FAST ROUTE 4.0 – GLOBAL ROUTER WITH EffiCIENT VIA MINIMIZATION FORM ASP_DAC09

Yue Xu, Yanheng Zhang and Chris Chu

Outline

- **Introduction**
- Preliminary
- Via Reduction Technique
  - Via Aware Steiner Tree
  - 3-Bend Routing
  - Layer Assignment with Careful Ordering
- Experimental Result
- Conclusion

Introduction

- Via is a major source for circuit failure in VLSI design.
- Via has large process variation that impacts the timing of circuits.
- Via minimization has become one of the major targets in back-end design.

Previous Work

- [1] Kastner , pattern base routing scheme.
- [2] Madden , amplify the congenstion map with a new congestion cost function.
- [3] BoxRouter , ILP based approach , can handle multiple nets.
- [4] FastRoute , explore congestion-driven RSMT.
- [5] FastRoute 2.0 , introduce the monotonic routing and multi-source and multi-sink maze routing.

Traditional global routers ignore via count but focus on the oldest yet most important criteria to judge the performance of a global router, the ability to generate solutions without any wire congestion.

Outline

- Introduction
- **Preliminary**
- Via Reduction Technique
  - Via Aware Steiner Tree
  - 3-Bend Routing
  - Layer Assignment with Careful Ordering
- Experimental Result
- Conclusion

