RegularRoute: An Efficient Detailed Router with Regular Routing Patterns

Yanheng Zhang and Chris Chu

Electrical and Computer Engineering Department

Iowa State University

Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion

Outline

- **Motivation and Overview**
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion

Previous Detailed Routing Techniques

- Iterative ripup and reroute
  - Mighty [Shin et al. TCAD-87]
- Multi-level methodology
  - DUNE [Cong et al. TCAD-01]
  - MR [Chang et al. TCAD-04]
- Boolean satisfiability
  - SAT Router for FPGA [Nam et al. TCAD-02]
- Track routing
  - Track Routing [Batterywala et al. ICCAD-02]
- Escape routing
  - Escape Routing for Pin Clusters [Ozdal TCAD-09]

Sequential in nature

Net ordering issue

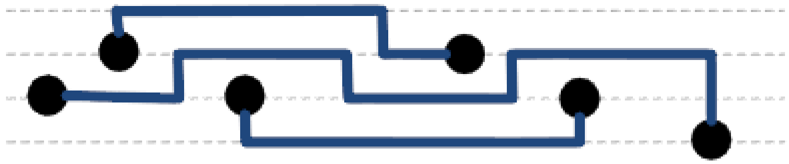
Concurrent approach
Long runtime

Pin access issue

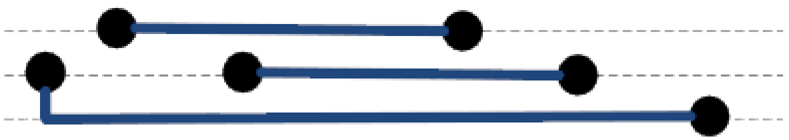
Not handle
full-chip routing

Apply Regular Routing Patterns

- Regular routing patterns
  - Potentially improve design rule satisfaction
  - Explore solution space more efficiently
  - Might affect routability due to restricted routing patterns



Non-trivial routing patterns

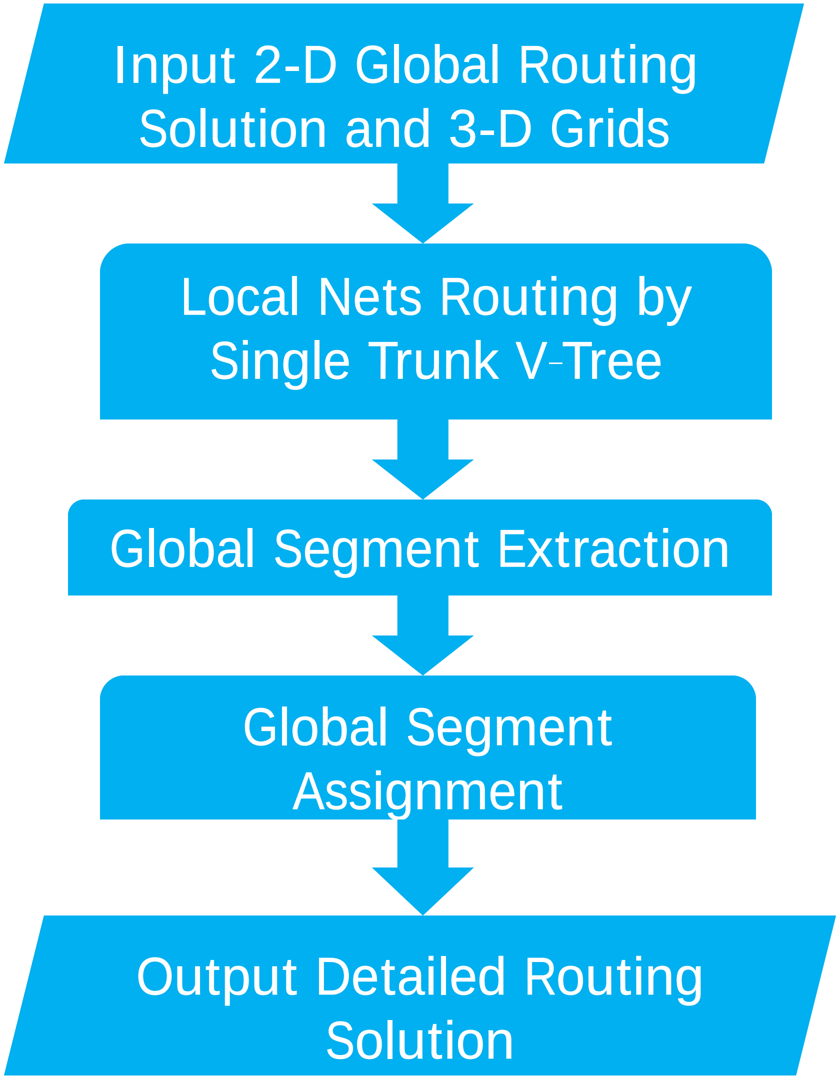


Regular routing patterns

Problem Formulation for Detailed Routing

- Input
  - 3-D detailed routing grids
  - 2-D global routing solution organized in global segments
  - Complete netlist
- Objective
  - Generate detailed routing solution to route as many nets as possible
  - Secondary objectives include minimizing wirelength, via count and non-preferred usage
- Assumptions
  - Each grid edge can accommodate exact one wire except blockage
  - Each layer has *preferred routing direction.* They are perpendicular for adjacent layers. Metal_1 is assumed to be *horizontal*
  - Pins are assumed to be on metal_1

