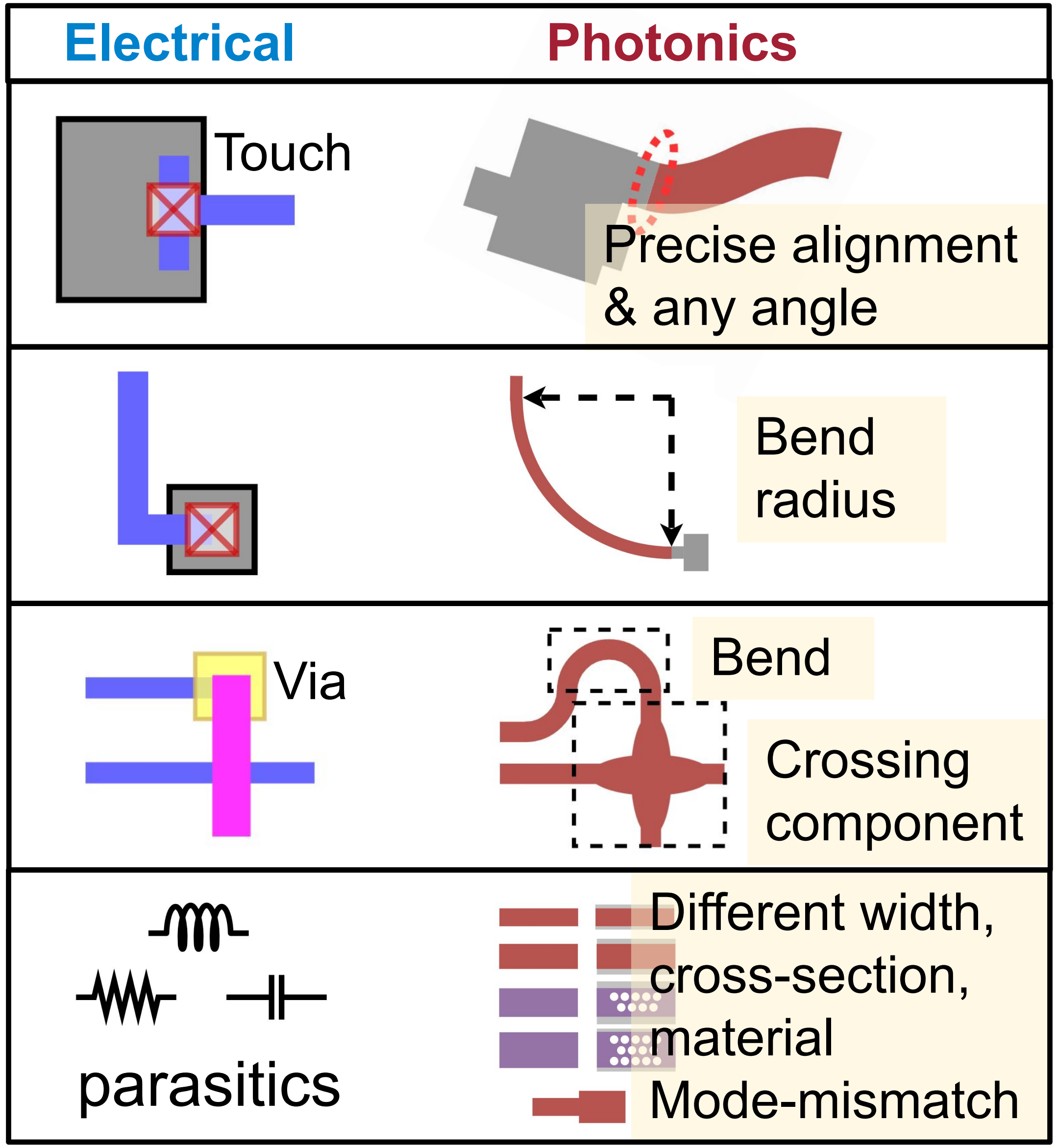
LiDAR: Automated Curvy Waveguide Detailed Routing for Large-Scale Photonic Integrated Circuits

Hongjian Zhou[1], Keren Zhu[2], Jiaqi Gu[1]

[1]Arizona State University, [2]Fudan University
School of Electrical, Computer and Energy Engineering
hzhou144@asu.edu
jiaqigu@asu.edu | scopex-asu.github.io

What Makes PIC Detailed Routing Different

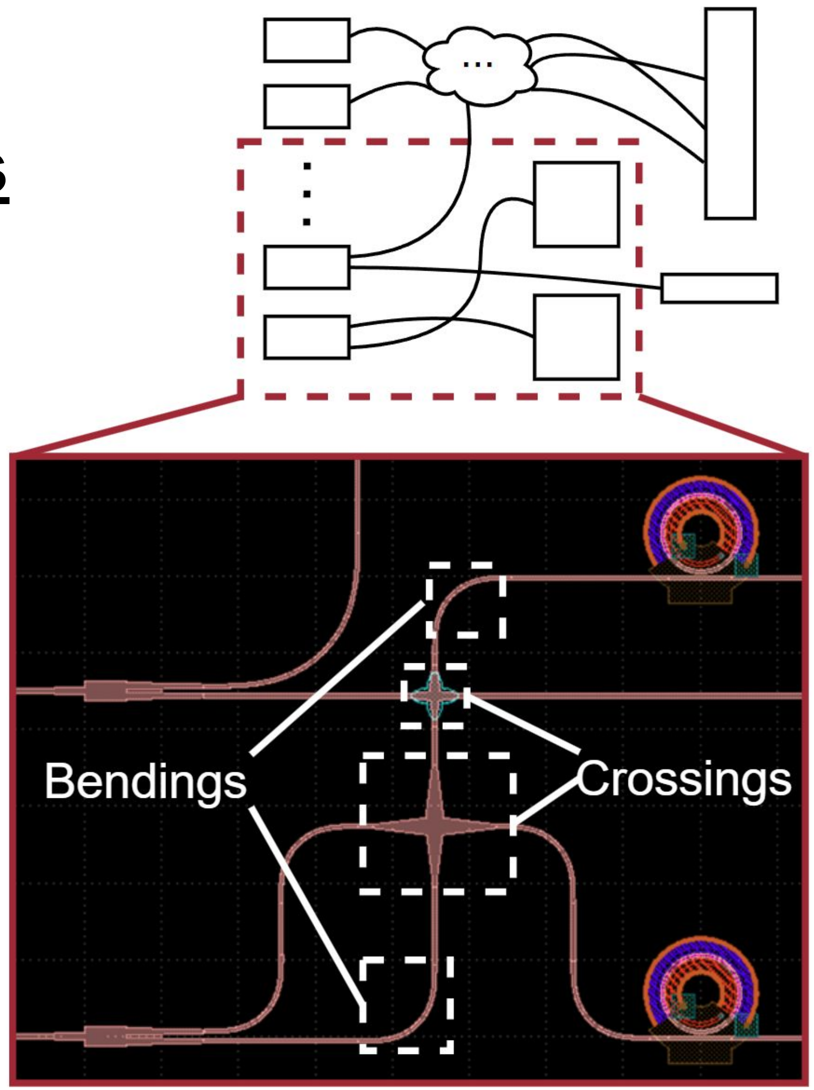
- Pin/Port Access
  - Need to consider port orientation
- Curvy bend
  - Need additional space
- Crossing
  - Single-layer, 90° intersection
  - Area-consuming
- Signal integrity
  - Model matching
  - Thermal crosstalk
  - ...