Grid Graph Model

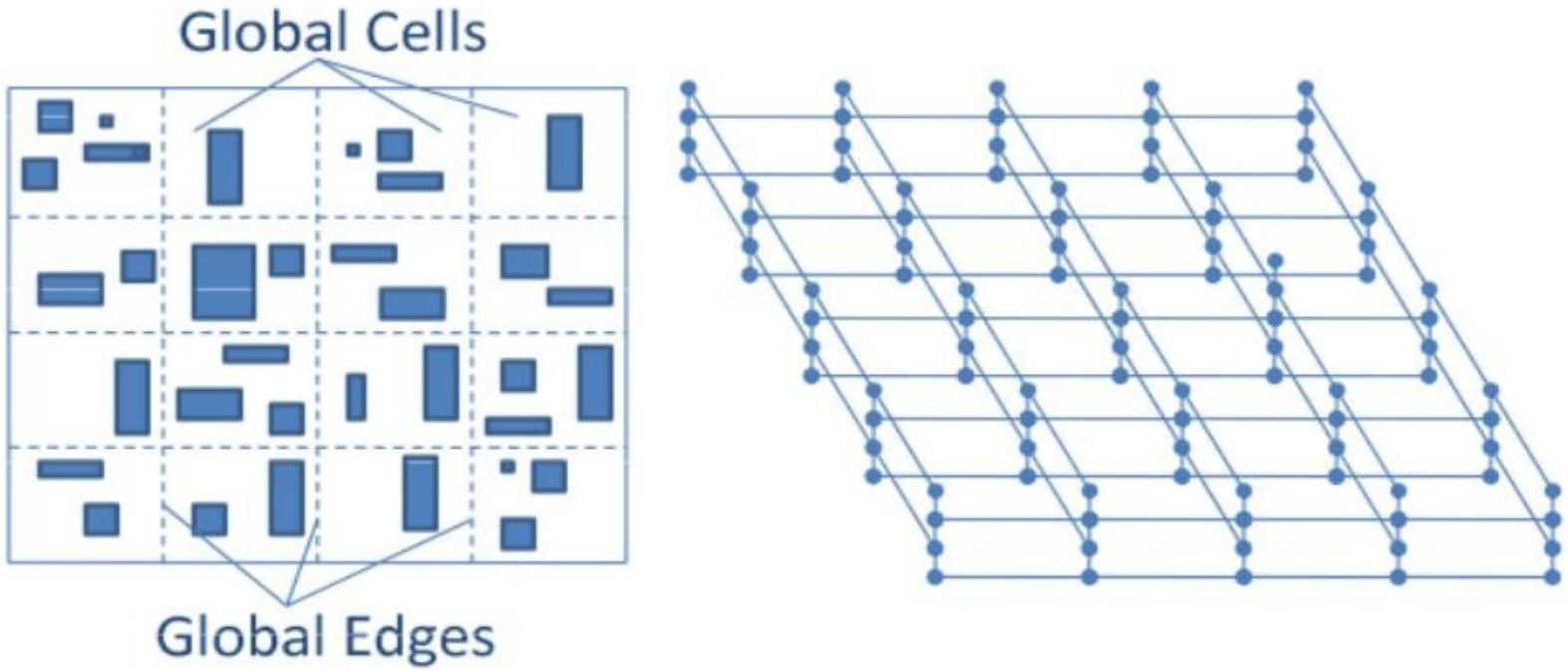


Fig. 1. Global Cells and Corresponding 3D Global Routing Grid Graph

Overview

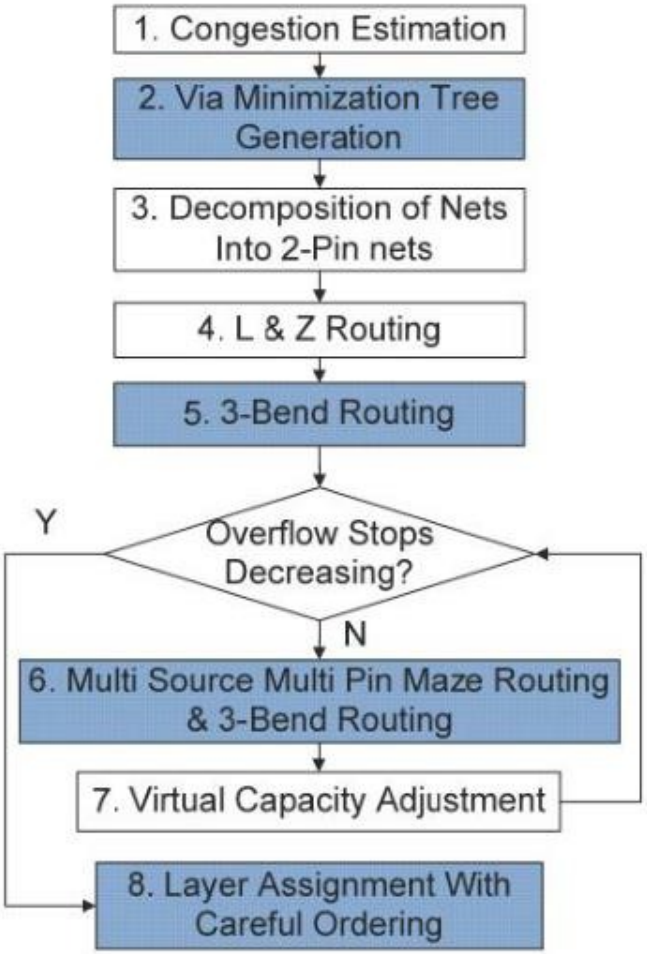


Fig. 2. FastRoute 4.0 Framework

Outline

- Introduction
- Preliminary
- **Via Reduction Technique**
  - Via Aware Steiner Tree
  - 3-Bend Routing
  - Layer Assignment with Careful Ordering
- Experimental Result
- Conclusion

Via Reduction Technique

- **Via aware Steiner tree generation:**

  produces tree topology that adjusts the number of vias based on congestion and layer information.

- **3-bend routing:**

  is an efficient routing algorithm with detouring option. It has congestion reduction capability like maze routing and effective control on via count like pattern routing.

Cont.

- **Layer assignment with careful net and edge ordering:**

  a sequential layer assignment algorithm that carefully orders nets and the edges in each net. It then uses efficient dynamic programming to assign layers to each edge.

