MCFRoute 2.0 : A Redundant Via Insertion Enhanced Concurrent Detailed Router

Xiaotao Jia[1]; Yici Cai[1]; Qiang Zhou[1]; Bei Yu[2]

1 TSINGHUA UNIVERSITY

2 The Chinese University of Hong Kong

Outline

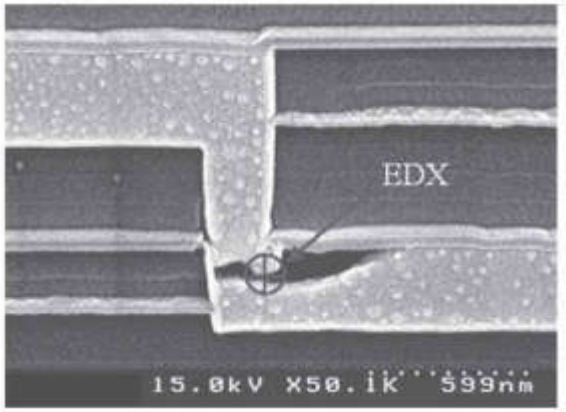
- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

Outline

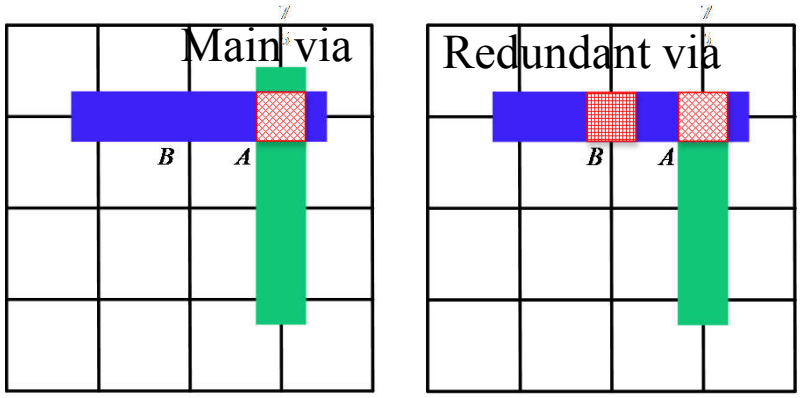
- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

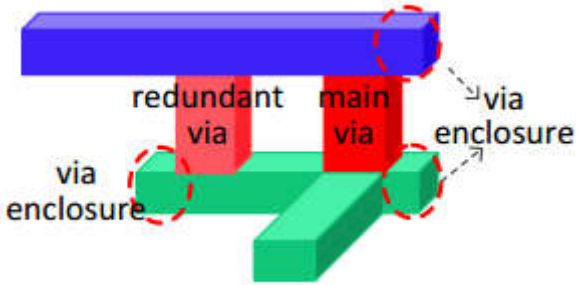
Redundant Via Insertion

- Why do we need to insert redundant via ?
  - via failure has a critical impact on chip performance functionality, and manufacturing yield.
  - Redundant via (also known as double via) insertion is one of the most effective techniques recommended by foundries to provide via reliability.



- What is a redundant via?