RegularRoute: Flow and Overview



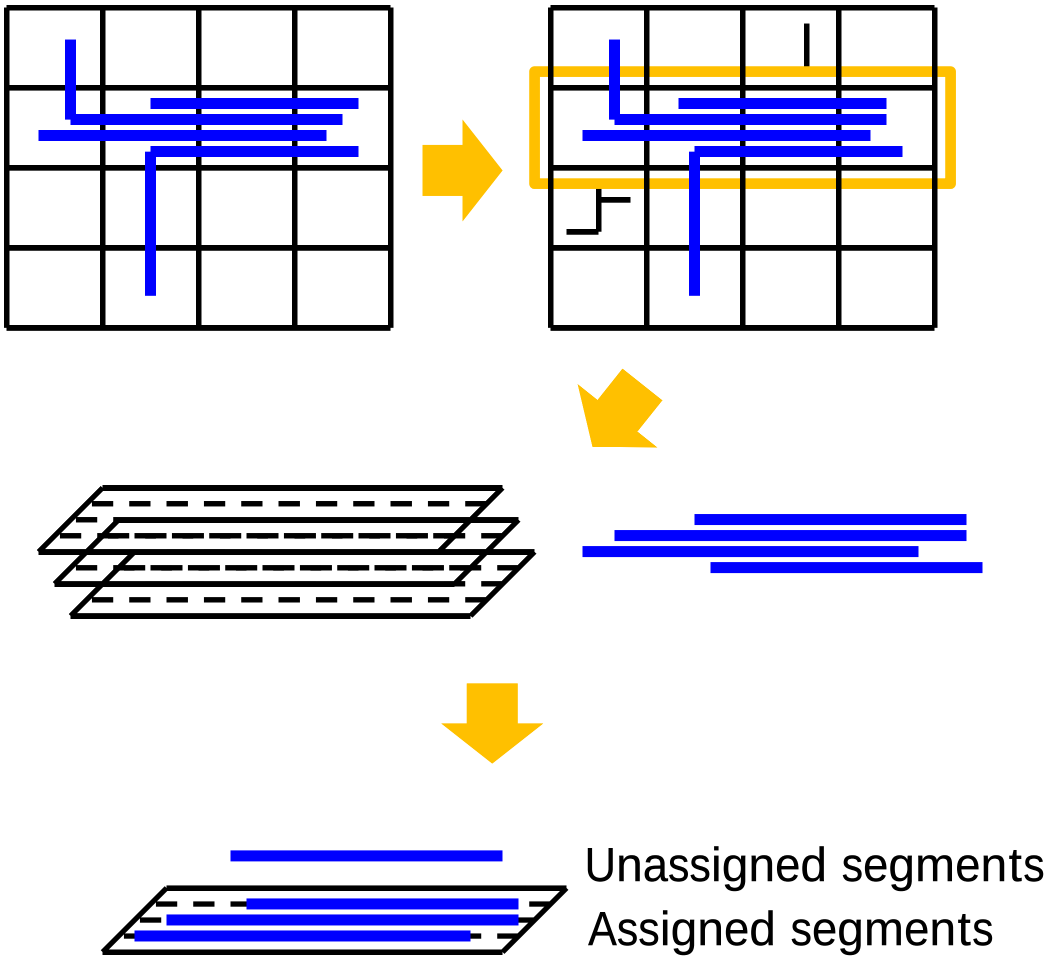
Input 2-D Global Routing Solution and 3-D Grids

Local Nets Routing by Single Trunk V-Tree

Global Segment Extraction

Global Segment Assignment

Output Detailed Routing Solution



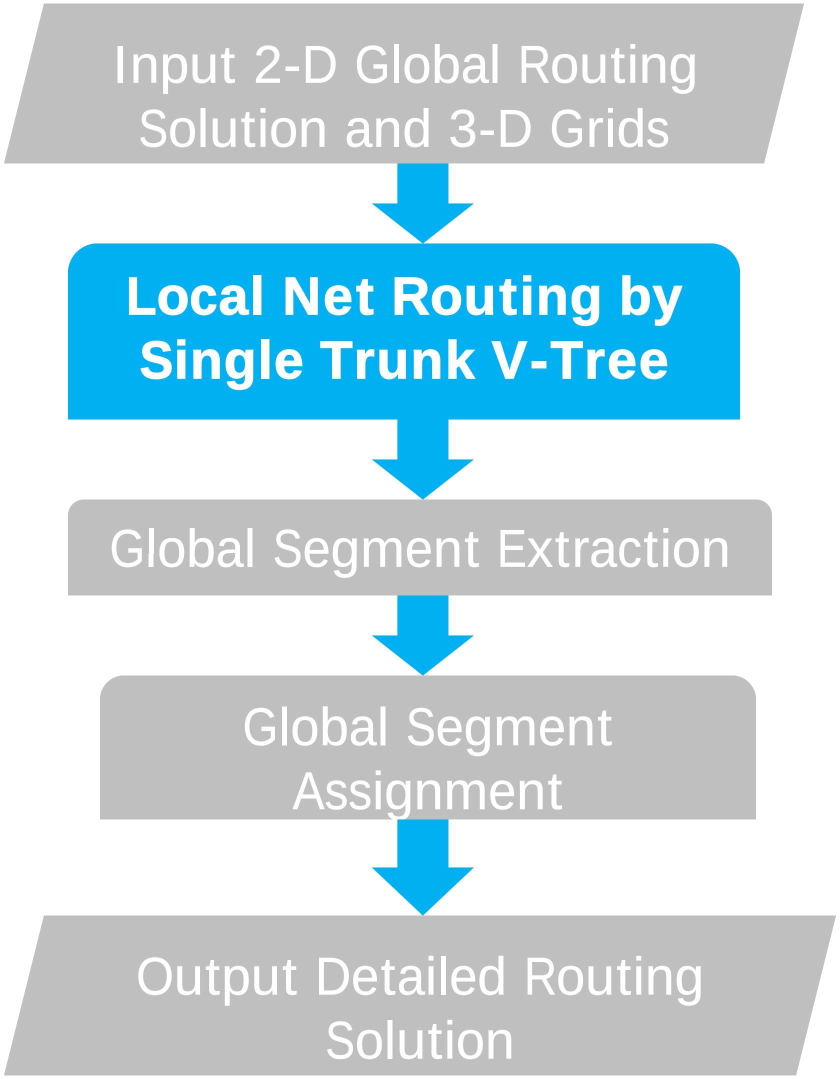
RegularRoute: Our Contributions

- Applying regular routing patterns
  - Use regular routing patterns instead of non-trivial patterns
  - Correct-by-construction for satisfying more design rules
- Panel based global segments allocation
  - Formulate assigning global segments in one panel as MWIS proble
  - All nets inside each panel are considered simultaneously
- Novel techniques to improve routability
  - Effective partial assignment for further assignment
  - Pin promotion to prevent pin access issue
- Fast computational time
  - Fast heuristic in solving the MWIS
  - Can easily be adapted to parallel version