Heavily relies on manual design!

Manual PIC Routing Method

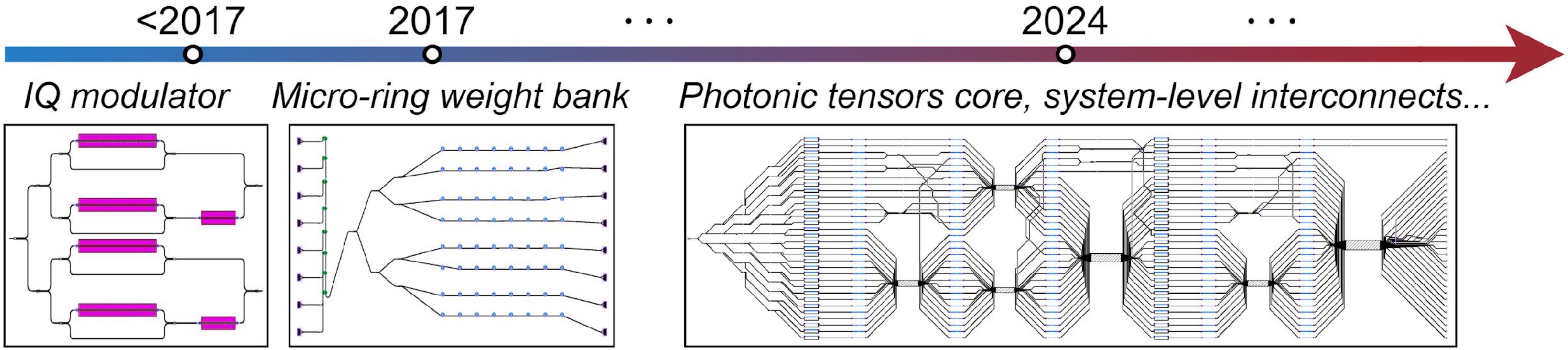
- Path is treated as **separate instances**
  - Segment, bending, crossing
- Place each instance carefully:
  - Consider bending radius constraint
  - Consider spacing constraint
  - Consider alignment constraint
  - ...
- Back-and-forth modifications
  - Instances are highly relative



Time-consuming & Only work for small scale circuits

PIC Scale and Design Complexity Grow Rapidly

- From **tens** to **hundreds** of instances/nets
- From **well-structured** designs to **irregular** designs
- From **basic geometry** to **stringent and multi-disciplinary** rules



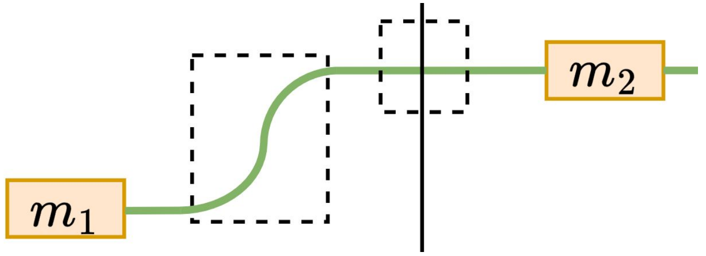
Small-scale, manually routable (~1day)

Large-scale, complex PICs (~weeks)

Require auto detailed routing tool to increase productivity and reduce time-to-market

Quality Metric and Problem Formulation

- ***Quality Metric***: Path with maximum insertion loss: IL_{max}
  - Determine the **needed laser power**
  - Path insertion loss: = device insertion losses + net insertion losses



  - Net insertion loss: waveguide connection loss
    - Propagation loss: proportional to **path length**
    - Bending loss: proportional to **bending angle**
    - Crossing loss: proportional to **crossing number**
- ***Problem formulation***:
  - Given a set of nets and placed devices, generate legal routing for each net