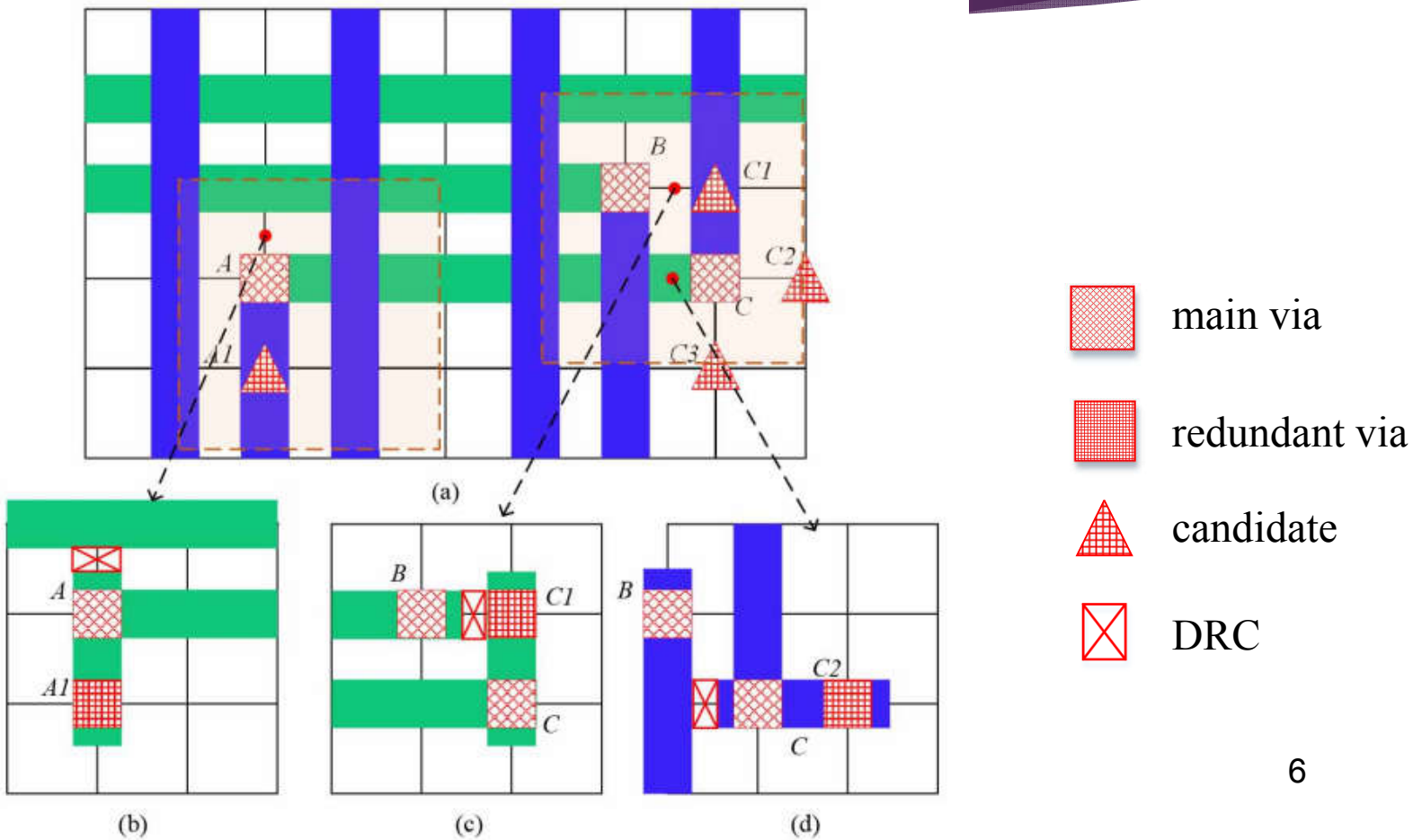




Previous Work

- Post-routing redundant via insertion
  - insert as many redundant vias as possible in post-routing stage
  - Lee[2006ASPDAC]; LEE[2006ICCAD];LEE[2016ISPD];LIN[ASPDAC2011];PAK[ASPDAC2011]
- Redundant via insertion aware routing
  - Considering RVI during routing stage
  - Xu[ASPDAC2005];YAO[GLSVLSI2005];CHEN[TCAD2008];LIN[ASPDAC2011]