Outline

- Motivation and Overview
- **Local Net Routing**
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion

Local Net Routing



Input 2-D Global Routing Solution and 3-D Grids

Local Net Routing by Single Trunk V-Tree

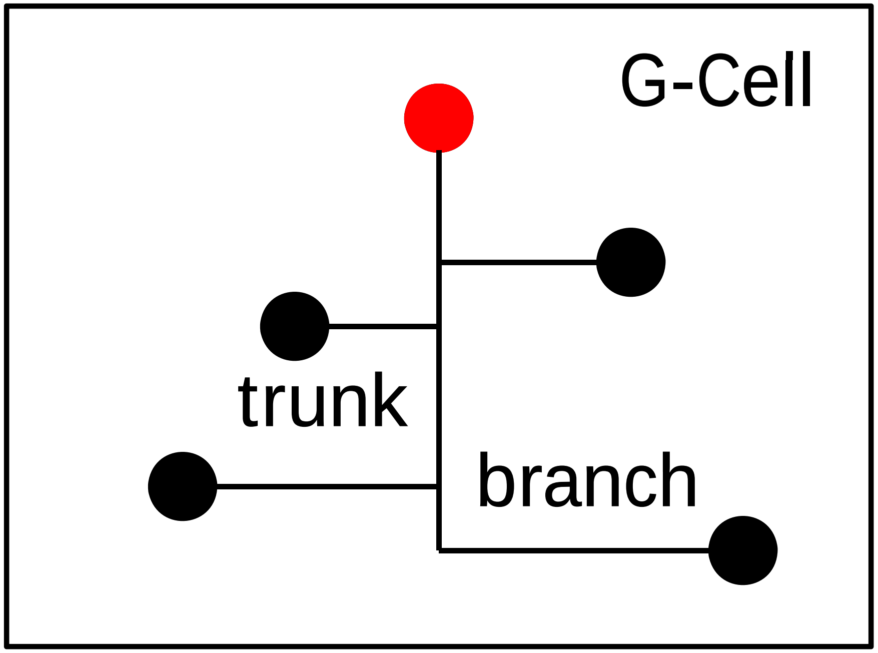
Global Segment Extraction

Global Segment Assignment

Output Detailed Routing Solution

Single-Trunk V-Tree

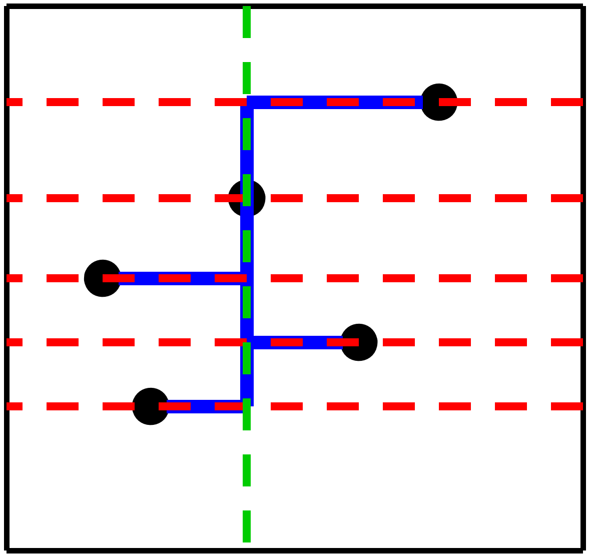
- Single-Trunk V-Tree
  - Find pin with *median* X coordinate
  - Construct *trunk* with vertical wire (metal_2)
  - Connect other pins to trunk as *branch*
  - Time complexity: O(n)



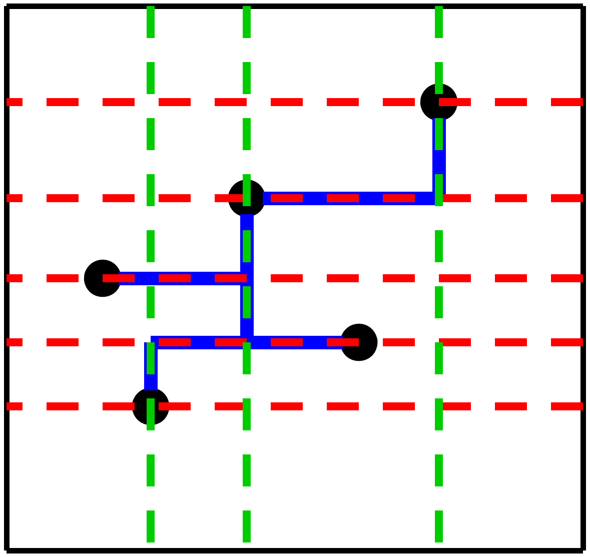
V-Tree vs. Arbitrary Tree

- V-Tree vs. Arbitrary Tree
  - Number of blocked horizontal tracks: V-Tree = Arbitrary Tree
  - Number of blocked vertical tracks: V-Tree < Arbitrary Tree

