IIBLAST: Speeding Up Commercial FPGA Routing by Decoupling and Mitigating the Intra-CLB Bottleneck

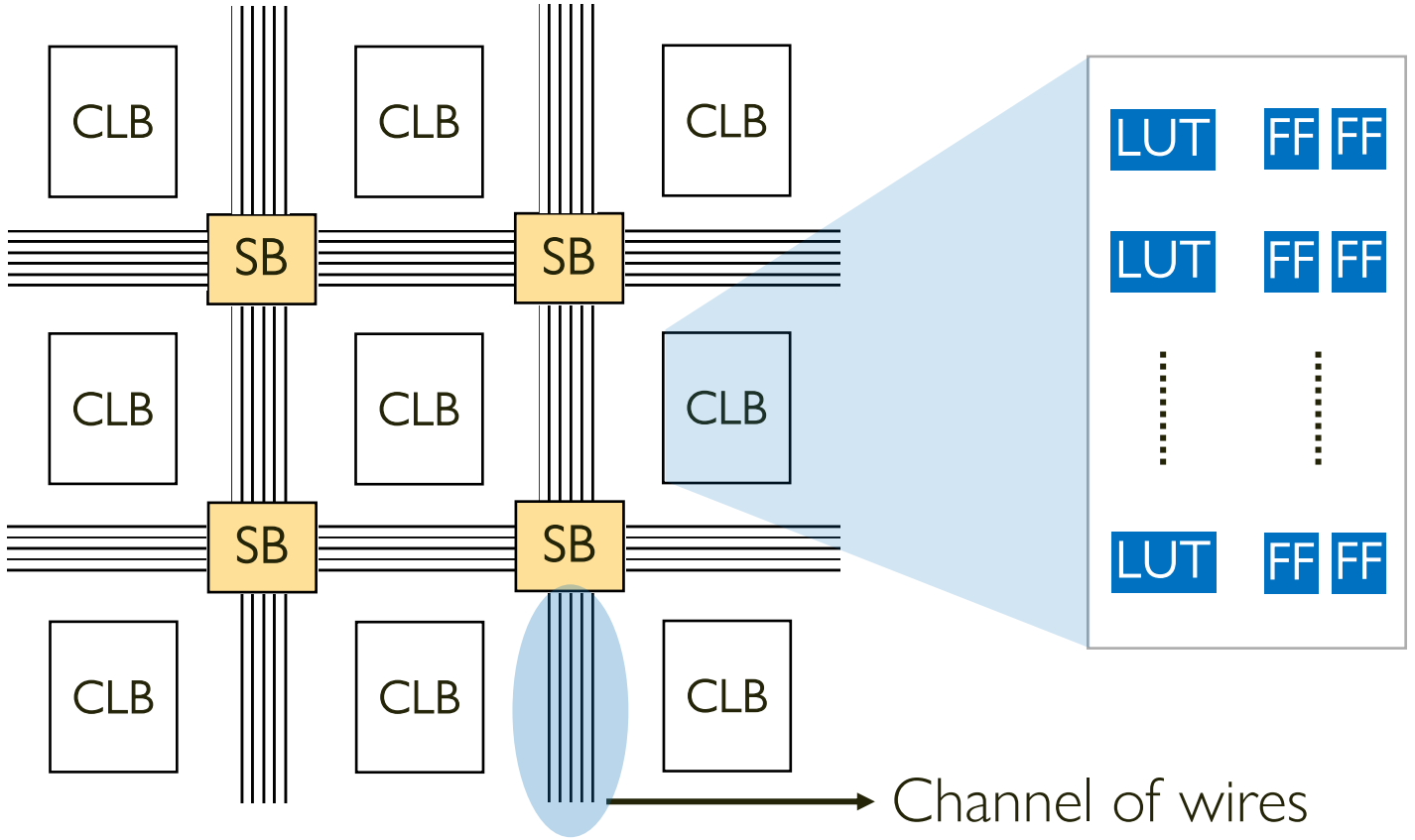
Shashwat Shrivastava, Stefan Nikolić, Chirag Ravishankar*, Dinesh Gaitonde*, Mirjana Stojilović

EPFL, *AMD

International Conference on Computer-Aided Design (ICCAD) 2023

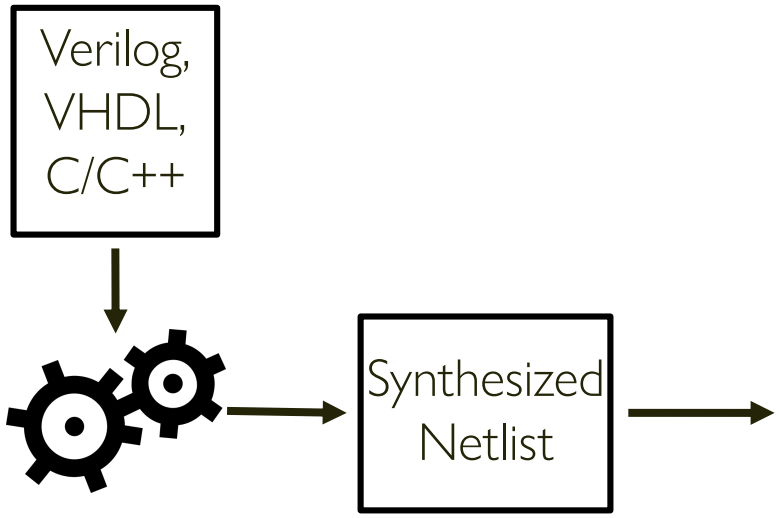
FPGA

- Reprogrammable
- Faster design to market time than ASICs

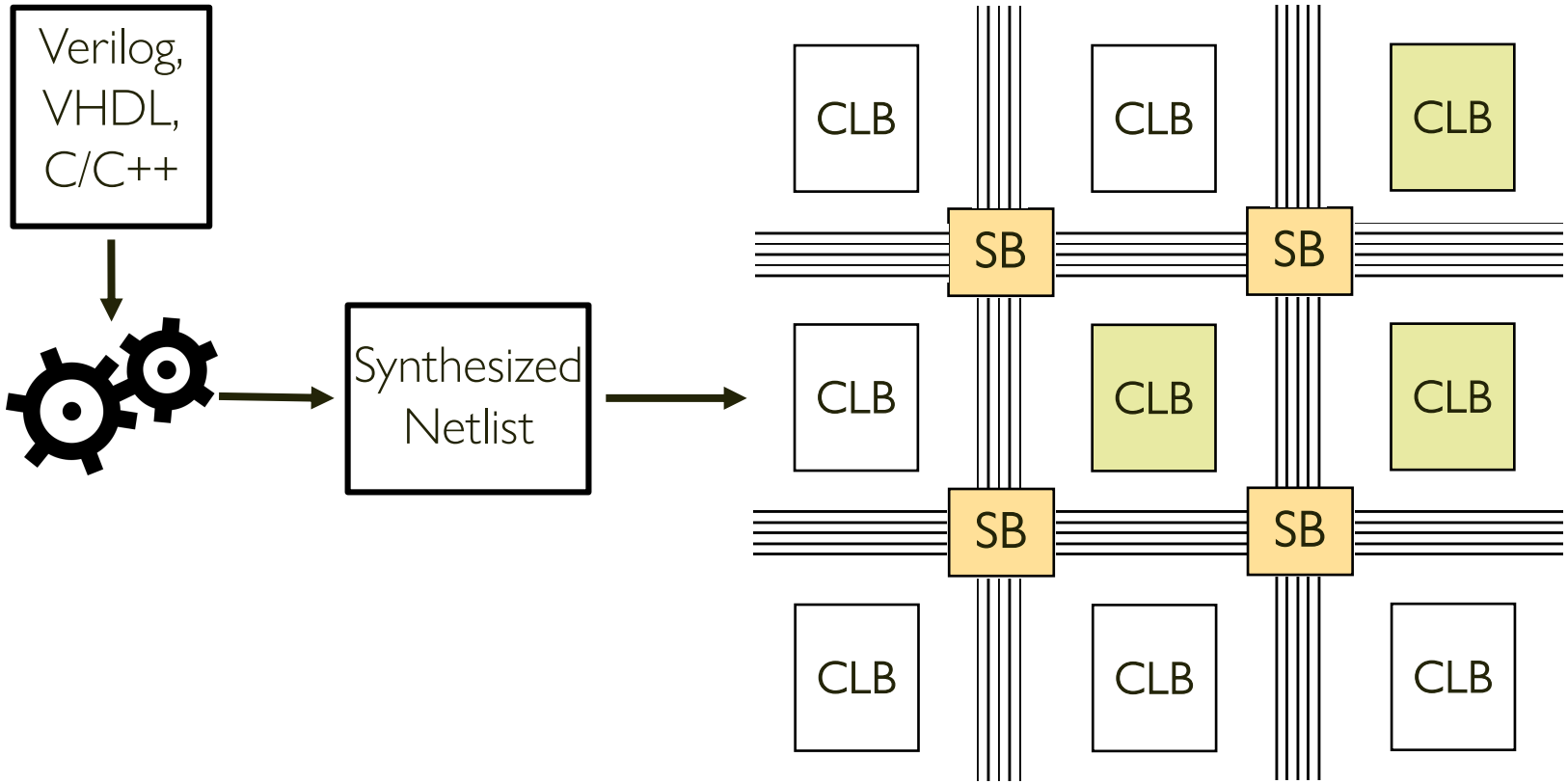


FPGA Compilation Flow

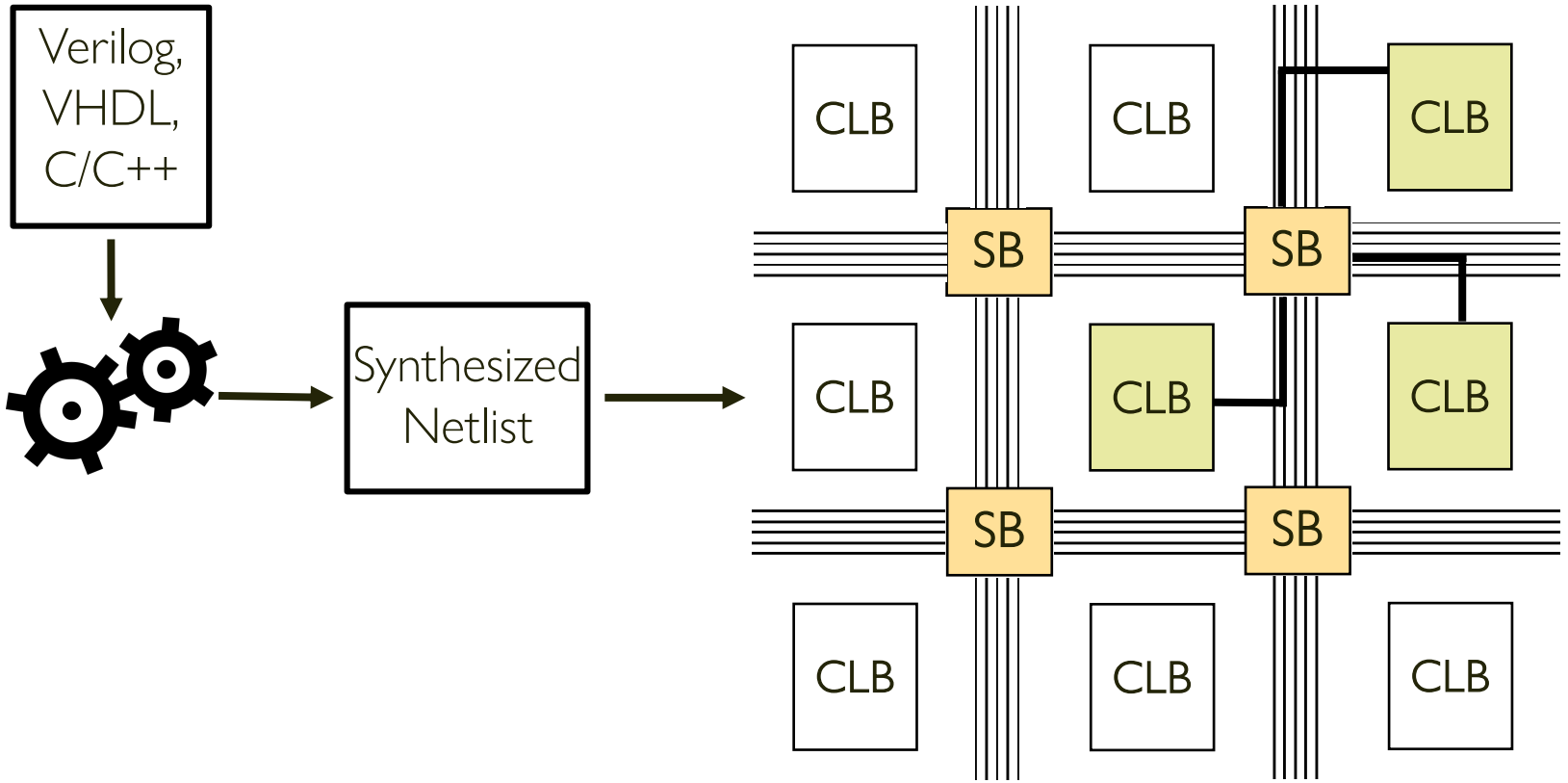
FPGA Compilation Flow



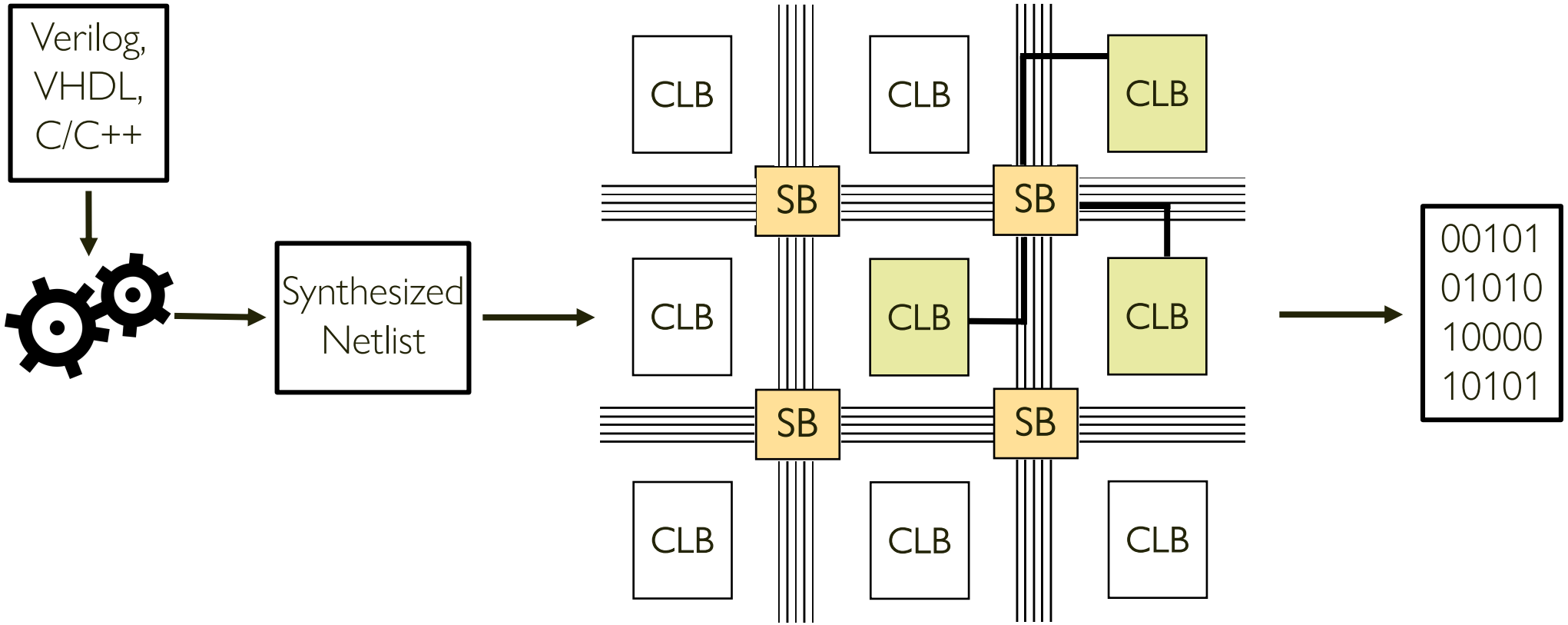
FPGA Compilation Flow



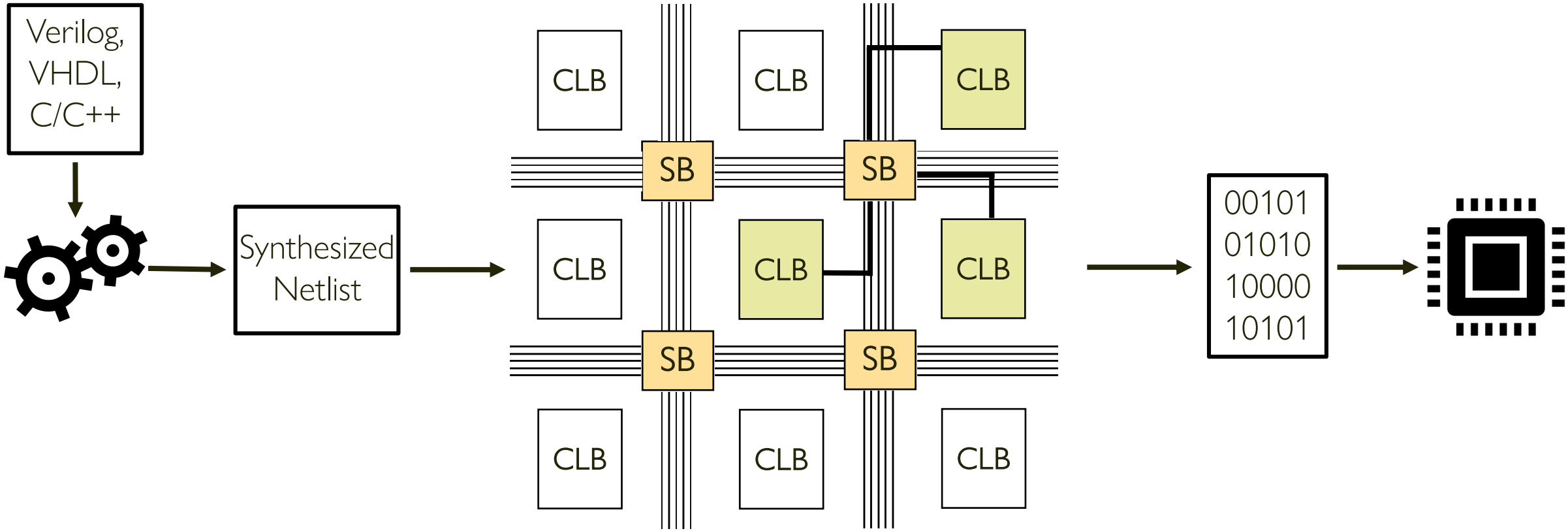
FPGA Compilation Flow



FPGA Compilation Flow



FPGA Compilation Flow

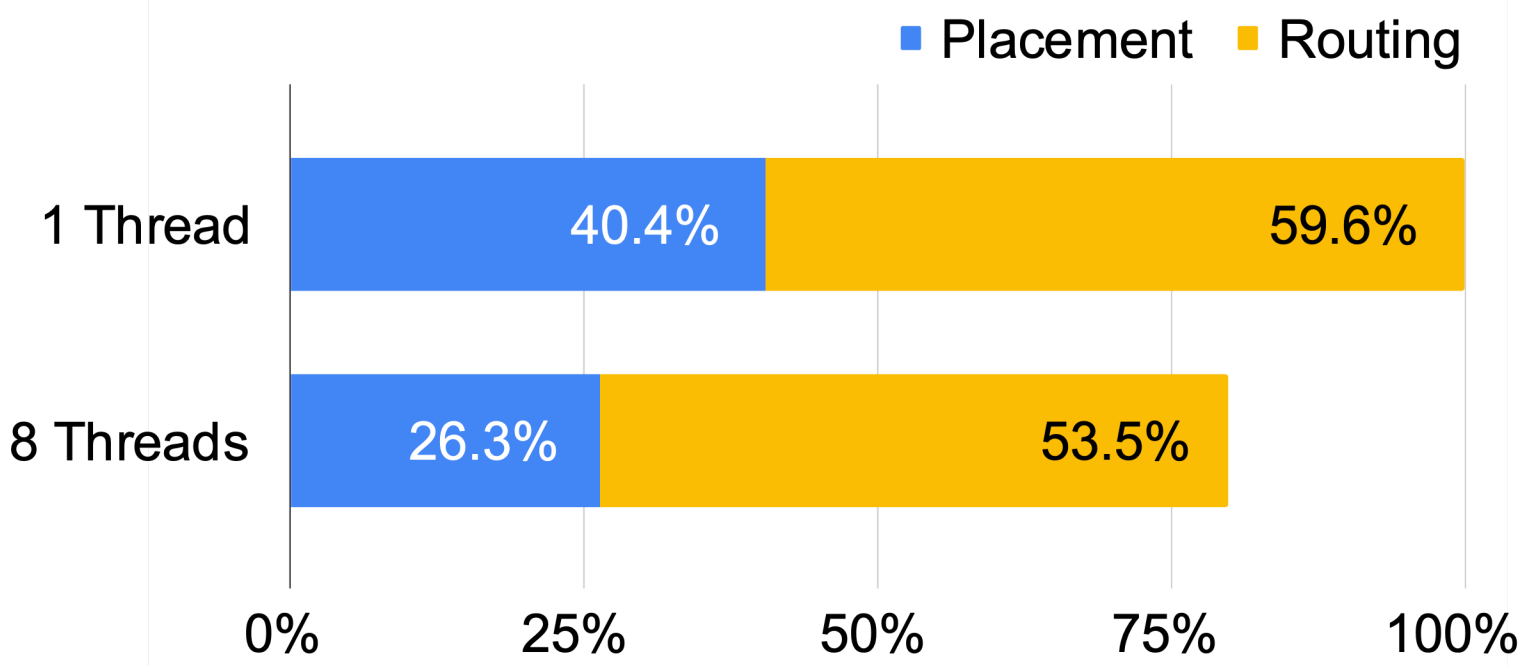


Slow FPGA Compilation

- The FPGA compilation flow is slow
  - Latest FPGAs have ~18M logic cells
  - A design with only a million cells → a long time to compile
  - Design space exploration → multiple iterations

Routing vs Placement Compilation Time

18 commercial designs ran in Vivado



Routing inherently non-parallelizable and a major runtime bottleneck

Why Speed Up Routing?