Motivation



main via

redundant via

candidate

DRC

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

Outline

- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

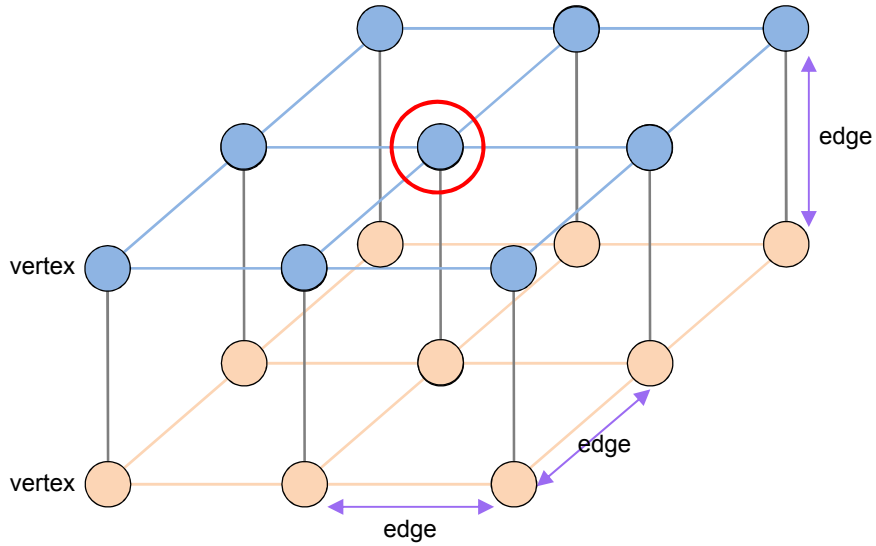
Contributions

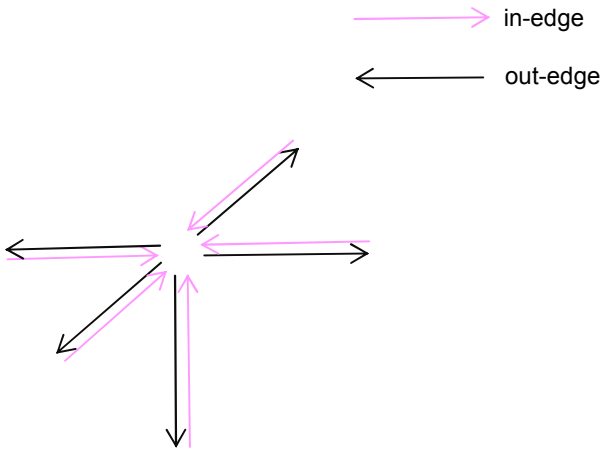
- The proposed router is able to search for redundant via position for every via, while all the nets are routed simultaneously.
- We propose a comprehensive analysis on the effect of via enclosure, which is rarely discussed by previous academic works. It makes the proposed algorithm much practical.
- We propose a routing resource aware heuristic technique, where an RVI aware pin access allocation algorithm is proposed to improve the via insertion rate on layer Via12.

Outline

- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

Graph Model





- Vertex: intersection point of routing grid
- Edge: sub-grid divided by vertex
- Routing graph is directed

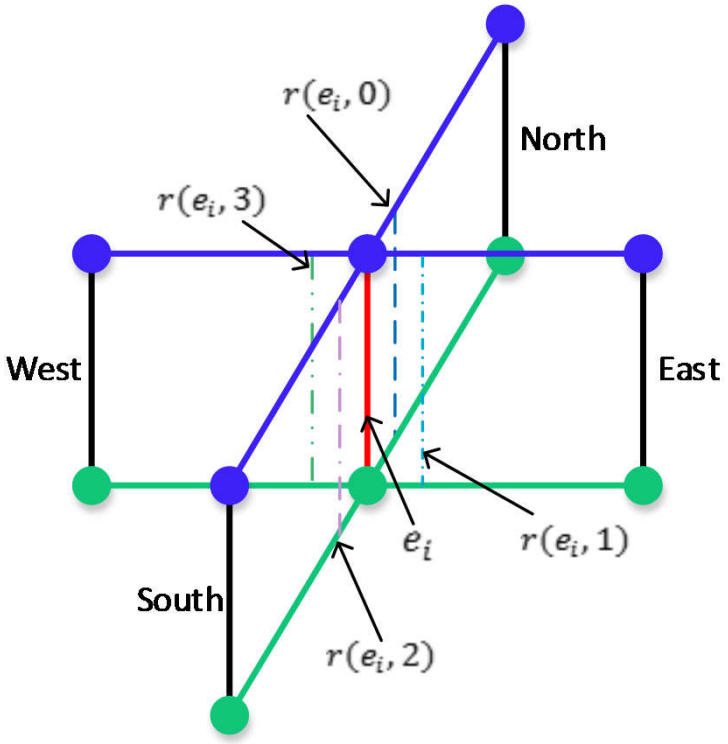
66 directed edges
18 vertices

Notions

notion	meaning
$E/V/N$	edge/vertex/net set (indexed by $i/j/k$)
E_{v_j}	adjacent edges set of vertex v_j
$E_{v_j,out}$	edges set whose start point is vertex v_j
$E_{v_j,in}$	edges set whose end point is vertex v_j
$E^{\eta}_{v_j}$	via edges set of vertex v_j
$dg(v_j, d)$	the vertex degree of redundant via on direction d
$u(e_i)$	binary variable, the capacity of edge e_i
$c(e_i)$	positive variable, the cost of edge e_i
$d(n_k, v_j)$	the flow command of vertex v_j with value of -1, 0, or 1 that commodity n_k demands
$f(n_k, e_i)$	binary variable, flow of commodity n_k by edge e_i