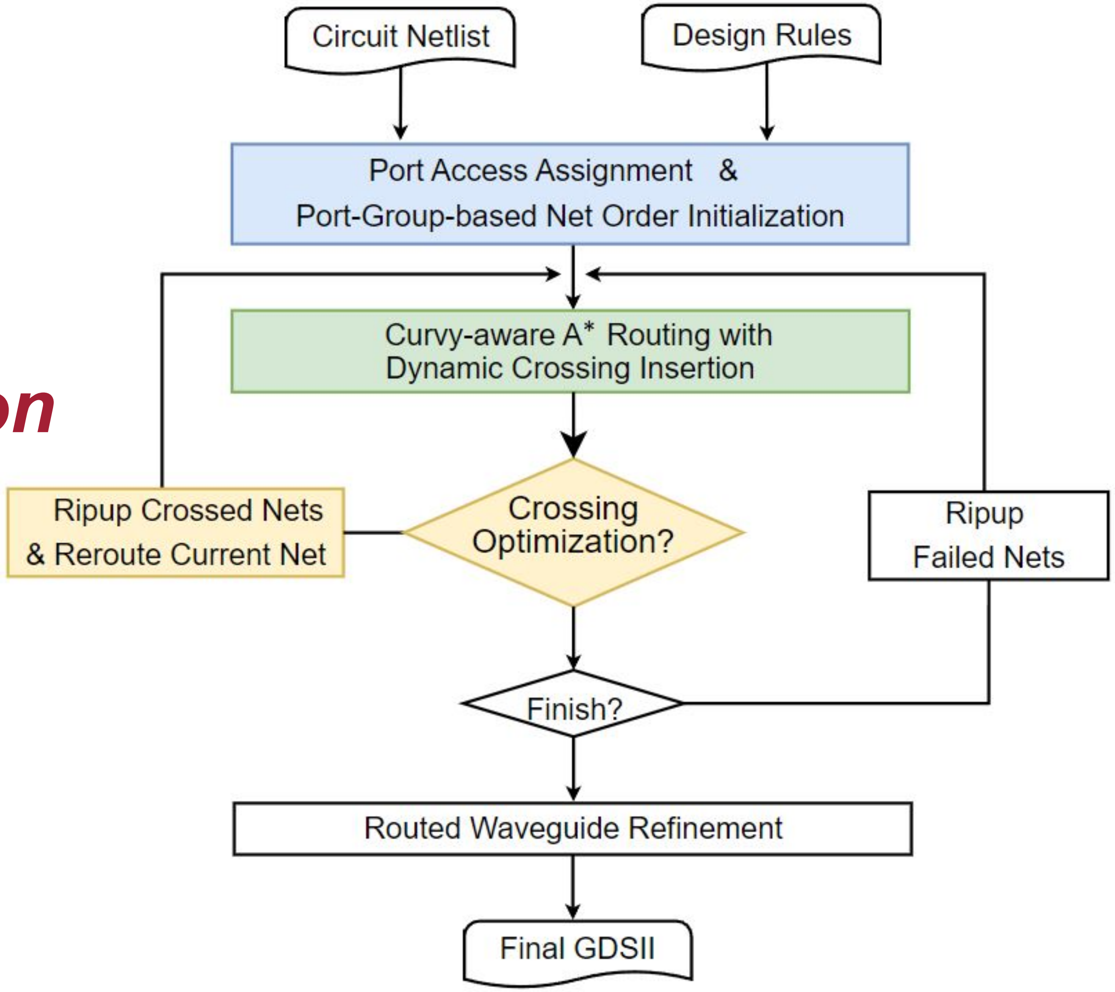
$$minimize \ \ IL_{max}$$

Related Work

- Focus on global route planning: **optimize #crossing**
  - Proton [ICCAD'13]: Adaptive crossing penalty
  - PlanarNoC [DAC'19]: Introduce flipping and rotation of devices
  - ToPro [ICCAD' 21]: Dynamic pushing algorithm
  - **Overlook physical implementation --- can not final GDS layout**
    - **Not curvy-aware bending**
    - **Not enough space for crossing insertion**
- Detailed channel routing:
  - Manhattan grid-based left-edge algorithm [Condrat+, MWSCAS'12]
  - Handle waveguide curves by non-Manhattan channel routing [Condrat+, SLIP'13]
  - **Cannot optimize crossing**
- Challenge: Generate implementable routing solution while

Proposed LiDAR Framework

- ***How to find a path that is physically implementable?***
  - *Sol:* Curvy-Aware *A** Search
    - Parametric neighbors' generation
    - Dynamic crossing insertion
- ***How to mitigate routing congestion on a single layer?***
  - *Sol:* Reserve routing resource
    - **Predictively** reserve space near ports
    - **Joint** planning for a group of nets
- ***How to optimize crossing?***
  - *Sol:* Local rip-up & reroute scheme

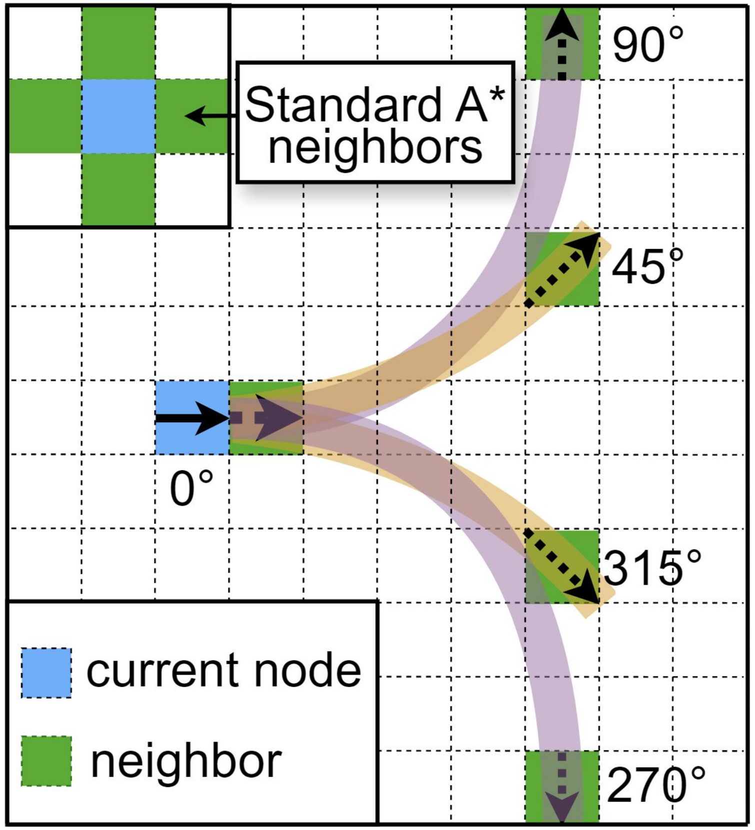


Curvy-Aware A^* Search

- We *augment* standard $\boldsymbol{A}^*$ search to support curvy waveguide:

$$f(n) = g(n) + h(n)$$

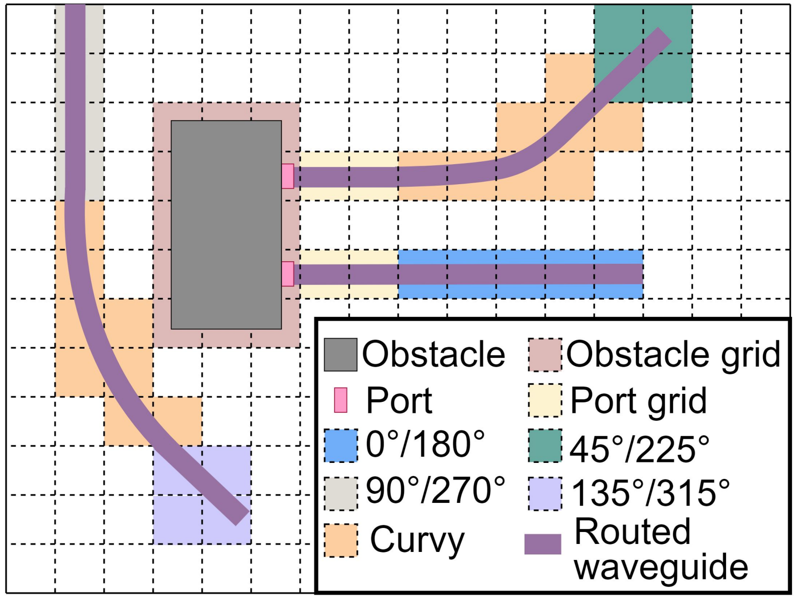
  - $g(n)$: cost from source to current node n
  - $h(n)$: estimated cost from current node to target
- ***How to find next neighbors to explore?***
- Depend on **<u>current path direction</u>**:
  - <u>*Sol*</u>: Extend $\boldsymbol{A}^*$ node state to remember orientation: $(x, y, \textbf{orientation})$
- Depend on **<u>bend radius</u>**:
  - <u>*Sol*</u>: redefine curvy-aware neighbors
  - Locations **adaptively calculated** based on:
    - Radius (r)
    - Grid size (s)