- To speed up FPGA compilation time

- For designs targeting ASIC emulation, prototyping, driver development, …
  - Designs are extremely large (millions of cells) → partitioned onto multiple FPGAs
  - Even partitioned designs are relatively large → making routing slow
  - Circuit's maximum clock frequency → a secondary objective

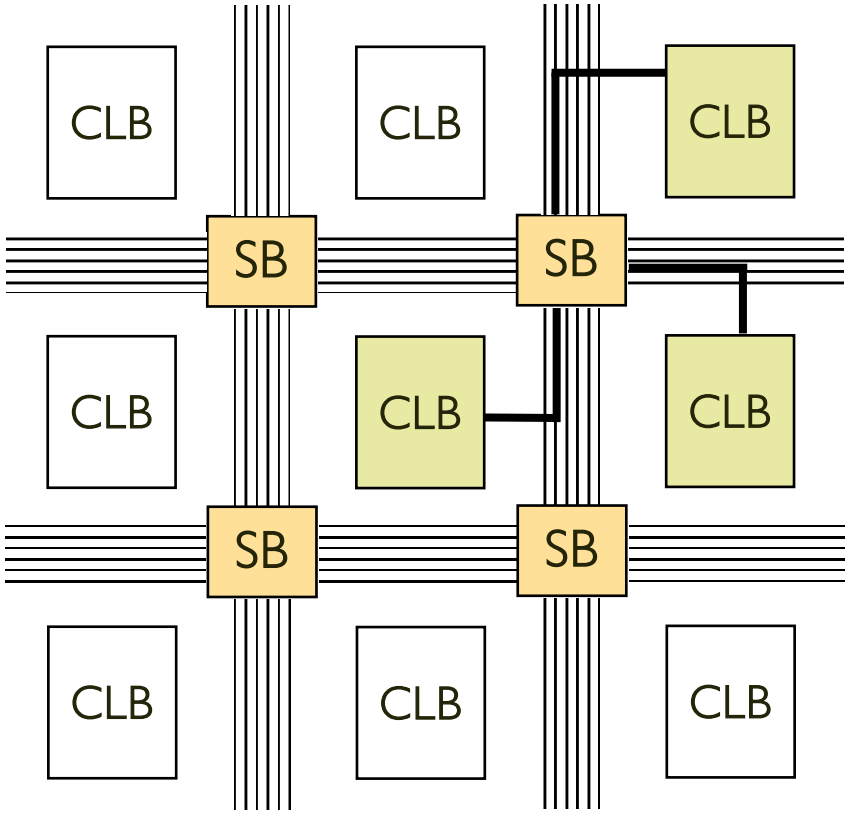
Routing runtime becomes a primary optimization objective

Outline

- Introduction
  - FPGA
  - FPGA Compilation Flow
  - Routing
- Multi-Stage Router
- Conclusion

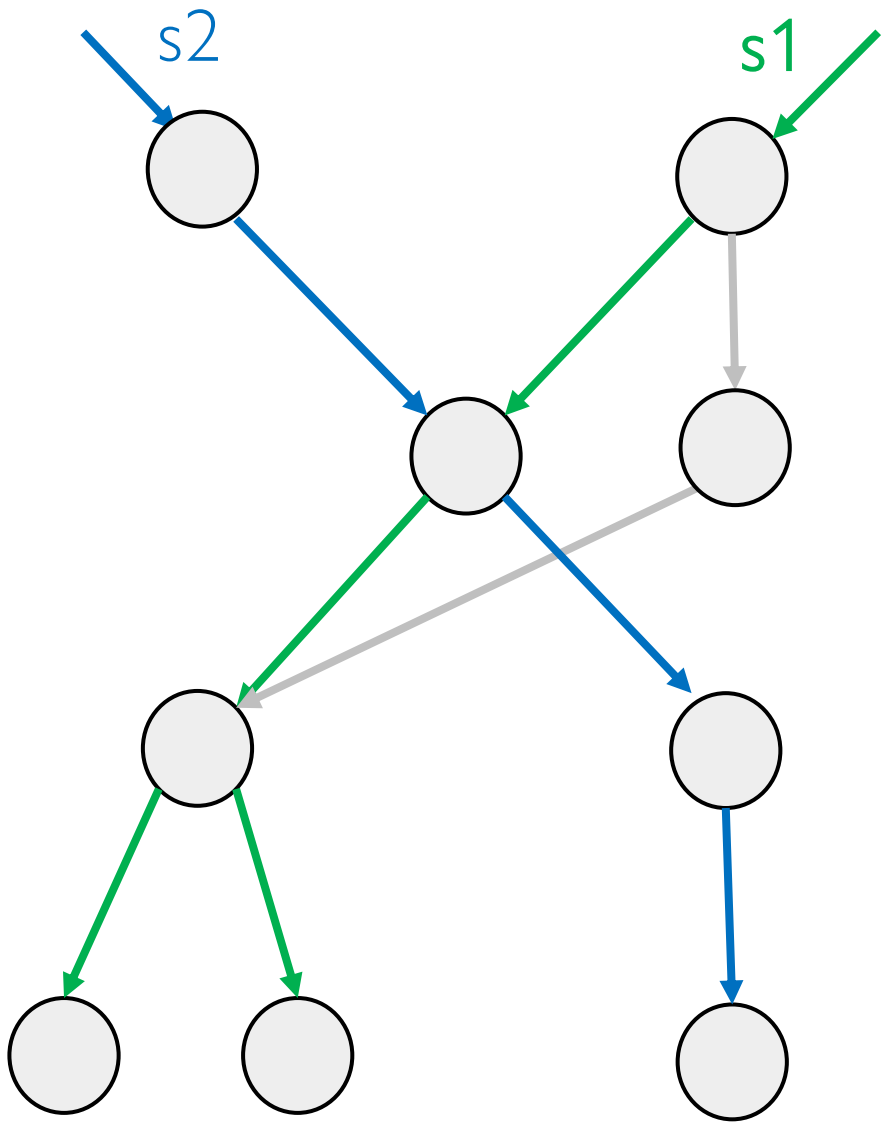
Routing

- **Aim:** Find shortest paths between every signals' source and its destinations
- **Given:** A netlist, placement, and a routing graph
- **Constraint:** Paths corresponding to different signals must be disjoint



Routing Algorithm: PathFinder

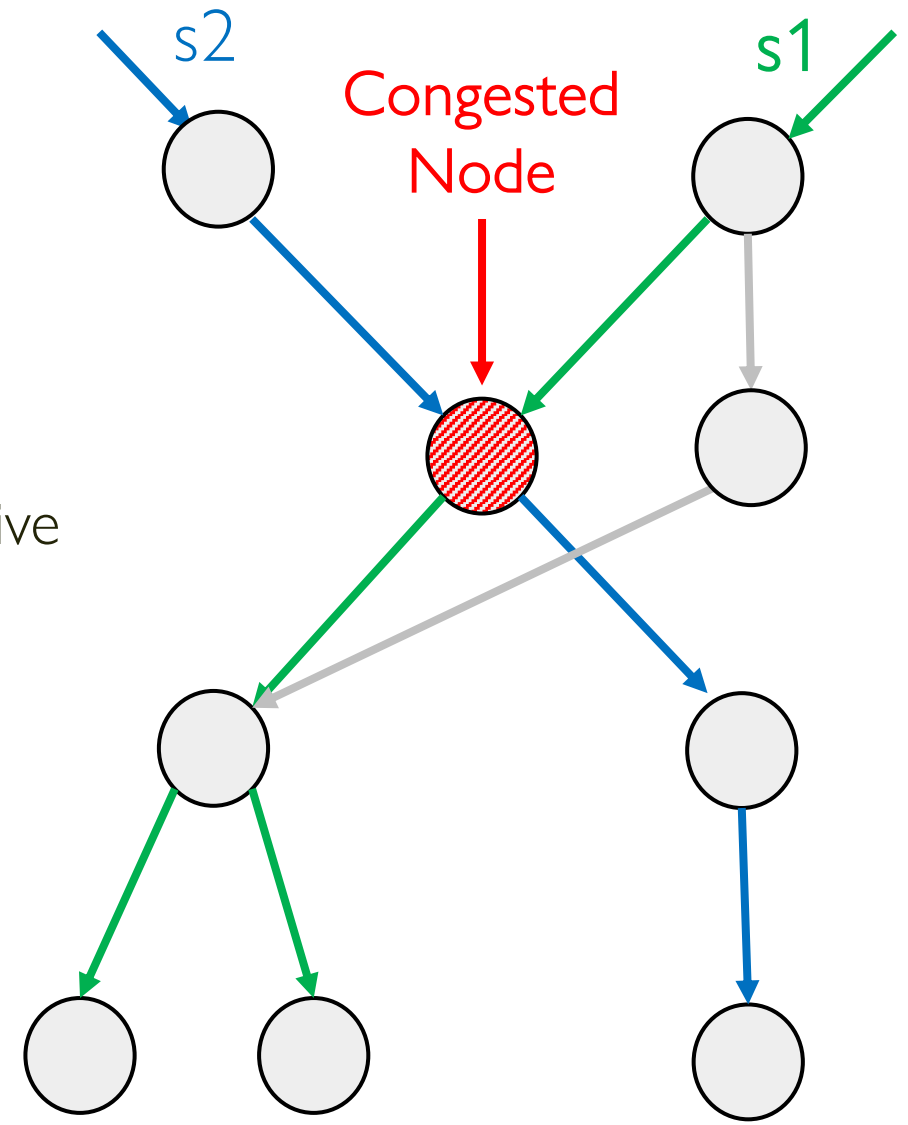
- Iterative algorithm
- In each iteration
  - Route the signals sequentially
  - Each signal finds the shortest path



McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In Proceedings of the 1995 ACM third International Symposium on FPGAs.

Routing Algorithm: PathFinder

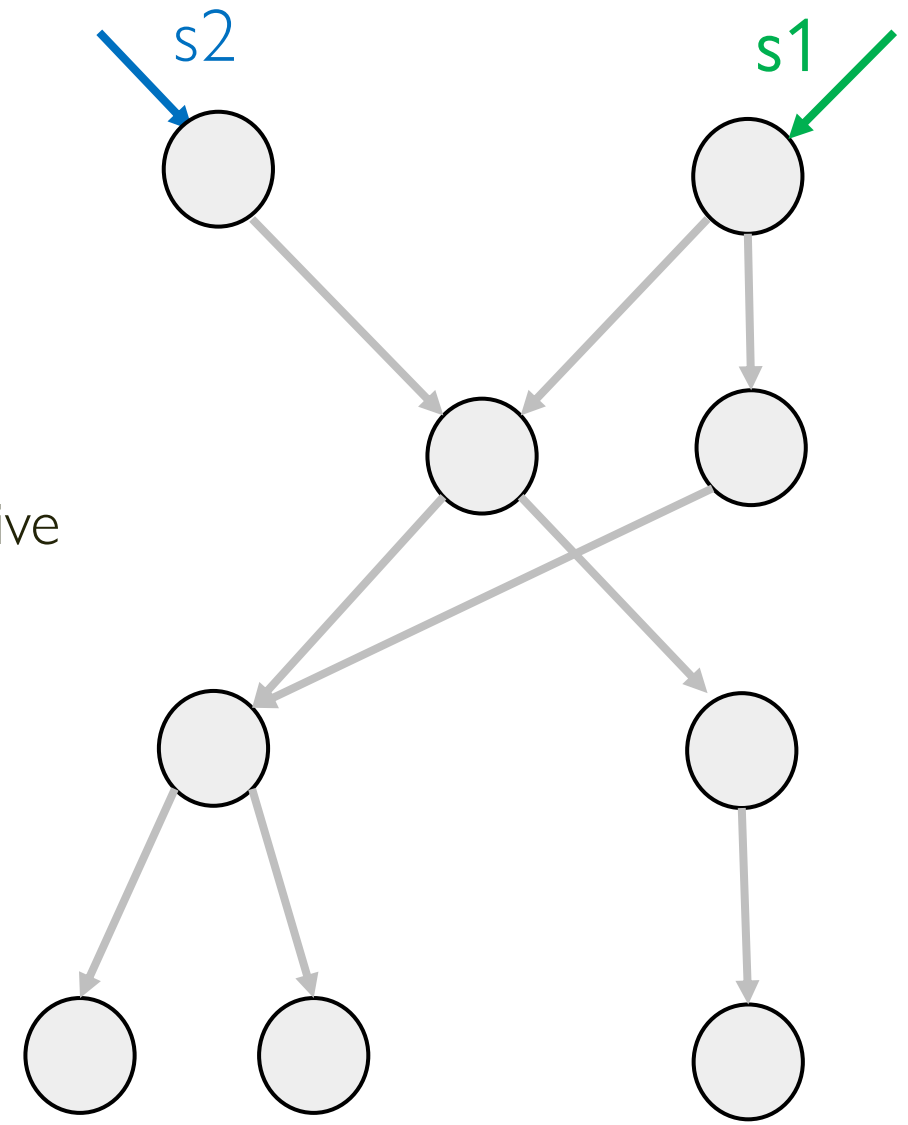
- Iterative algorithm
- In each iteration
  - Route the signals sequentially
  - Each signal finds the shortest path
  - Signals can use the same nodes → congestion
  - To remove congestion → make the node expensive



McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In Proceedings of the 1995 ACM third International Symposium on FPGAs.

Routing Algorithm: PathFinder

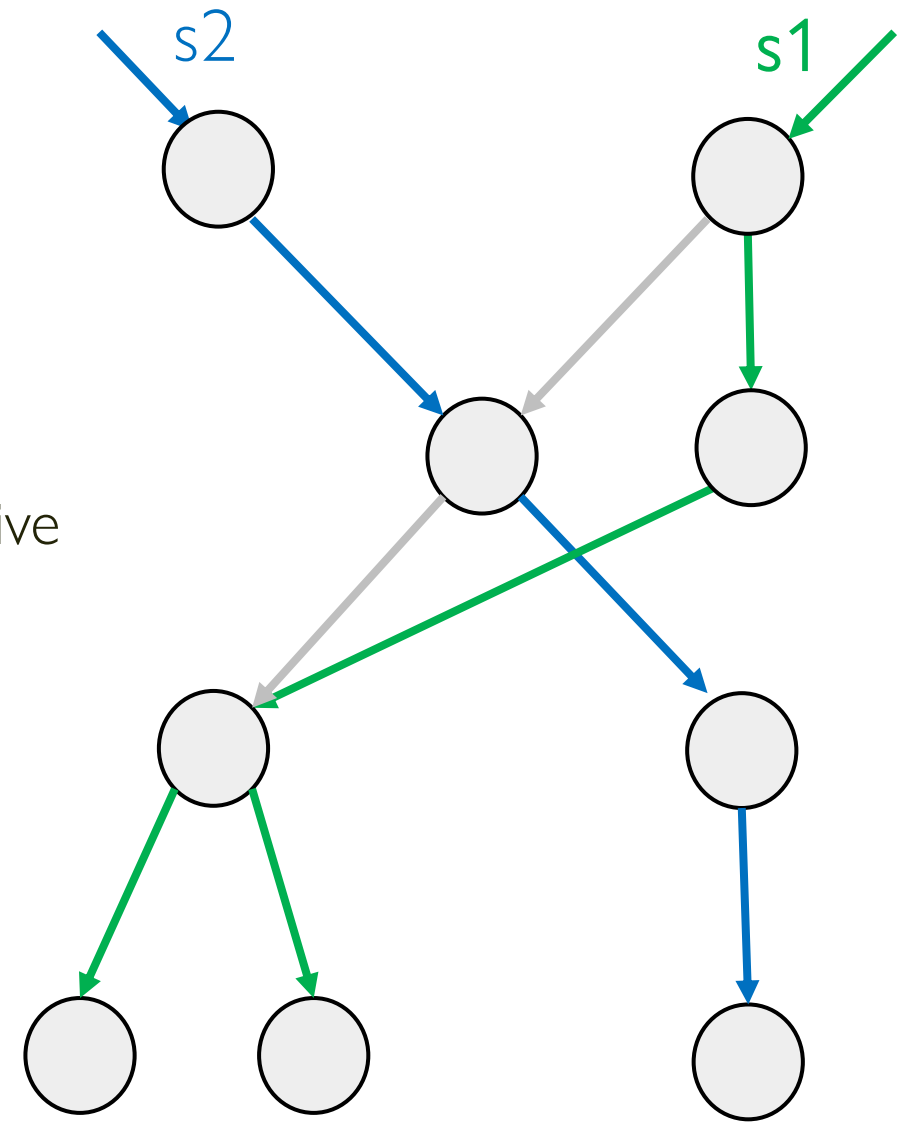
- Iterative algorithm
- In each iteration
  - Route the signals sequentially
  - Each signal finds the shortest path
  - Signals can use the same nodes → congestion
  - To remove congestion → make the node expensive
- At the end of each iteration
  - Rip up signals that use the congested node



McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In Proceedings of the 1995 ACM third International Symposium on FPGAs.

Routing Algorithm: PathFinder

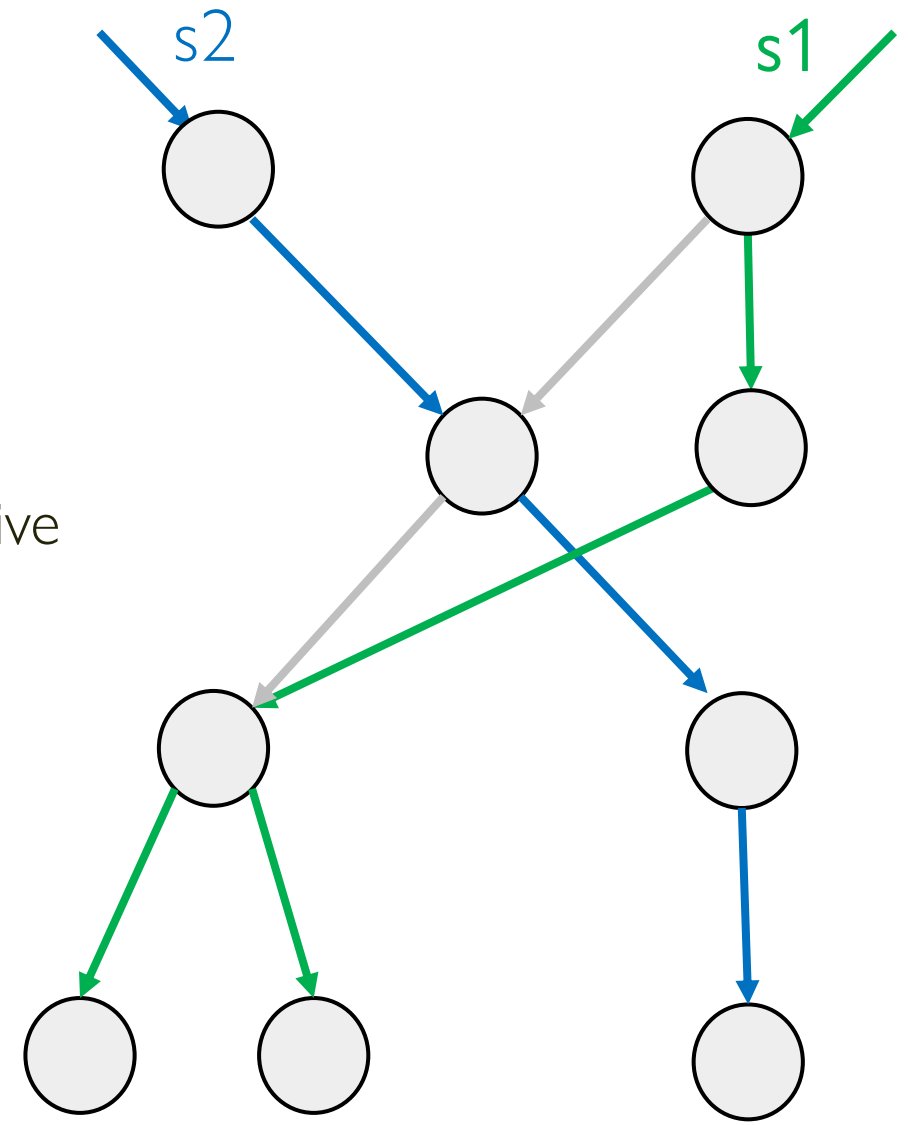
- Iterative algorithm
- In each iteration
  - Route the signals sequentially
  - Each signal finds the shortest path
  - Signals can use the same nodes → congestion
  - To remove congestion → make the node expensive
- At the end of each iteration
  - Rip up signals that use the congested node
- In the next iteration, reroute ripped up signal



McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In Proceedings of the 1995 ACM third International Symposium on FPGAs.

Routing Algorithm: PathFinder

- Iterative algorithm
- In each iteration
  - Route the signals sequentially
  - Each signal finds the shortest path
  - Signals can use the same nodes → congestion
  - To remove congestion → make the node expensive
- At the end of each iteration
  - Rip up signals that use the congested node
- In the next iteration, reroute ripped up signal
- Terminates when all congestion is resolved



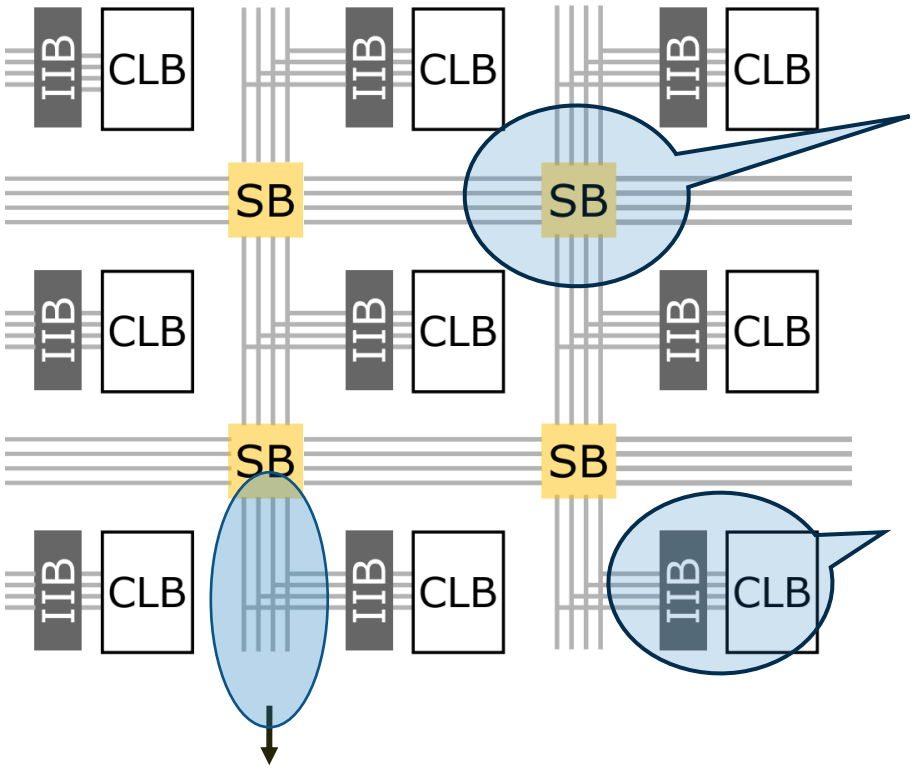
McMurchie L, Ebeling C. PathFinder: A negotiation-based performance-driven router for FPGAs. In Proceedings of the 1995 ACM third International Symposium on FPGAs.