Notions

- Redundant via variable
  - the conventional variables are unable to build an effective model under the practical redundant via design rules.
  - Direction-based redundant via variable
    - Each via edge has four redundant via variables
    - $d=0\sim3$: represents direction



Problem Definition

- **RVI enhanced detailed routing**
- *Given global routing (GR) and track assignment (TA) results, as well as routing region, RVI enhanced detailed routing is to route all nets in given region simultaneously. The objective is to insert as many redundant vias as possible without introducing DRC violations.*

Outline

- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

Objective Function

- To minimize total routing cost

$$Cost = \sum_{k=1}^{K} \sum_{e \in E} (f(k,e) \cdot c(e)) + penalty$$

  - With different cost $c(e)$ assignment, we can achieve different routing goal.
  - Penalty terms
    - Spacing constraint penalty
    - Capacity constraint penalty
    - Redundant via insertion penalty

Connectivity Constraints

- Inspired by [1], the connectivity constraint is based on flow conservation theory
- Flow conservation

$$f_{out}(k, v_j) - fin(k, v_j) = d(k, v_j)$$

[1] E. Shragowitz and S. Keel. A global router based on a multicommodity flow model. the VLSI journal, 5(1):3–16,1987.

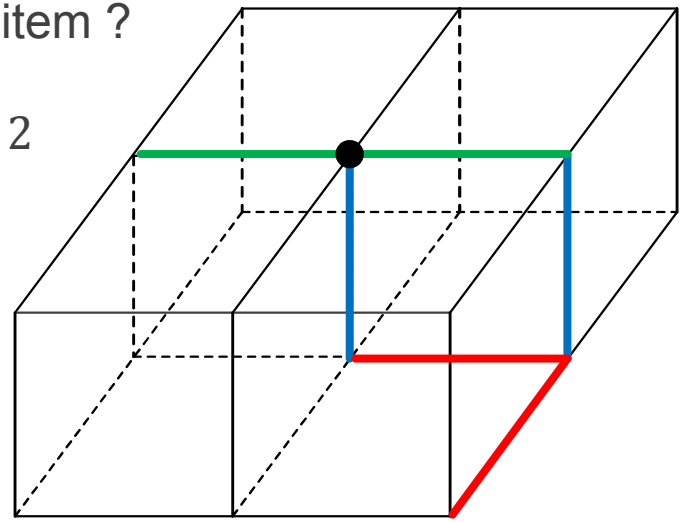
Capacity Constraints

- No shorts !
- Without redundant via variables

$$\sum_{k=1}^{K} \sum_{e_i \in Evj} f(k, e_i) \leq 2$$

- Add redundant via variables to left item ?

$$\sum_{k=1}^{K} \sum_{e_i \in Evj} f(k, e_i) + \sum_{d=0}^{3} dg(v_j, d) \leq 2$$



[15] X. Jia,, et. al, "MCFRoute: a detailed router based on multi-commodity flow method," in *ICCAD*, 2014, pp. 397–404.

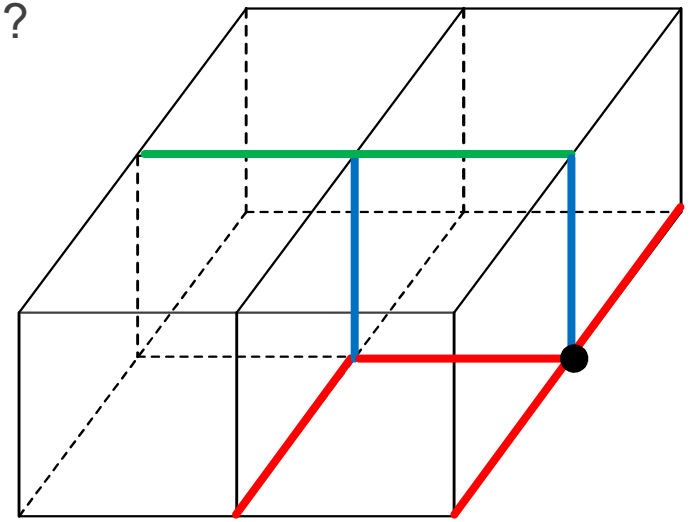
Capacity Constraints

- No shorts !
- Without redundant via variables

$$\sum_{k=1}^{K} \sum_{e_i \in Evj} f(k, e_i) \leq 2$$

- Add redundant via variables to left item ?
- Change right hand value ?