Via Aware Steiner Tree

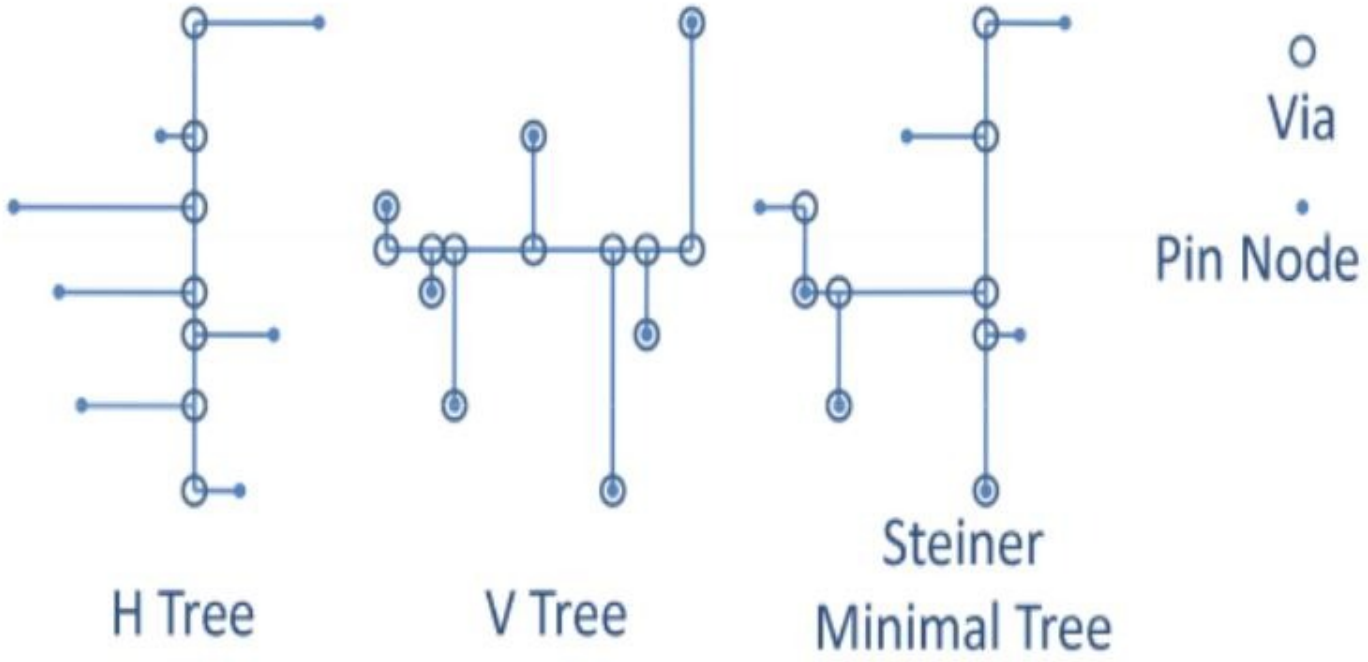


Fig. 3. Via Aware Steiner Tree

Congestion map

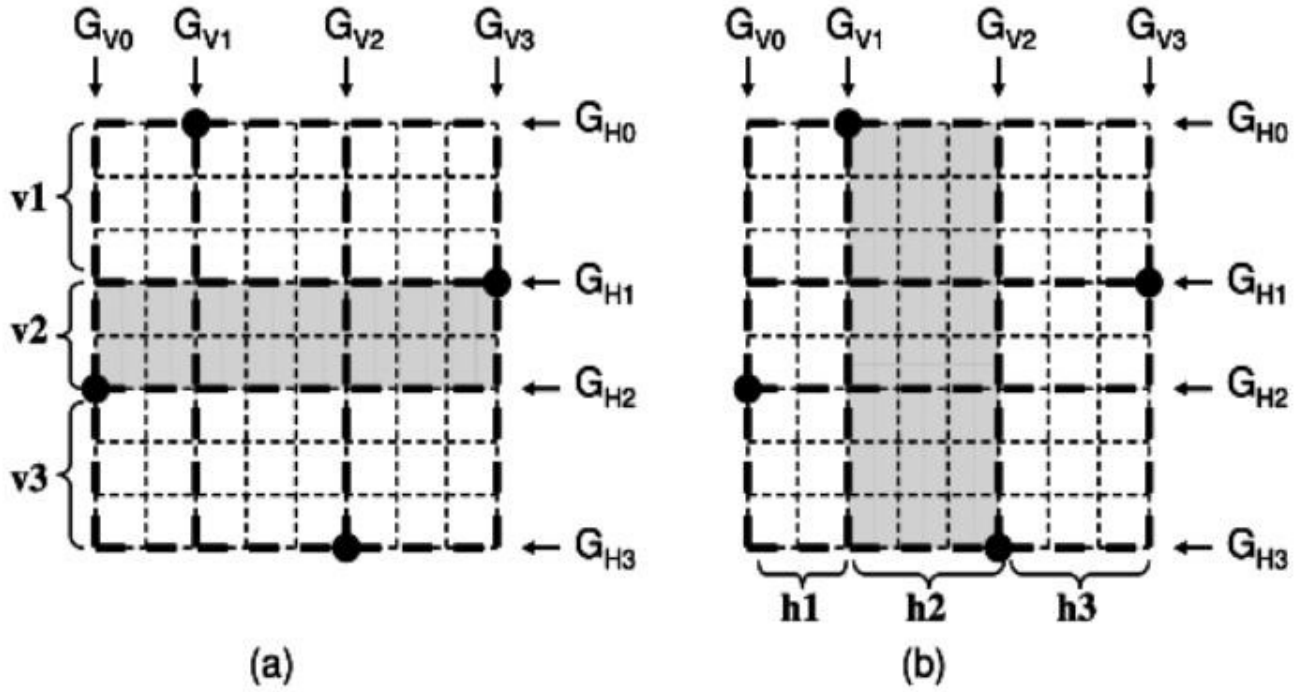


Figure 5: (a) The row region between G_{V1} and G_{V2}. (b) The column region between G_{H1} and G_{H2}.

3-Bend Routing

fewer vias , more congestions

