Delay-Driven Layer Assignment in Global Routing under Multi-tier Interconnect Structure

ISPD-2013

Jianchang Ao*, Sheqin Dong*
Song Chen†, Satoshi Goto‡

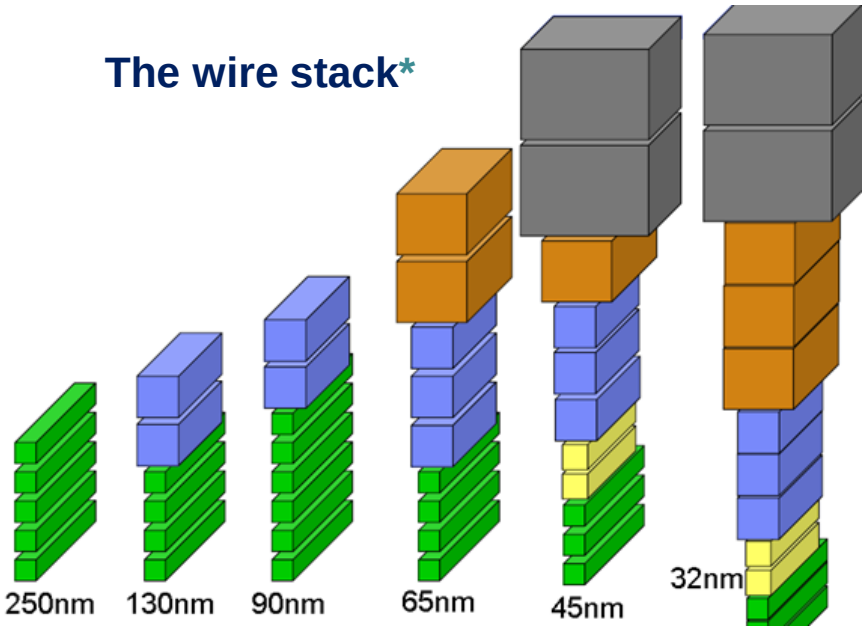
***Dept. of Computer Sci. & Tech., Tsinghua U**
†China U of Sci. and Tech., ‡Waseda U

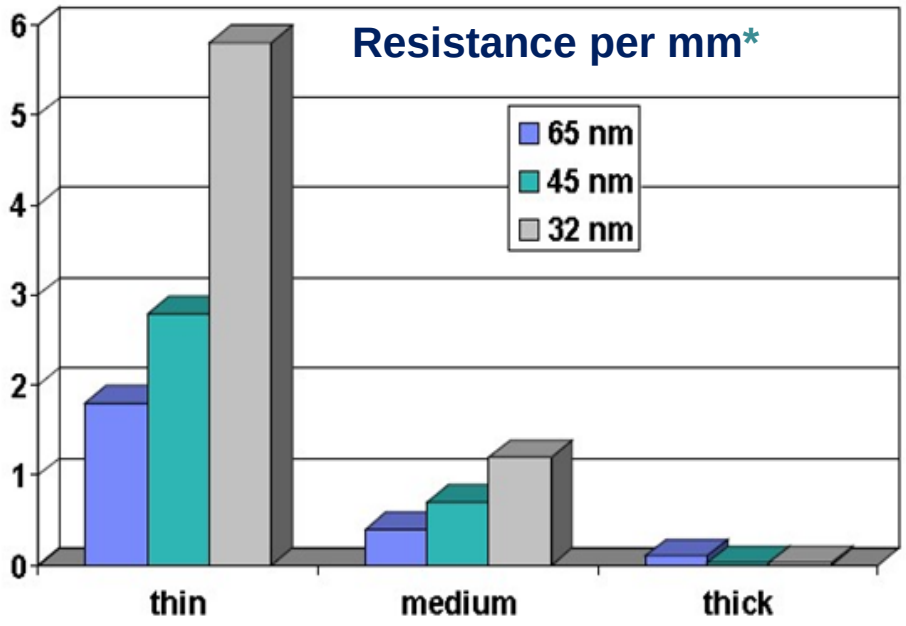
Outline

- **Introduction**
  - **Motivation**
  - **Previous work**
  - **This work**
- Problem Formulation
- Proposed Algorithm
- Experimental Results
- Conclusion

Trends of Routing Technology

- Interconnect delay determines system performance [ITRS08].
- More and more metal layers are available for routing.
- The gap of conductivity is expanding fast for metals with different sizes.

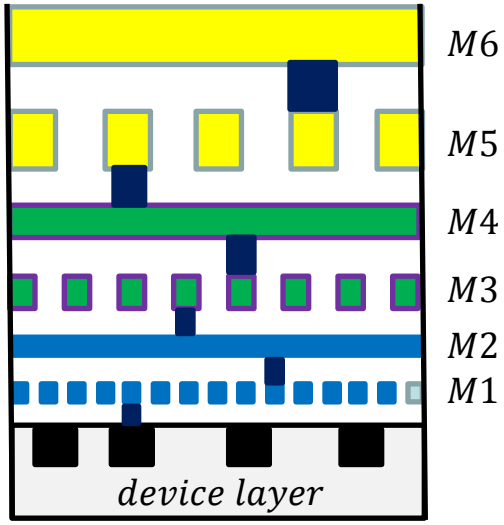




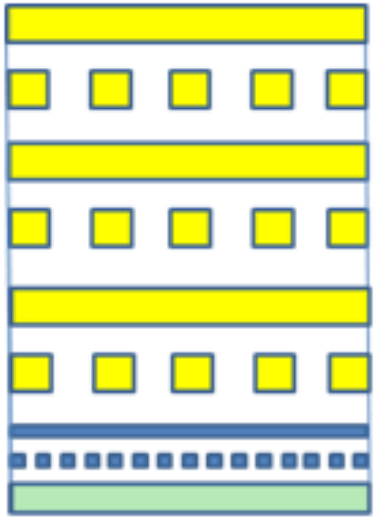
*[Alpert+ISPD10] "What makes a design difficult to route"

Multi-tier Interconnect Structure

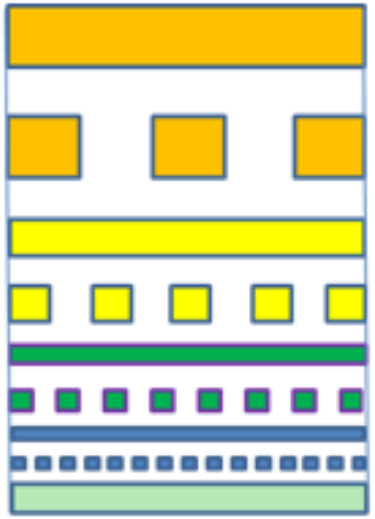
- Multi-layer routing system usually adopts multiple interconnect configuration with diverse specifications of wire sizes for metal layers.
- Fatter / thicker wires on higher metals are less resistive, which induces smaller Interconnect Delays.