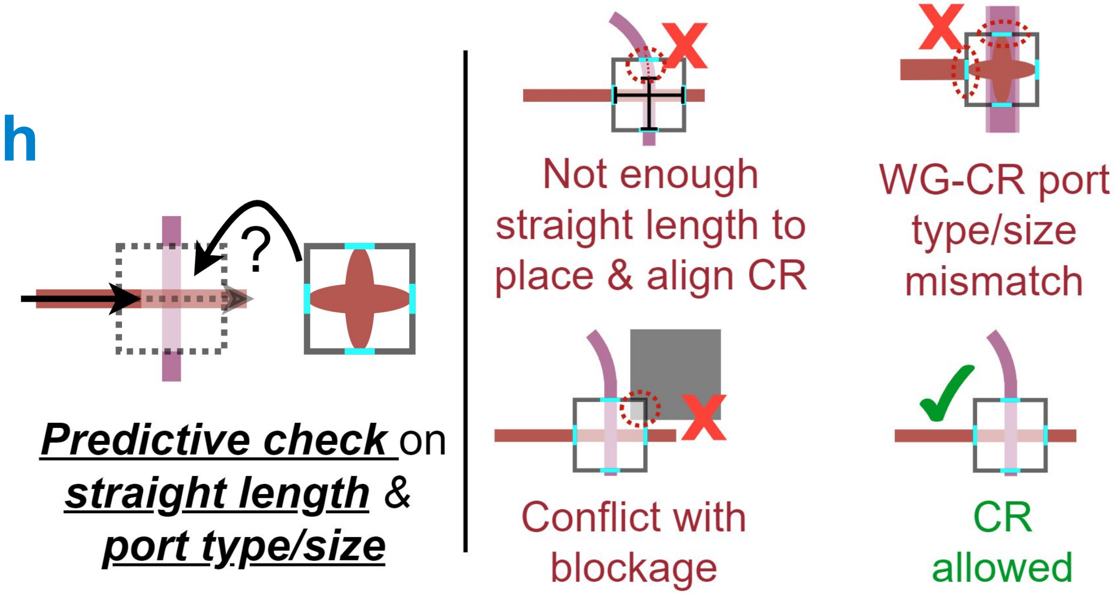


How to Ensure Neighbors' Legality

- Proposed **oriented grid map**:
  - Enable correct connection direction
  - Enable crossing insertion

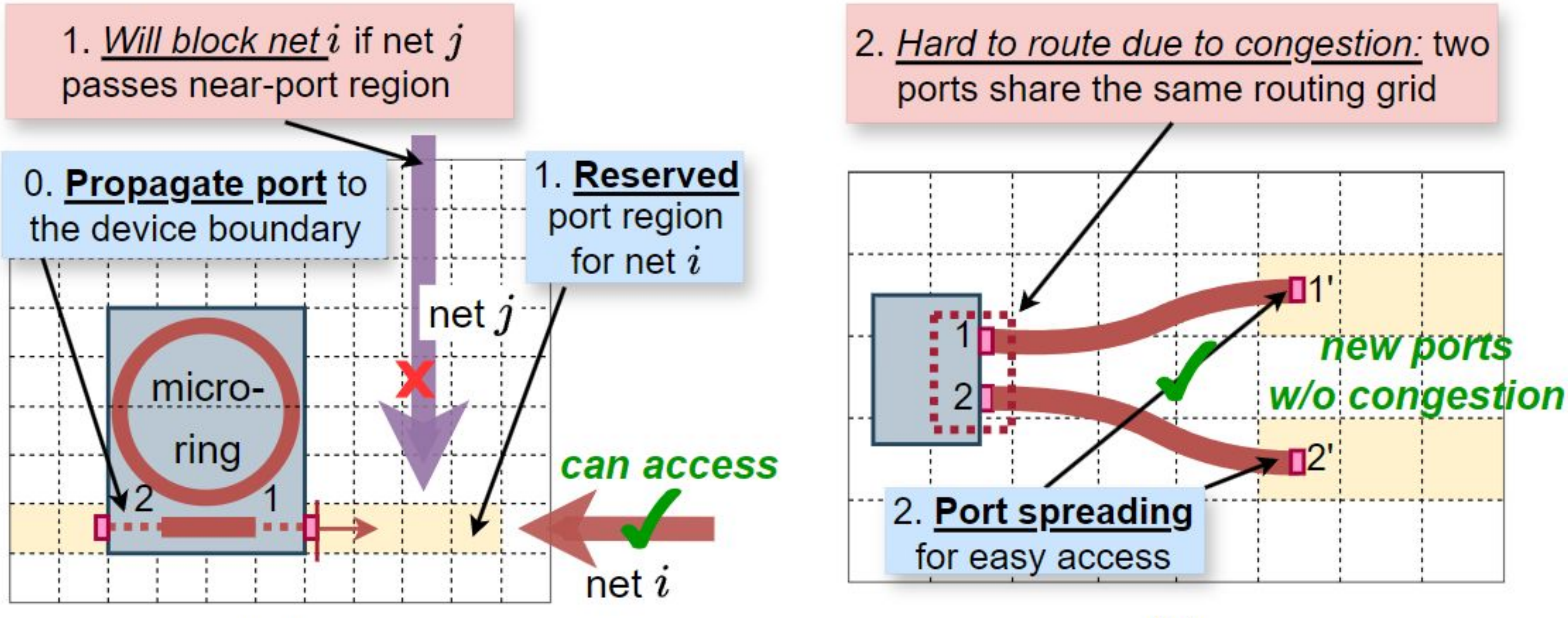
- Handle four neighbors' state:
  - Empty
    - Add neighbor to priority queue
  - Target node
    - Check **its direction before finish**
  - Routed nets
    - See whether to insert a crossing
  - Others
    - Discard illegal neighbor





How to Mitigate Waveguide Conflict on One Layer

- Waveguide Conflict: routing resource competition among waveguide
- Basic idea: Reserve routing resource