detour to strengthen the via reductions , but run time …

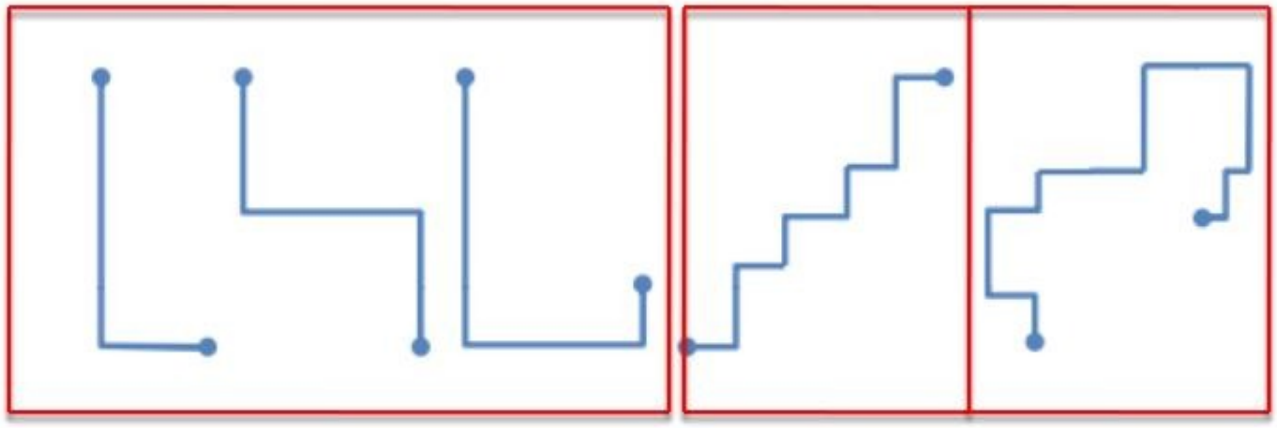


Fig. 4. L/Z/U, Monotonic and Maze Routing

fewer congestions , more vias

Cont.

- A 3-bend route is a 2-pin rectilinear connection that has at most three bends and possible detour.