3-tier 6-layer



2-tier 8-layer

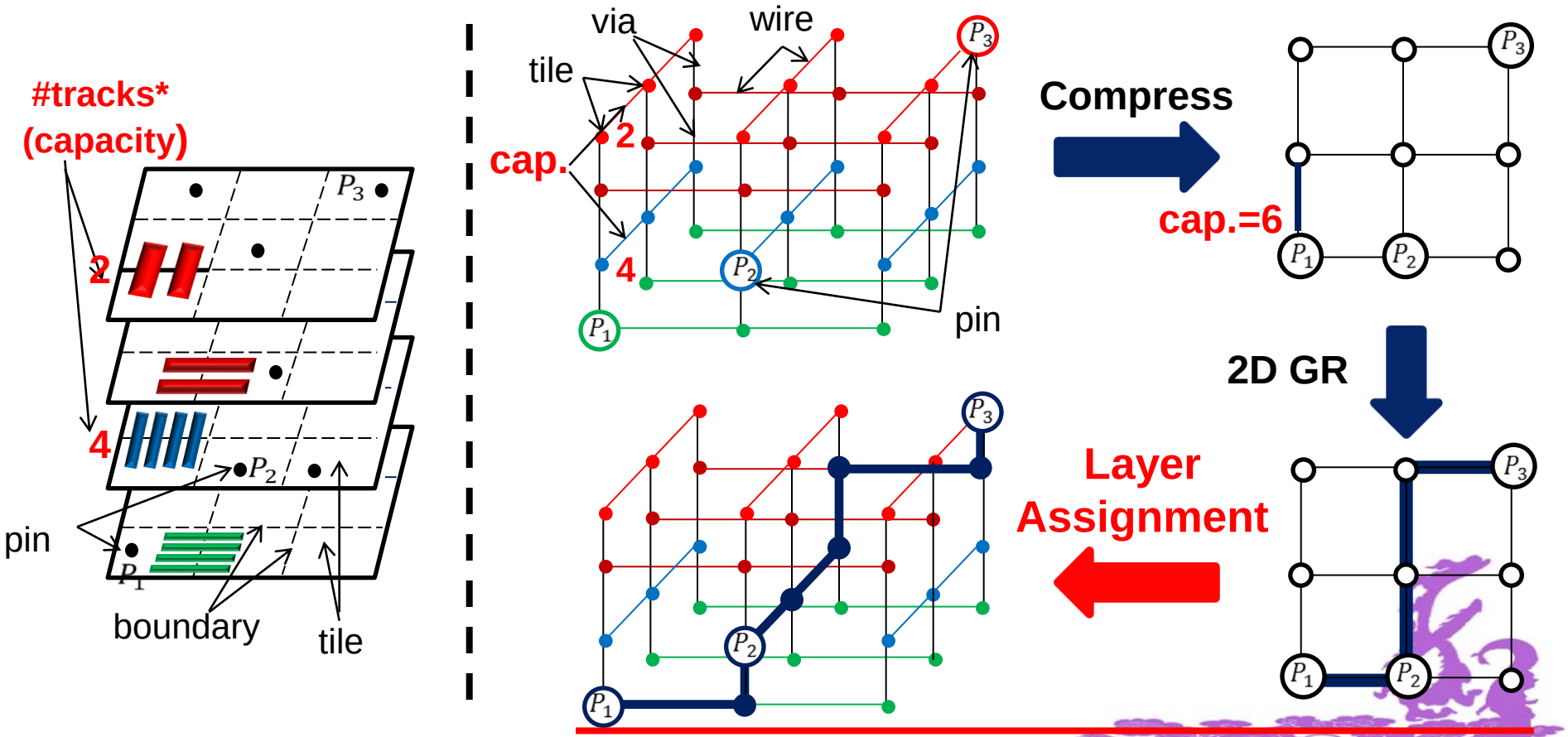


4-tier 8-layer

*Layer: metal routing layers used, Tier: number of metal sizes used

Layer Assignment in Global Routing

- Layer Assignment (LA) is a major step of multi-layer (3D) global routing (GR).



*Tiles on each layer may have Different track count due to Different wire sizes / pitches.

Previous Work

- Via (antenna, crosstalk, etc.) optimization in layer assignment of 3D global routing
  - [RoyICCAD07], [LeeTCAD08], [LiuASP-DAC11], [LiuISPD12] etc.
  - IGNORE the delay optimization due to layer dependent characteristics
  - Maybe because ISPD07/08 routing contests do NOT specify different wire sizes / pitches on metal layers
- Timing constrained minimum cost layer assignment or buffer insertion
  - [Hu+ICCAD08], [Li+ISPD08], etc.
  - Regard multi-tier interconnect structure, but
  - Deal primary with a single tree, NOT tree sets
  - Assign wires to thick metals or insert buffers SUCH THAT timing constraint of a net is satisfied and the usage of thick metals or buffers is minimized, while GR LA assign nets to metals SUCH THAT wire congestion constraints of 3D global routing are satisfied and via count (or delay, etc.) of all nets is minimized

Previous Work (cont.)

- Global routers honoring layer directives
  - [Chang+ICCAD10], [Lee+ISPD11], etc.
  - Specify candidate routing layers (higher / thicker metals) for the appointed timing-critical nets
  - NO actual calculation of delays
- Classical performance driven layer assignment
  - [Chang+TCAD99], [Saxena+TCAD01],etc.
  - Handle delay optimization in the POST-layout stage, NOT global routing stage
  - Handle the strict constraints of design rules on the layout, NOT the wire congestion constraints of 3D global routing
- Timing optimization for coupling capacitance in layer assignment
  - [Wu+ISPD05] etc.
  - NOT consider multi-tier interconnect structure

This Work

- Study the DELAY-driven layer assignment under MULTI-TIER interconnect structure, which arises from 3D Global routing.

- Delay-driven Layer Assignment (DLA) algorithm
  - Single-net Delay-driven Layer Assignment (SDLA) by DP : minimize net delay, via count and wire congestion
  - 2-stage algorithm framework based on SDLA : minimize total delay, maximum delay and via count simultaneously

- Significantly reduce the total delay and maximum delay while keeping roughly the same via count, compared to the state-of-the-art via count minimization layer assignment.

Outline

- Introduction
- **Problem Formulation**
  - **Problem formulation**
  - **Delay model**
- Proposed Algorithm
- Experimental Results
- Conclusion