PathFinder's Runtime Analysis

- Depends broadly on two factors
- **Runtime of each iteration:** Time to find the shortest path for each signal
  - Algorithm → Dijkstra → min heap data structure
- **Total iterations:** Congestion levels of the design
  - Designs with high congestion → repeated re-routing of signals → high runtime

```
for each iteration in max_allowed_iterations
    for each signal in all_signals
        call find_shortest_path on a signal
    if no congestion remains
        break
```

Routing Architecture



Inter-CLB routing → Switch Block (SB)

- Connections between channel wires

Intra-CLB routing → Input Interconnect Block (IIB)

- Connections between channel wires and LUT and FF pins

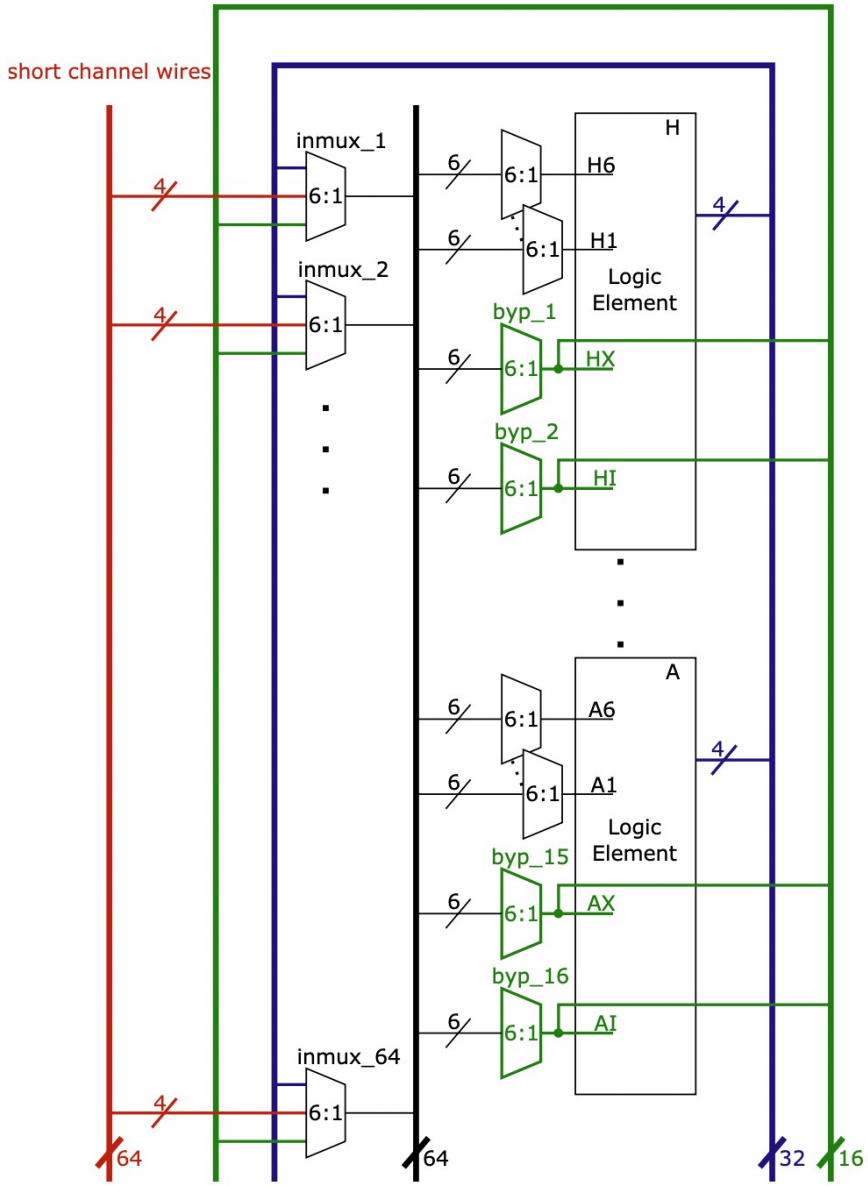
Intra-CLB Routing: Commercial IIB

- 2 layers of 6:1 mux → IIB-6-2L
  - Outer layer → 4 wires connect to each mux
  - Inner layer → to further improve the connectivity

\+ Compact and fast

\- Limited routing flexibility

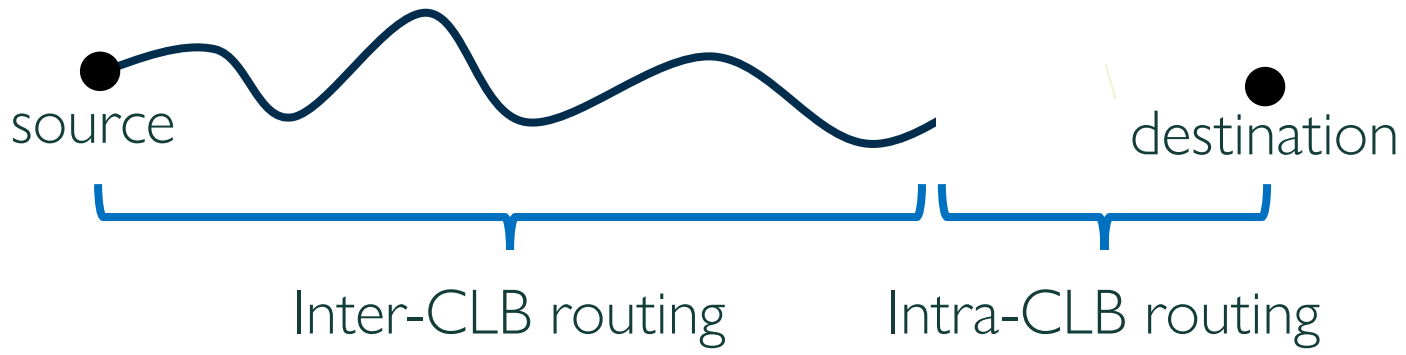
But approximates a full crossbar with limited resources



Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

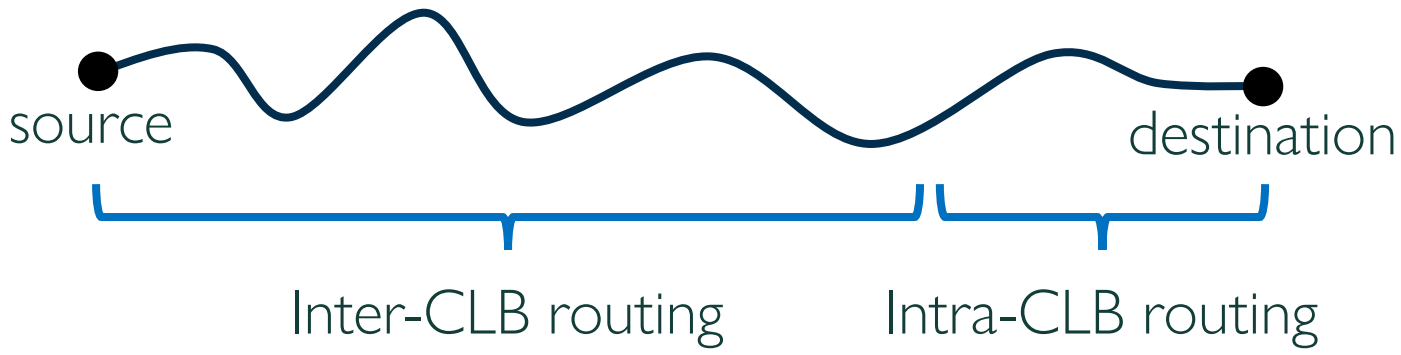
- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region



Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

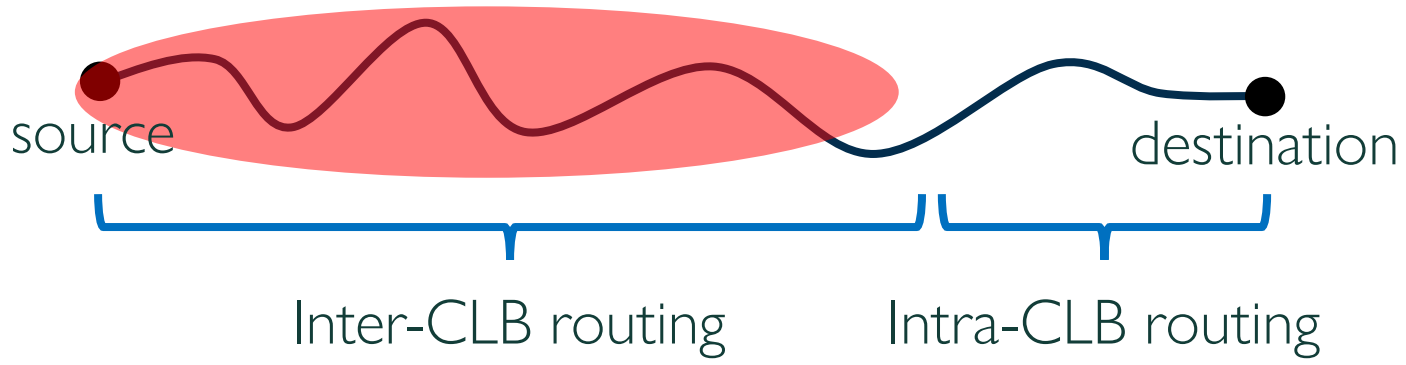
- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region



Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up



Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up

Intra-CLB Routing: Commercial IIB

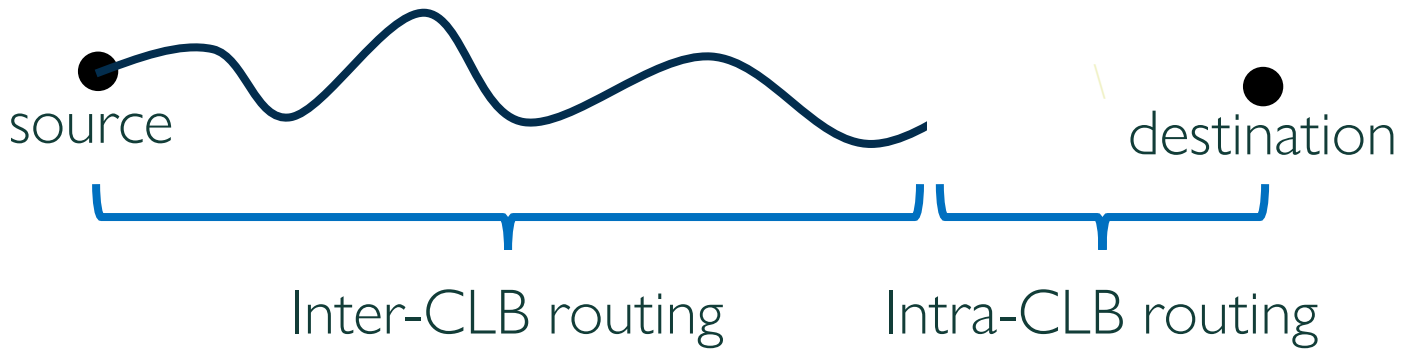
- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant

Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant

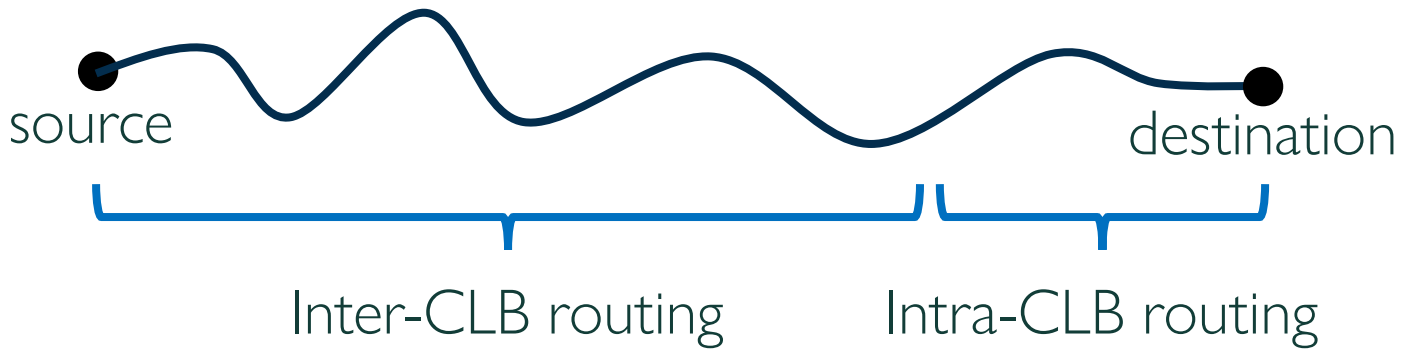


The same repeats in subsequent iterations

Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant

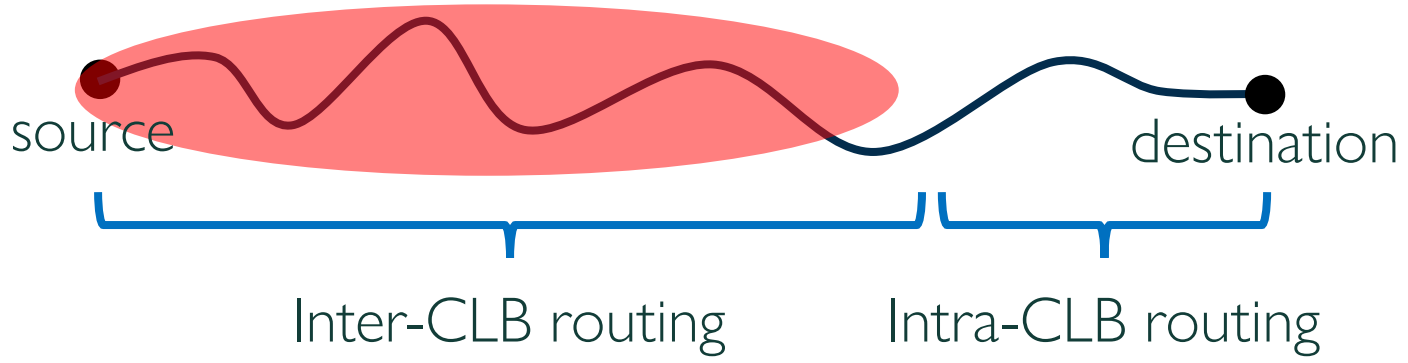


The same repeats in subsequent iterations

Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant



The same repeats in subsequent iterations

Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant

The same repeats in subsequent iterations

Intra-CLB Routing: Commercial IIB

- Limited routing flexibility IIBs → neglected in academic research

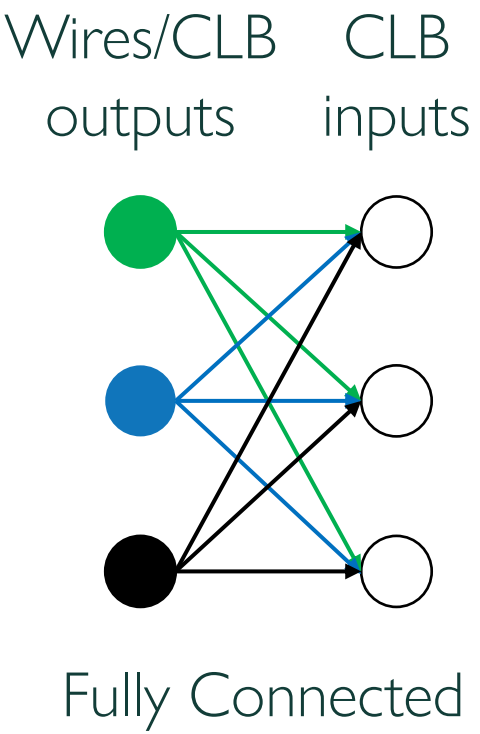
- During routing, for every re-routing of a signal
  - Signal is routed in inter and intra-CLB region
  - For congestion in inter-CLB routing → signal is ripped up
  - Previous routing inside IIB → redundant
  - Redundant expansions inside IIB → accumulation of expansions → increase in runtime

Impact of Commercial IIB

- To quantify the impact of routing inside IIB
  - Routing on the commercial IIB → IIB-6-2L
  - Routing on full IIB
    - Fully connected → all channel wires and all output pins connect to each LUT and FF pin

- Measuring metric
  - Runtime
  - Heap pushes
  - Heap pops

} Machine agnostic



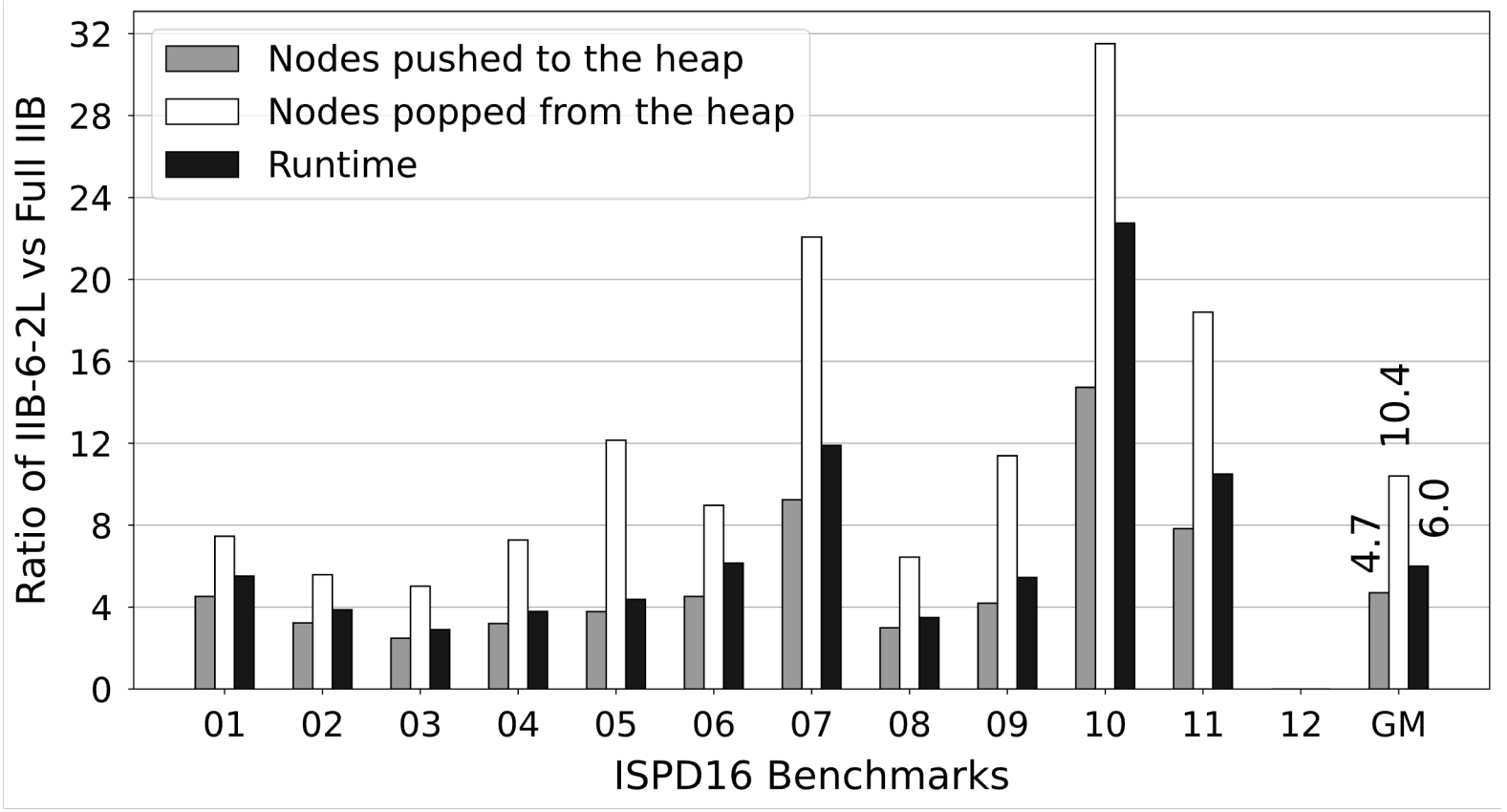
Fully Connected

Experimental Setup

- Versatile Place and Route (VPR) tool for routing → latest version of VTR8
- Target architecture: Closely resembling AMD UltraScale (without timing information)
- ISPD16 benchmarks → routability driven placements
- UTPlaceF placements
- Intel(R) Xeon(R) CPU E5-2680 v3 (48 logical cores)