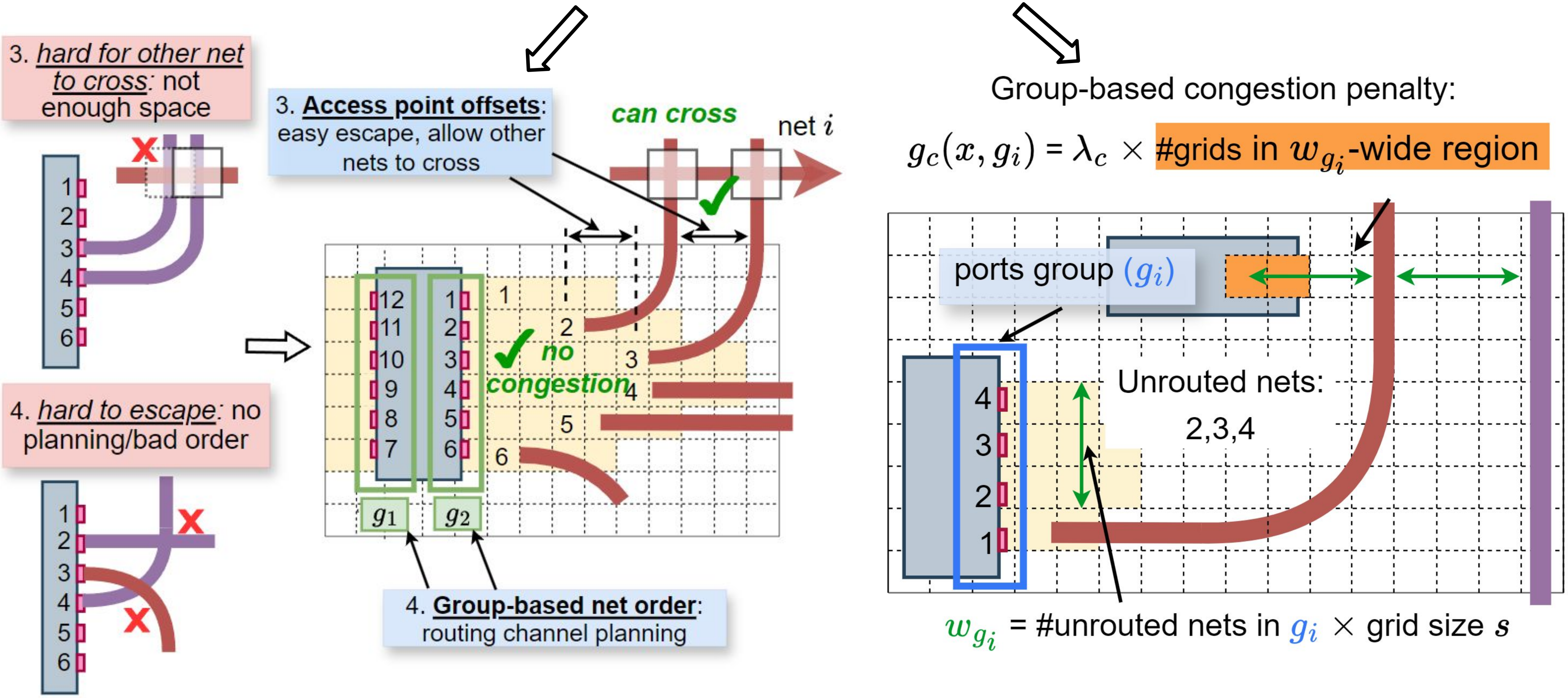


(a) (b)

Predictively reserve space near ports

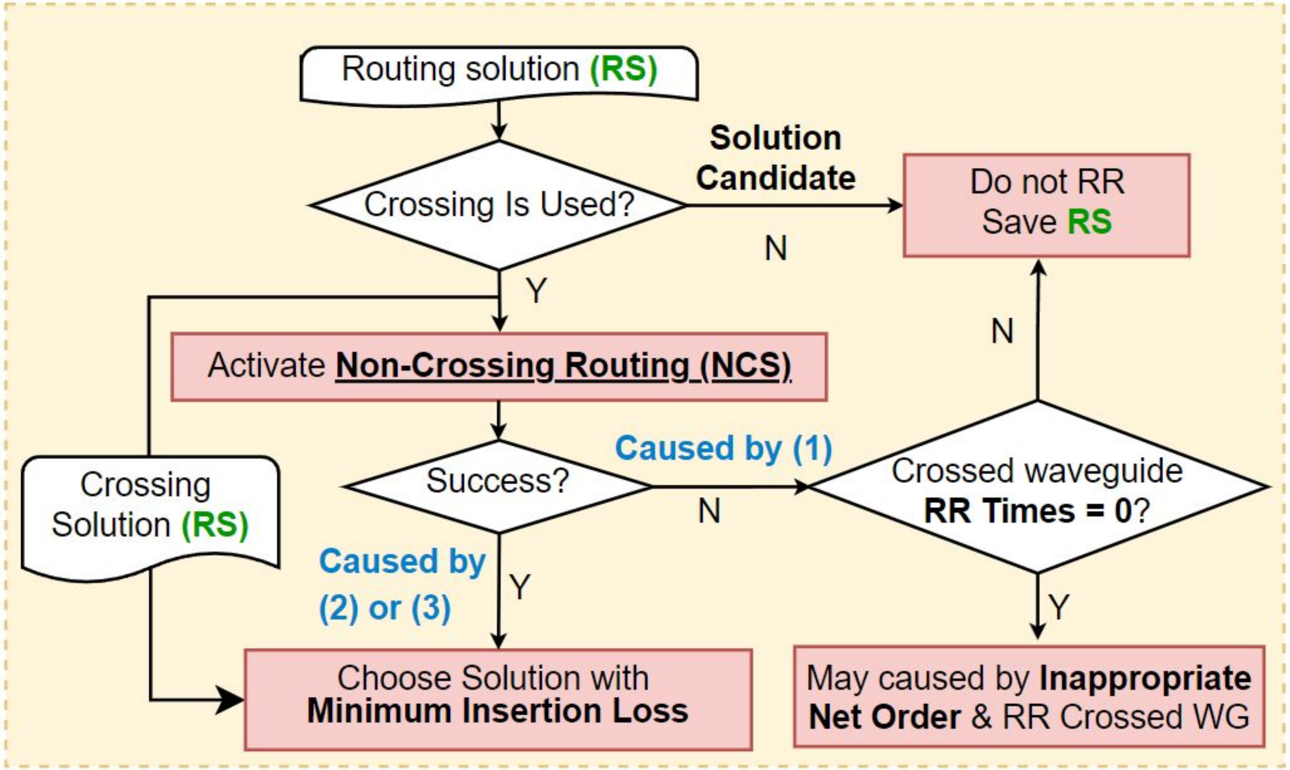
Reserve Resource for Unrouted Nets (Cont'd)

- **Joint** planning for a group of nets
  - Port-group-based routing order & congestion penalty