Minimize metal_2 usage



Single Trunk V-Tree

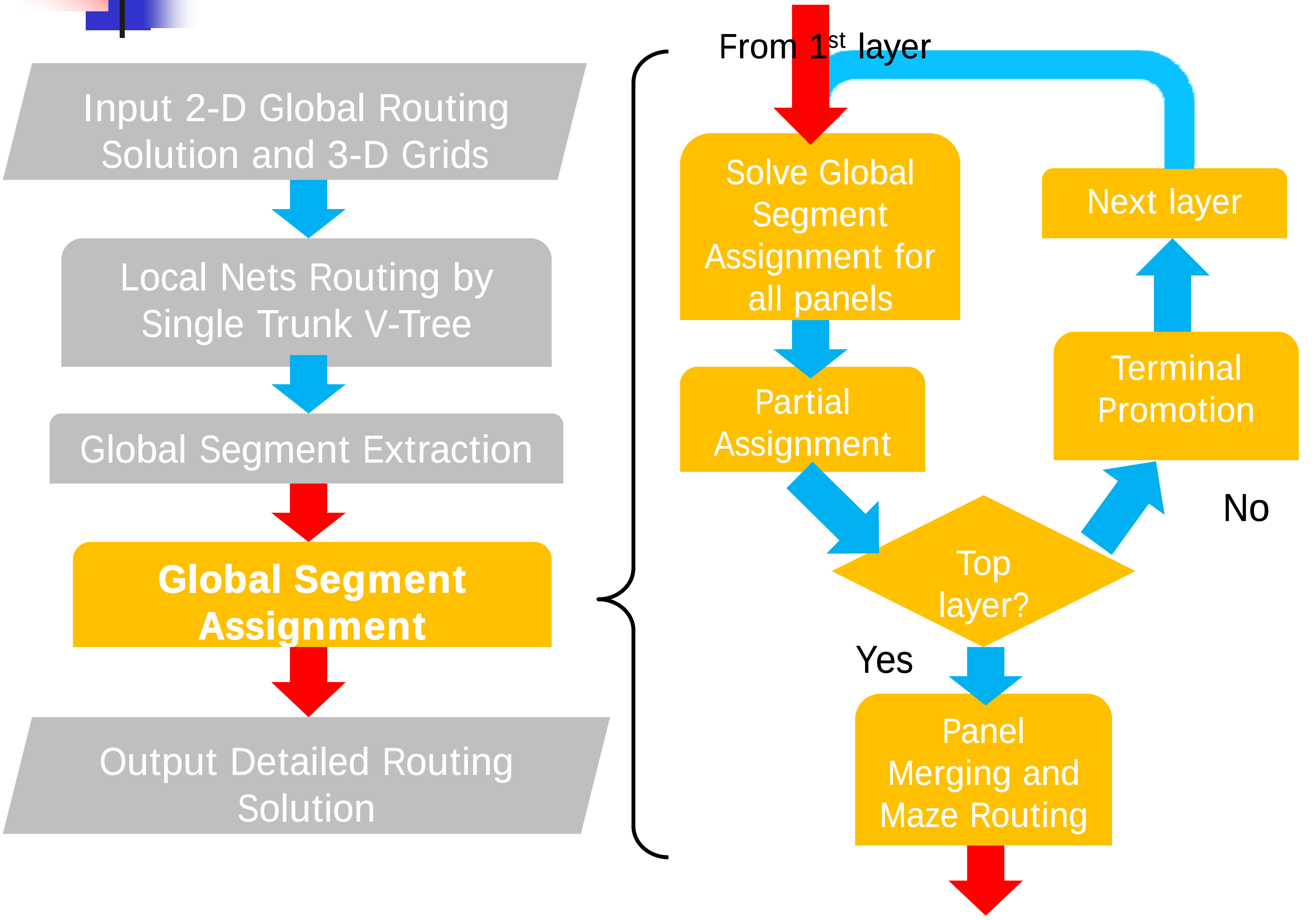


Arbitrary Tree

Outline

- Motivation and Overview
- Local Net Routing
- **Global Segment Assignment**
- Experimental Results
- Conclusion and Discussion

Global Segment Assignment



Input 2-D Global Routing Solution and 3-D Grids

Local Nets Routing by Single Trunk V-Tree

Global Segment Extraction

Global Segment Assignment

Output Detailed Routing Solution

From 1st layer

Solve Global Segment Assignment for all panels

Partial Assignment

Top layer?