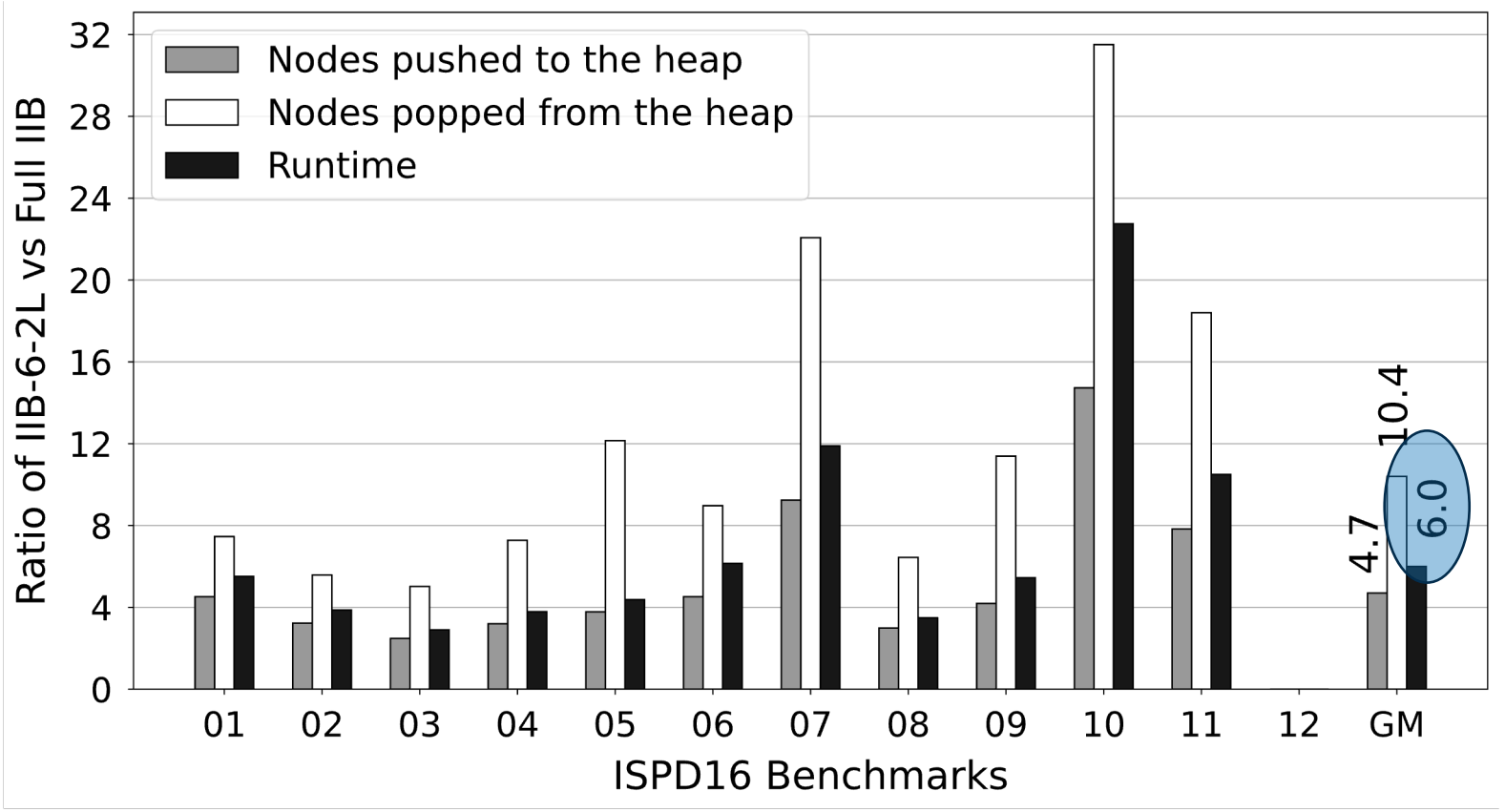
Routing on Full IIB vs IIB-6-2L

Heap pushes ↓ 4.7× Heap pops by ↓ 10.4× Runtime ↓ 6×



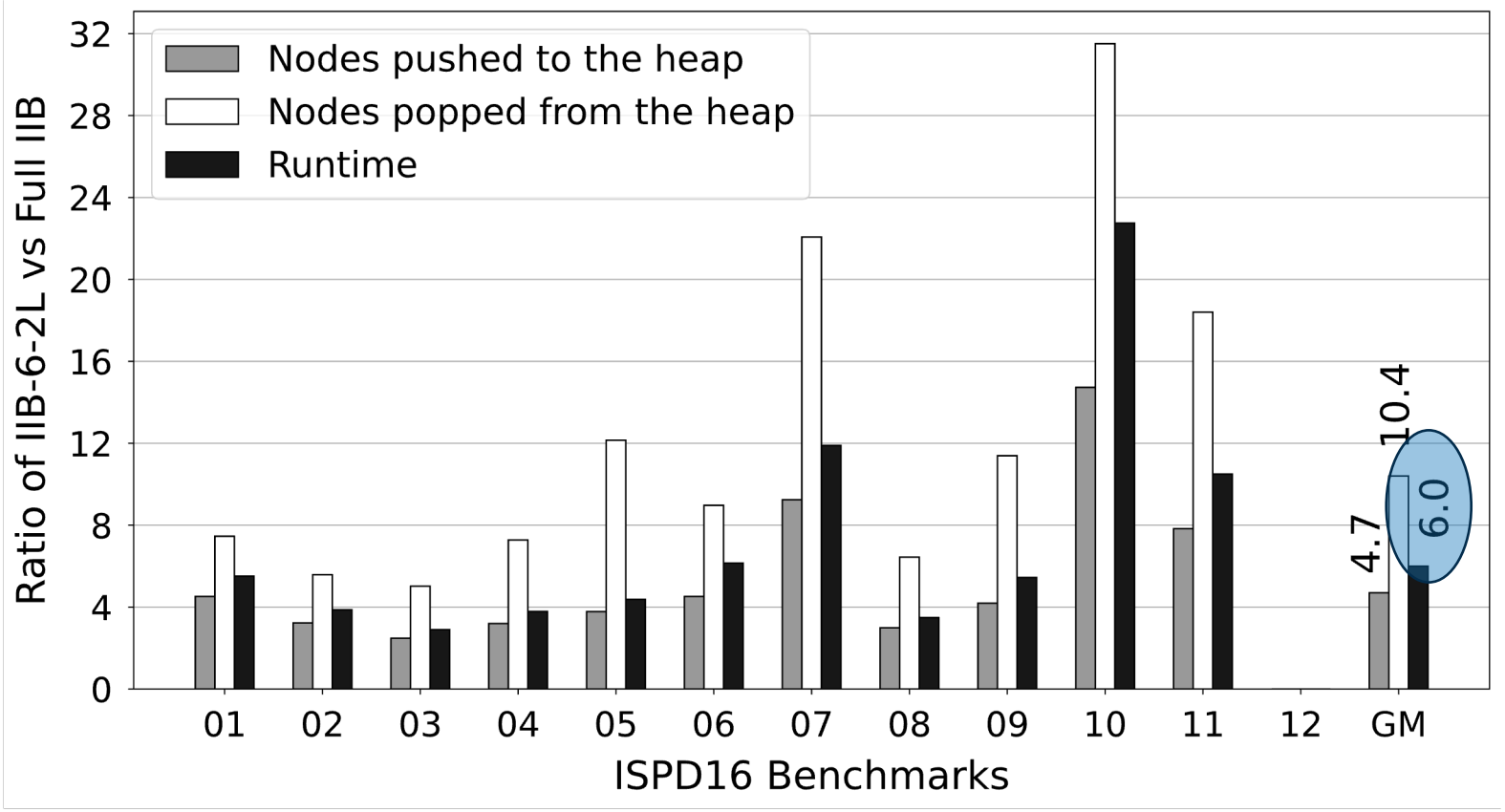
Routing on Full IIB vs IIB-6-2L

Heap pushes ↓ 4.7× Heap pops by ↓ 10.4× Runtime ↓ 6×



Routing on Full IIB vs IIB-6-2L

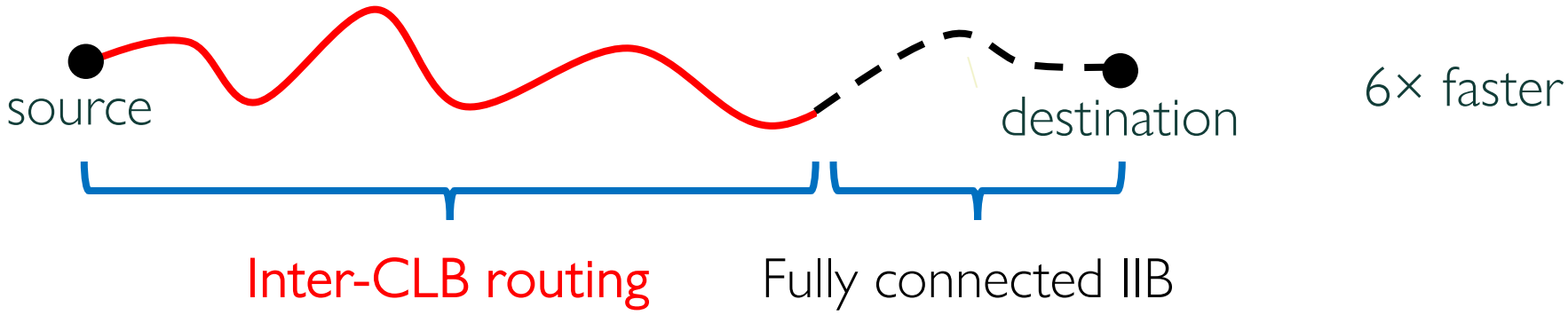
Heap pushes ↓ 4.7× Heap pops by ↓ 10.4× Runtime ↓ 6×



Routing through IIB-6-2L is the major routing bottleneck

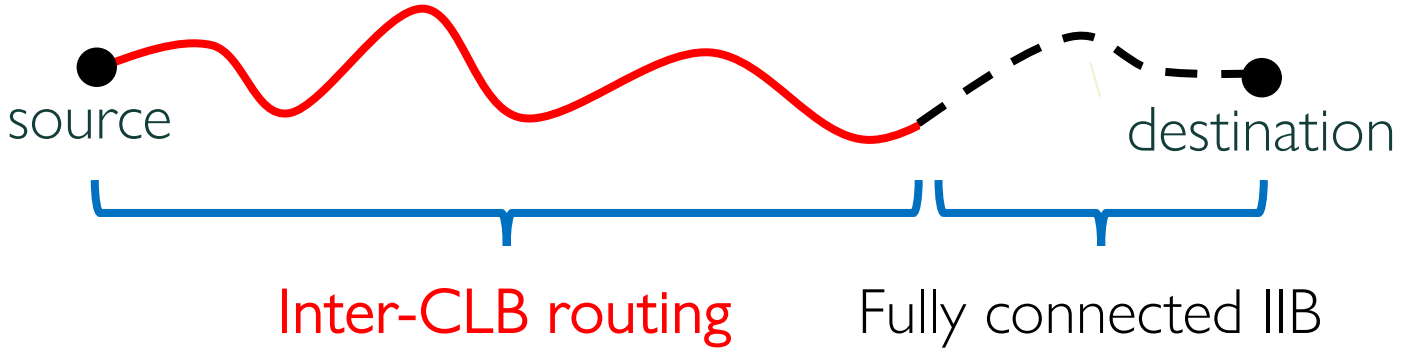
Aiming for 6× Runtime Improvement

- Perform inter-CLB routing assuming a full IIB



Aiming for 6× Runtime Improvement

- Perform inter-CLB routing assuming a full IIB



6× faster

- Perform intra-CLB routing separately

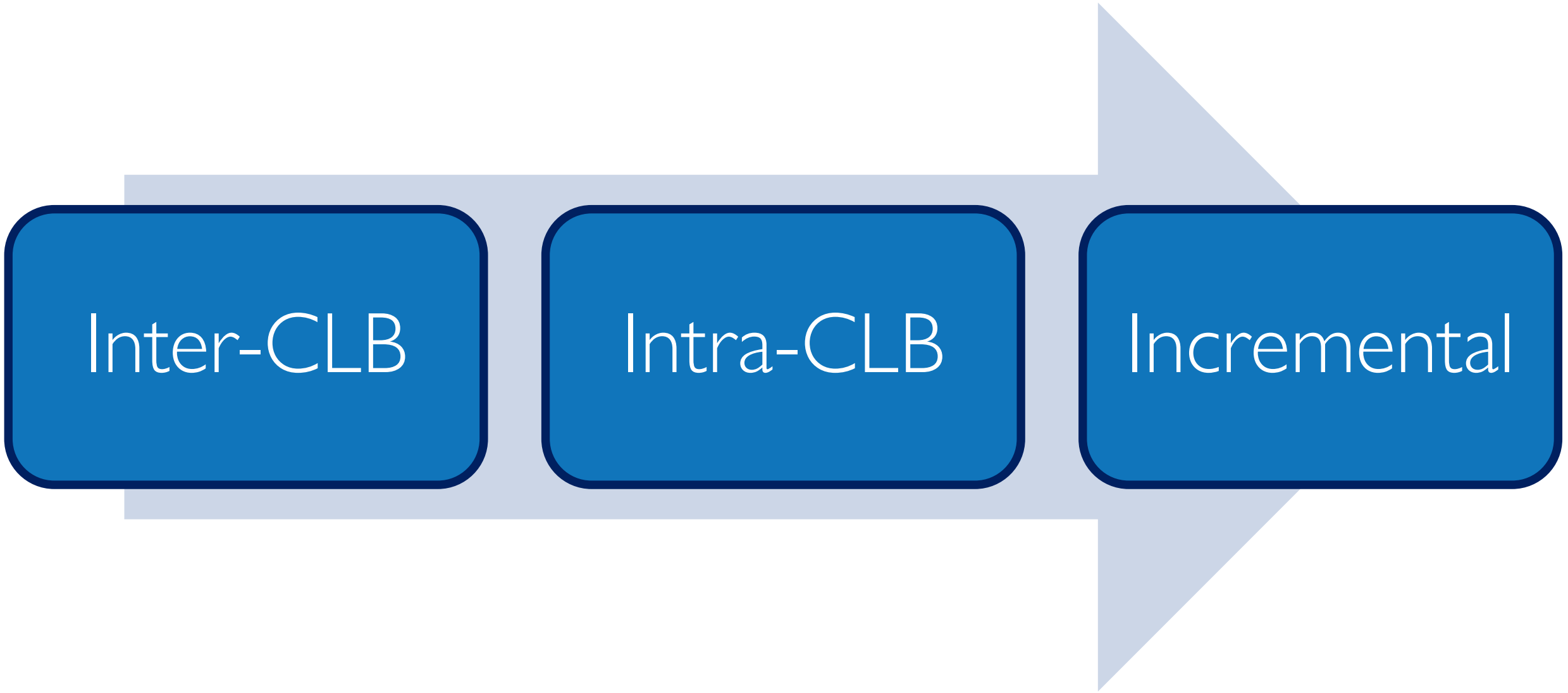
Intra-CLB Routing
on IIB-6-2L

Decouple the inter and intra-CLB routing

Outline

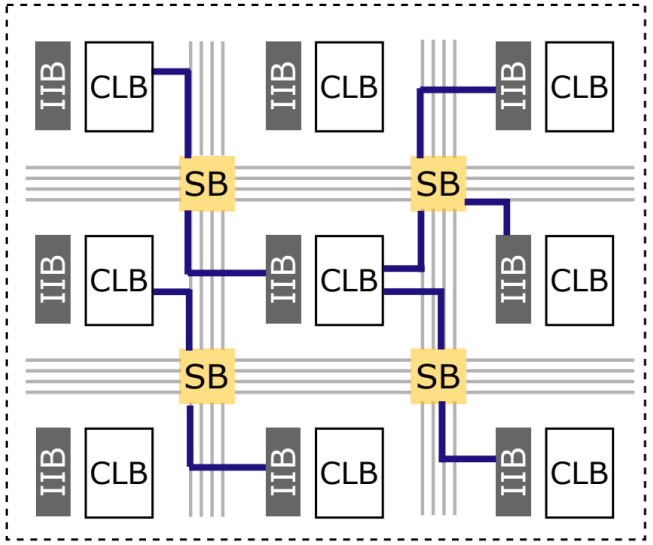
- Introduction
- Multi-Stage Router
  - Introduction
  - Analysis
  - Mitigations
- Conclusion

Multi-Stage Routing



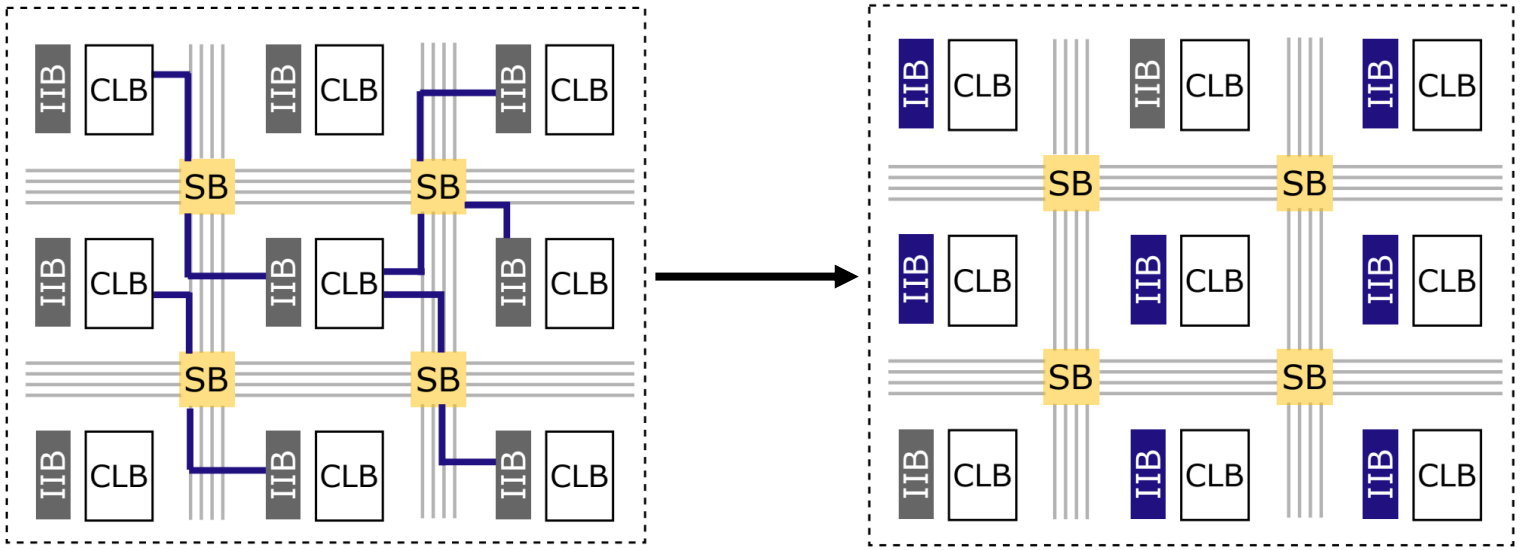
Step 1: Inter-CLB Routing

- Route assuming a full IIB
- Analogous to routing until the inputs of the IIB
- Route until all the congestion is resolved



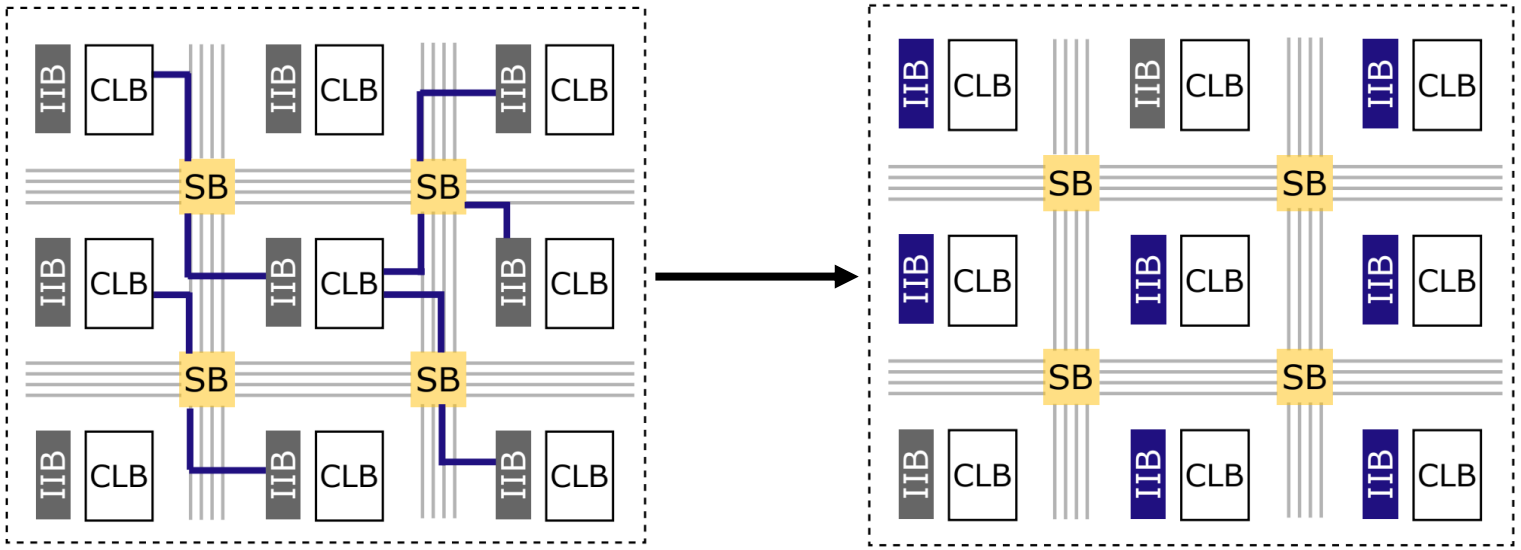
Step 2: Intra-CLB Routing

- Extract independent routing subproblems
  - Small routing problems → low runtime



Step 2: Intra-CLB Routing

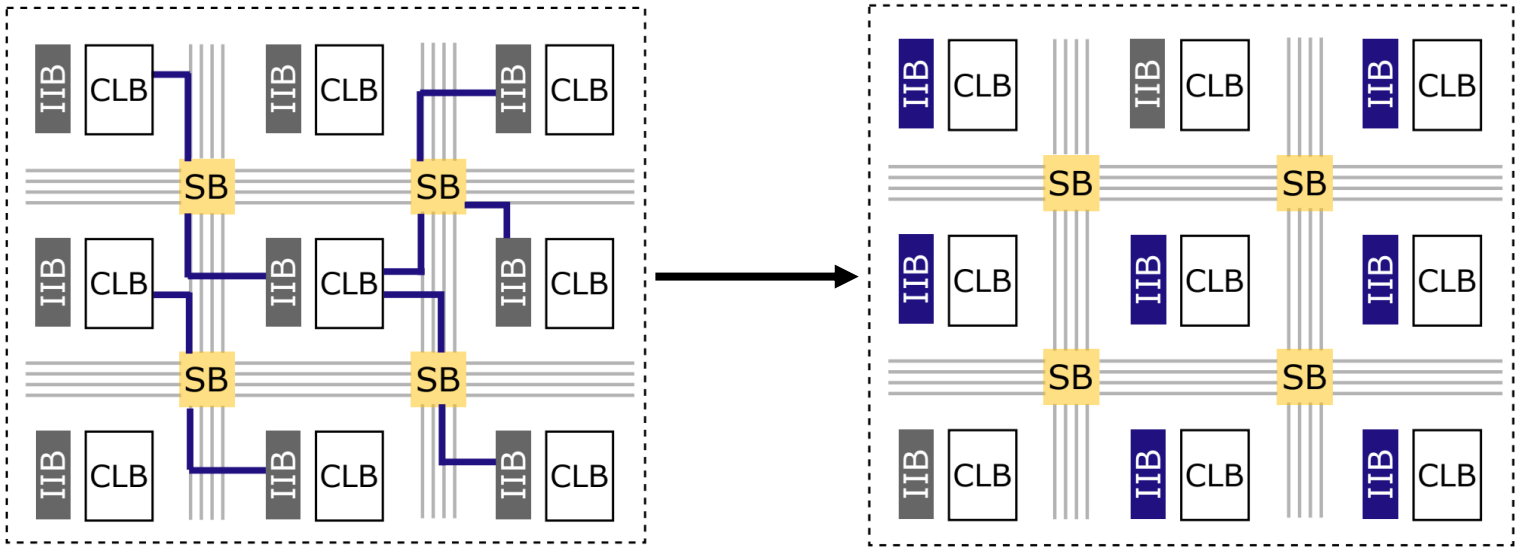
- Extract independent routing subproblems
  - Small routing problems → low runtime
- Route IIBs in parallel, in any order



Step 2: Intra-CLB Routing

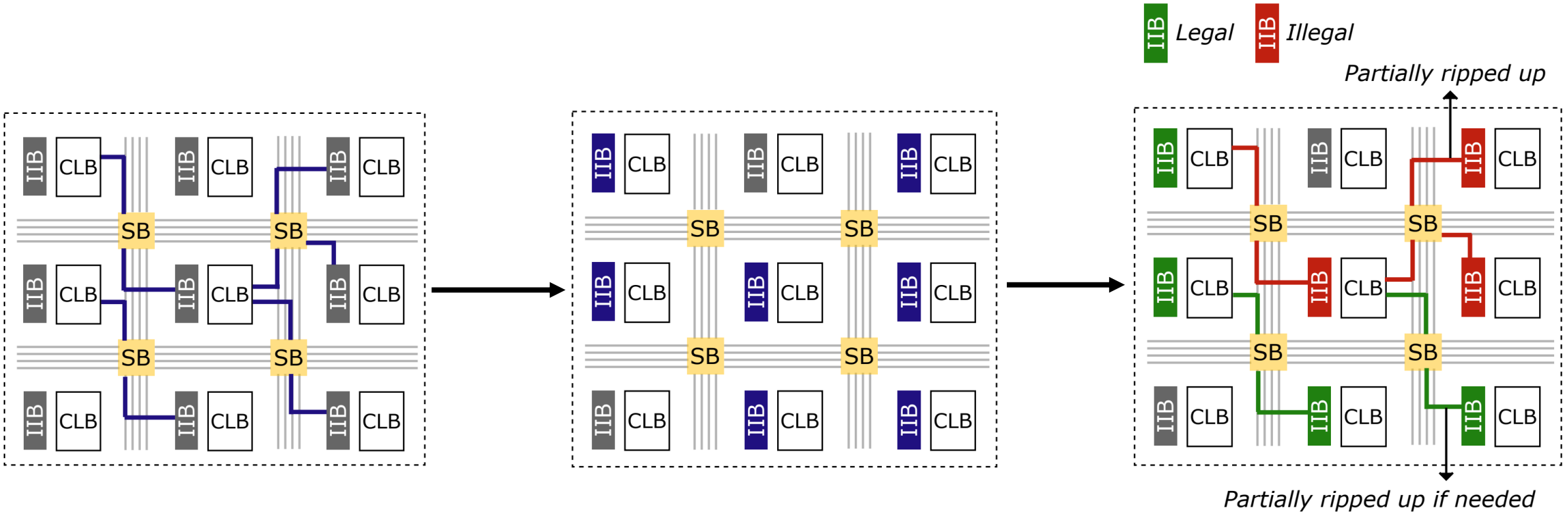
- Extract independent routing subproblems
  - Small routing problems → low runtime
- Route IIBs in parallel, in any order
- Because the IIBs are not actually full → some subproblems may not legalize