$$\sum_{k=1}^{K} \sum_{e_i \in Evj} f(k, e_i) + \sum_{d=0}^{3} dg(v_j, d) \leq 3$$

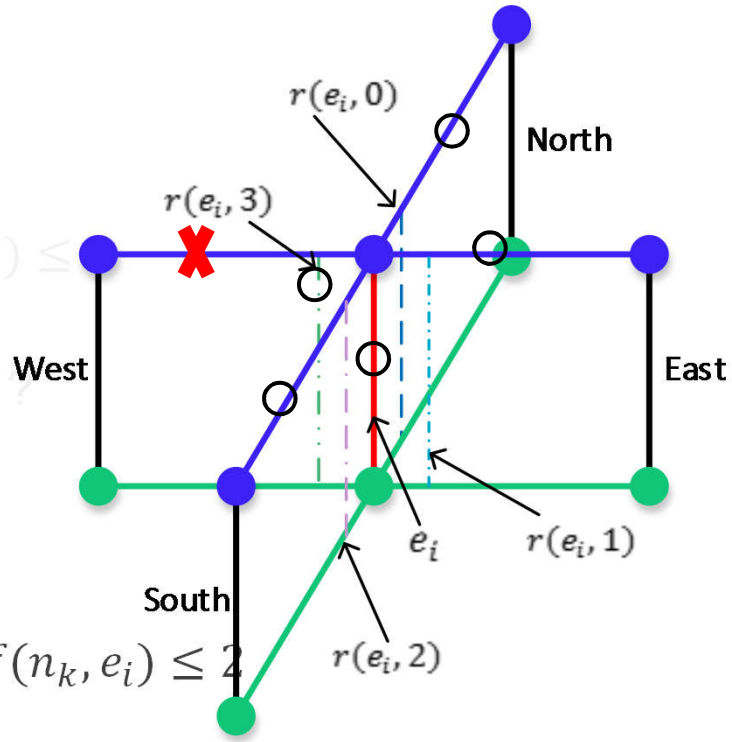


Capacity Constraints

- No shorts !
- Without redundant via variables

$$\sum_{k=1}^{K} \sum_{e_i \in Evj} f(k, e_i) \leq$$

- Add redundant via variables to left item ?
- Change right hand value ?
- Direction-based capacity constraints



$$dg(v_j, d) + \sum_{k=1}^{|N|} \sum_{e_i \in E_{v_j}^d} f(n_k, e_i) \leq 2$$

$$E_{v_j}^d = E_{v_j} \backslash \{(v_j, v_{j_d}), (v_{j_d}, v_j)\}$$

Redundant Via Insertion Constraints

- Ideal Solution
  - 100% redundant via insertion rate

$$\sum_{k=1}^{|N|} f(n_k, e_i) = \sum_{d=0}^{3} r(e_{i_d}, opp(d))$$

- Realistic Solution
  - If the insertion of a redundant via violates with around geometries, the redundant via should be given up to avoid DRC violations generation.
  - Penalty theory

$$\sum_{k=1}^{|N|} f(n_k, e_i) = \sum_{d=0}^{3} r\left(e_{i_d}, opp(d)\right) + \lambda$$

Spacing Constraints

- Preliminaries
  - Via enclosure
    - A *via enclosure* measures the distance from the cut array edge to the metal edge that encloses the cut array [1]
  - Vertices distance
    - For two vertices $v_{j1}(x_1, y_1, z)$ and $v_{j2}(x_2, y_2, z)$, the distance of them is defined as $\Gamma(v_{j1}, v_{j2}) = \max(|x_1 - x_2|, |y_1 - y_2|)$
  - Spacing constraints of work[2]

$$\sum_{k=1}^{|N|}\left(\sum_{e_i \in E^{\eta}_{v_{j1}}} f(n_k, e_i) + \sum_{e_i \in E'_{v_{j2}}} f(n_k, e_i)\right) \leq 2$$

    - The constraint indicates that if the via edges of v_{j1} is occupied by net n_k, all adjacent edges of vertex v_{j2} can not be used by other nets again except for net n_k.