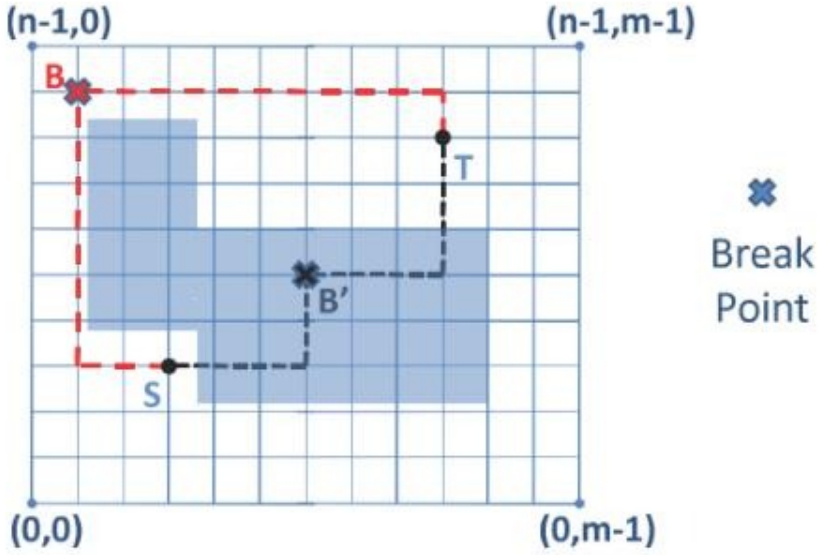


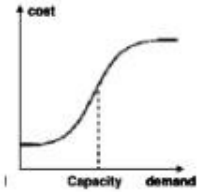
Fig. 5. 3-Bend Routing

Time complexity: O(n*m)

Algorithm 3-Bend Routing

1. $C_{best} = +\infty$
2. **for** $y = 0$ to $n - 1$
3. $\quad d_h(y, 0) = 0$
4. $\quad$ **for** $x = 1$ to $m - 1$
5. $\quad\quad d_h(y, x) = d_h(y, x - 1) + cost_h(y, x - 1)$
6. **for** $x = 0$ to $m - 1$
7. $\quad d_h(0, x) = 0$
8. $\quad$ **for** $y = 1$ to $n - 1$
9. $\quad\quad d_v(y, x) = d_v(y - 1, x) + cost_v(y - 1, x)$
10. **for** $y = 0$ to $n - 1$
11. $\quad$ **for** $x = 0$ to $m - 1$
12. $\quad\quad B = (x, y)$
13. $\quad\quad d_{L1}(B) = |d_h(S) - d_h(y_s, x)| + |d_v(y_s, x) - d_v(B)|$
14. $\quad\quad d_{L2}(B) = |d_h(S) - d_h(y, x_s)| + |d_v(y, x_s) - d_v(B)|$
15. $\quad\quad d_{L3}(B) = |d_h(T) - d_h(y_t, x)| + |d_v(y_t, x) - d_v(B)|$
16. $\quad\quad d_{L4}(B) = |d_h(T) - d_h(y, x_t)| + |d_v(y, x_t) - d_v(B)|$
17. $\quad\quad$ Compute the cost of four possible 3-bend paths from 4 paths above plus via cost and compare it to C_{best}. If better, update best 3-bend path.