(congestion remains in some IIBs)



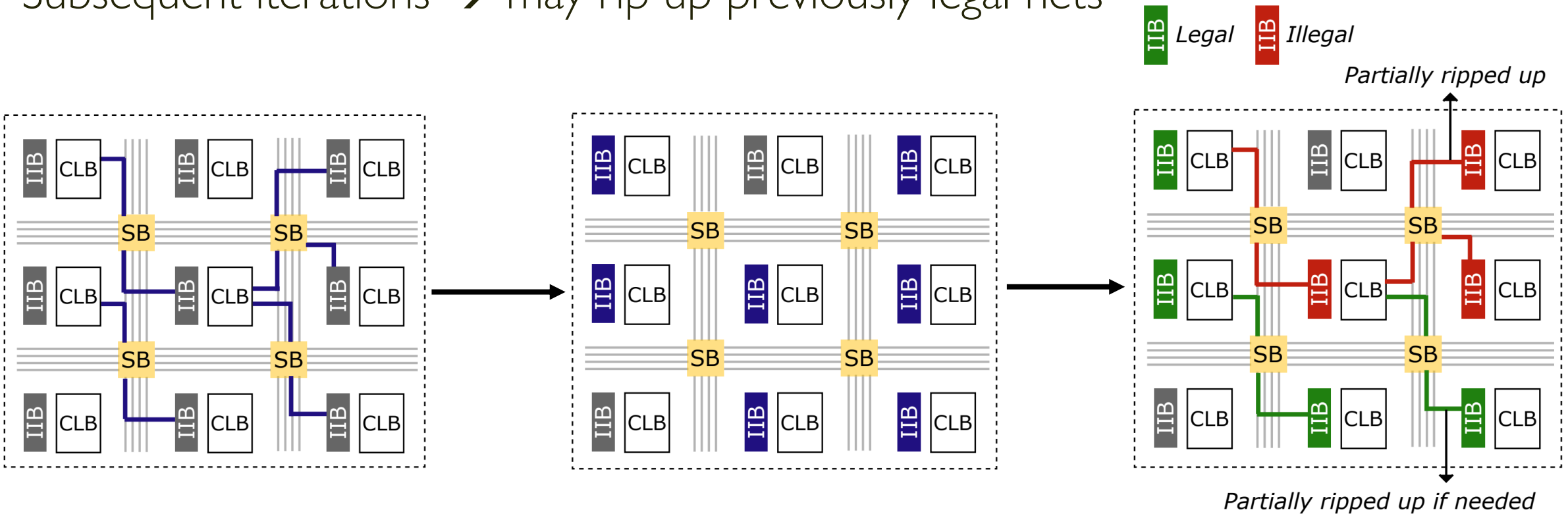
Step 3: Incremental Routing

- At the start of step 3
  - Combine routing from step-1 and step-2
  - Rip up congested nets within illegal IIBs



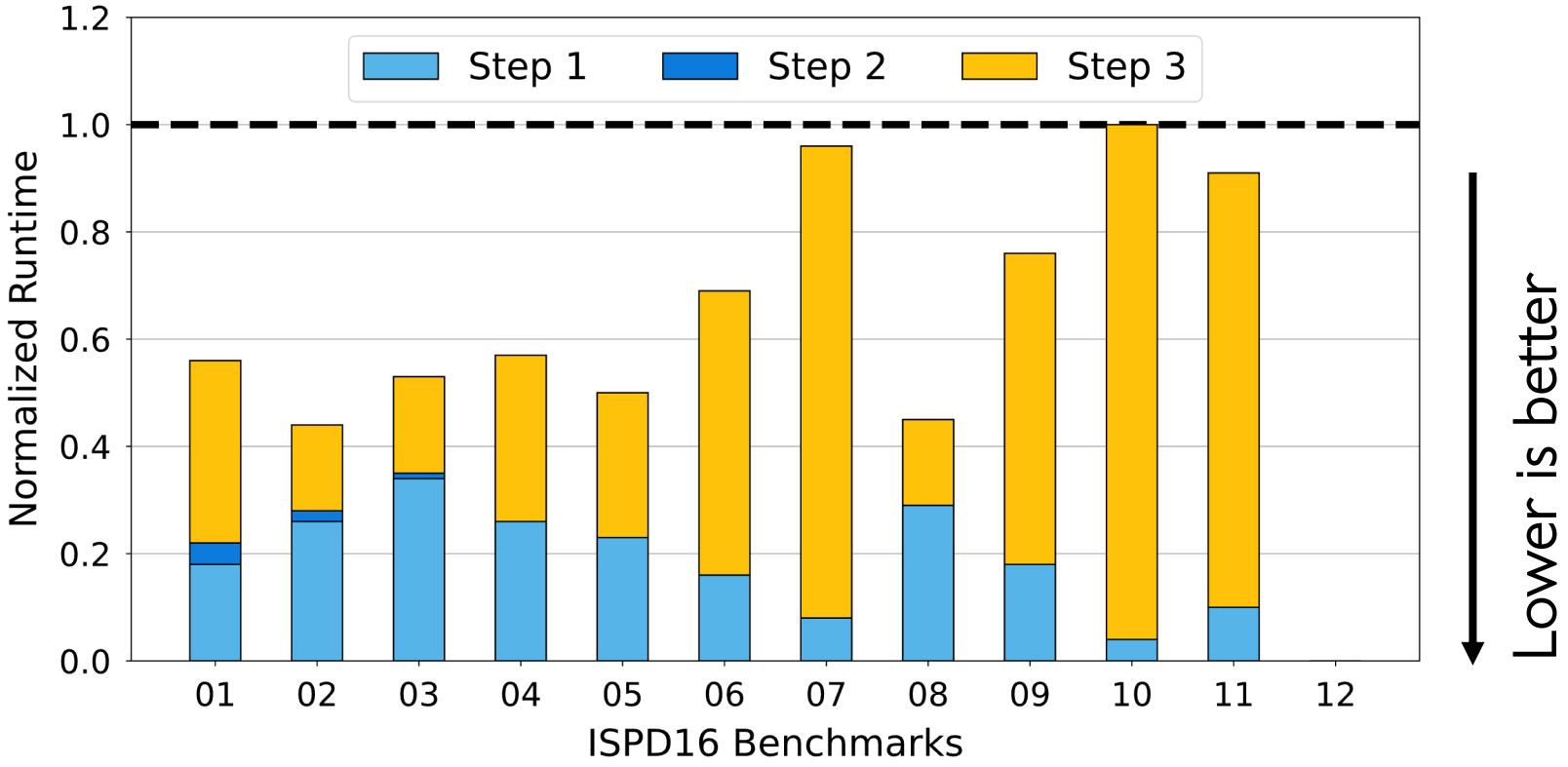
Step 3: Incremental Routing

- At the start of step 3
  - Combine routing from step-1 and step-2
  - Rip up congested nets within illegal IIBs
- Subsequent iterations → may rip up previously legal nets



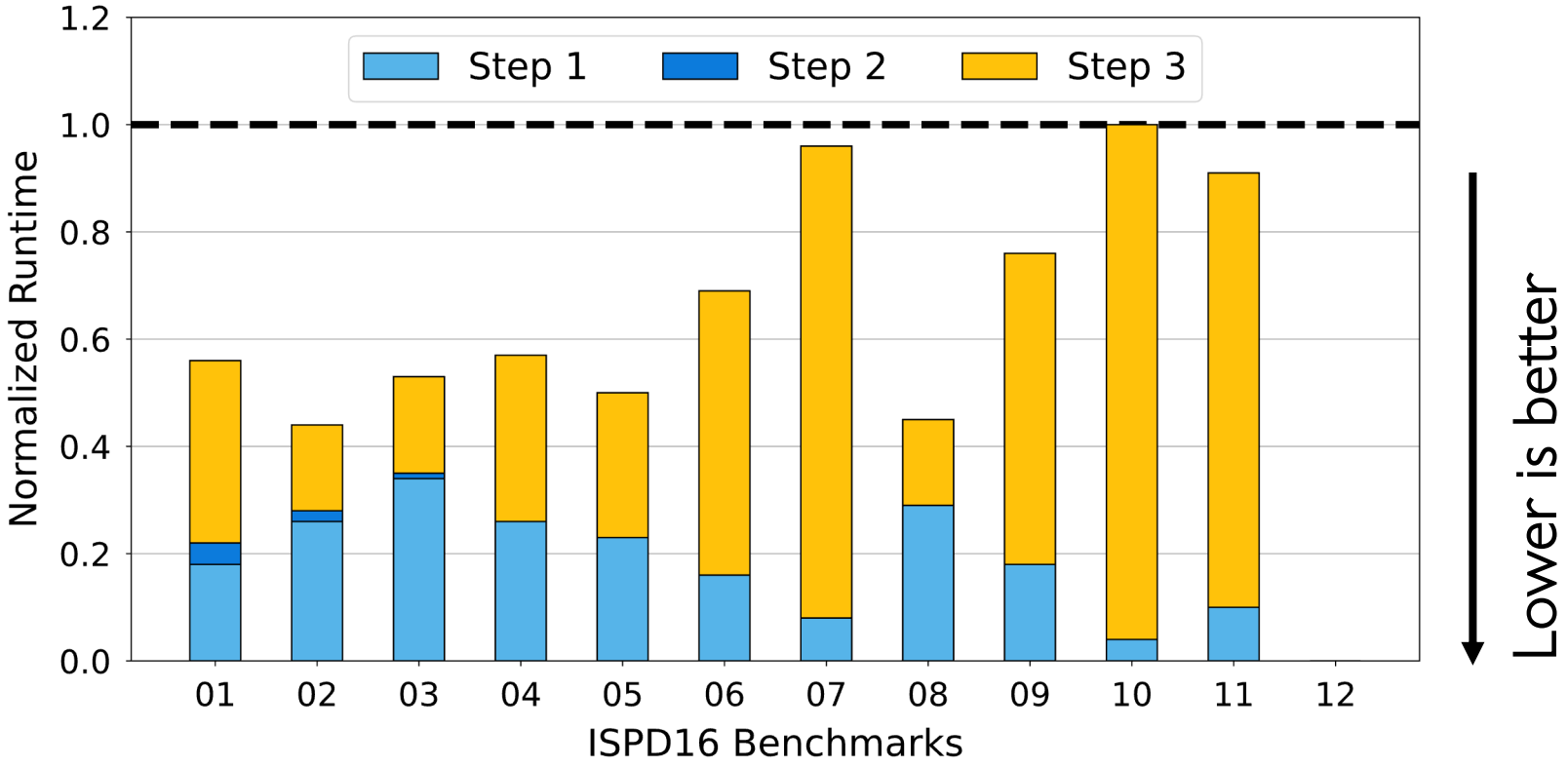
Multi-Stage vs Single-Stage Routing

- Single-Stage Routing → routing all the way to CLB pins through IIB in one step
- Runtime normalized with respect to Single-Stage Routing
- In step 2, IIBs routed in parallel



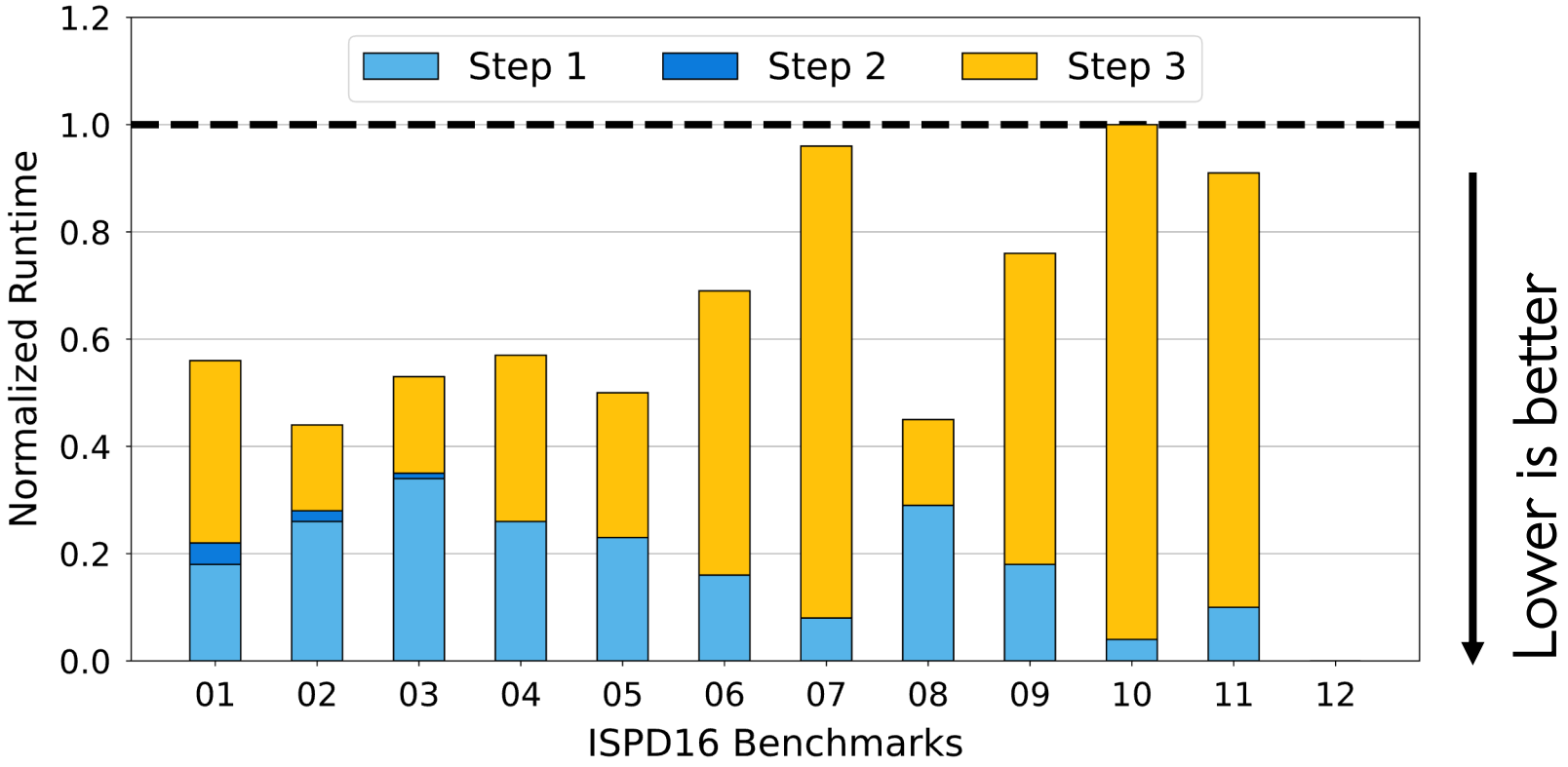
Multi-Stage vs Single-Stage Routing

- Geomean runtime improvement 1.5×



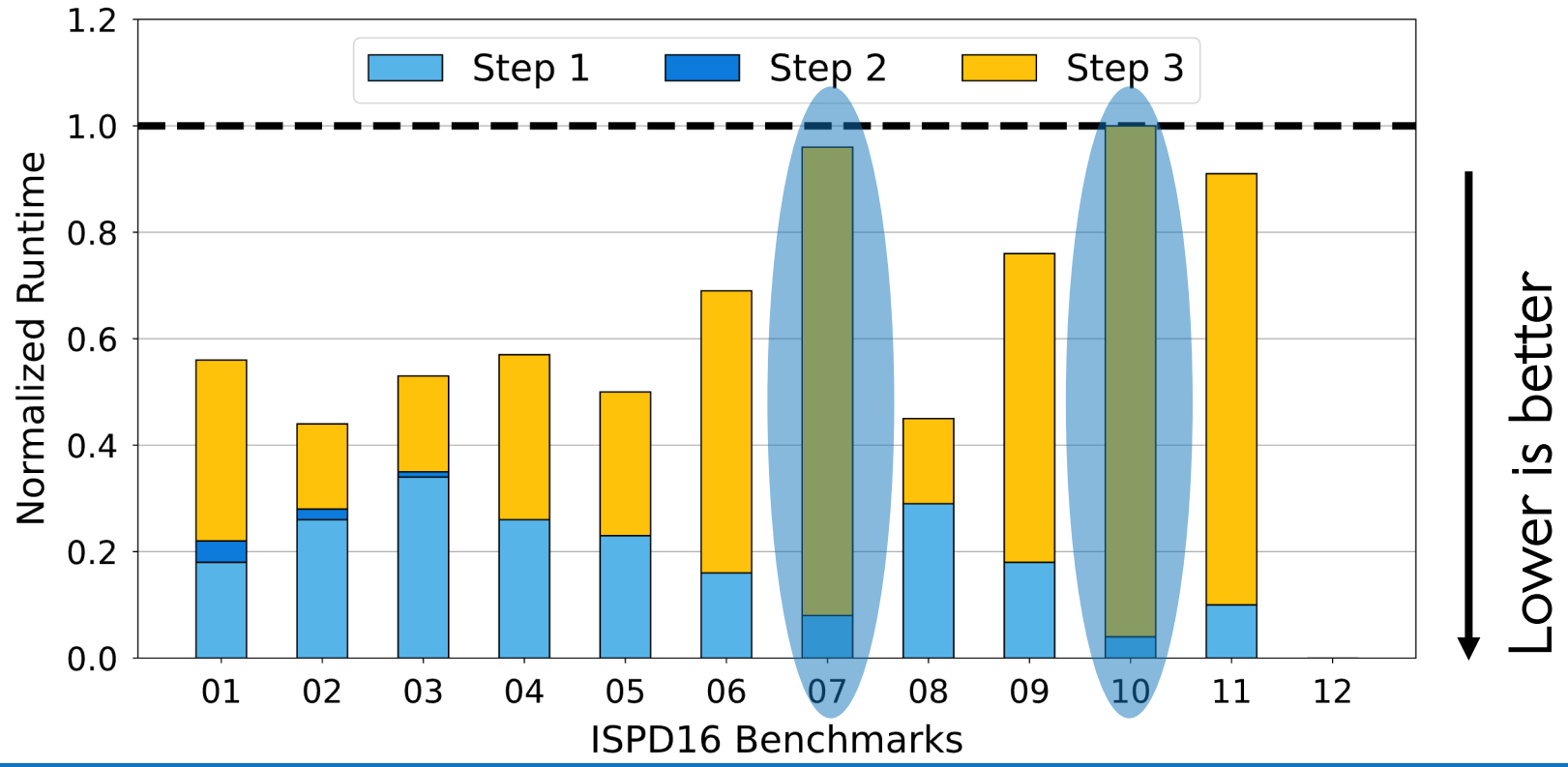
Multi-Stage vs Single-Stage Routing

- Geomean runtime improvement 1.5×
- If the assumption of full connectivity held → runtime improvement would be 6×



Multi-Stage vs Single-Stage Routing

- Benchmarks 07 and 10, step 3 consumes all the runtime gain



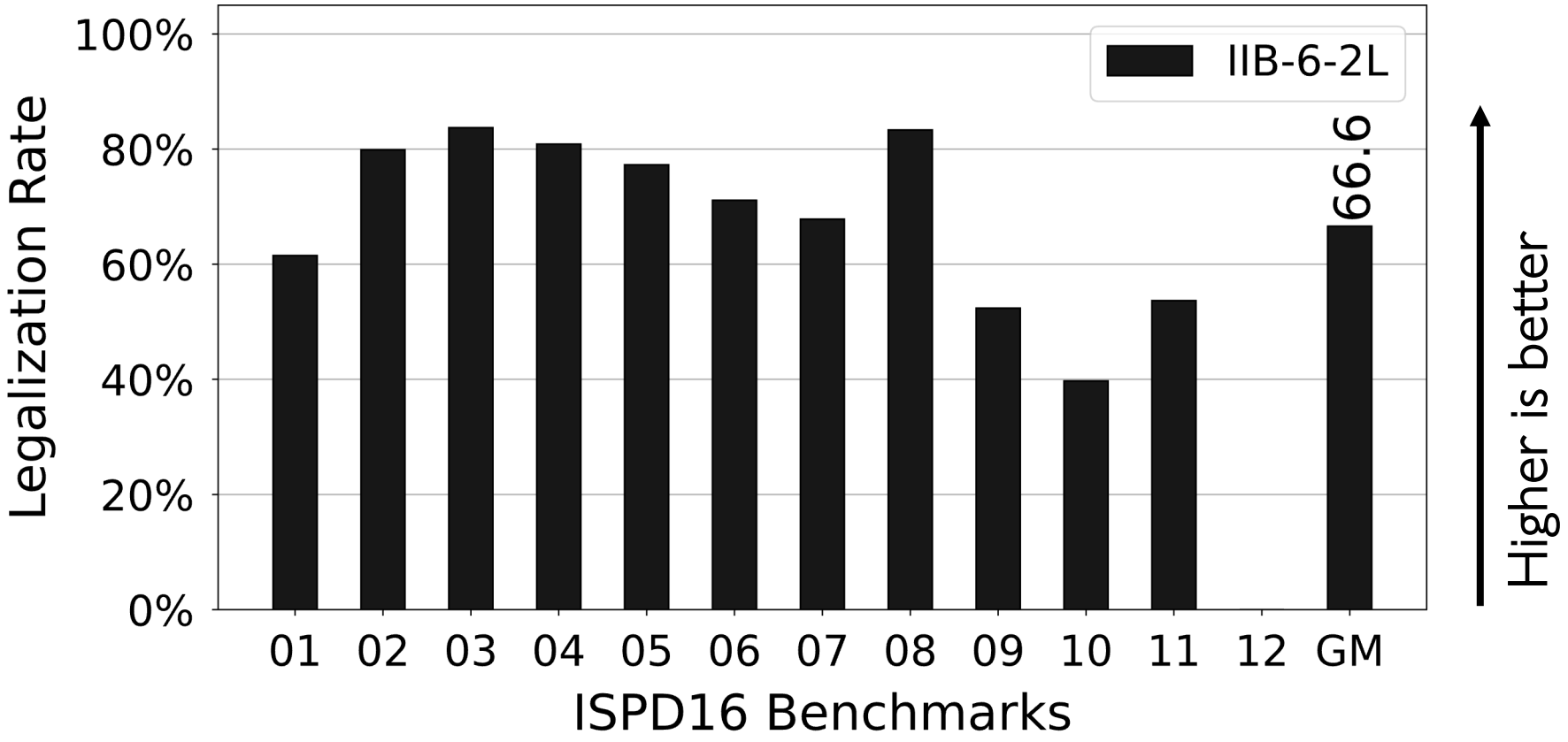
Step 3 consumes significant fraction of runtime