[12]"LEF/DEF language reference,"http://www.ispd.cc/contests/14/web/doc/lefdefref.pdf.

[15] X. Jia,, et. al, "MCFRoute: a detailed router based on multi-commodity flow method," in *ICCAD*, 2014, pp. 397–404.

Spacing Constraints

- **Case1** Vertex v_j is occupied by a single via.
  - Constraint in work[2] still works

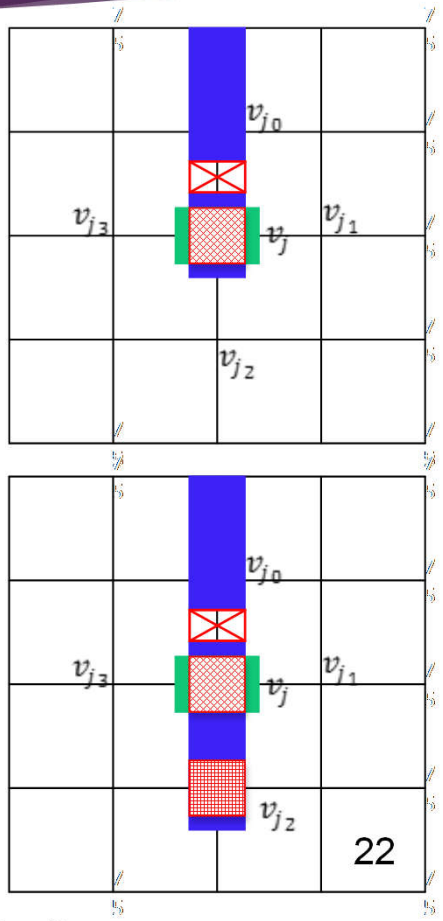
$$\sum_{k=1}^{|N|}\left(\sum_{e_i \in E_{v_j}^{\eta}} f(n_k, e_i) + \sum_{e_i \in E'_{v_{j0}}} f(n_k, e_i)\right) \leq 2$$

- **Case2** Vertex v_j is a redundant via for neighbor vertex on vertical direction

$$dg(v_j, 2) + \sum_{k=1}^{|N|} \sum_{e_i \in E'_{v_{j0}}} f(n_k, e_i) \leq 2$$

- Constraints of case1 and case2 could be merged together:

$$dg(v_j, 2) + \sum_{k=1}^{|N|} \sum_{e_i \in E_{v_j}^{\eta}} f(n_k, e_i) + \sum_{k=1}^{|N|} \sum_{e_i \in E'_{v_{j0}}} f(n_k, e_i) \leq 2$$



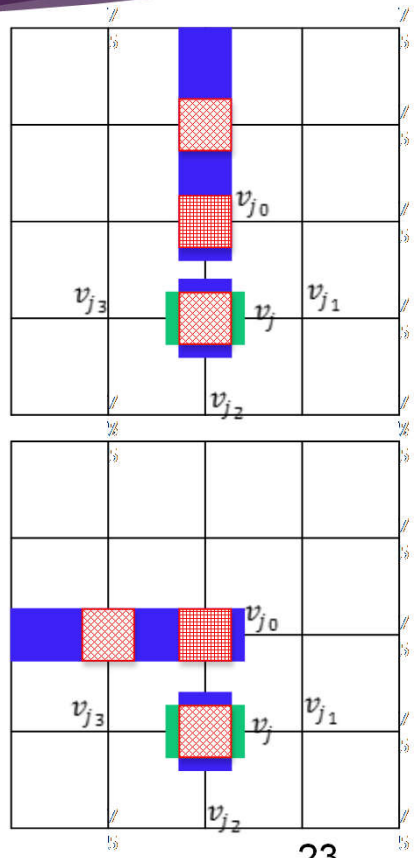
[12]"LEF/DEF language reference,"http://www.ispd.cc/contests/14/web/doc/lefdefref.pdf.