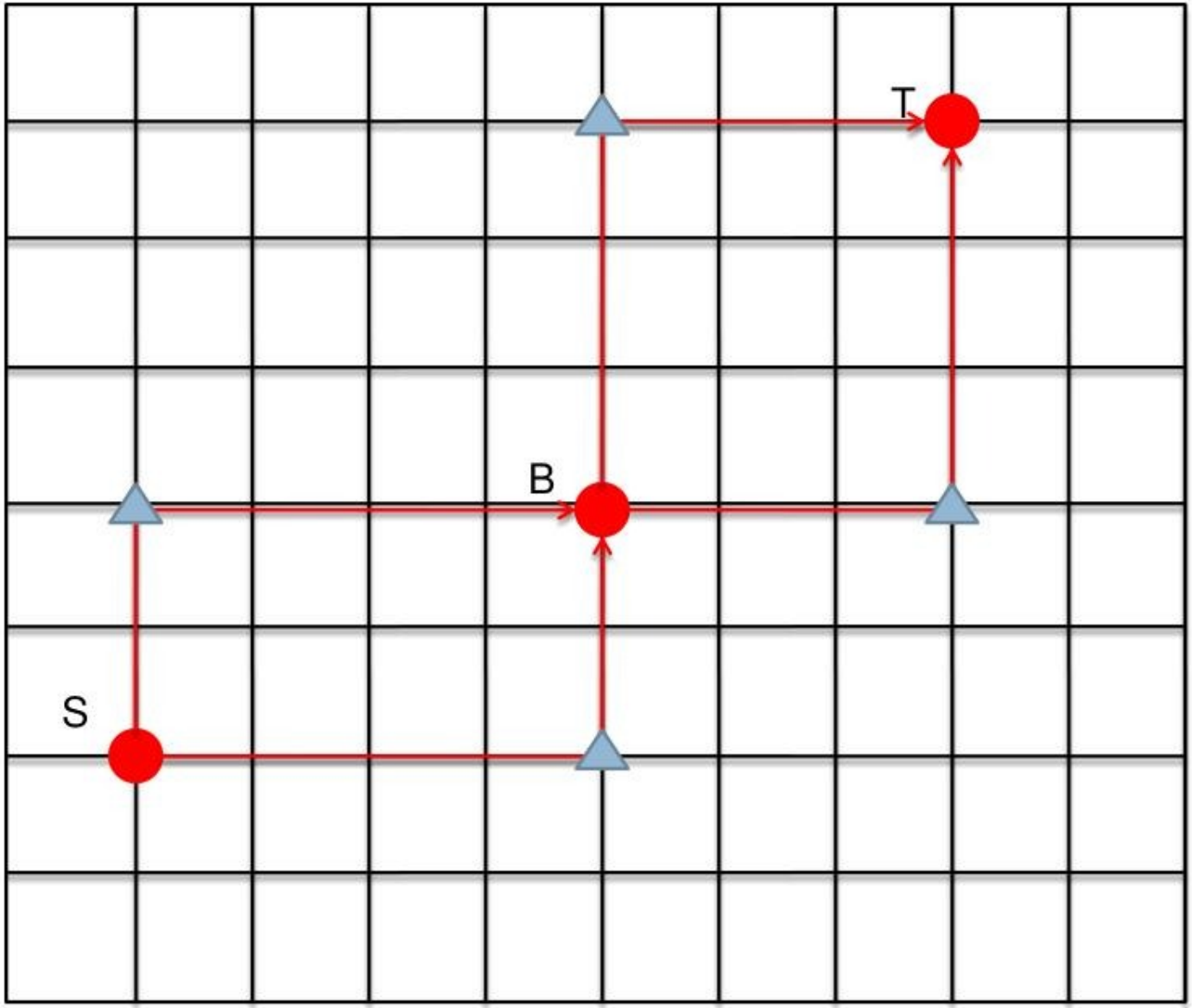
$$cost = 1 + \frac{h}{1 + e^{-k(demand - capacity)}}$$



Time complexity : O(m*n)

$$\frac{\sum cap_{(h)}}{\sum usg_{(h)}} / \frac{\sum cap_{(v)}}{\sum usg_{(v)}}$$

dh(y, x) and dv(y, x) denote the costs for a path going from the point (y, x) horizontally to the left boundary and vertically to the bottom boundary respectively.


T
B
S

Layer Assignment with Careful Ordering

- There are generally two methods to generate solution for 3D global routing benchmarks:
- 1. running routing techniques and layer assignment concurrently
- 2. first projects the 3D benchmarks from aerial view, finds a solution for the 2D problem and expands the solution to multiple layers

Observations

- Smaller nets connecting nearby global cells are considered relatively local and should use lower metal layers.
- longer nets assigned to upper layers will encounter less hopping between layers
- higher number of pins tend to cause more vias

Order net by increasing order of $\sum wl/\#Pins$

Order edges in each net in increasing order of their distance to the pin nodes

Layer Assignment for 2-Pin Net

1. Initial the cost for all the via nodes to $+\infty$
2. For every source, $C(j, 0) = 0$
3. Update the cost for other via nodes on the first column
4. **for** $x = 1$ to $n - 1$
5. **for** $y = 0$ to $L - 1$
6. **if** $cap(j, x - 1) > usg(j, x - 1)$
7. $C(j, x) = C(j, x - 1)$
8. Update the cost from vertical neighbors.
9. Find the least cost for any sink node and trace back using $C(j, x)$

Cost represents the least number of vias on the paths from the node to any source nodes.

