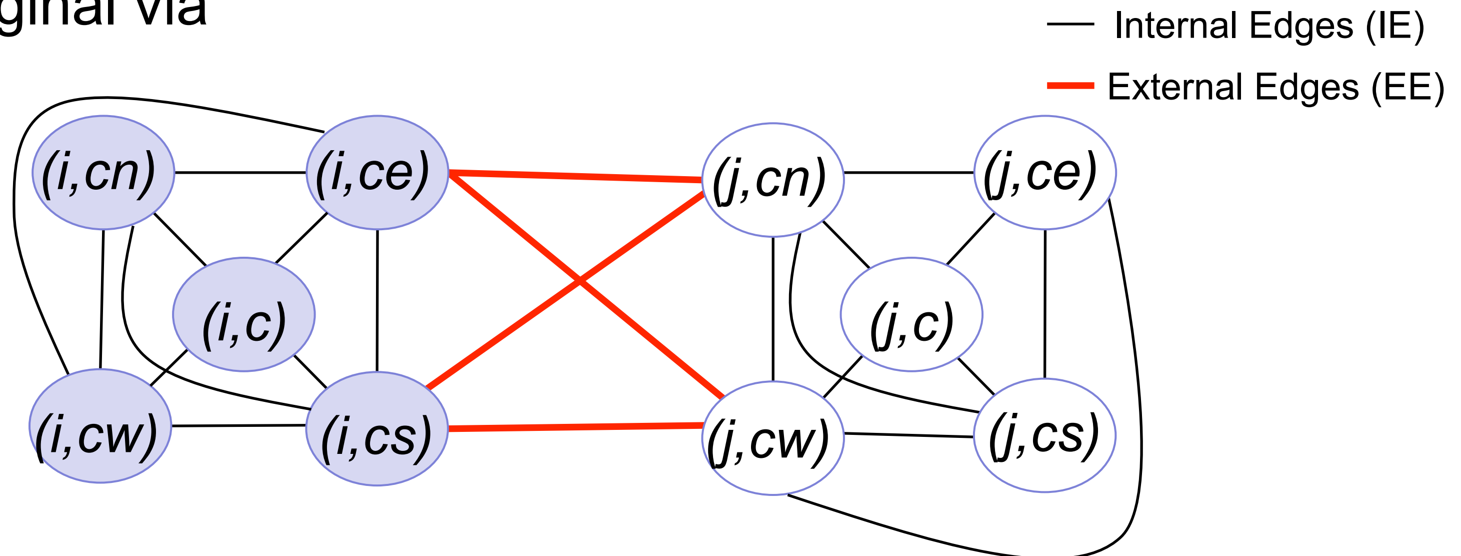
Flow of EM-aware Via Insertion

- Overall flow of EM-aware via insertion



Conflict Graph Construction

- For each unit structure,
  - Add vertices for EM-prone layout cases, if any
  - Each vertex has estimated failure time (Tf) of EM
  - Internal Edges (IE): conflicts with a vertex from the same 'original via'



  - External Edges (EE): conflicts with a vertex from the neighboring unit structures

Formulation for EM-aware Via Insertion

- EM safeness (EMS)*: based on look-up table of Tf

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

- ILP** formulation with conflict graphs

$$R_{(id,case)} = \begin{cases} 0 & \text{, if } (id,case) \text{ is not used} \\ 1 & \text{, if } (id,case) \text{ is used} \end{cases}$$

$$Maximize \sum_{(id,case)\in V} EMS_{(id,case)} \cdot R_{(id,case)}$$

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

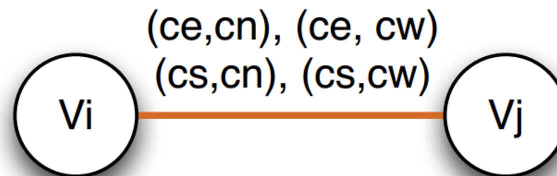
$$R_{(id,case)} + R_{(id',case')} \leq 1, \ \forall((id,case),(id',case')) \in EE$$