[15] X. Jia,, et. al, "MCFRoute: a detailed router based on multi-commodity flow method," in *ICCAD*, 2014, pp. 397–404.

Spacing Constraints

- **Case3** Vertex v_{j0} is a redundant via, but is not for vertex v_j.
  - Two conclusion:
    - $dg(v_{j0}, 2) = 0$ $\quad$ $dg(v_{j0}, 0) + dg(v_{j0}, 1) + dg(v_{j0}, 3) = 1$
  - if the usage of vertex v_j belongs to *case 1* or *case 2*, spacing violation is introduced.
  - New constraint:

$$dg(v_j, 2) + \sum_{k=1}^{|N|} \sum_{e_i \in E_{v_j}^{\eta}} f(n_k, e_i) + \sum_{d=0; d \neq 2}^{3} dg(v_{j0}, d) \leq 1$$



[15] X. Jia,, et. al, "MCFRoute: a detailed router based on multi-commodity flow method," in *ICCAD*, 2014, pp. 397–404.

Spacing Constraints

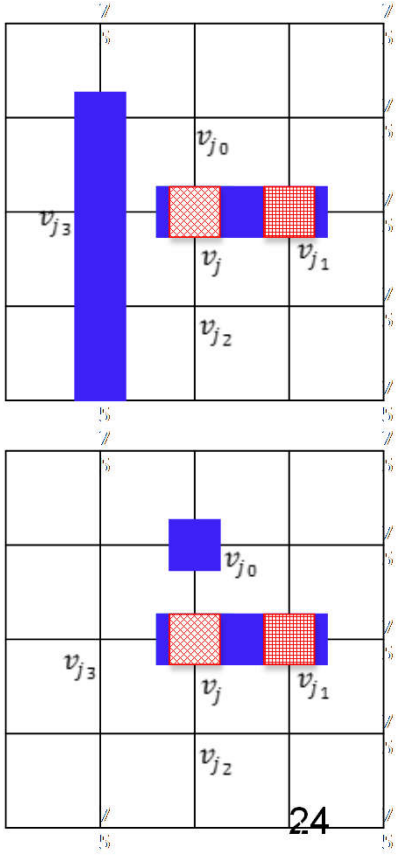
- **Case4** Vertex v_j and one of its neighbor vertex on horizontal direction construct a double via.
  - The usage of edges adjacent to vertex v_{j3} is restricted by via enclosure.

$$dg(v_j, 1) + dg(v_{j1}, 3) + \sum_{k=1}^{|N|} \sum_{e_i \in E'_{v_{j3}}} f(n_k, e_i) \leq 2$$

  - The usage of adjacent vertical vertices *i.e.* v_{j0} and v_{j2} is no longer limited by via edges on vertex v_j.

$$dg(v_j, 2) + \sum_{k=1}^{|N|} \sum_{e_i \in E^{\eta}_{v_j}} f(n_k, e_i) + \sum_{k=1}^{|N|} \sum_{e_i \in E'_{v_{j0}}} f(n_k, e_i) - dg(v_{j1}, 3)$$

$$- dg(v_{j3}, 1) \leq 2$$



RVI aware Pin Access Allocation Algorithm

- Pin access
  - Points which are covered by pin and can be served as the candidates for
    via creation are called *pin accesses*

- About 50% vias are generated on the Via12 which locates between Metal1 and Metal2.

RVI aware Pin Access Allocation Algorithm

Algorithm 1 RVI aware pin access allocation algorithm

Input: Pins and blockages;
Output: Via candidates;

```
1: C = searchCandidates();
2: for each candidate in C do
3:     SgVias = createSingleVias();
4:     for each via in SgVias do
5:         valid = checkDRCAndAssignCost();
6:         if valid = true then
7:             RdntVias = createRdntVias();
8:             for each redundant via in RdntVias do
9:                 checkDRCAndUpdateCost();
10:            end for
11:        end if
12:    end for
13: end for
```

Outline

- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

Environment and Benchmarks

- Environment
  - MCFRoute 2.0 is implemented with c++ on Linux server with 2.4 GHZ Intel Xeon CPU and 24GB memory.
- Benchmarks
  - 5 industrial benchmarks mapped to 45nm library
  - 8 academic benchmarks mapped to 65nm library
  - Compared with Cadence router, Encounter and MCFRoute
  - Work on and exchange data through OA database
- Routing Engine
  - Encounter
  - MCFRoute + RVI(post-layout)
  - MCFRoute 2.0