Fig. 8. Layer Assignment Algorithm for 2-Pin Net

Time complexity: O(n*L^2)

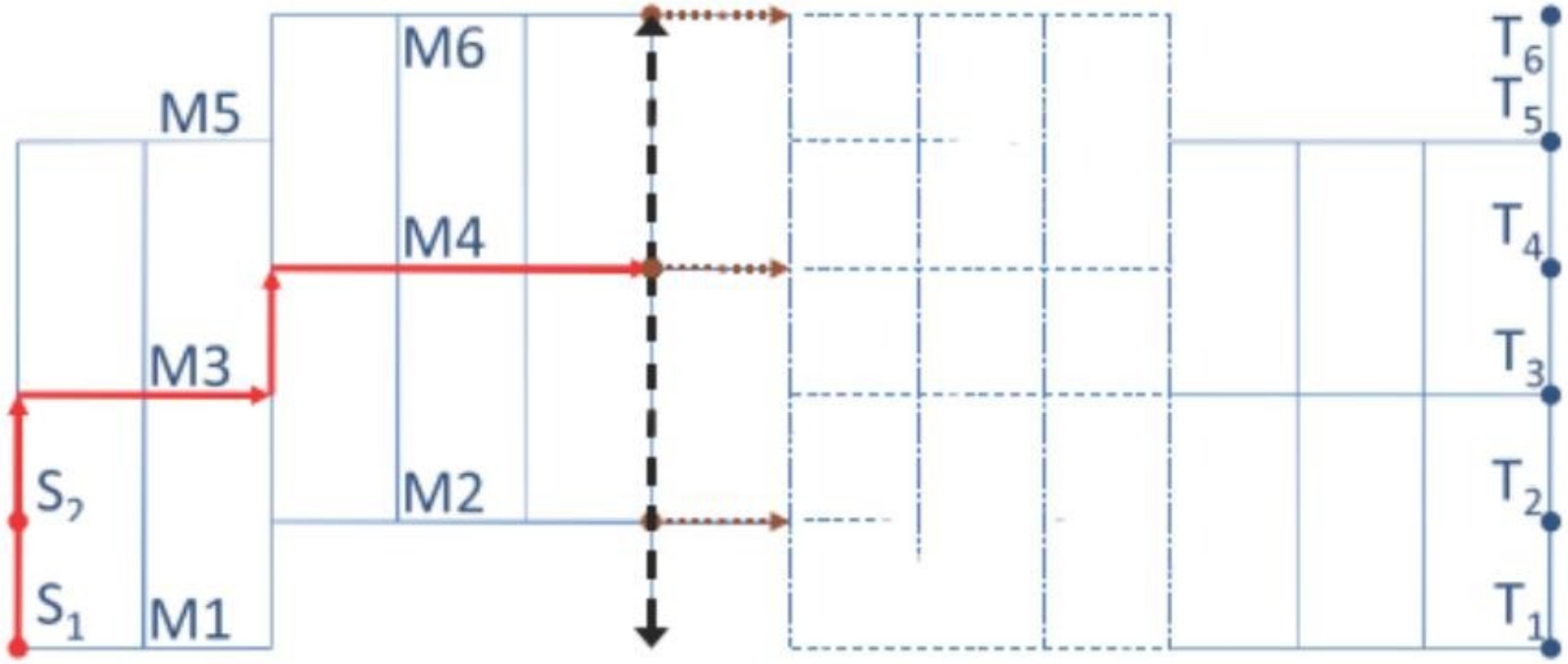


Fig. 7. Dynamic Programming Layer Assignment

Outline

- Introduction
- Preliminary
- Via Reduction Technique
  - Via Aware Steiner Tree
  - 3-Bend Routing
  - Layer Assignment with Careful Ordering
- **Experimental Result**
- Conclusion

Experimental Result

3.6% less number of vias
0.5% less wirelength
48% runtime reduction.