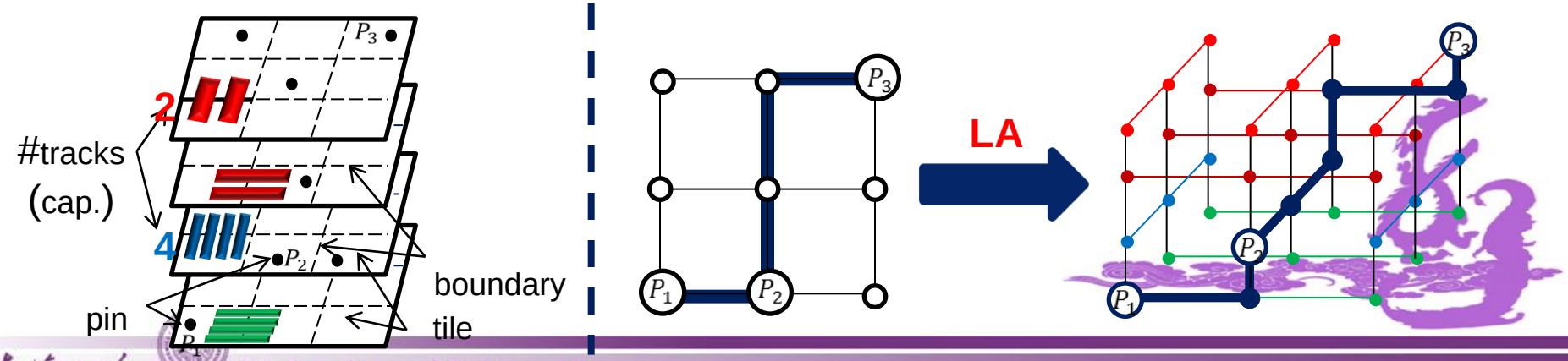
Layer Assignment in Global Routing

- Layer Assignment for Via Count Minimization
  - Minimize: Vias
  - Subject: Wire congestion constraints
    - The total overflow does not increase after layer assignment
    - Overflows are evenly distributed to each layer
- Delay-Driven Layer Assignment under Multi-tier Interconnect Structure
  - Minimize: delays and via count

    $min: \sum_{each\ net\ i} (\lambda \cdot delay_i + \#via_i)$

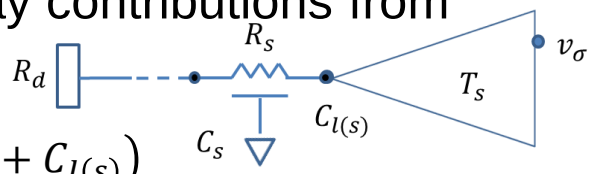
  - Subject: Wire congestion constraints

λ: 1) specify the relative importance of net delay and via count; 2) is specified for selected nets to emphasize their critical role



Delay Model

- Delay model
  - Elmore distributed RC delay model
  - A net tree has *one source* and *multiple sinks*, with *resistance of the driver driving the source* and *load capacitance* at each *sink*.
  - For an arbitrary net tree, each *wire segment* or *via segment* is viewed as an *individual RC conductor segment*.
- Delay calculation
  - Signal transmission line is seen as a series circuit composed by series of these RC conductor segments
  - The delay at any sink v_σ is the sum of delay contributions from each of its ancestors



$$delay(v_\sigma) = \sum_{s \in ans(v_\sigma)} delay(s) = \sum_{s \in ans(v_\sigma)} R_s \cdot (C_s/2 + C_{l(s)})$$

  - Elmore delays are incorporated at multiple sinks by attaching priority a_σ to $delay(v_\sigma)$ at sink v_σ. Assume $\sum_{\sigma=1}^{m} a_\sigma = 1$, m is the number of sinks.

$$delay(T) = \sum_{\sigma=1}^{m} [a_\sigma \cdot delay(v_\sigma)] = \sum_{s \in T} [wt_s \cdot R_s \cdot (C_s/2 + C_{l(s)})]$$

wt_s: delay weight of segment s

Outline

- Introduction
- Problem Formulation
- **Proposed Algorithm**
  - **SDLA: Single-net Delay-driven Layer Assignment**
  - **DLA: Delay-driven Layer Assignment**
- Experimental Results
- Conclusion

Overview of *S*DLA

- Minimize: Total Cost of delay, via count and wire congestion of net T

$$min: cost(T) = \lambda \cdot delay(T) + \#via(T) + \sum_{e \in T} congestion\_cost_e *$$

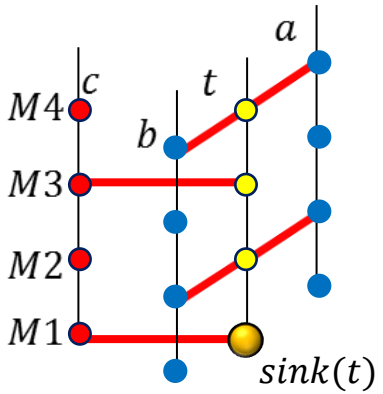
- Base on dynamic programming

*[McMurchie+FPGA95]
[Liu+ASP-DAC11]

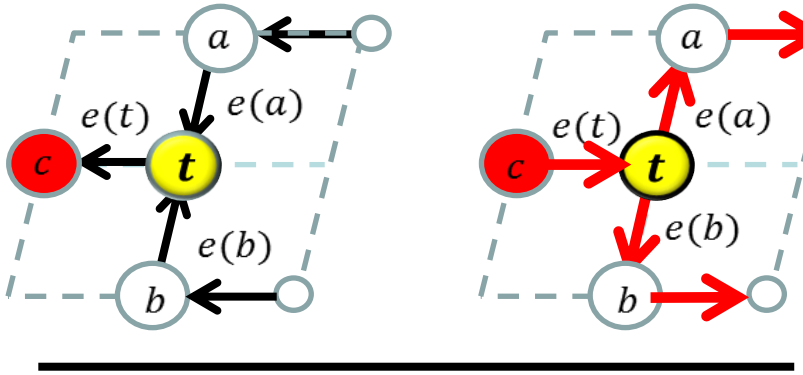
- Treat the tree source as root, processes each tree edge from sinks to source.
- Partition stage by tree edges, assign one edge at a time, and place vias after the assignment of edges.
- For each stage, record the Minimum Total Costs (TC) and the corresponding downstream Load Capacitance (LC), and propagate the results to the next stage.
  - LC is used for delay calculation of a segment for next stage
- Finally, after the root has been handled, the layer assignment with minimum total cost is the required solution.

Single-net Layer Assignment

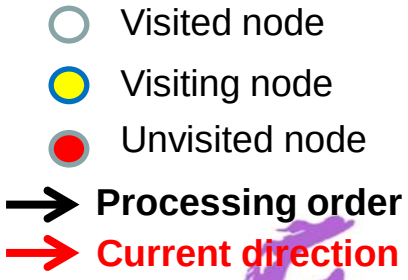
- Let $par(t)$ be the parent of vertex t of tree T, $ch(t)$ be the set of children of t, $e(t)$ be the edge $(t, par(t))$.
- Let $TC(t,r)$ and $LC(t,r)$ be the minimum Total Cost and the corresponding Load Capacitance among all possible layer assignment for the sub-tree rooted at t, with edge $e(t)$ assigned to layer r.
- $TC(t,r)$ and $LC(t,r)$ can be computed by considering all possible combinations of $TC(t_j, r_j)$'s and $LC(t_j, r_j)$'s for all $t_j \in ch(t)$.