Outline

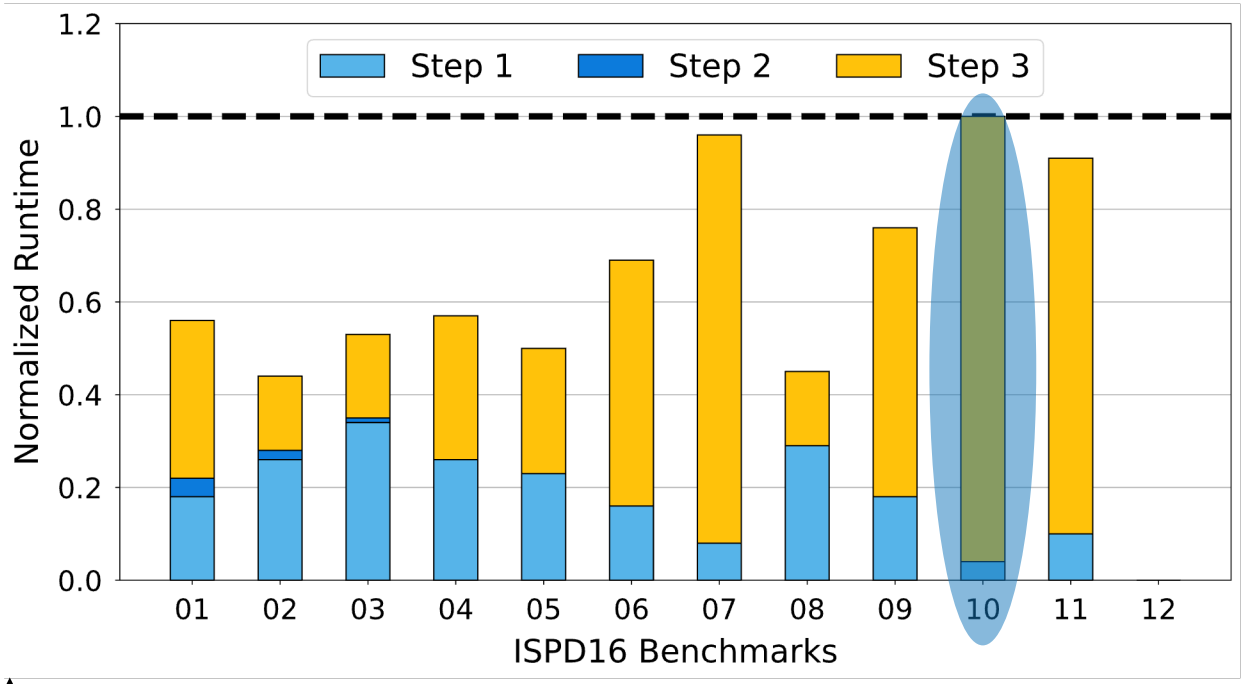
- Introduction
- Multi-Stage Router
  - Introduction
  - Analysis
  - Mitigations
- Conclusion

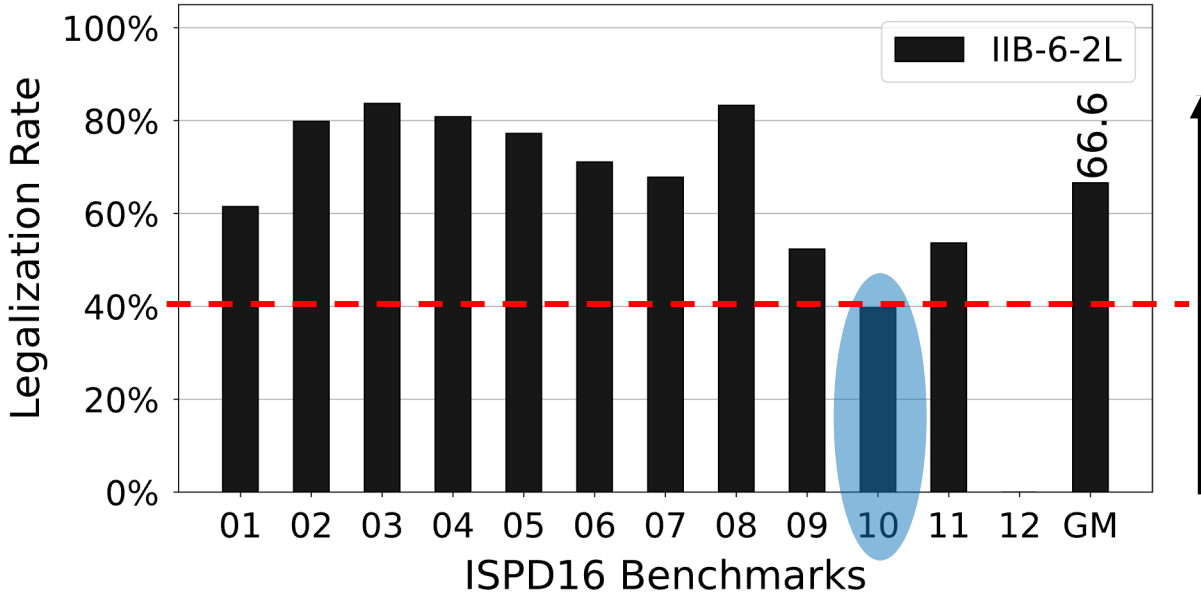
Step 2 IIB Legalization Rate

- Only 66% of the IIBs actually legalized under our assumption of full IIB



Step 2 IIB Legalization Rate



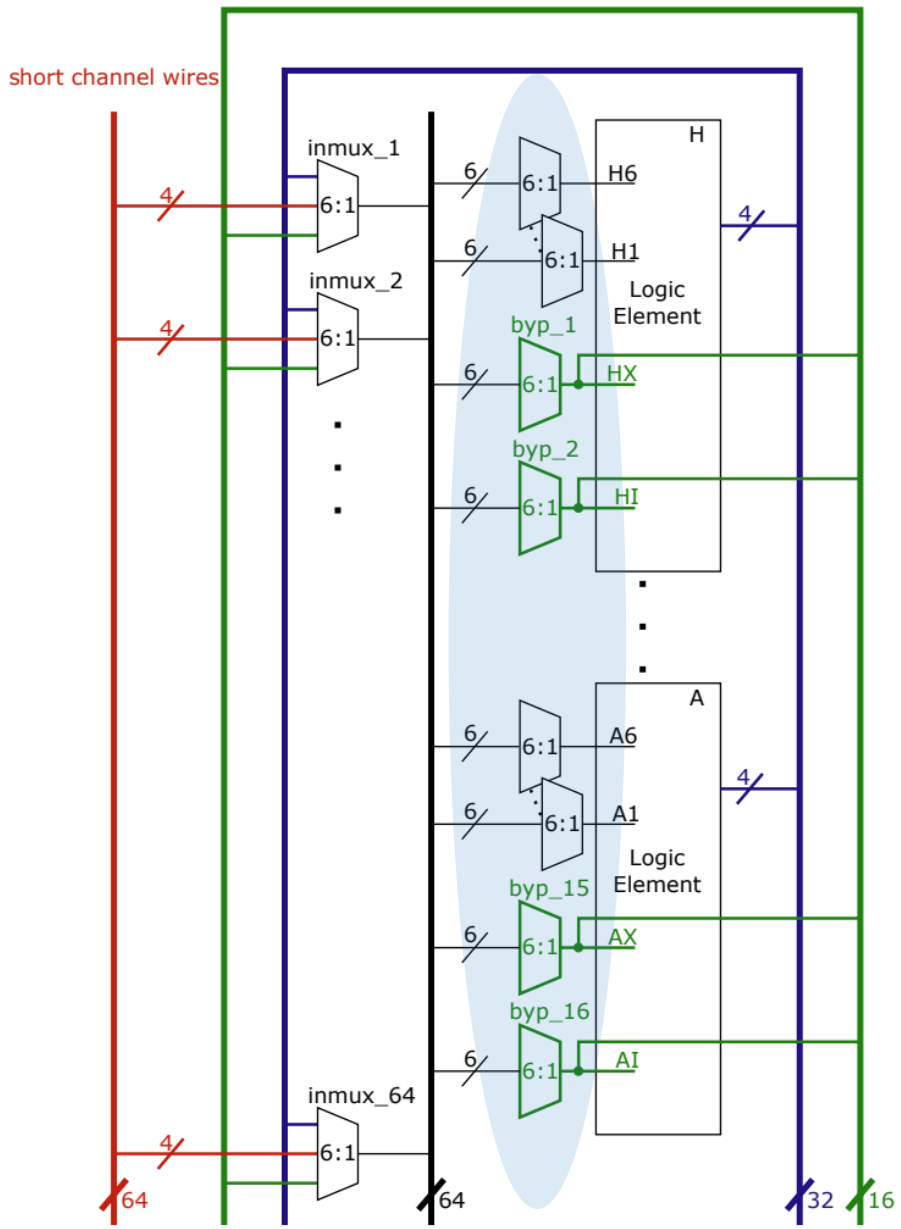


Higher is better

What causes low legalization rate?

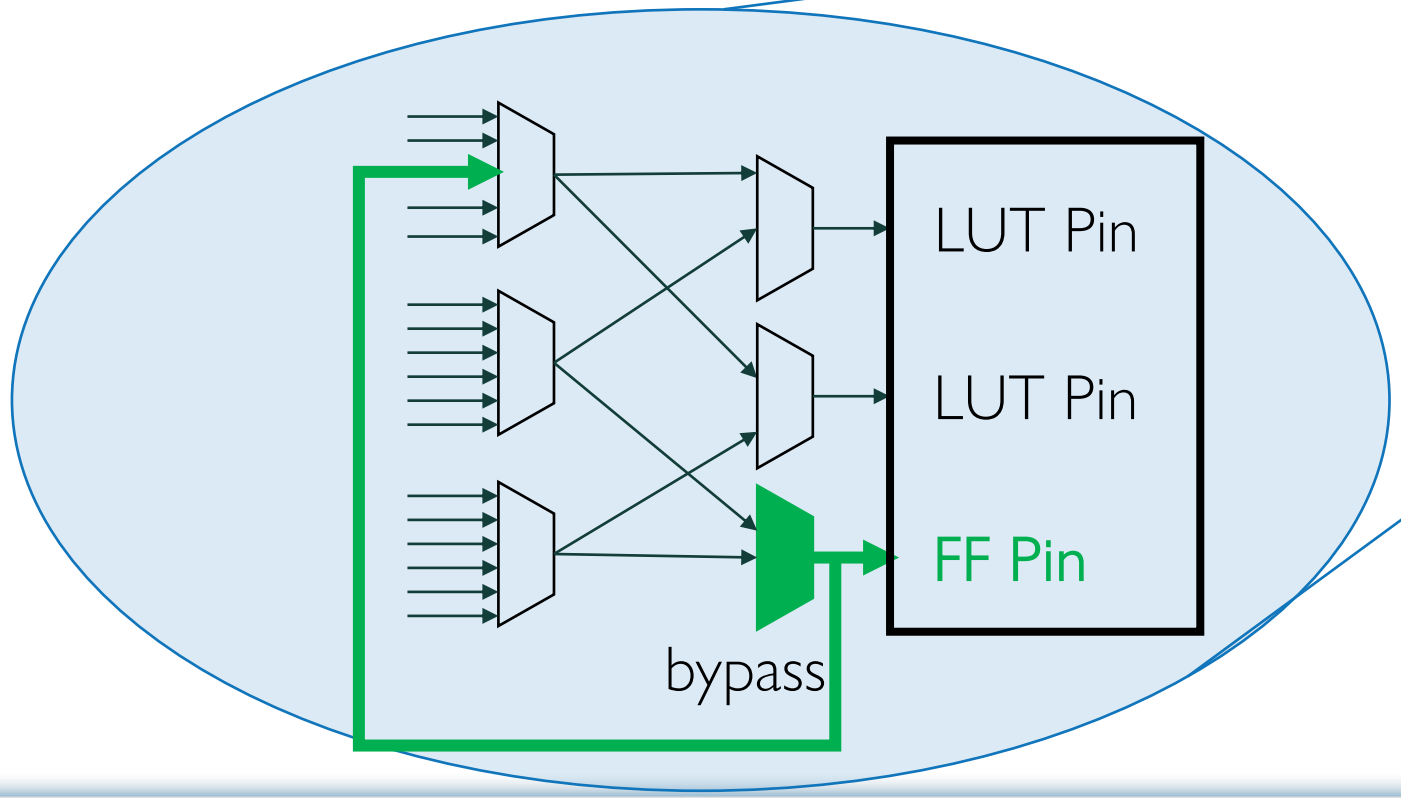
Bypass Mux: The Routing Slayer

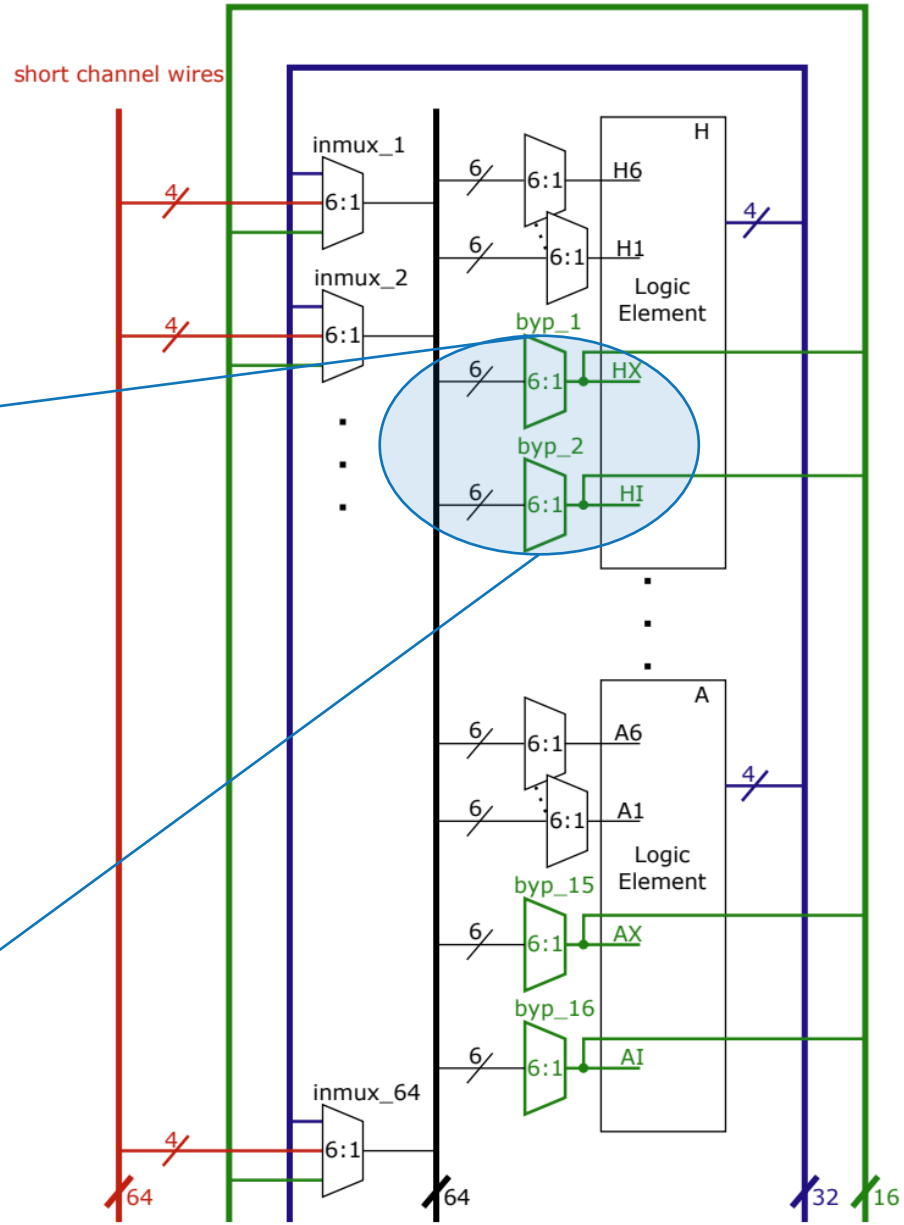
- Inner layer → improves connectivity
- Bypass mux serves two purposes
  - Connection to FF input pin
  - Improves connectivity within IIB



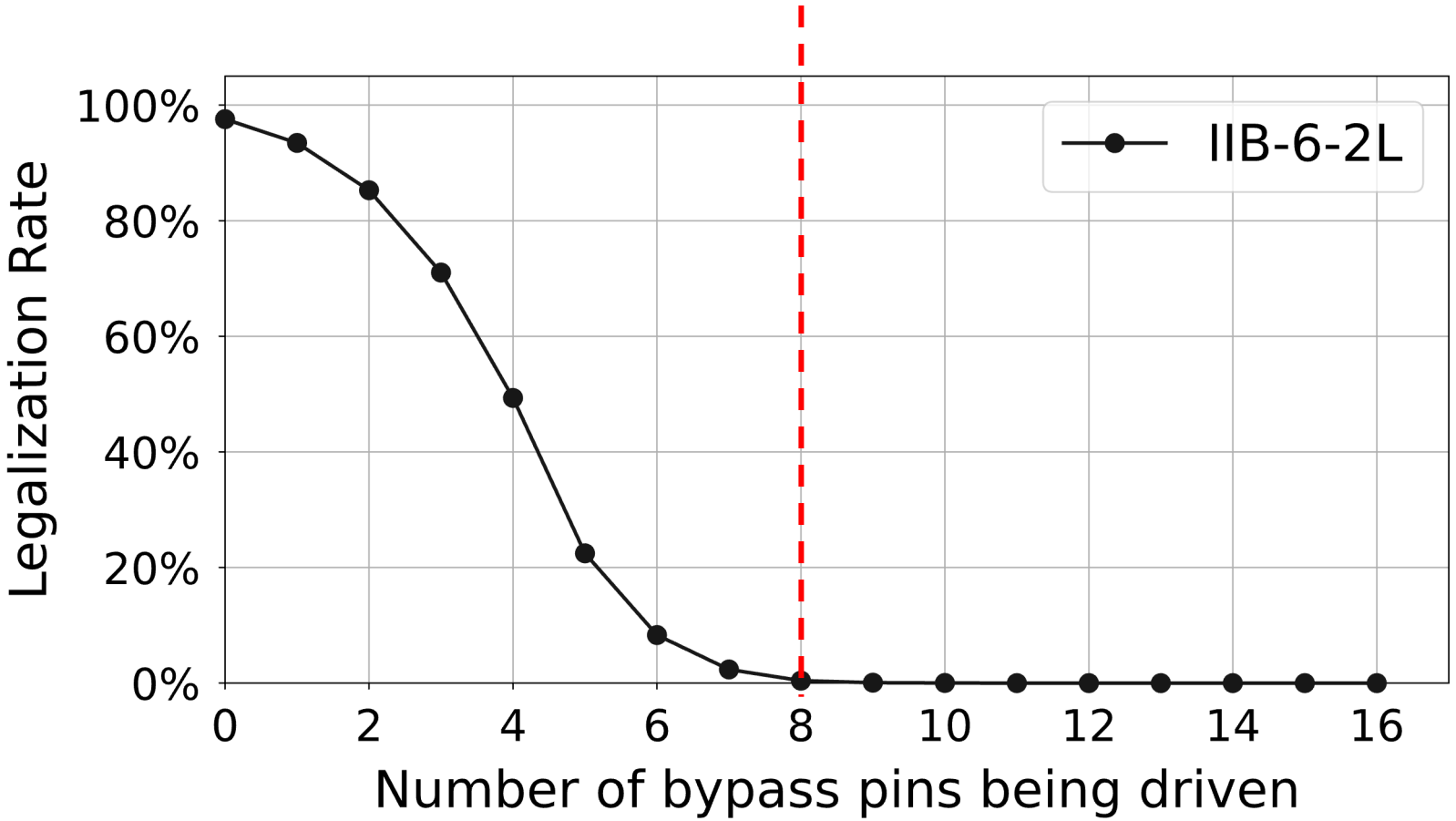
Bypass Mux: The Routing Slayer

- Inner layer → improves connectivity
- Bypass mux serves two purposes
  - Connection to FF input pin
  - Improves connectivity within IIB





Legalization Rate vs Number of Driven Bypass Pins



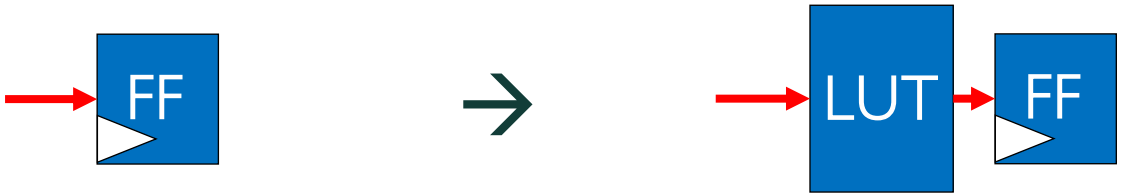
Legalization rate drops to zero if ≥ 8 bypass muxes are used as pins

Outline

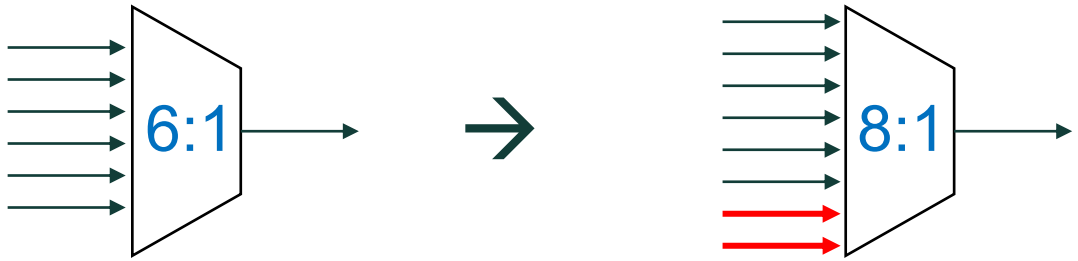
- Introduction
- Multi-Stage Router
  - Introduction
  - Analysis
  - Mitigations
- Conclusion

Mitigations

- Reduce load on bypass muxes (CAD optimization)
  - Use LUT-1 [Intel Quartus Prime Pro'23]



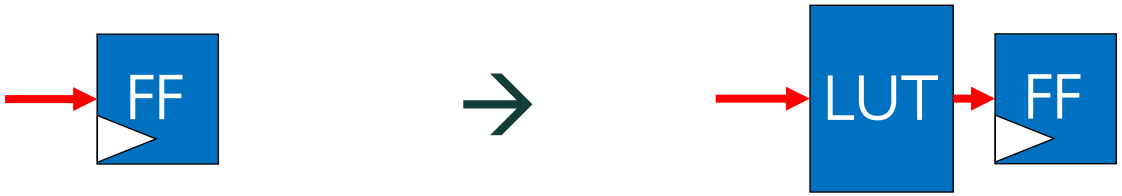
- Increase routing flexibility (architectural enhancement)
  - Increase IIB mux size from 6:1 to 8:1



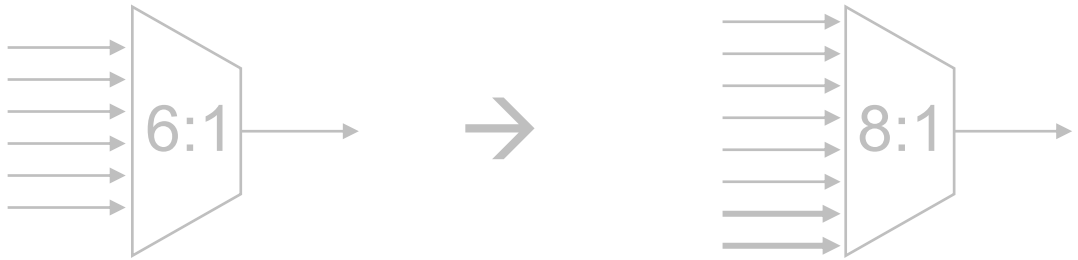
I. Corporation, "Intel® Quartus® Prime Pro Edition Help version 21.4," 2023, retrieved May 2023.