*MaxCost, MinCost are constants

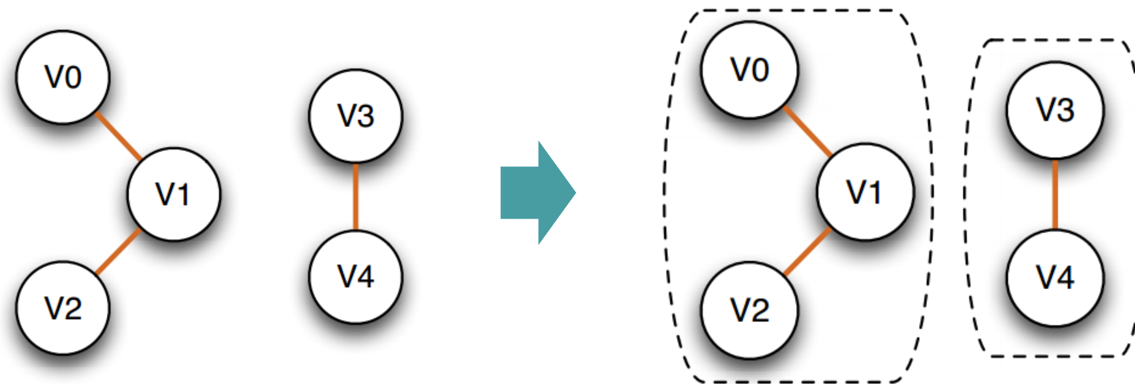
**ILP: integer linear programming

Speed-up Techniques

1. Simplify conflict graphs
   - Show external edges between unit structures only

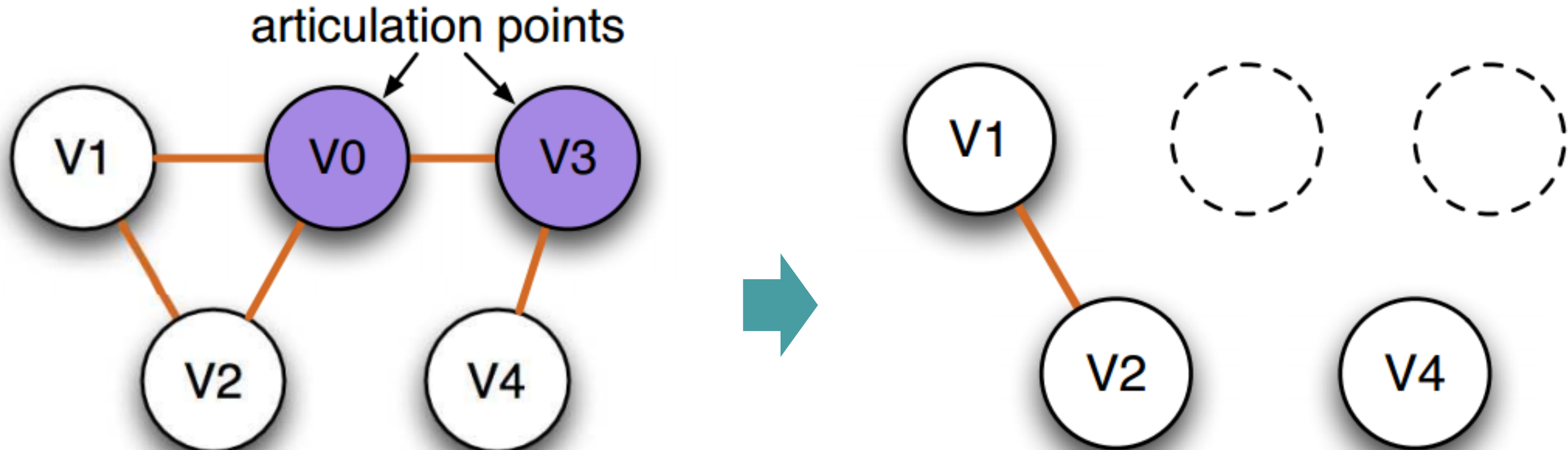


2. Independent component computation
   - Take union of sub-solutions



Speed-up Techniques (cont')

3. Articulation point computation
   - Check if the redundant via case can be pre-selected
     - If one via case can be pre-selected without harm to EM, we can assign a vertex as an articulation point
     - By removing edges of articulation points, graph size can be reduced