4-layer routing graph



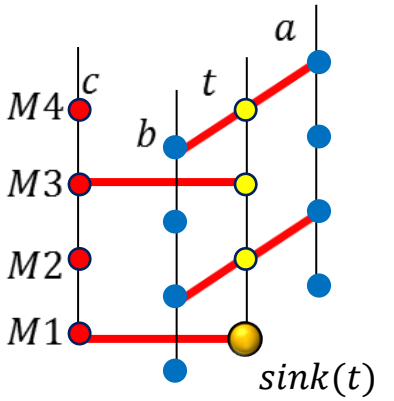
A part of a 2D routed net



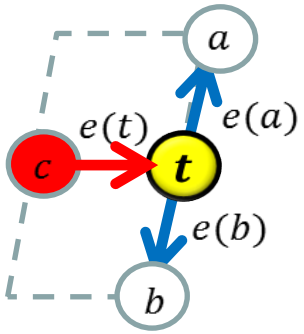
Processing order: from sinks (leafs) to source(root)

Single-net Layer Assignment

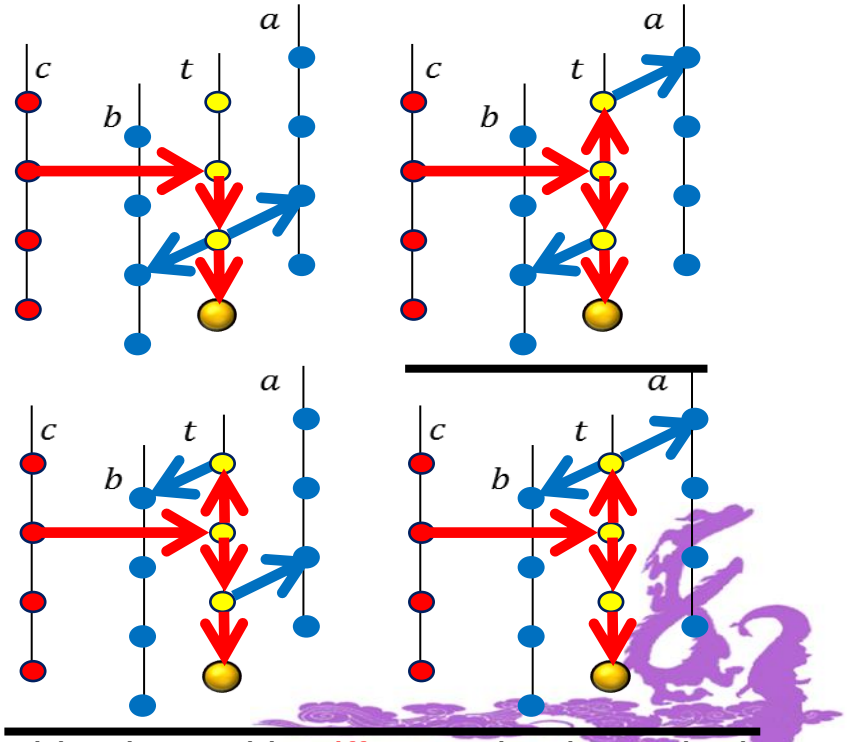
- Assume $TC(a, r_a)$, $LC(a, r_a)$, $TC(b, r_b)$, and $LC(b, r_b)$ for all combinations of r_a and r_b have been computed, where r_a and r_b can be layer $M2$ or $M4$. Now compute $TC(t, M3)$ and $LC(t, M3)$.
- For each combination, place vias to connect the 3 related 3D edges and the 3D pins projected to 2D pin t, then compute the Cost Increase.



4-layer routing graph



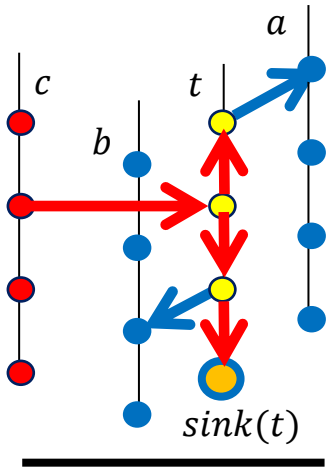
Part of a 2D routed net



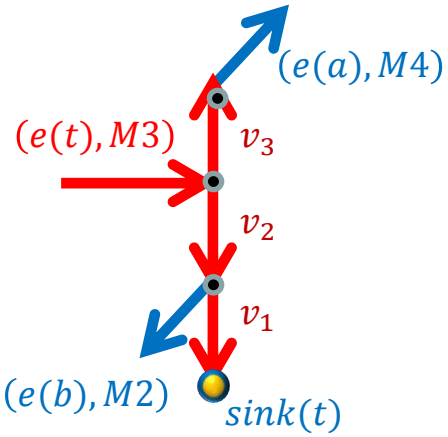
4 combinations with Different circuit topologies

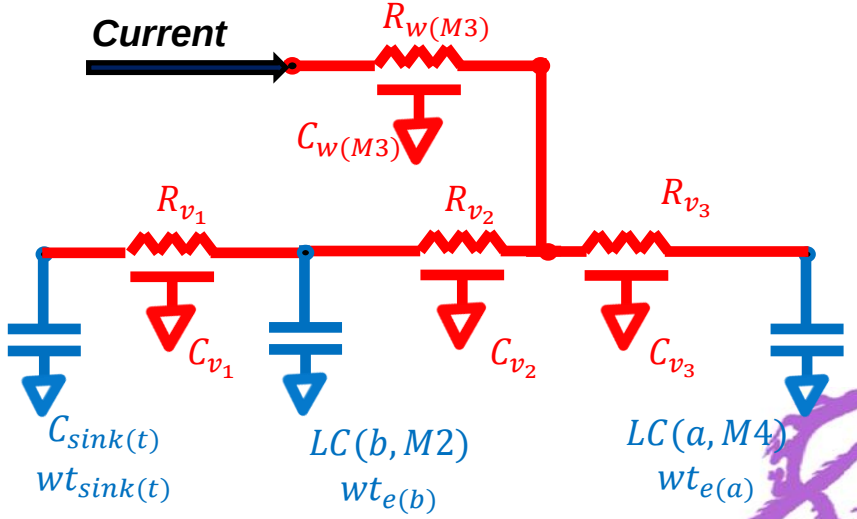
Calculation of Cost Increase

- Let $iND(t)$, $iLC(t)$, $iVC(t)$, and $iTC(t)$ denote the respective Increase of Net Delay, Load Capacitance, Via Count, and Total Cost due to vias v_1, v_2, v_3, and wire $w(M3)$.
- Let $congestion\_cost_{w(M3)}$ *be* the *congestion cost* of edge $w(M3)$.
- Load capacitance and total cost for This combination are $LC(a,M4) + LC(b,M2) + iLC(t)$, $TC(a,M4) + TC(b,M2) + iTC(t)$.