No

Terminal Promotion

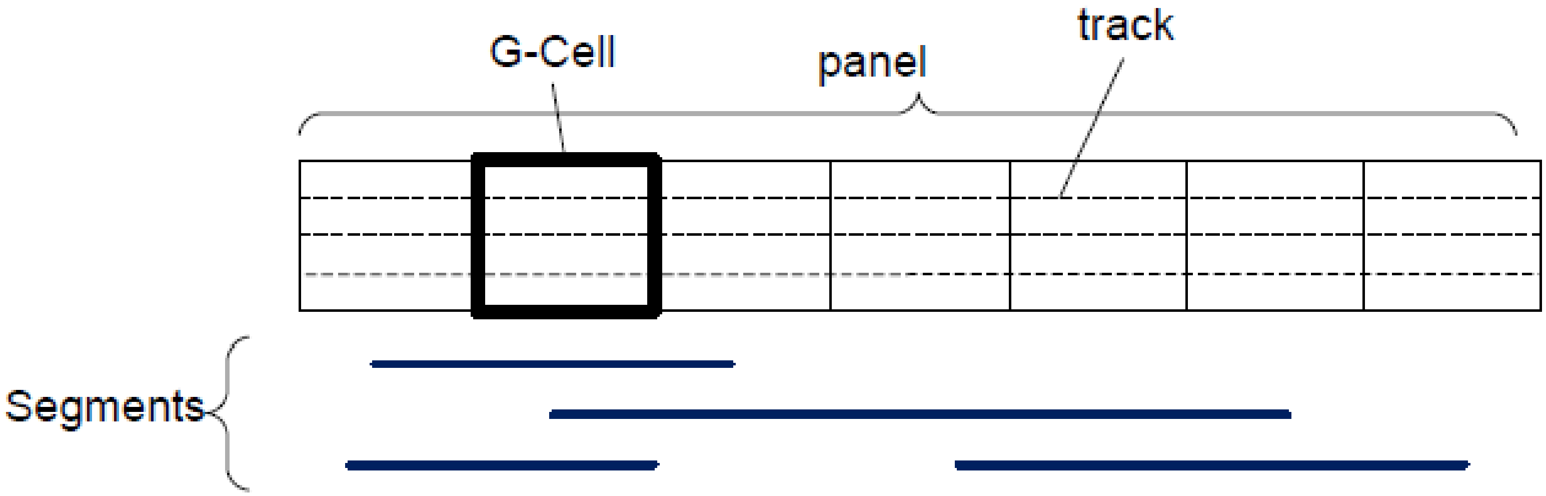
Next layer

Yes

Panel Merging and Maze Routing

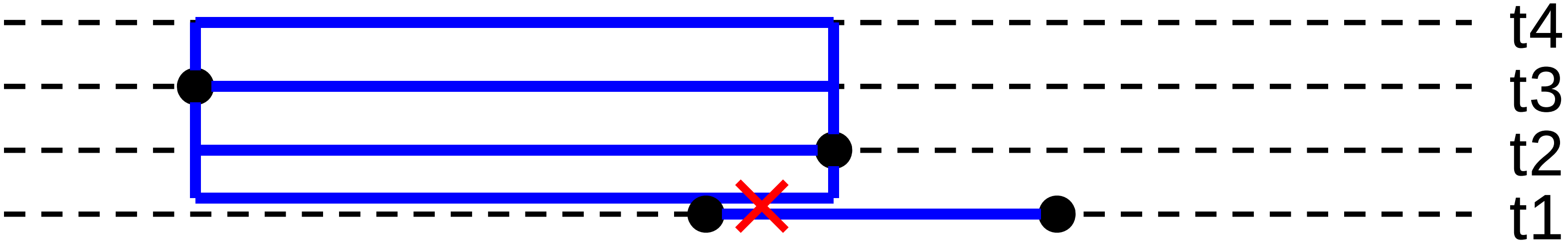
Global Segment Assignment in one Panel

- Input
  - A set of global segments that have not been assigned
  - A set of routing tracks inside one panel
- Objective
  - Assign as many segments as possible in regular routing patterns
  - Minimize wirelength, via count, non-preferred usage
- Concepts
  - *Track*: A sequence of grids in preferred routing direction
  - *Panel*: A collection of tracks in one column/row of G-Cells



Concept of a Choice

- Choice
  - A valid regular routing solution for one segment
  - Number of choice reflects the flexibility of assignment for one segment
  - Two choices that cannot co-exist cause a *conflict*