TABLE I

FastRoute 4.0 performance comparison for the three techniques on ISPD08 global routing contest benchmarks

name	FastRoute 4.0				Without the Techs			
	ovfl	#via	seg wl	cpu(s)	ovfl	#via	seg wl	cpu(s)
bigblue1	0	1990K	3795K	423	0	2645K	3866K	773
bigblue2	0	4455K	5104K	913	34	4628K	5138K	1797
bigblue3	0	5179K	7891K	278	0	5521K	7940K	342
bigblue4	152	11340K	12825K	674	150	12767K	12841K	3711
newblue4	144	4892K	8502K	1135	178	5334K	8521K	2459
newblue5	0	8659K	15013K	607	0	10223K	15184K	1419
newblue6	0	7701K	10561K	574	0	9266K	10611K	1357
newblue7	72	16949K	18742K	11060	158	19238K	18789K	18084

Cont.

TABLE II

COMPARISON BETWEEN FASTROUTE AND PUBLISHED GLOBAL ROUTERS ON 3D BENCHMARKS OF ISPD07 GLOBAL ROUTING CONTEST

	FastRoute 4.0			NTHU-R [10]			Archer [9]		FGR [12]		BoxRouter 2.0 [8]		MaizeRouter [11]	
name	ovfl	wlen	cpu(s)	ovfl	wlen	cpu(s)	ovfl	wlen	ovfl	wlen	ovfl	wlen	ovfl	wlen
adaptec1	0	9100K	190	0	9056K	5613	0	11380K	0	8845K	0	9204K	0	10000K
adaptec2	0	9157K	45	0	9217K	1010	0	11256K	0	8989K	0	9428K	0	9800K
adaptec3	0	20461K	300	0	20504K	3893	0	24408K	0	19966K	0	20741K	0	21400K
adaptec4	0	18695K	40	0	18843K	604	0	22157K	0	17936K	0	18642K	0	19400K
adaptec5	0	27064K	568	0	26503K	16104	0	33409K	0	25998K	0	27041K	0	30500K
newblue1	0	9170K	283	352	9091K	2280	494	11608K	526	9426K	400	9294K	1348	10200K
newblue2	0	13563K	22	0	13601K	257	0	16650K	0	12940K	0	13464K	0	14000K
newblue3	31634	18211K	1280	31800	16840K	21465	31928	19877K	39908	17371K	38958	17244K	32588	18400K

TABLE III

FASTROUTE 4.0 RESULTS ON 3D VERSION OF ISPD08 GLOBAL ROUTING CONTEST BENCHMARKS

	FastRoute 4.0					NTHU-R [7]			NTUgr [7]			BoxRouter2.0 [7]			FGR [7]		
name	ovfl	seg wlen	#via	twl	cpu(s)	ovfl	twl	cpu(s)	ovfl	twl	cpu(s)	ovfl	twl	cpu(s)	ovfl	twl	cpu(s)
bigblue1	0	3795K	1990K	5789K	423	0	5631K	586	0	5784K	839	0	5698K	1147	0	5733K	4194
bigblue2	0	5104K	4455K	9559K	913	0	9059K	594	0	9718K	15862	0	9042K	2346	0	9143K	14287
bigblue3	0	7891K	5179K	13070K	278	0	13075K	259	0	13573K	296	0	13133K	380	0	13201K	5256
bigblue4	152	12825K	11340K	24165K	674	182	23076K	7533	188	24282K	24785	472	23156K	52644	414	23163K	35513
newblue4	144	8502K	4892K	13394K	1135	152	12990K	4023	142	14378K	67087	200	12947K	78225	262	12959K	35220
newblue5	0	15013K	8659K	23672K	607	0	23166K	854	0	24578K	1679	0	23294K	1700	0	23296K	9963
newblue6	0	10561K	7701K	18262K	574	0	17696K	818	0	18556K	935	0	17975K	1785	0	18030K	6194
newblue7	62	18742K	16949K	35691K	11060	68	35357K	8433	310	37222K	86732	208	35859K	84743	1458	35023K	86068

Outline

- Introduction
- Preliminary
- Via Reduction Technique
  - Via Aware Steiner Tree
  - 3-Bend Routing
  - Layer Assignment with Careful Ordering
- Experimental Result
- **Conclusion**

Conclusion & Future Work

- This paper develop a new global routing tool that focuses on reducing the number of vias.
- It reduce 13.6% of via count of FastRoute 3.0 with enhancement in saving runtime.
- Their future work will focus on how to control maze routing so that it can make more effective balance between reducing congestion and keeping wirelength well.