Mitigations

- Reduce load on bypass muxes (CAD optimization)
  - Use LUT-1 [Intel Quartus Prime Pro'23]



- Increase routing flexibility (architectural enhancement)
  - Increase IIB mux size from 6:1 to 8:1

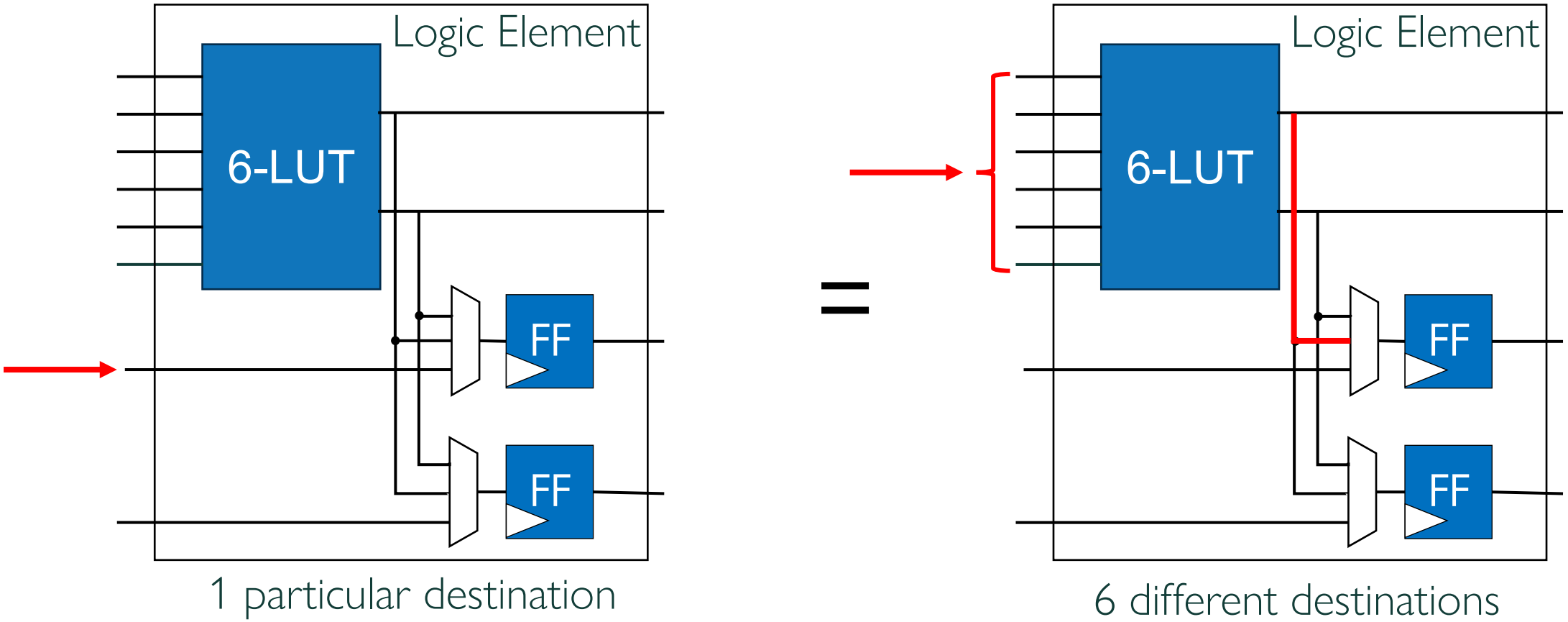


I. Corporation, "Intel® Quartus® Prime Pro Edition Help version 21.4," 2023, retrieved May 2023.

Advantages of LUT-1

- Exploiting LUT pin equivalence
- Frees up bypass mux to be used by other signals

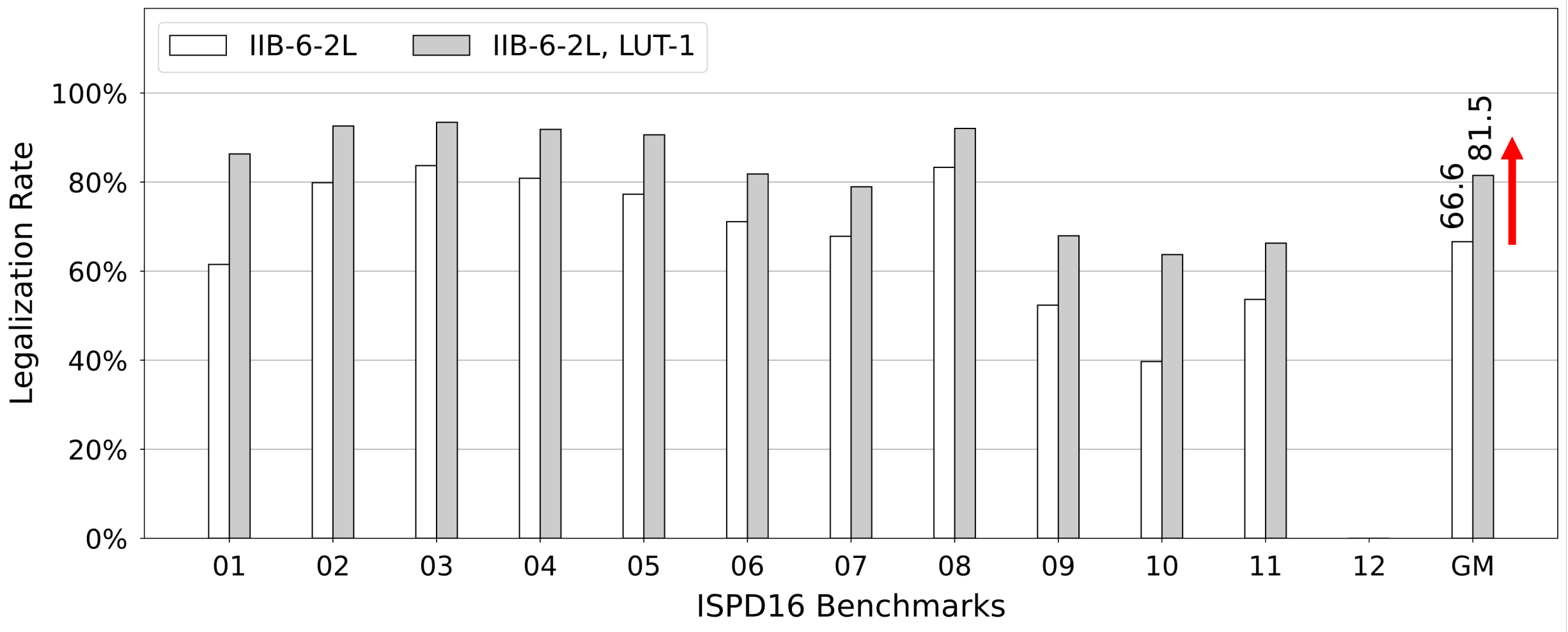
} increases routing flexibility



1 particular destination

6 different destinations

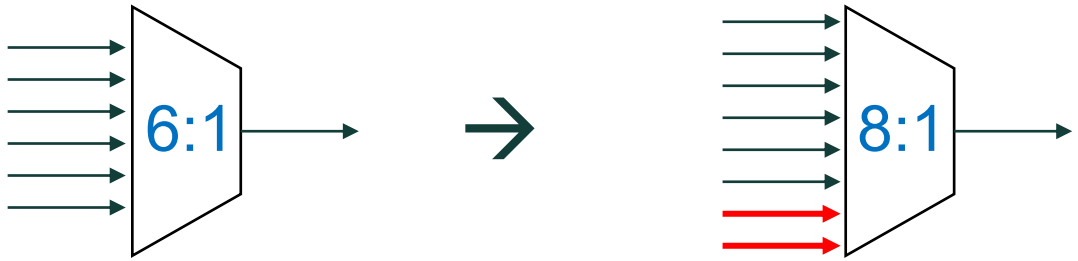
LUT-1: Legalization Rate Improvement



Mitigations

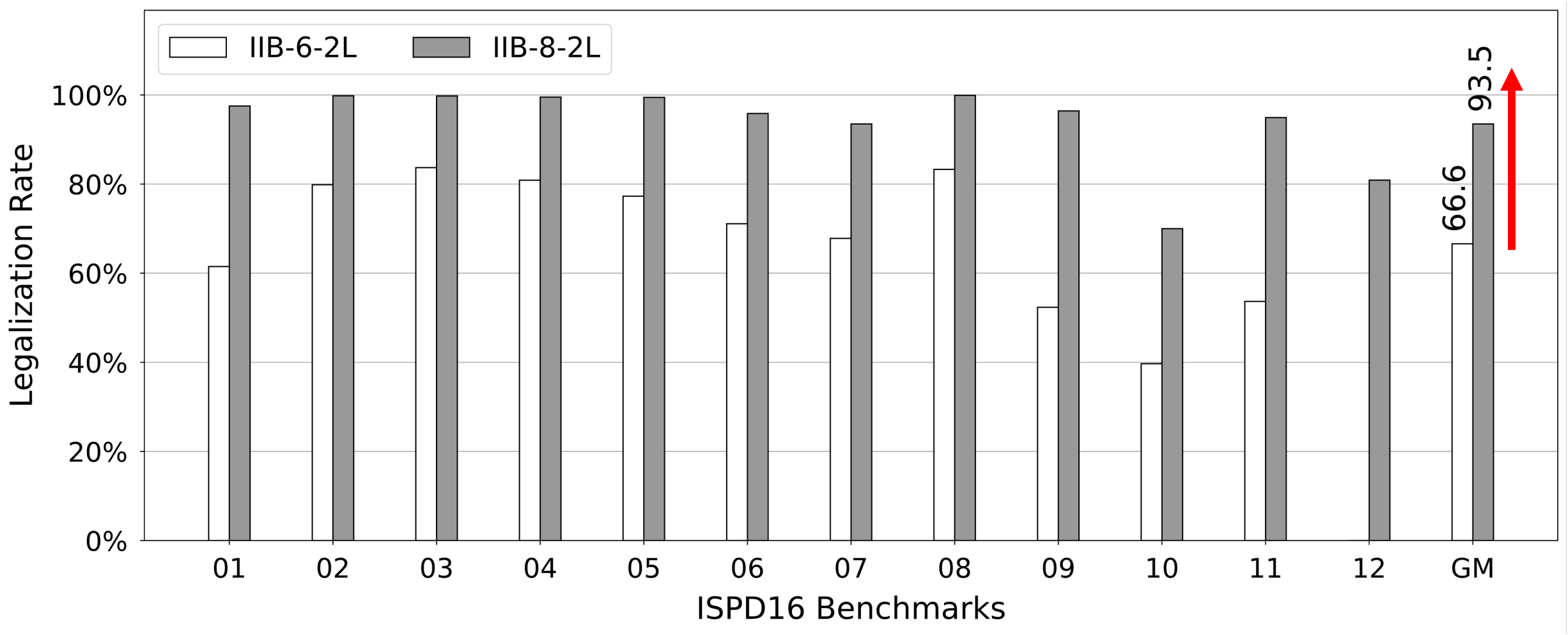
- Reduce load on bypass muxes → placement optimization
  - Use LUT-1 [Intel Quartus Prime Pro'23]

- Increase routing flexibility → architectural enhancement
  - Increase IIB mux size from 6:1 to 8:1



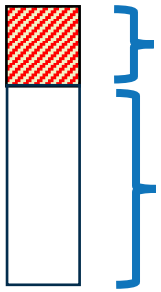
I. Corporation, "Intel® Quartus® Prime Pro Edition Help version 21.4," 2023, retrieved May 2023.

IIB-8-2L: Legalization Rate Improvement



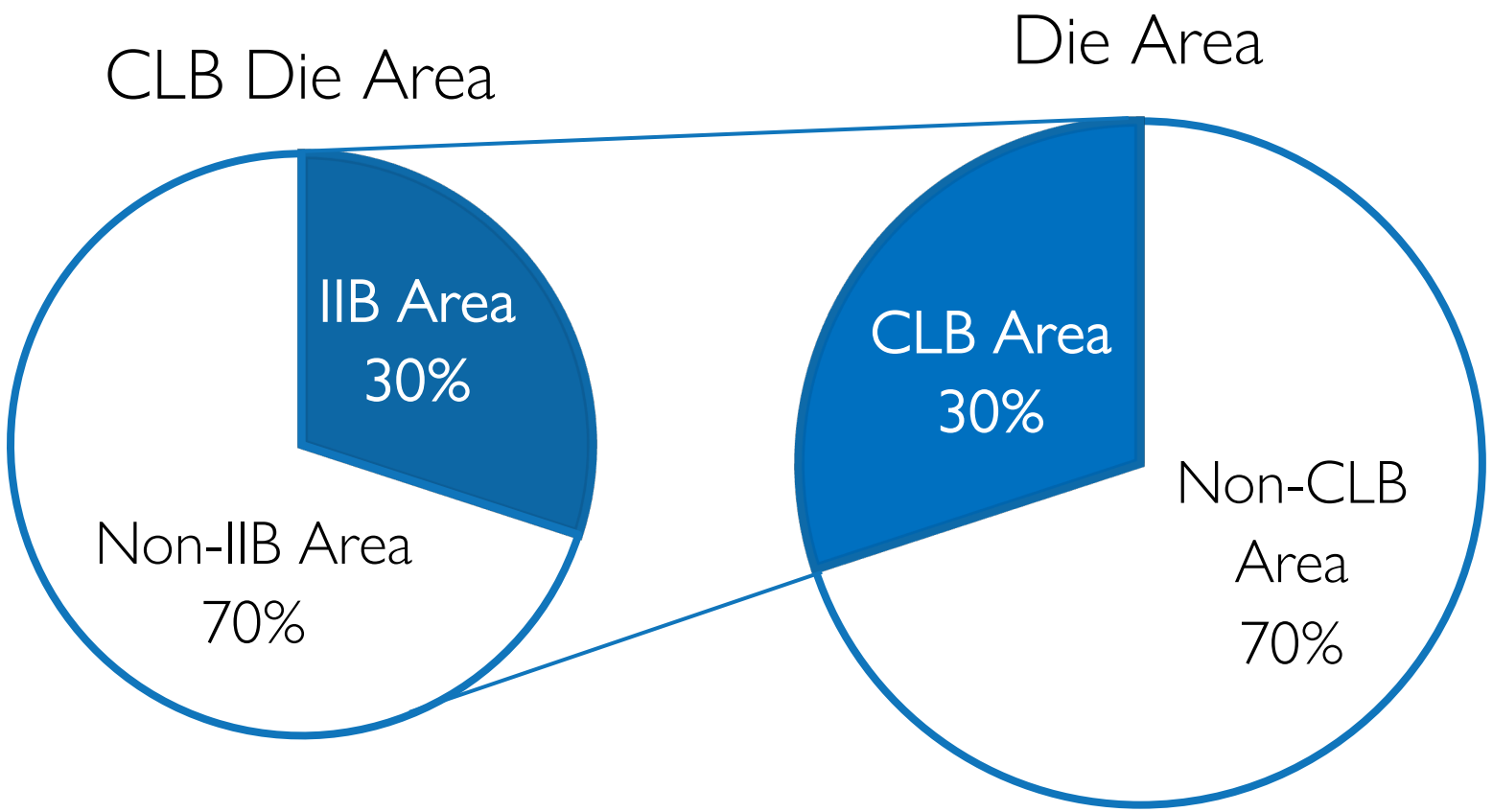
Impact of Architectural Enhancement on Area