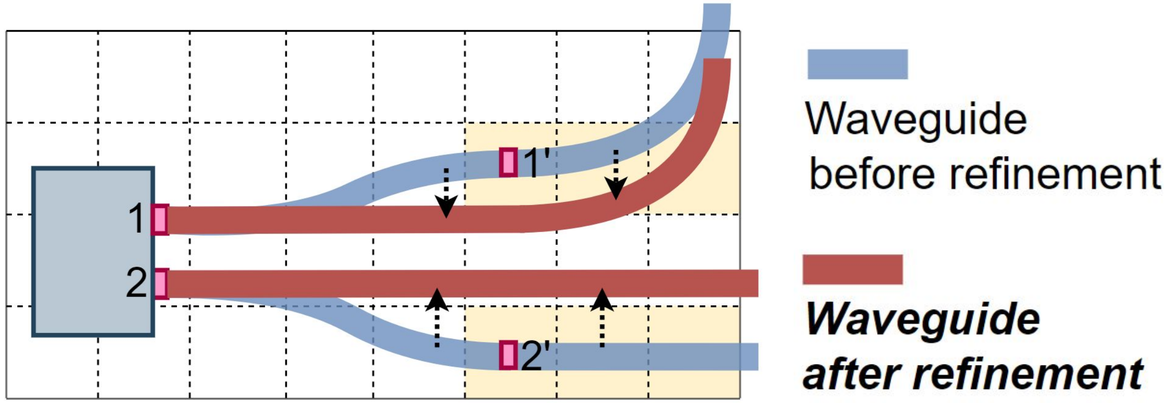


Crossing Optimization & Waveguide Refinement

- Crossing Optimization:
  - Find out why crossings are introduced by **non-crossing routing (NCS)**
    1. Obstructed by other net
    2. High congestion penalty
    3. Long detour w/o crossing
  - Adjust the routing solution accordingly



- Waveguide refinement
  - Revise offset due to grid resolution



Experimental Setup

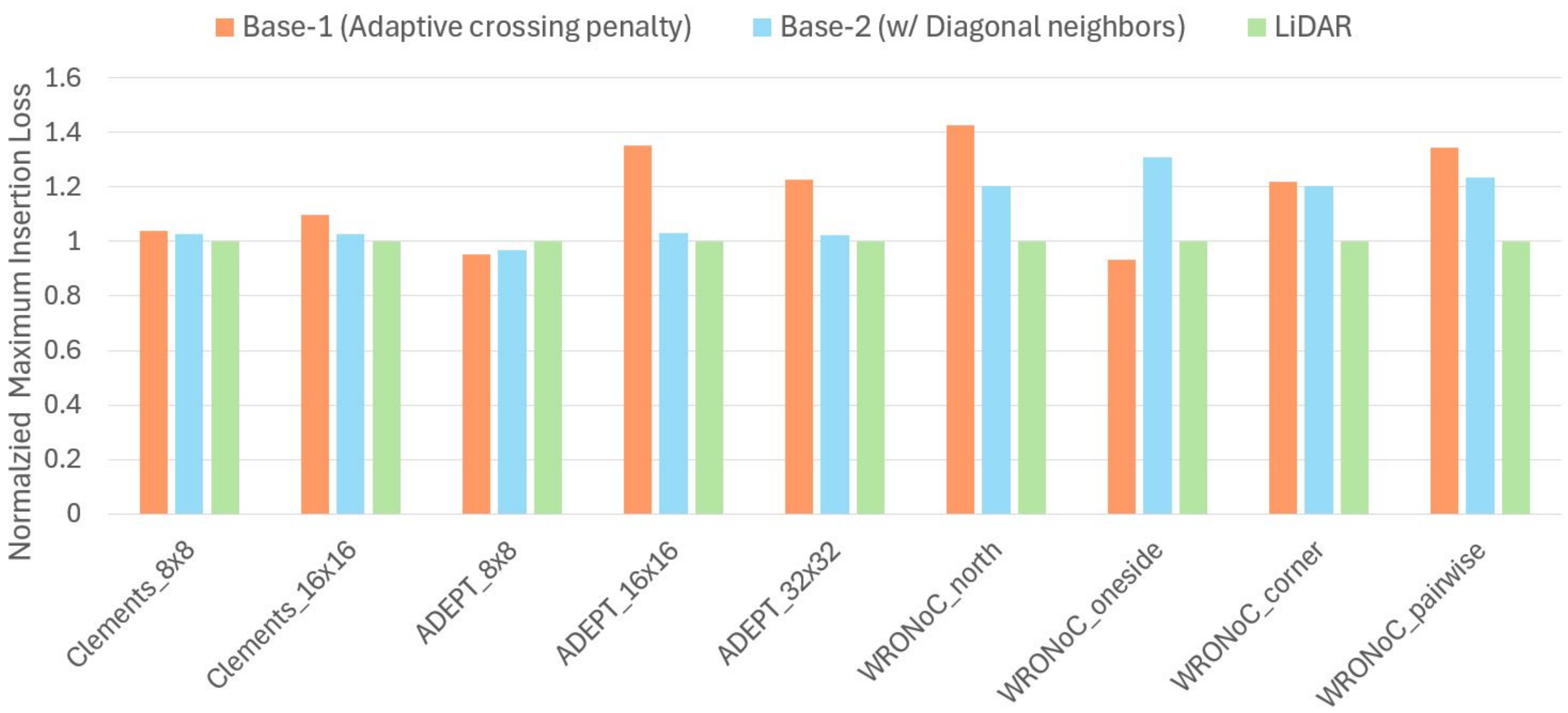
- Machine
  - Intel i5-125600KF 3.7GHz CPU
  - 32 GB RAM
- Benchmark suits
  - Photonic tensor core
    - Clements **(w/o crossing)**
    - ADEPT **(w/ crossing)**
  - Wavelength-routed Optical Network-on-Chip (WRONOC)
- Device insertion loss

			Y-branch	MZI	MMI
1.5 dB/cm	0.005 dB	0.52 dB	0.3 dB	1.2 dB	0.1 dB

- Routers for comparison
  - Base-1: Proton [ICCAD'13] with rip-up & reroute
  - Base-2: Proton [ICCAD'13] with diagonal neighbors