case 2





$$iVC = 3$$

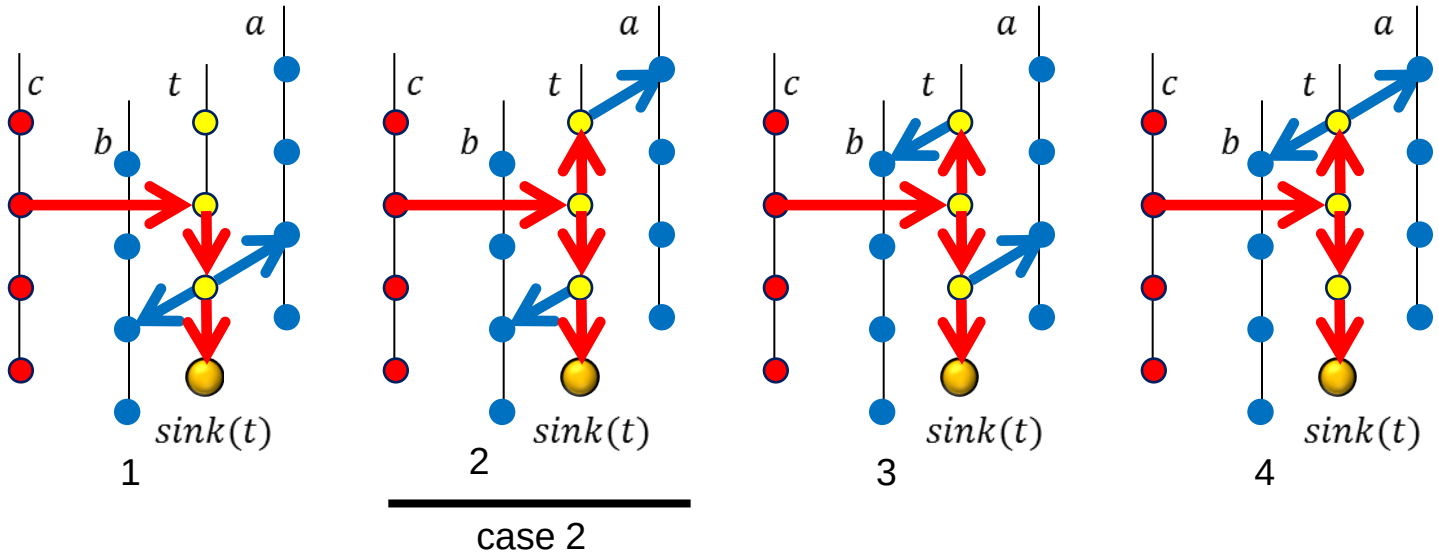
$$iND = \sum_{i=1}^{3} delay_{v_i} + delay_{w(M3)}$$

$$iLC = \sum_{i=1}^{3} C_{v_i} + C_{sink(t)} + C_{w(M3)}$$

$$iTC = \lambda \cdot iND + iVC + congestion\_cost_{w(M3)}$$

Calculation of Cost Increase

- The increase amount of load capacitance and total cost for each of other combinations are computed similarly.
- Among ALL these combinations, the one with minimum amount of total cost is chosen as the value $TC(t, M3)$
  - with the corresponding local capacitance $LC(t, M3)$

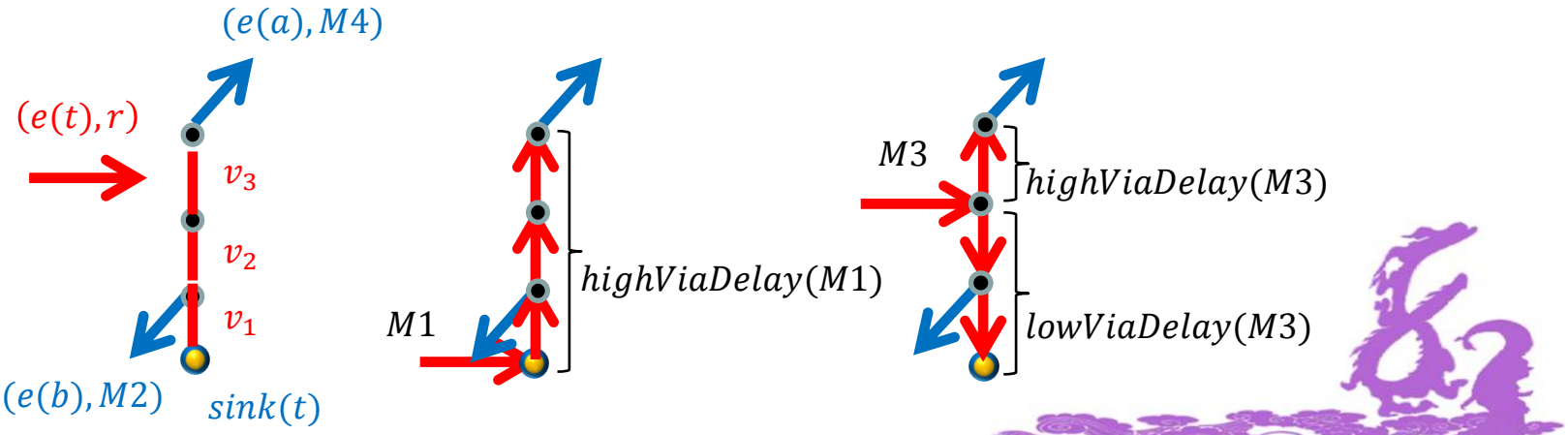


Fast Calculation of Delay Increase

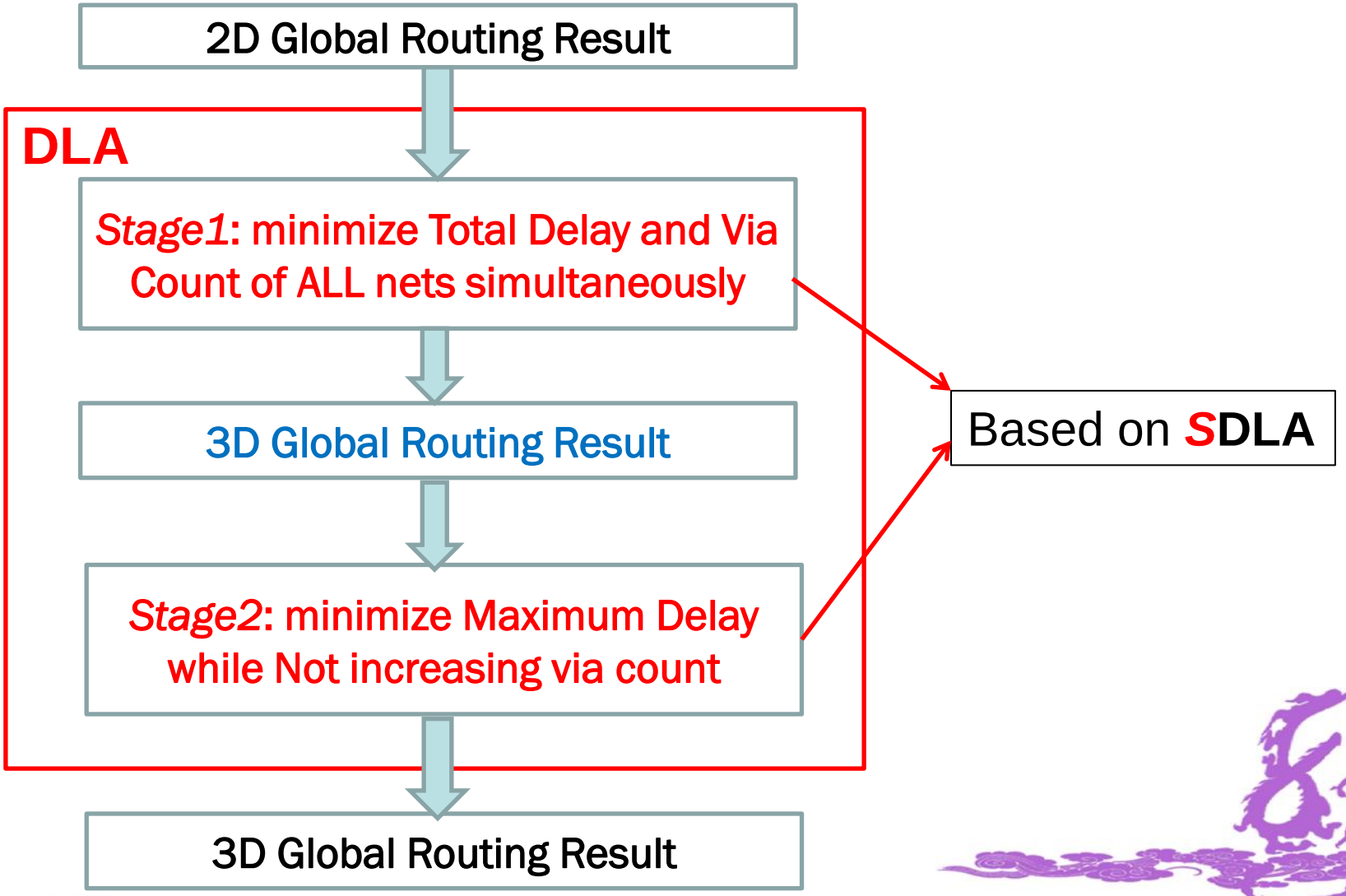
- For each sub-combination of all children of t, $iND(t,r)$'s and $iLC(t,r)$'s for All layer r's are computed in $O(3M)$ time.
- Delay Increase contains three parts: wire segment delay on layer r, and vias delay below / above layer r.