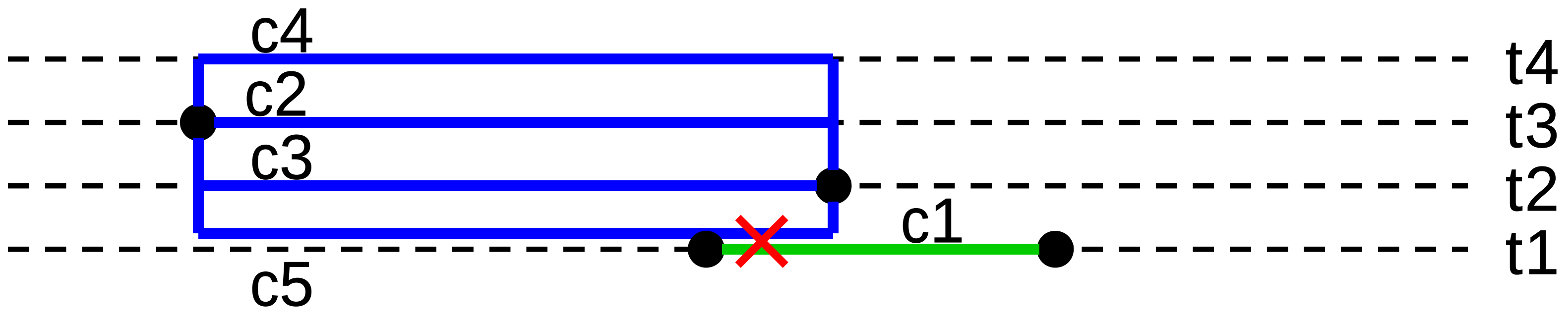


conflict

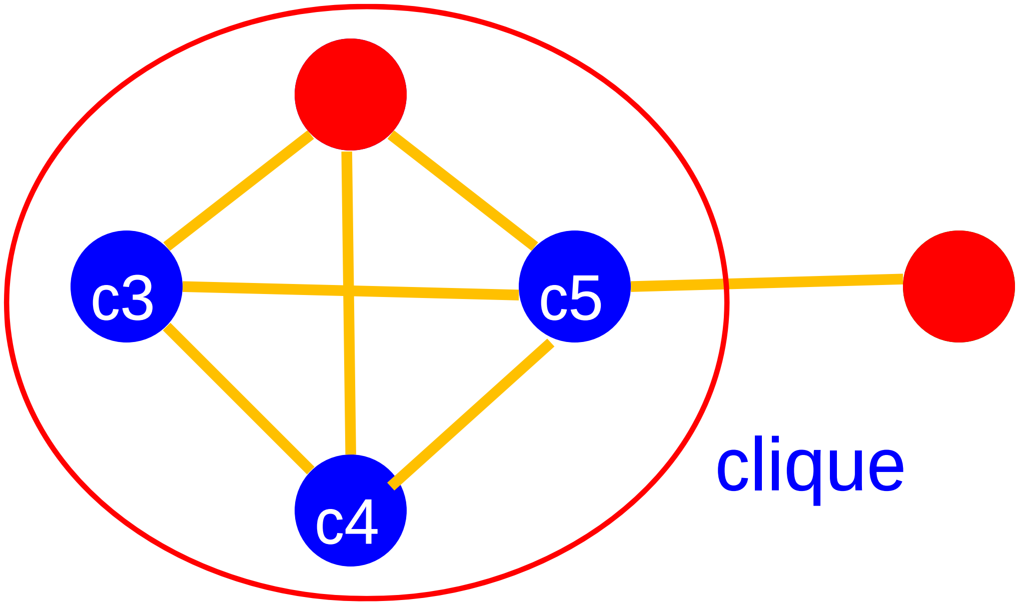
MWIS problem

- Maximum Weighted Independent Set (MWIS)
  - Formulate Global Segment Assignment in one Panel as MWIS problem
  - Introduce conflict graph ***G*** with vertex set ***V*** and edge set ***E***: each vertex represents one choice, each edge represents conflict between two choices
  - Each vertex is assigned a weight representing assignment priority
  - Objective: find the independent set of vertices to maximize total weight

Example of Conflict Graph



conflict



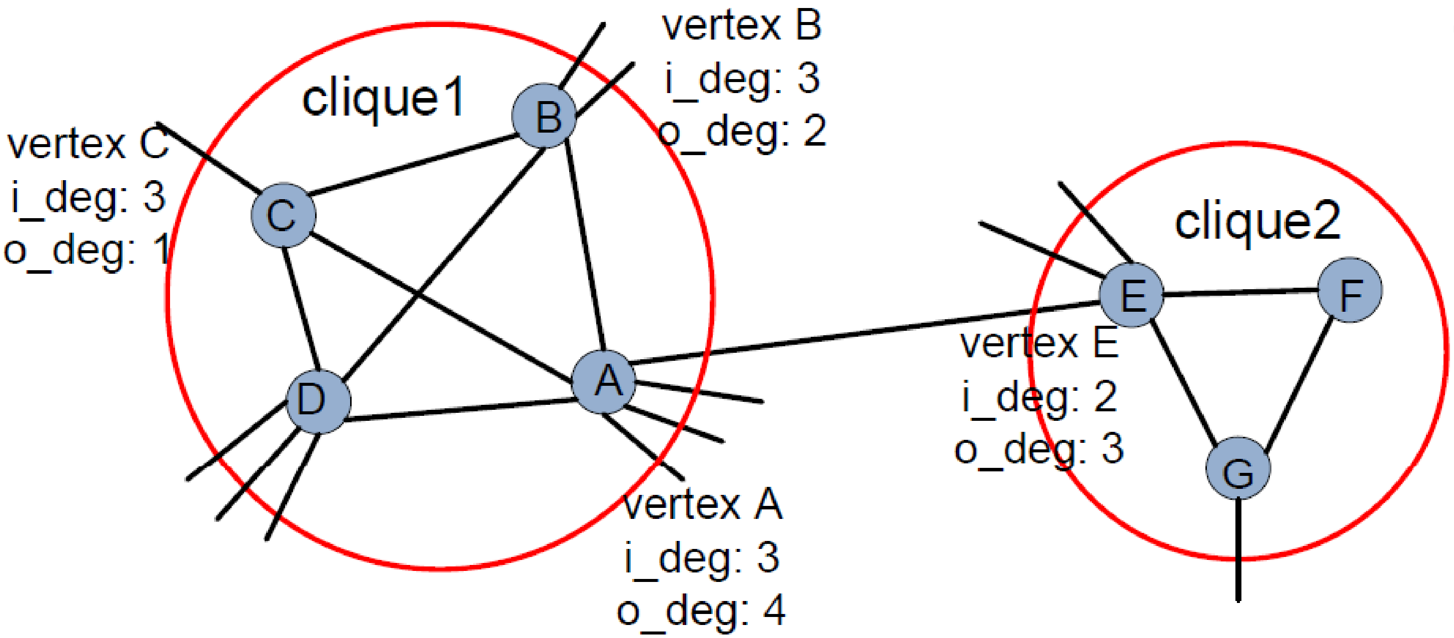
clique

Calculate weight for vertices

$$W(v) = \underline{L} - \alpha_1 \times \underline{\|R\|} + \alpha_2 \times \underline{AvD} \\ + \alpha_3 \times \underline{(F_1 + F_2)}$$

- Contains five components
  - Segment length (number of spanned G-Cells)
  - Terminal connection
  - G-Cell boundary density
  - Flexibility component for ending G-Cell with pending segment

Solve MWIS

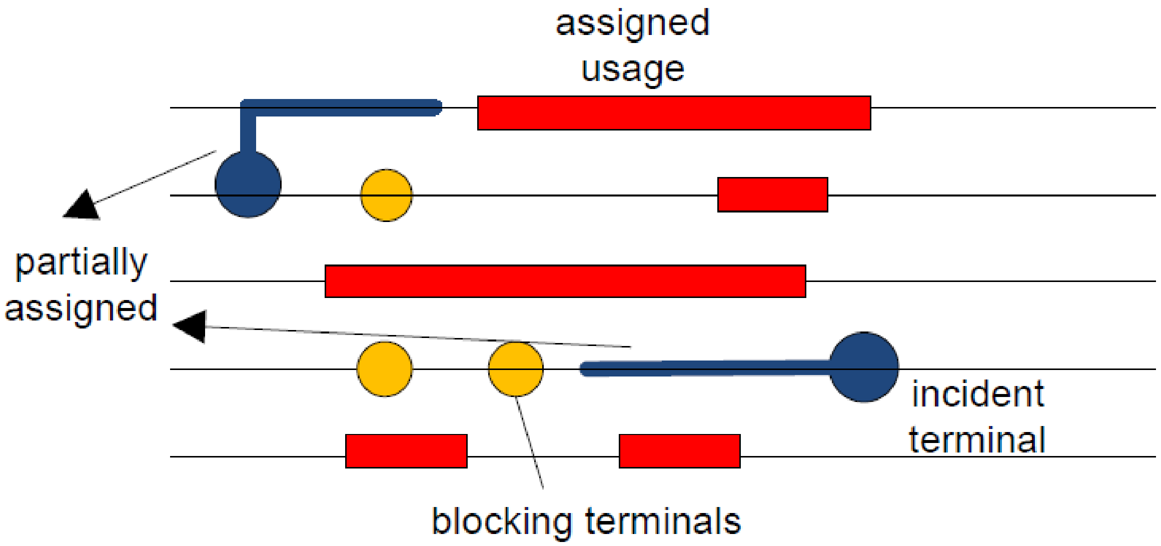


$$C(v) = W(v) - \beta \times i\_deg(v) - \gamma \times o\_deg(v)$$

- Solve MWIS problem
  - Rank vertices based on cost
  - Extract vertex with largest weight and do assignment
  - Update incident vertices and in/out degrees
  - Use *heap* for efficient extraction and update