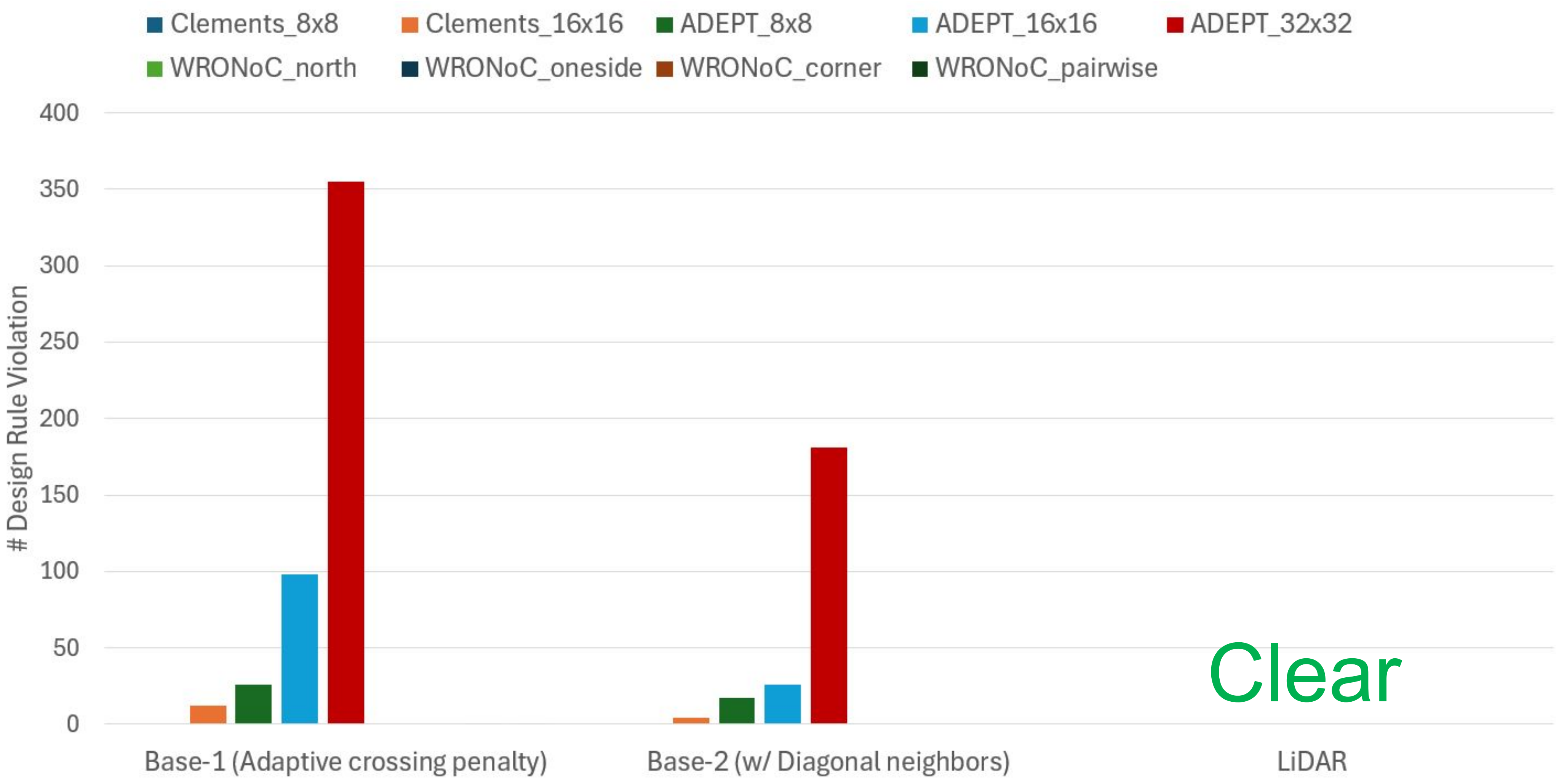
Maximum Insertion Loss Comparison

- LiDAR outperforms other routers
  - 14% better than Base-1 5% better than Base-2