$$iND(r) = wireDelay(r) + lowViaDelay(r) + highViaDelay(r)$$

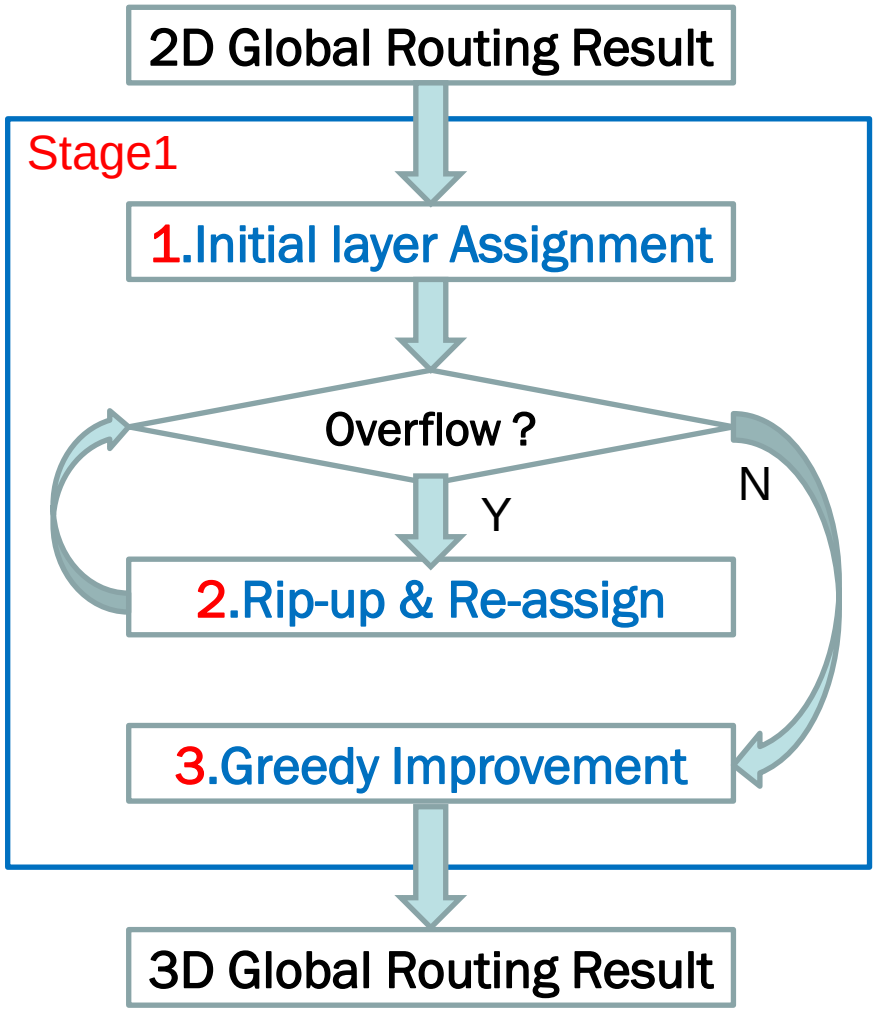
- ALL $lowViaDelay(r)$'s and $highViaDelay(r)$'s values can be computed on one trip scanning.
  - $iLC(t,r)$ is calculated along with $iND(t,r)$.



Overview of DLA



S1: Minimize Total Delay and Via Count



- ***SDLA*** (Single-net Delay-driven Layer Assignment) is presented in this work, it finds a solution with the minimum total costs of net delay, via count and wire congestion.
- Stage1 adopts the common negotiation-based RRR flow* with 3 steps.
- At each step, ***SDLA*** is used to perform layer assignment for each net repeatedly, until all nets are processed.

*similar to [McMurchie+FPGA95], [Liu+ASP-DAC11]

S2: Minimize the Maximum Delay

- Reduce the maximum delay of the set of nets while not increasing via count as much as possible
- Idea: continue to reduce the delay of the net with the maximum delay currently, until no improvement can be achieved
  - Make maximum delay decrease monotonically along with iterations

- Algorithm: For the net T with the maximum delay currently
  - Step1. rip-up-and-re-assign T by ***SDLA*** with large λ^* and *without* considering wire congestion constraint. If new_delay(T) does NOT decrease, break.
  - Step2. traverse all 3D edges of the new path of T: if a 3D edge causes congestion violation, select the net with minimum delay from the nets that pass through this 3D edge, and add it to set S
  - Step3. rip-up-and-re-assign the nets of S by ***SDLA*** *under* wire congestion constraint to eliminate the violation induced by Step1.