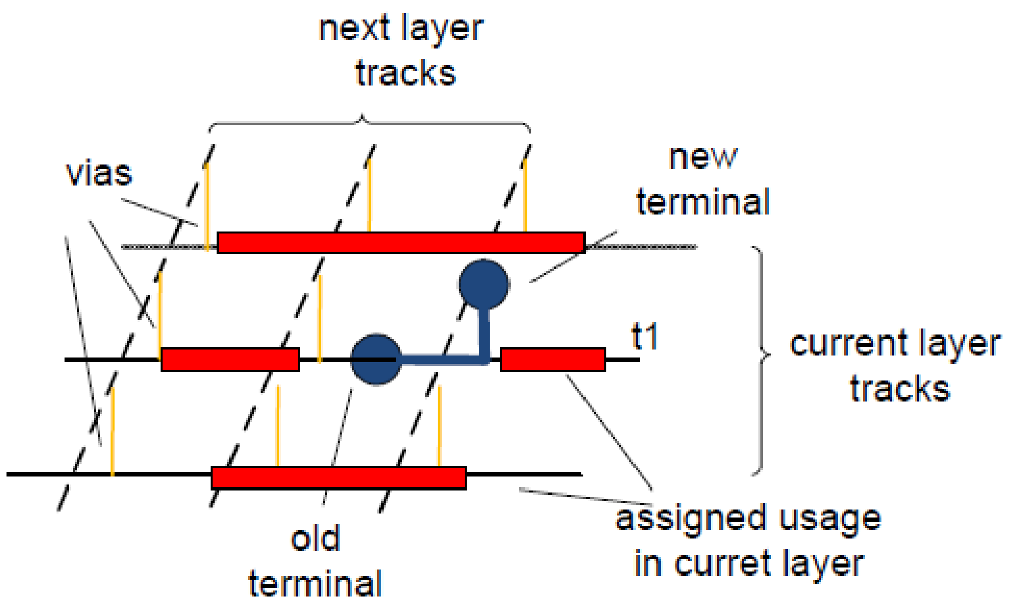
Partial Assignment

- Partial Assignment
  - Improve resource utilization after MWIS
  - Assign partial segment starting from terminals
  - Post-processing after MWIS



Terminal Promotion

- Terminal Promotion
  - Terminal connection issue: segment is assigned in upper layer while terminals are on lower layers
  - Promote terminals after processing current layer
  - Treat new terminals *as if* they are on upper layer



Unassigned Segments on Top Layer

- Panel Merging
  - Allow violation of the input global routing solution
  - Offers more flexibility
  - Can be applied in lower layers
- Maze Routing
  - Line probe based maze routing
  - 3-D maze routing
- Optimal MWIS Solver
  - Last resort for better solving the problem
  - Generally slow and solution quality is not guaranteed

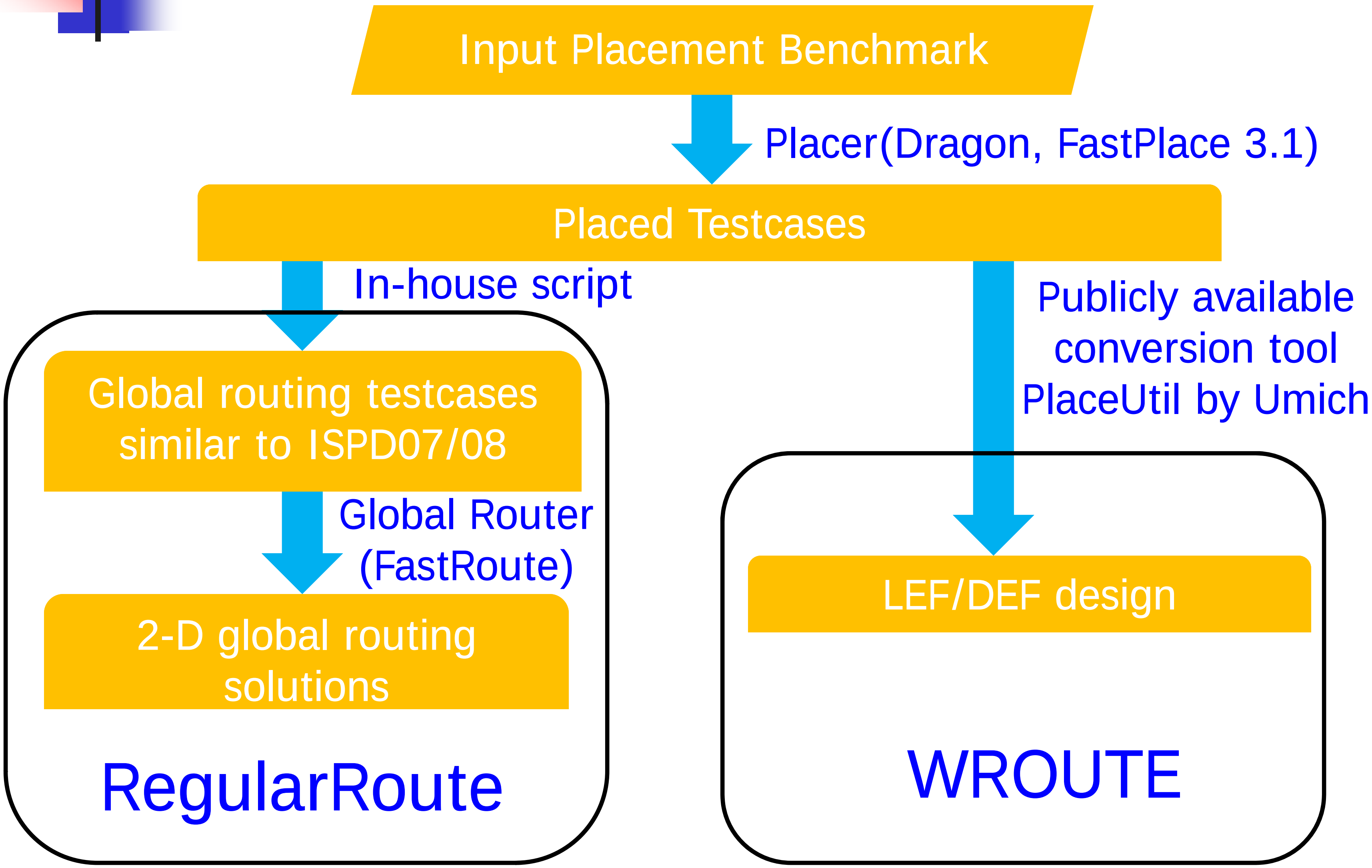
Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- **Experimental Results**
- Conclusion and Discussion