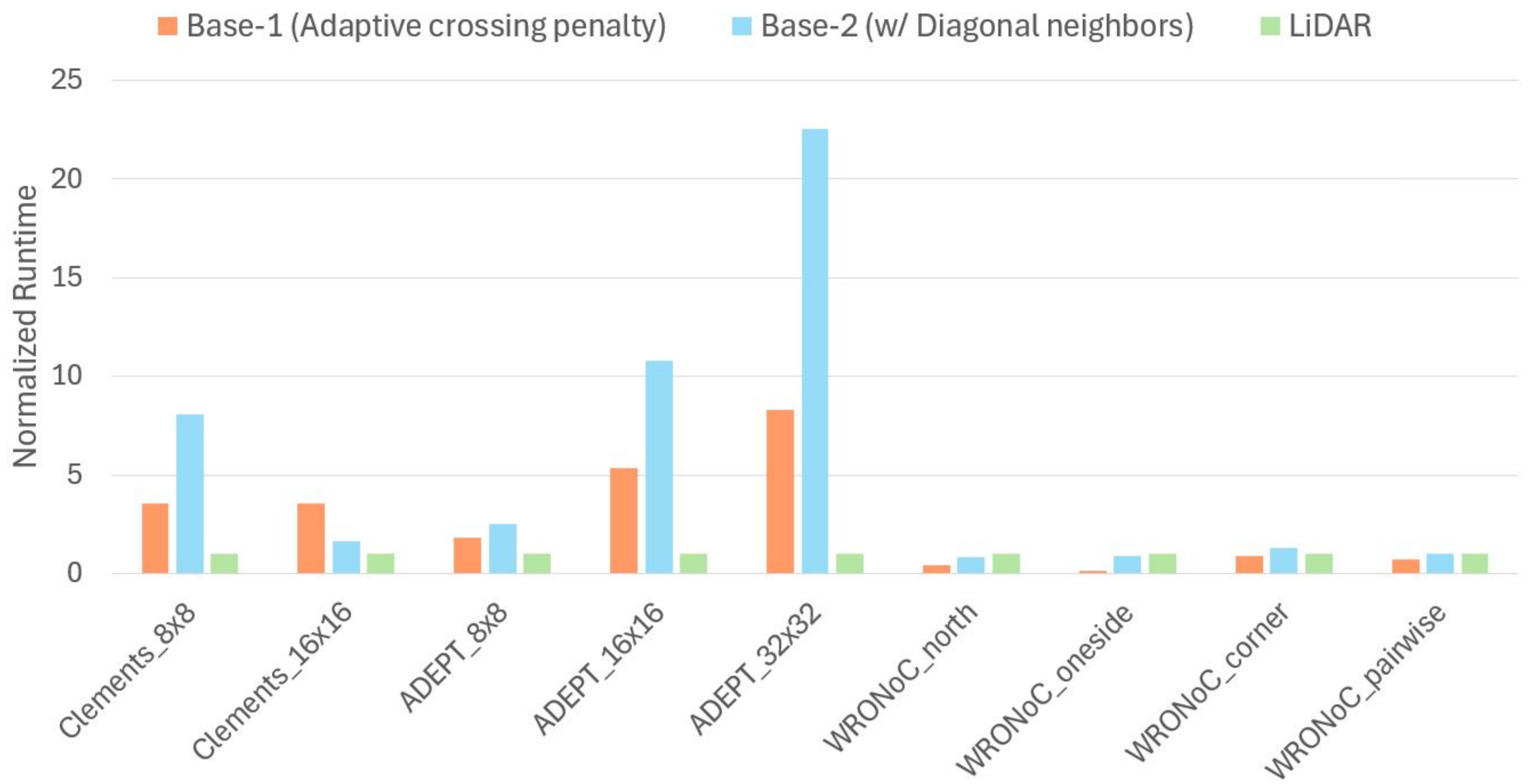
Design Rule Violation Comparison

- LiDAR generates DRV-free solutions on all benchmarks



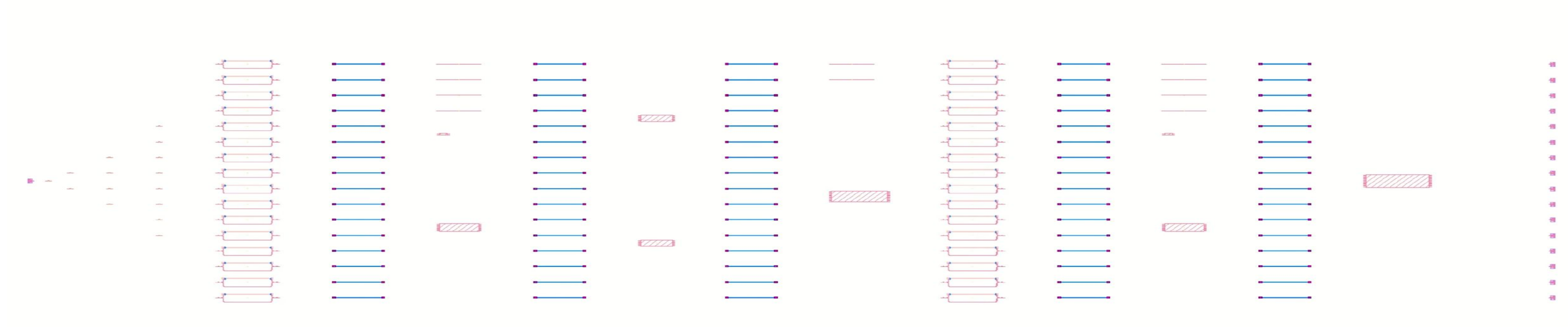
Runtime Comparison

- LiDAR is 2.75× faster than Base-1 and is 5.51× faster than Base-2

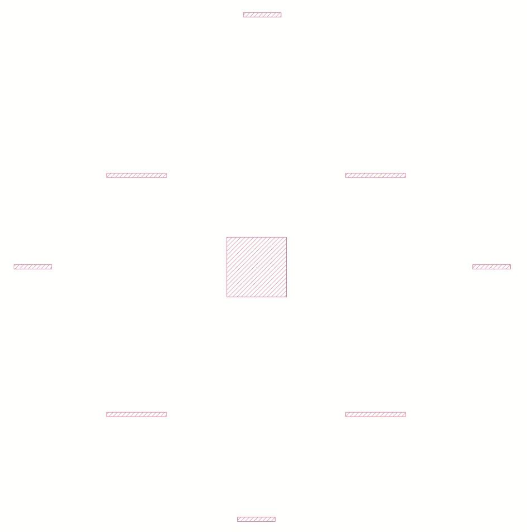


Animation

- ADEPT_16x16

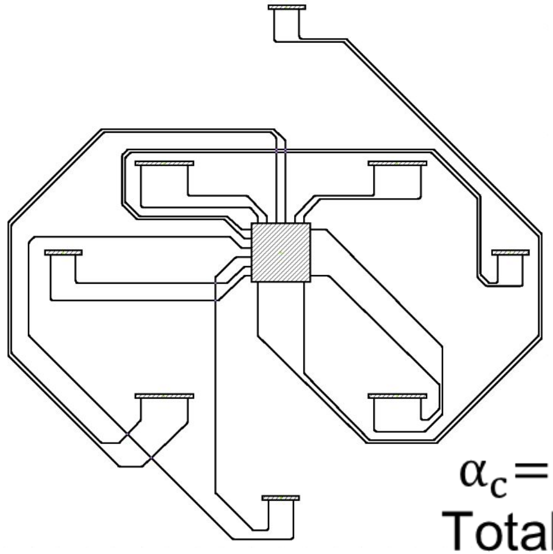


- WRONoC_north



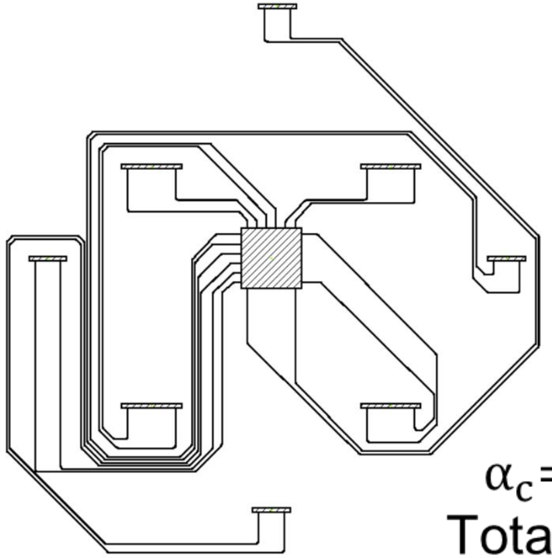
Evaluation of Group-based Congestion Penalty (GCP) and Crossing Parameter α_c

- $\alpha_c\ (dB)$: Insertion loss per crossing:
  - $\boldsymbol{\alpha_c}\uparrow \rightarrow$ **#CR**$\downarrow$
- GCP: Improve routability



$\alpha_c = 0.3$
Total $\#CR = 8$

Metrics	High Crossing Cost $\alpha_c = 1$		Low Crossing Cost $\alpha_c = 0.3$	
	w/o GCP	**LiDAR**	w/o GCP	**LiDAR**
#CR	6	0	5	5
WL (mm)	20.72	31.11	25.11	26.04
$IL_{max}\downarrow$	15.21	10.78	7.18	7.31
DRV	0	0	1	0
Time (s)	129	73	261	197



$\alpha_c = 1$
Total $\#CR = 0$

Conclusion and Future Direction

- Conclusion
  - **Curvy-Aware *A** Search**: parametric neighbors and crossing insertion
  - **Routability optimization**: access port assignment & group-based routing
  - **Crossing optimization**: local rip-up & reroute scheme
  - **>14% better insertion loss and 2.75× speedup** than prior approach
  - **DRV-free routing solution**

- Future direction
  - Honor more routing constraints
  - Algorithm acceleration by C\C++
  - Large-scale optical interconnect network

Thank you!