* Larger λ leads to smaller delay but more via count

Outline

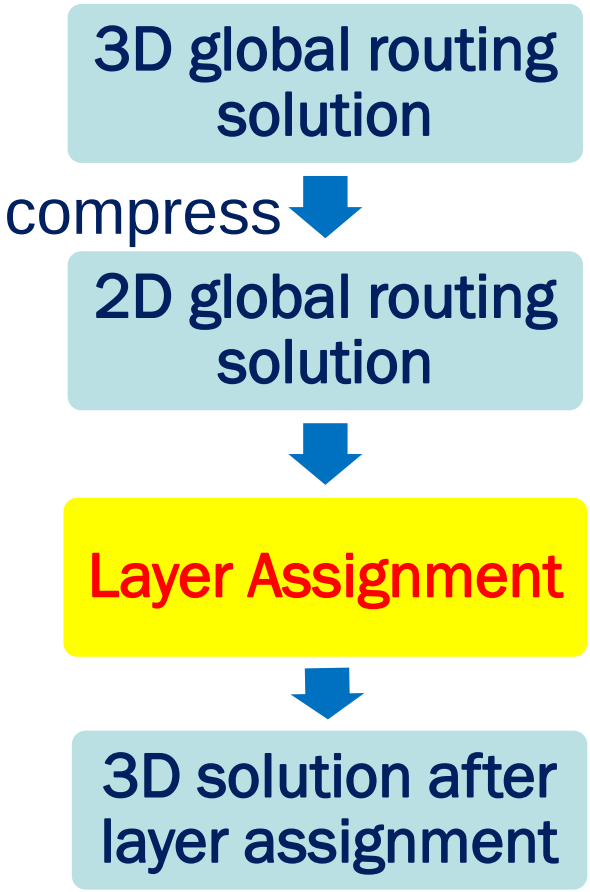
- Introduction
- Problem Formulation
- Proposed Algorithm
- **Experimental Results**
  - **Experimental setup**
  - **Delay V.S. via count**
  - **Algorithms comparison**
- Conclusion

ICCAD09 Circuits

Circuit	#nets	#tiles	#pins	#layers
adaptec1	219794	324*324	942705	6
adaptec2	260159	424*424	1063632	6
adaptec3	466295	774*779	1874576	6
adaptec4	515304	774*779	1911773	6
adaptec5	867441	465*468	3492790	6
bigblue1	282974	227*227	282974	6
bigblue2	576816	468*471	2121863	6
bigblue3	1122340	555*557	3832388	8
bigblue4	2228903	403*405	8899095	8
newblue1	331663	399*399	1237104	6
newblue2	463213	557*463	1771849	6
newblue4	636195	455*458	2498322	6
newblue5	1257555	637*640	4931147	6
newblue6	1286452	463*464	5305603	6
newblue7	2635625	488*490	10103725	8

- ICCAD09 circuits [Moffit+ICCAD09] are modified from ISPD07 / 08 3D global routing benchmarks.
- Higher metal layers have less routing tracks, showing that wires become thicker and wider on higher metals.
- 3-tier metal sizes is assumed:
- 6-layer circuits, wire width / spacing from M1-M6: 0.07, 0.07, 0.14, 0.14, 0.4, 0.4 um.
- 8-layer circuits, M1-M8: 0.07, 0.07, 0.07, 0.07, 0.14, 0.14, 0.4, 0.4 um.
- Wire thickness is twice wire width. Resistance of a driver is 100ohm, sink load capacitance is 1fF.
- Priority of each sink of a net is 1/m (m is the number of sinks of the net).

Experimental Setup

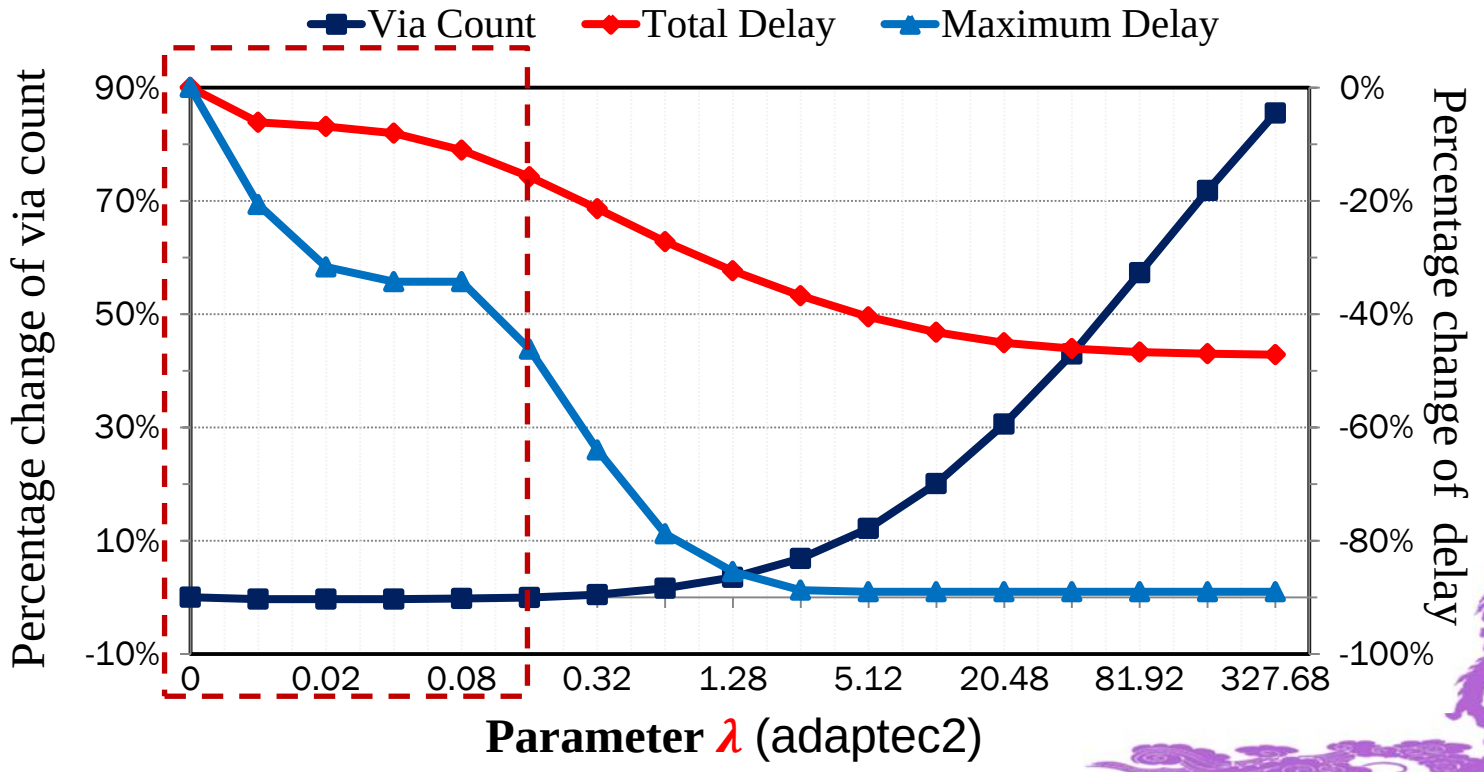


Experimental flow

- Proposed algorithms are implemented in C++ language.
- Machine: Linux PC with 2.27GHz CPU and 8GB memory
- ICCAD09 global routing circuits are used.
- To fairly compare this work with previous layer assignment works, each algorithm reads the same 2D global routing results of NTHU-Route 2.0 [Chang+ICCAD08].

Delay V.S. Via Count

- With larger λ, total delay and maximum delay decrease, but via count increases.
- When λ is small (<=0.16), as λ increases, total / maximum delay decrease quickly, while via count still keeps the same.