Experimental Set-up

- Testcases
  - ISPD98 placement benchmark suite derived testcases
  - ISPD05 placement contest benchmark suites derived testcases
- Computing Platform
  - 3.16 GHz Intel Xeon processor with 32G memory
- Input to RegularRoute
  - Global routing testcases with similar format to ISPD07/08 global routing contest benchmark suites
  - 2-D global routing solutions
- Input to WROUTE
  - LEF/DEF design for placed testcases

Flow for making testcases



Input Placement Benchmark

Placer(Dragon, FastPlace 3.1)

Placed Testcases

In-house script

Global routing testcases similar to ISPD07/08

Global Router (FastRoute)

2-D global routing solutions

RegularRoute

Publicly available conversion tool PlaceUtil by Umich

LEF/DEF design

WROUTE

Results for Local Net Routing

		Single Trunk V-Tree				RSMT			
	# Local Nets	# un. Local	CPU (Sec.)	Metal_2 usage	# un. Global	# un. Local	CPU (Sec.)	Metal_ 2 usage	# un. Global
ibm01	1081	0	0.04	6.3	0	0	0.02	9.6	0
ibm02	1750	0	0.09	12.8	0	0	0.04	15.3	0
ibm07	4479	0	0.18	22.3	0	7	0.05	32.6	5
ibm08	5539	0	0.23	27.8	0	0	0.11	39.6	0
ibm09	5429	0	0.20	28.2	0	9	0.08	37.9	0
ibm10	2984	0	0.27	17.4	0	0	0.12	29.4	1
ibm11	6983	0	0.26	38.9	0	4	0.07	50.1	7
ibm12	2433	0	0.32	14.5	0	0	0.12	26.8	0

Full Results for ISPD98 Derived Testcases

	FR4.0	RegularRoute				WROUTE(Encounter)			
	CPU (Sec.)	# un. assign	CPU (Sec.)	Via ×10e5	wlen ×10e5	Violation	CPU (Sec.)	Via ×10e5	wlen ×10e5
ibm01	0.47	0	3.17	0.84	6.9	0	47	0.84	7.1
ibm02	2.71	0	14.4	2.9	15.9	3	155	3.0	16.1
ibm07	8.51	0	34.3	3.8	39.9	12	190	3.8	40.6
ibm08	10.1	0	54.6	4.4	44.5	0	193	4.4	44.1
ibm09	6.11	0	43.1	3.9	37.0	0	184	3.9	37.4
ibm10	8.97	0	66.9	6.0	68.5	0	290	6.2	69.5
ibm11	15.7	0	68.1	4.8	53.2	23	287	5.1	53.8
ibm12	25.4	0	112.1	7.0	97.4	9	422	7.2	98.3
Sum	77.97	0	396.7	33.6	363.3	47	1768	34.4	366.9
Norm	0.25	/	1	1	1	/	4.45	1.02	1.01

Full Results for ISPD05 Derived Testcases

	FR4.0	RegularRoute				WROUTE(Encounter)			
	CPU (Sec.)	# un. assign	CPU (Sec.)	Via ×10e6	wlen ×10e7	Viola-tion	CPU (Sec.)	Via ×10e6	wlen ×10e7
a1	141	0	622	1.5	8.4	0	1201	1.5	8.5
a2	189	0	558	1.9	10.2	221	1344	2.0	10.4
a3	342	0	1176	3.5	21.8	0	3939	3.6	22.1
a4	289	4	1330	3.0	19.8	324	4424	3.2	20.4
b1	134	0	911	2.2	9.8	0	1802	2.2	9.7
b2	249	0	1177	3.7	21.2	54	2856	3.9	22.0
Sum	3384	4	5774	15.8	91.2	599	15566	16.4	93.1
Norm	0.22	1	1	1	1	150	2.69	1.04	1.02

Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- **Conclusion and Discussion**

Conclusion and Future Work

- Conclusion
  - We proposed RegularRoute for routing with regular routing patterns in detailed routing
  - Propose a layer by layer and panel by panel strategy to solve global segment assignment
  - Formulate MWIS and solved by fast heuristic
  - Proposed other effective methods for improving QoR

- Future Work
  - Continue improve performance of RegularRoute
  - Incorporate more design-related objectives
  - Develop parallel version of RegularRoute

Thank You !

Questions?