Comparison between MCFRoute with post-route RVI and MCFRoute 2.0

Benchmark		MCFRoute + RVI					MCFRoute 2.0				
Name	#Net	WL	#Via	#Rt	Rate	Time	WL	#Via	#Rt	Rate	Time
b20	9309	120	61006	33962	55.7%	160	121	58656	41466	70.7%	215
b21	9575	129	62763	33779	53.8%	213	129	60879	42774	70.3%	282
b22	10046	177	67962	42029	61.8%	174	179	65133	46094	70.8%	235
dma	12475	303	101648	51575	50.7%	301	304	96781	62310	64.4%	406
frisc	15869	292	116974	56966	48.7%	332	294	112463	71634	63.7%	429
dsp	24129	550	196311	92565	47.2%	735	550	187095	115116	61.5%	926
pci	30835	328	157879	96476	61.1%	345	329	158281	125445	79.3%	481
eth	43169	1006	331514	182887	55.2%	1025	1009	319116	211094	66.1%	1395
Ratio		0.99	1.04	0.82	0.79	0.75	1	1	1	1	1

#net: total net count
WL: total wirelength (*millimeter*)
#Via: total via count

#Rt: total redundant via count
Rate: redundant via insertion rate
Time: measured in *second*

Post-routing RVI: By Encounter

Comparison between Encounter and MCFRoute 2.0

Benchmark		Encounter					MCFRoute 2.0				
Name	#Net	WL	#Via	#Rt	Rate	Time	WL	#Via	#Rt	Rate	Time
b20	9309	126	60910	37475	61.5%	50	121	58656	41466	70.7%	215
b21	9575	133	62764	38409	61.2%	52	129	60879	42774	70.3%	282
b22	10046	186	67487	46500	68.9%	50	179	65133	46094	70.8%	235
dma	12475	314	102352	58679	57.3%	99	304	96781	62310	64.4%	406
frisc	15869	304	116943	68709	58.8%	112	294	112463	71634	63.7%	429
dsp	24129	572	197653	105660	53.5%	191	550	187095	115116	61.5%	926
pci	30835	339	158535	115389	72.8%	141	329	158281	125445	79.3%	481
eth	43169	1049	332394	206694	62.2%	380	1009	319116	211094	66.1%	1395
Ratio		1.04	1.04	0.95	0.91	0.25	1	1	1	1	1

#net: total net count
WL: total wirelength (*millimeter*)
#Via: total via count

#Rt: total redundant via count
Rate: redundant via insertion rate
Time: measured in *second*

Post-routing RVI: By Encounter

Comparison between Encounter and MCFRoute 2.0

benchmark		Encounter						MCFRoute 2.0					
Name	#Net	WL	#Via	#Rt	Rate	DRC	Time	WL	#Via	#Rt	Rate	DRC	Time
chip1	37584	684	400311	162436	40.6	115	1534	628	369980	221666	59.9	72	2392
chip2	53400	832	462701	250055	54.0	159	1478	808	401074	260785	65.0	33	2003
chip3	101817	1696	908599	514129	56.6	101	1849	1639	779529	506149	64.9	54	3660
chip4	92046	1550	828688	464708	56.8	81	1592	1504	703770	457310	65.0	46	3357
chip5	101729	1731	907323	510295	56.2	93	1865	1674	778938	505296	64.9	79	3732
Ratio		1.04	1.15	0.98	0.83	1.89	0.55	1	1	1	1	1	1

#net: total net count
WL: total wirelength (*millimeter*)
#Via: total via count

#Rt: total redundant via count
Rate: redundant via insertion rate
DRC: design rule violation count
Time: measured in *second*

Outline

- Background
- Contributions
- Problem Formulation
- RVI Enhanced MCF Model
- Experimental Results
- Summary

Summary

- A concurrent routing algorithm considering redundant via insertion
- A comprehensive analysis on the effect of via enclosure, which makes the proposed algorithm much practical.
- A redundant via insertion aware pin access allocation algorithm

Thank you