DLA (Stage1) is used. Cost function = λ * delay + #via

Algorithms Comparison

circuit	NVM			Greedy [d]			DLA (S1 [d])			DLA (S1 [d] +S2) [e]		
	VC [a] *(e4)*	*TD* [b] *(e5)*	*MD* [c]	*VC (e4)*	*TD (e5)*	*MD*	*VC (e4)*	*TD (e5)*	*MD*	*VC (e4)*	*TD (e5)*	*MD*
adaptec1	85.43	3.86	2767	117.26	3.45	1131	85.34	3.34	873	85.39	3.33	270
adaptec2	94.97	3.95	764	130.84	3.35	500	94.91	3.35	411	95.04	3.34	84
adaptec3	170.63	11.4	1689	238.02	9.70	1016	171.33	8.47	283	171.42	8.46	156
adaptec4	156.35	10.56	3352	212.24	8.77	3386	156.63	7.84	417	156.66	7.83	268
adaptec5	243.61	18.92	2792	337.09	15.98	1182	244.31	14.92	439	244.37	14.91	271
bigblue1	93.37	6.25	517	128.89	5.38	369	93.29	5.27	227	95.37	5.16	67
bigblue2	187.75	3.62	1984	257.81	3.32	1014	187.43	3.38	476	187.48	3.38	209
bigblue3	259.47	17.22	1054	379.71	15.40	998	260.30	12.08	235	261.54	12.05	108
bigblue4	518.67	32.51	10224	758.27	28.61	2299	519.65	24.76	859	519.69	24.76	737
newblue1	105.18	2.05	123	141.14	1.91	160	105.01	1.91	69	105.36	1.89	34
newblue2	126.31	7.88	764	171.40	6.95	1198	126.38	6.68	242	126.40	6.68	184
newblue4	229.82	9.08	1256	316.18	7.82	828	228.95	7.81	287	229.00	7.81	173
newblue5	407.69	22.17	991	549.79	18.97	870	407.41	18.36	274	408.20	18.31	114
Newblue6	342.82	20.91	997	466.64	18.09	997	343.32	16.90	311	343.46	16.89	110
Newblue7	847.27	56.13	6116	1246.3	41.87	1285	846.38	40.04	907	847.77	39.96	484
Ratio	1	1	1	1.409	0.837	0.487	1.000	0.773	0.178	1.002	0.772	0.092

NVM: [LiuASP-DAC11]; DLA: Delay-driven Layer Assignment; Greedy: greedy version of DLA (S1)

[a] **VC**: via count, [b] **TD**: total delay, [c] **MD**: maximum delay, [d] λ=0.15, [e] **S1/S2**: Stage 1/2

Conclusion

- DELAY-driven Layer Assignment (DLA) under MULTI-TIER interconnect structure, arising from 3D Global routing.
- Propose a 2-stage algorithm to minimize the total delay, maximum delay and via count simultaneously, and resistances and capacitances of metal wires and vias are considered in the RC model.
- Significantly reduce the total delay and maximum delay with roughly the same via count, compared to the state-of-the-art via count minimization layer assignment.
- DLA can be especially applicable to circuits in which the interconnecting layers have drastically different electrical characteristics.

Thank You!

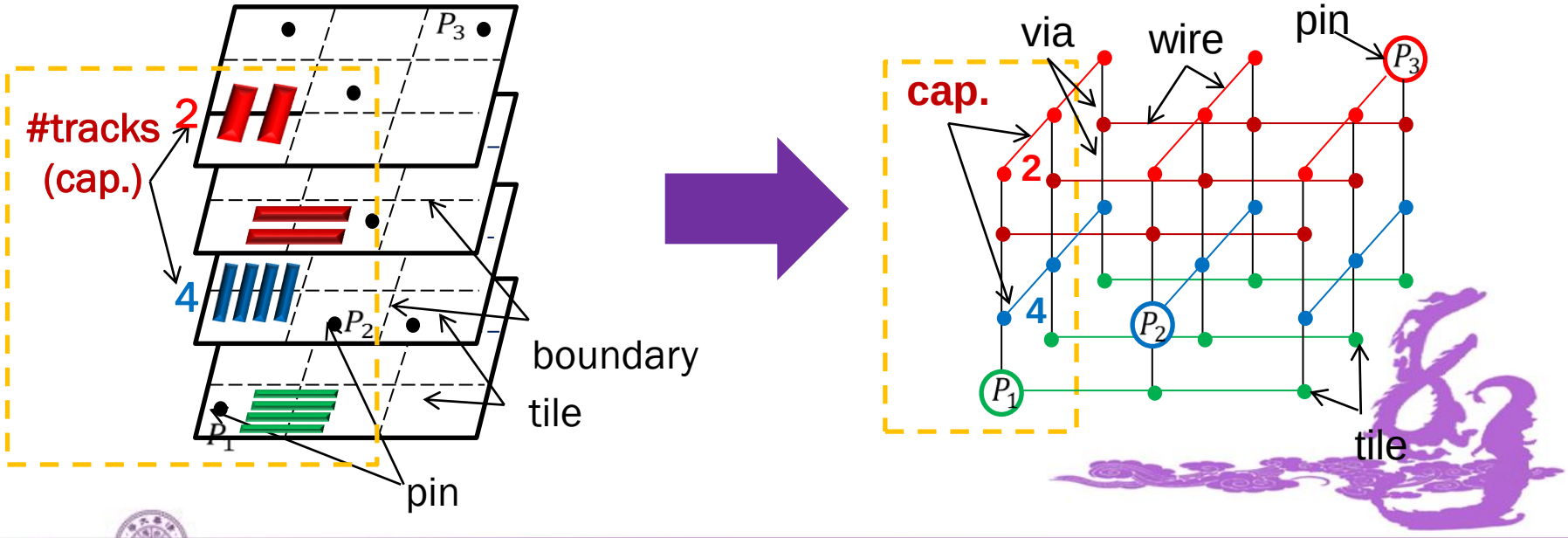
Q&A

Comparison of CPU Time (seconds)

circuit	NVM	Greedy	DLA(S1)	DLA(S1+S2)
adaptec1	121	137	195	199
adaptec2	96	93	153	157
adaptec3	325	312	471	481
adaptec4	234	205	321	333
adaptec5	338	452	519	529
bigblue1	131	273	213	221
bigblue2	196	162	309	316
bigblue3	312	286	515	525
bigblue4	556	488	980	993
newblue1	74	69	121	124
newblue2	123	101	171	176
newblue4	244	244	376	384
newblue5	486	482	718	733
newblue6	354	386	534	544
newblue7	965	1212	1677	1730
ratio	**1**	**1.076**	**1.60**	**1.634**

On Wire Capacity

- The wire overflow of a boundary edge indicates the wire usage locally exceeds the wire capacity.
- Wire capacity (detail routing track count) may be Different on each layer due to Different wire sizes and pitches.
- Wire usage is the net count assigned to the boundary edge.



Delay V.S. Via Count

- Different segments (of the same net or different nets) have *a wide range of delay weights and load capacitances*, some wire segments can generate much smaller delay when assigned to proper layers, which leads to big delay improvement.
- Given a net, even for multiple LA results with the same via count, different via poses induces *diverse circuit topologies* of the tree, and then induce diverse delays