IIB-8-2L Area



33% ↑ in IIB area

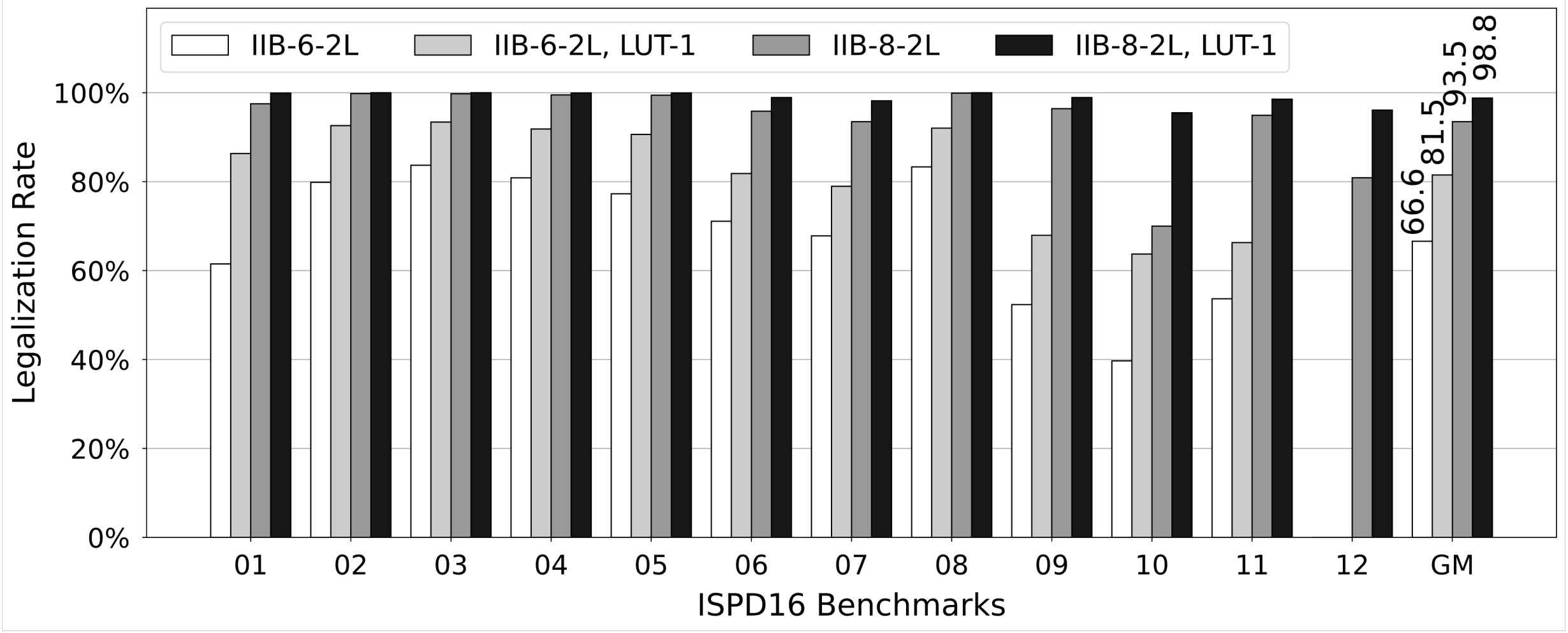
IIB-6-2L Area



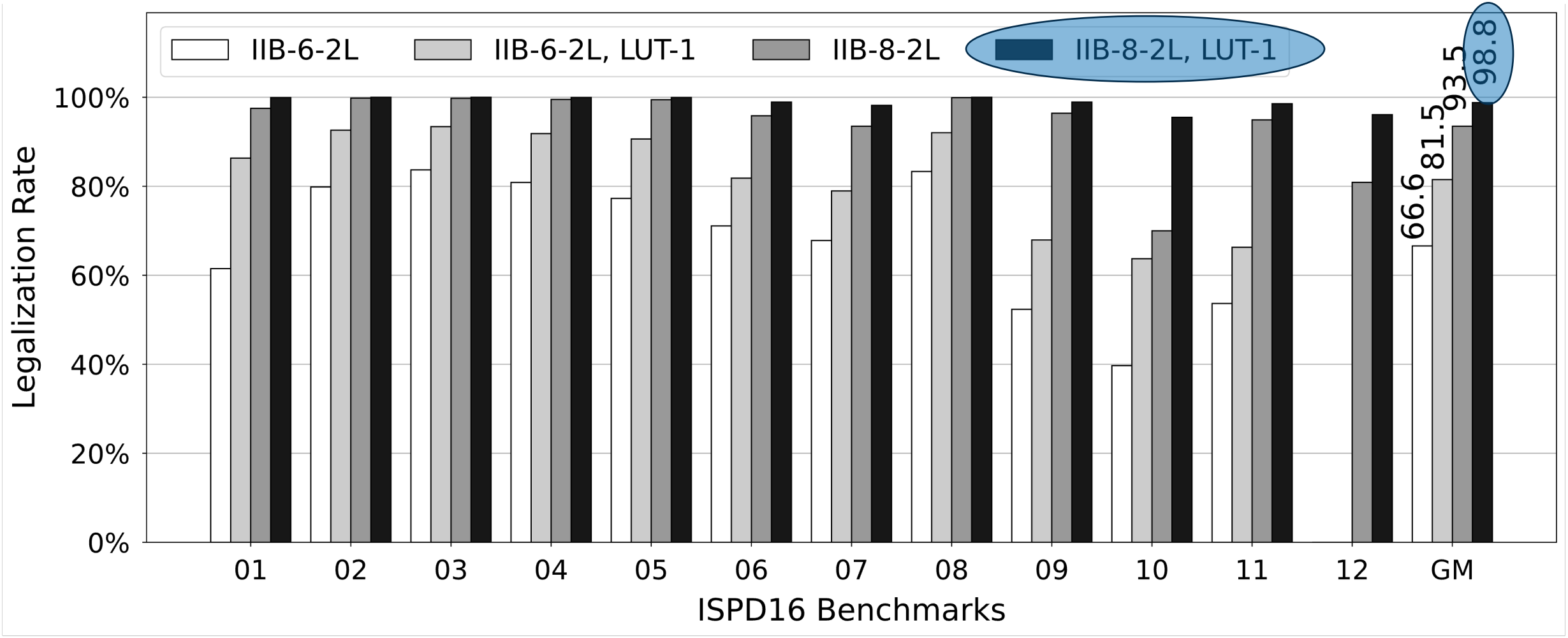
*Rough approximation

IIB-8-2L causes ~3% increase in die area

Comparing Legalization Rate

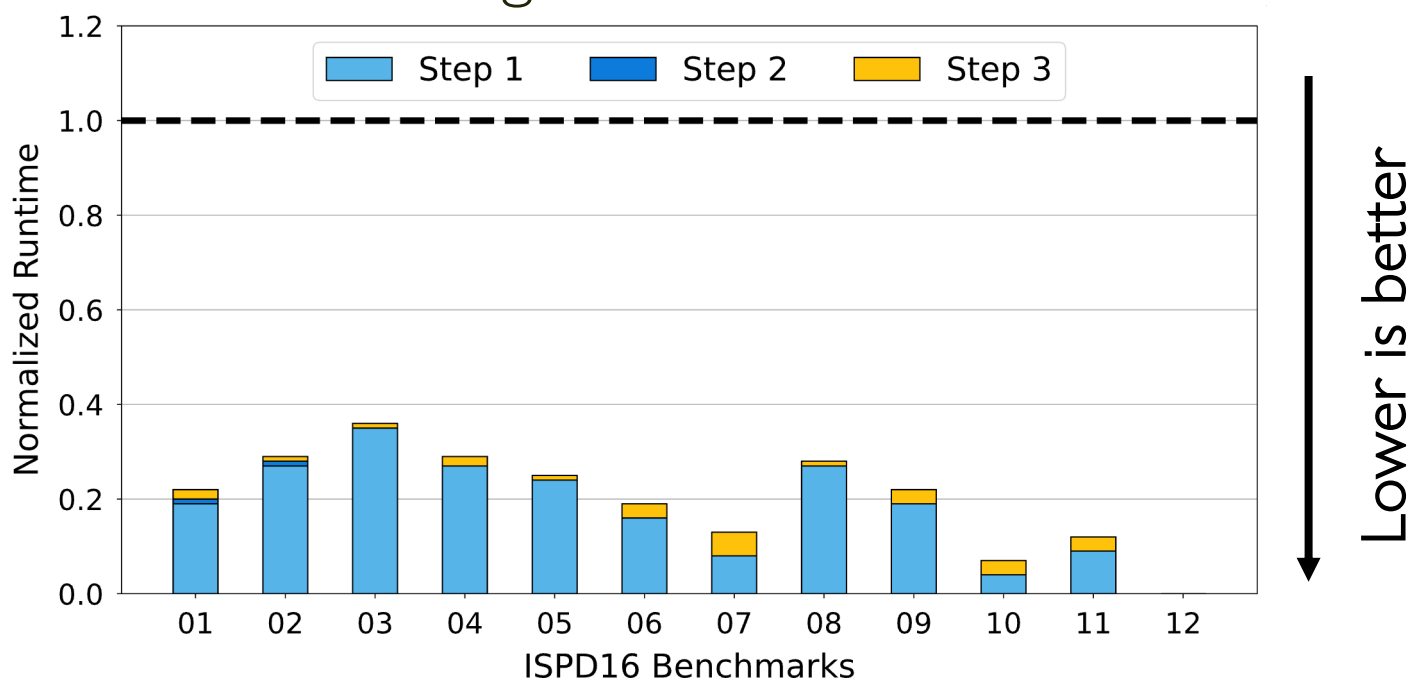


Comparing Legalization Rate



Multi-Stage Router with IIB-8-2L & LUT-1

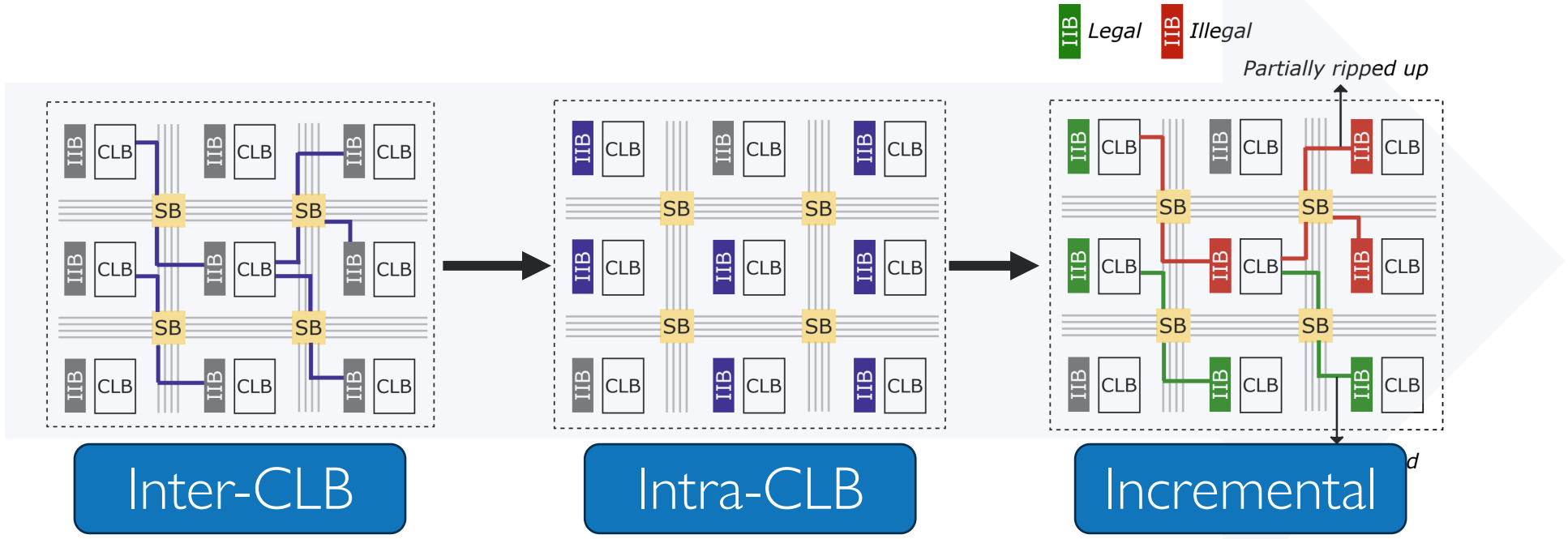
- Normalized with respect to Single-Stage router with IIB-6-2L
- Geomean of runtime improvement 4.9×
- At no deterioration of total wirelength



Recovered the runtime improvement from 1.5× to ~5×

Conclusions

- Identified and quantified intra-CLB routing as a major routing bottleneck
- Proposed a Multi-Stage Router that decouples inter and intra-CLB routing
- Introduced complementary techniques to mitigate intra-CLB routing bottleneck
- Multi-Stage Router achieves 4.9× speed up compared to Single-Stage Router



Copyright and Disclaimer

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS." AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

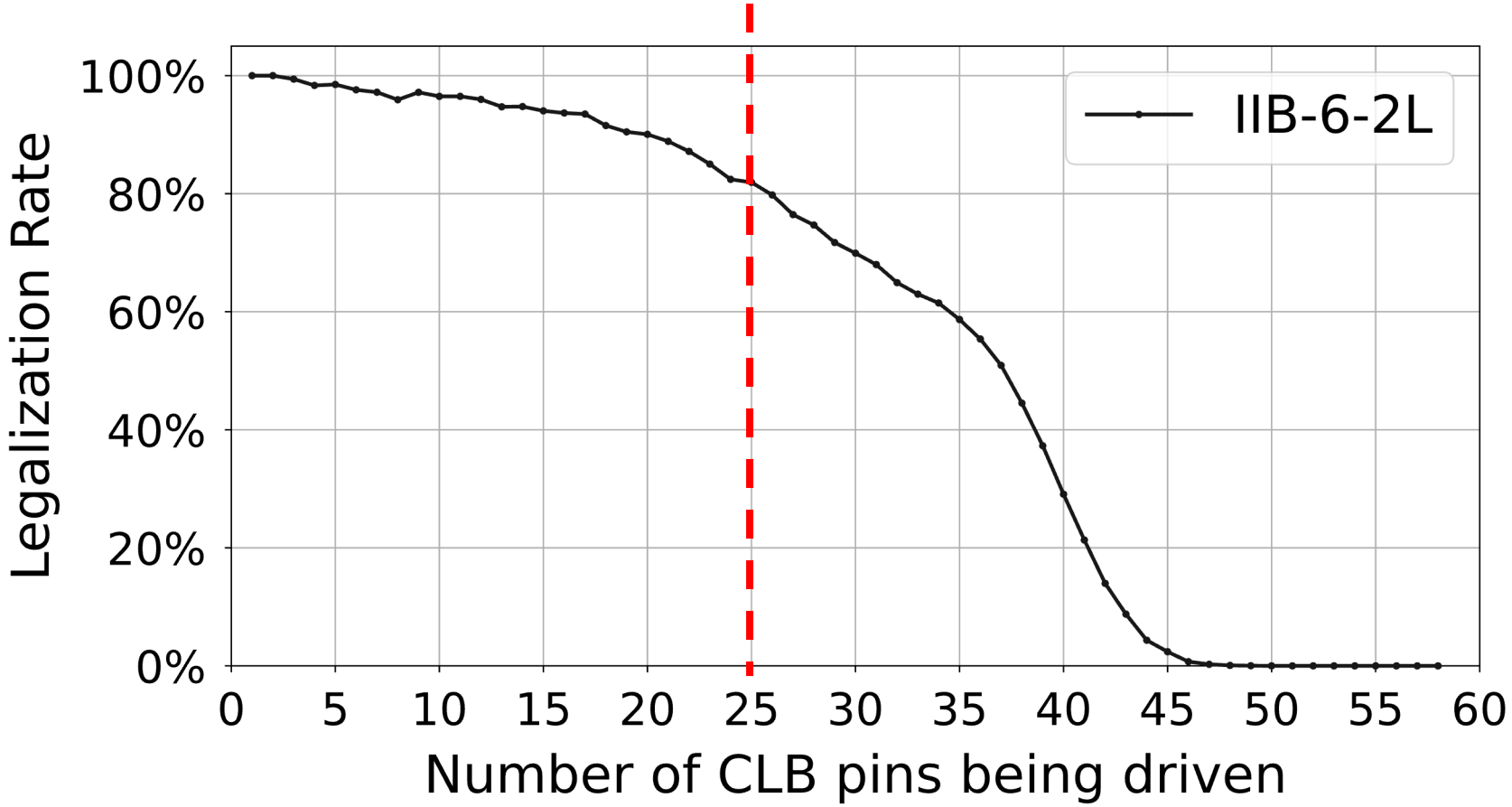
Backup Slides and Extra Slides

Runtime Comparison

TABLE II: Speedup and wirelength (WL) for different configurations of our Multi-Stage Router with respect to the baseline Single-Stage Router (Enhanced VTR, IIB-6-2L).

Benchmark	Single-Stage Router		Multi-Stage Router							
	IIB-6-2L		IIB-6-2L		IIB-6-2L, LUT-1		IIB-8-2L		IIB-8-2L, LUT-1	
	Runtime (minutes)	WL	Speedup	WL (norm.)	Speedup	WL (norm.)	Speedup	WL (norm.)	Speedup	WL (norm.)
01	2.29	601,476	1.78	0.91	3.12	0.84	4.29	0.81	**4.64**	**0.81**
02	2.85	1,017,353	2.26	0.91	2.97	0.89	3.50	0.88	**3.42**	**0.88**
03	12.82	4,757,537	1.89	0.95	2.30	0.94	2.78	0.93	**2.71**	**0.93**
04	28.09	7,772,544	1.75	0.97	2.18	0.96	3.43	0.96	**3.36**	**0.96**
05	142.55	13,006,369	1.98	0.99	2.81	0.98	4.11	0.98	**4.05**	**0.98**
06	50.74	8,817,464	1.44	0.97	2.28	0.94	4.33	0.91	**5.22**	**0.91**
07	384.13	14,034,613	1.04	0.98	1.59	0.96	4.95	0.94	**7.50**	**0.94**
08	35.46	11,433,301	2.20	0.96	2.69	0.96	3.27	0.95	**3.56**	**0.95**
09	278.95	16,930,863	1.31	0.99	1.76	0.97	4.25	0.95	**4.61**	**0.94**
10	339.90	11,935,708	1.00	0.99	3.18	0.88	5.91	0.84	**13.68**	**0.82**
11	387.78	14,617,538	1.11	0.98	1.77	0.96	5.81	0.92	**8.27**	**0.92**
Geomean	52.60	6,704,329	1.55	0.96	2.36	0.93	4.13	0.91	**4.94**	**0.91**

Legalization Rate vs Number of Driven CLB Pins



Legalization rate starts to drop significantly if more than 25 CLB pins are driven

Experimental Setup

- Intel(R) Xeon(R) CPU E5-2680 v3 (48 logical cores)
- AMD UltraScale architecture
- ISPD16 benchmarks → routability driven placements
- UTPlaceF placements
- Versatile Place and Route (VPR) tool for routing
- Non-timing driven mode
- VPR parameters
  - Fixed present congestion factor = 5
  - min incremental reroute fanout = 1

Heap pushes ↓ 2.3x
Heap pops ↓ 3.6x
Runtime ↓ 4.9x

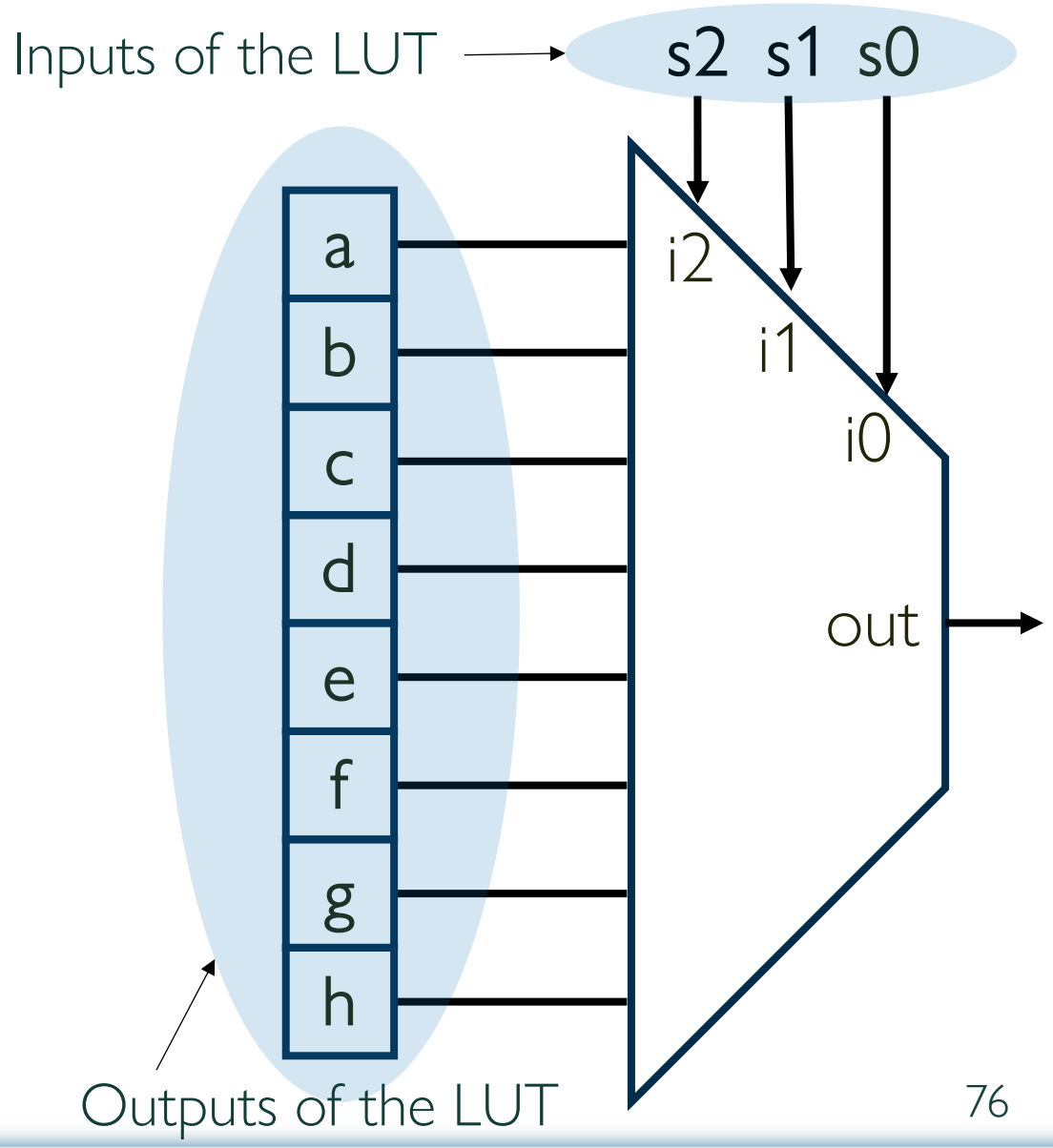
IV. Enhancing the Baseline

To ensure a comprehensive evaluation of the runtime improvements discussed in this paper, a fair baseline is required. Through experimentation, we have found that adjusting some of the VPR router parameters reduces its runtime. This section explains our findings.

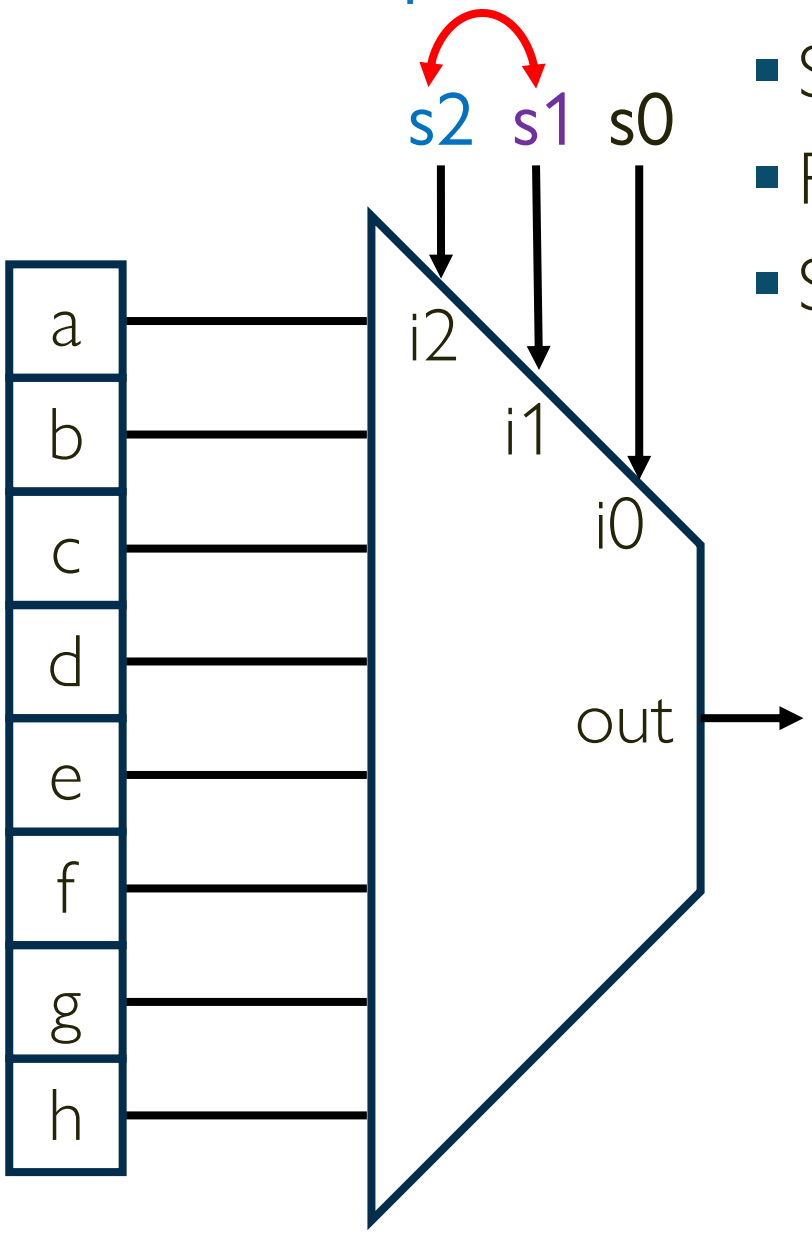
During routing, nodes of the RRG are assigned costs which help resolve congestion by increasing the cost of congested nodes and determining the shortest path by considering lower-cost nodes. In VPR, the cost is a function of present and historical congestion; the former is updated after a signal is routed, while the latter is updated between two subsequent routing iterations. The overall cost of a node, which determines where the node will be inserted in the ordered heap, depends not only on the two previously mentioned factors but also on a *router lookahead*—that is, the A* heuristic. The lookahead estimates the cost of a path from a node until the target pin. The node

Look Up Table (LUT)

- Physical implementation → MUX
- For example: 3-input LUT
- **Interesting property**: LUT pin equivalence
  - All input pins → logically equivalent

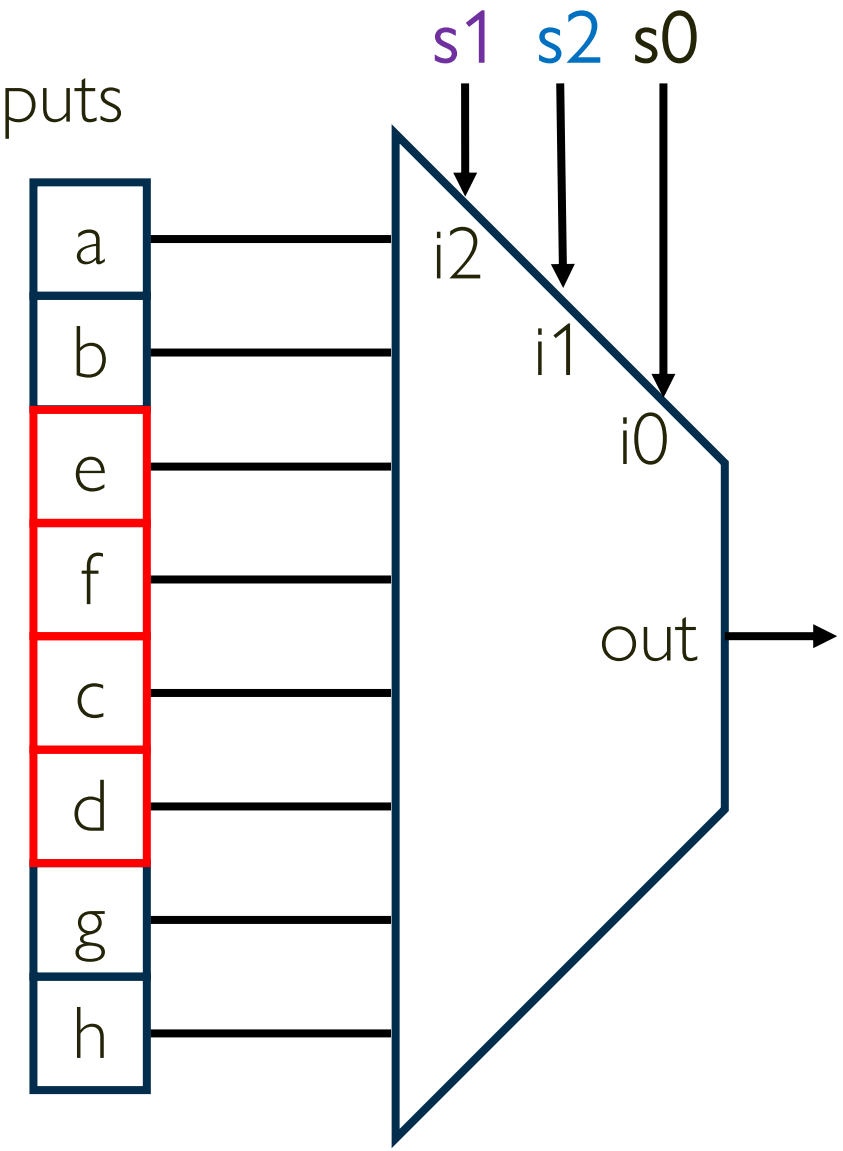


LUT Pin Equivalence



- Swap signals
- Rearrange stored outputs
- Same functionality

=



Comparative Analysis

- Perform inter-CLB after intra-CLB routing [Moctar et al., FPL'12]
  - Fixes the entry points at IIB → makes inter-CLB routing harder

  Our approach fixes entry points at IIB, but for intra-CLB routing → which are smaller and parallelizable problems

- Parallelizing intra-CLB [Wang et al., ASPDAC'23]
  - Intra-CLB routing dependent on each other → not fully parallelizable

  Our intra-CLB routing fully parallelizable

- Other parallel routing approaches
  - Workload-partitioning method
  - Too many conflicting nets → limits scalability

  Our intra-CLB routing fully scalable

Future Work

- Explore other IIB architectures conducive for Multi-Stage Router
- Evaluate the performance in timing-driven mode
- Placement optimizations improving pin density and load balancing on bypass mux

Why not use full IIB?

- Each channel → 64 wires connecting to IIB
- Each CLB → 8 LUT, 16 FFs → 32 output pins
- 96:1 mux → connect each channel wires and CLB outputs to a CLB input pin

\+ High routing flexibility

\- High capacitive load → ↑ wire delay → slow circuits

Full IIB makes the circuits slow

Multi-Stage Router with LUT-1

Geomean = 2.6x