Outline

1. Introduction
2. EM modeling for redundant vias
3. EM-aware via insertion
4. Results
5. Summary

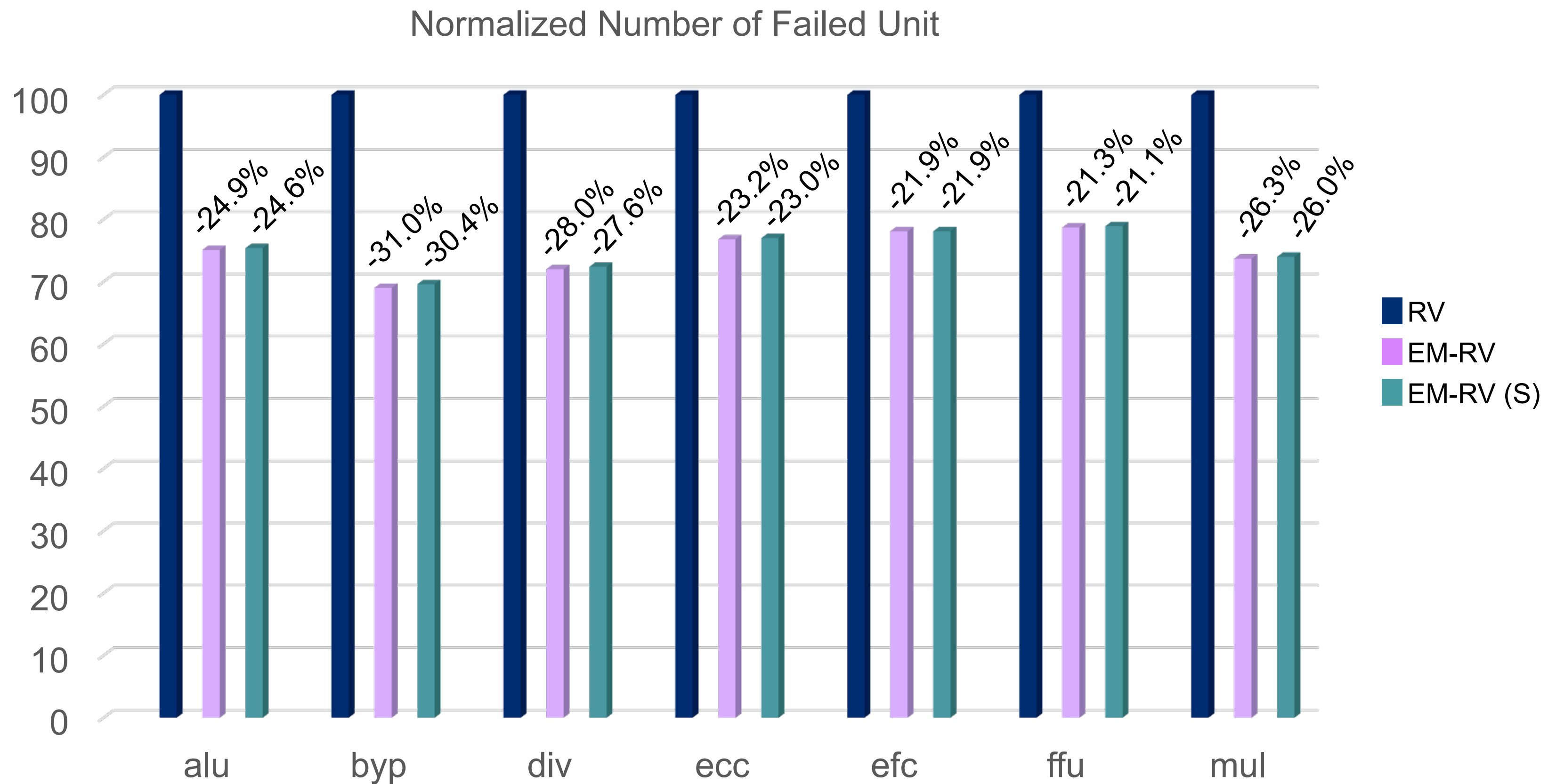
Results: EM-aware Via Insertion

- Benchmark circuits: OpenSparc T1 (Nangate 45nm)
- EM-aware via insertion can achieve better EM reliability with smaller routing resources

Ckt	Mode	# Unit	# EM-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
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

Comparison of Failed Units

- The suggested method reduces EM-failed units up to -31%



1. Introduction
2. EM modeling for redundant vias
3. EM-aware via insertion
4. Results
5. Summary

Summary: EM-aware Redundant Via Insertion

- Modeled and analyzed electromigration (EM) for various redundant-via structures
- Found that the degree of current imbalance in redundant vias affects EM reliability of the whole structure
- Proposed a via-insertion algorithm that can maximize EM reliability than the conventional via insertion, with similar number of total vias
- Investigated a set of speed-up techniques for ILP formulation

BACK UP SLIDES

Algorithm of EM Modeling for Redundant Vias

```
For each case in RV cases
        ti = 0;                         /* discrete time */
        while (1) do
                for each via i in case
                        void_size [case][i] = get_void_growth(dt);
                        resistance [case][i] = R_LUT (void_size[case][i]);
                        if (void_size[case][i] > critical_size)
                                Tf [case][i] = ti;
                if all the vias in case failed
                        return Tf [case];          /* failure time of case */
                update_current (resistance [case])
                ti = ti + dt;                   /* dt = time step */
```