Routing Algorithms for FPGAs with Sparse Intra-cluster Routing Crossbars

Yehdhih Ould Mohammed Moctar[1]

Guy G.F. Lemieux[2]   Philip Brisk[1]

[1]University of California Riverside

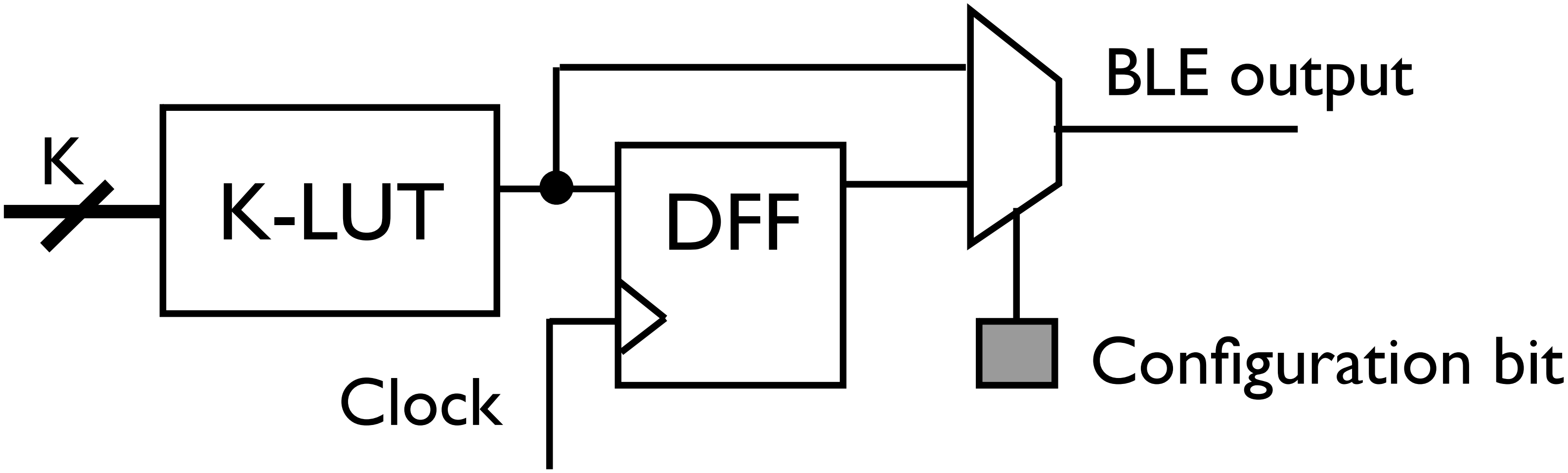
[2]University of British Columbia

22nd International Conference on Field Programmable Logic and Applications
Oslo, Norway, August 29-31, 2012

Outline

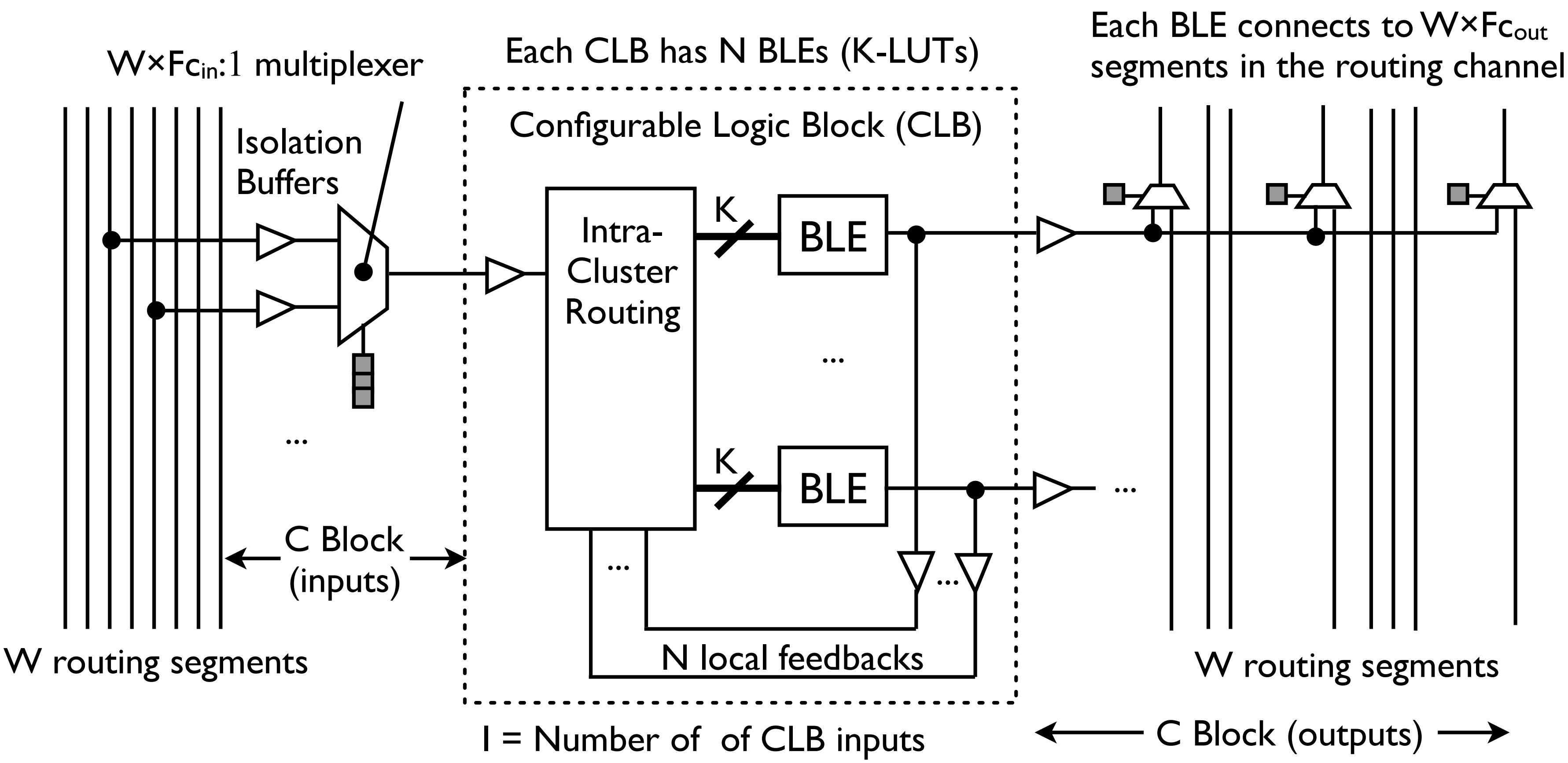
- Background
  - FPGA architecture
  - FPGA routing
- FPGAs w/full intra-cluster routing crossbars
  - Implications for routing
- FPGAs w/sparse intra-cluster routing crossbars
  - Routing challenges
  - New routing algorithms
- Experimental Results
- Conclusion

FPGA Architecture (1/3)



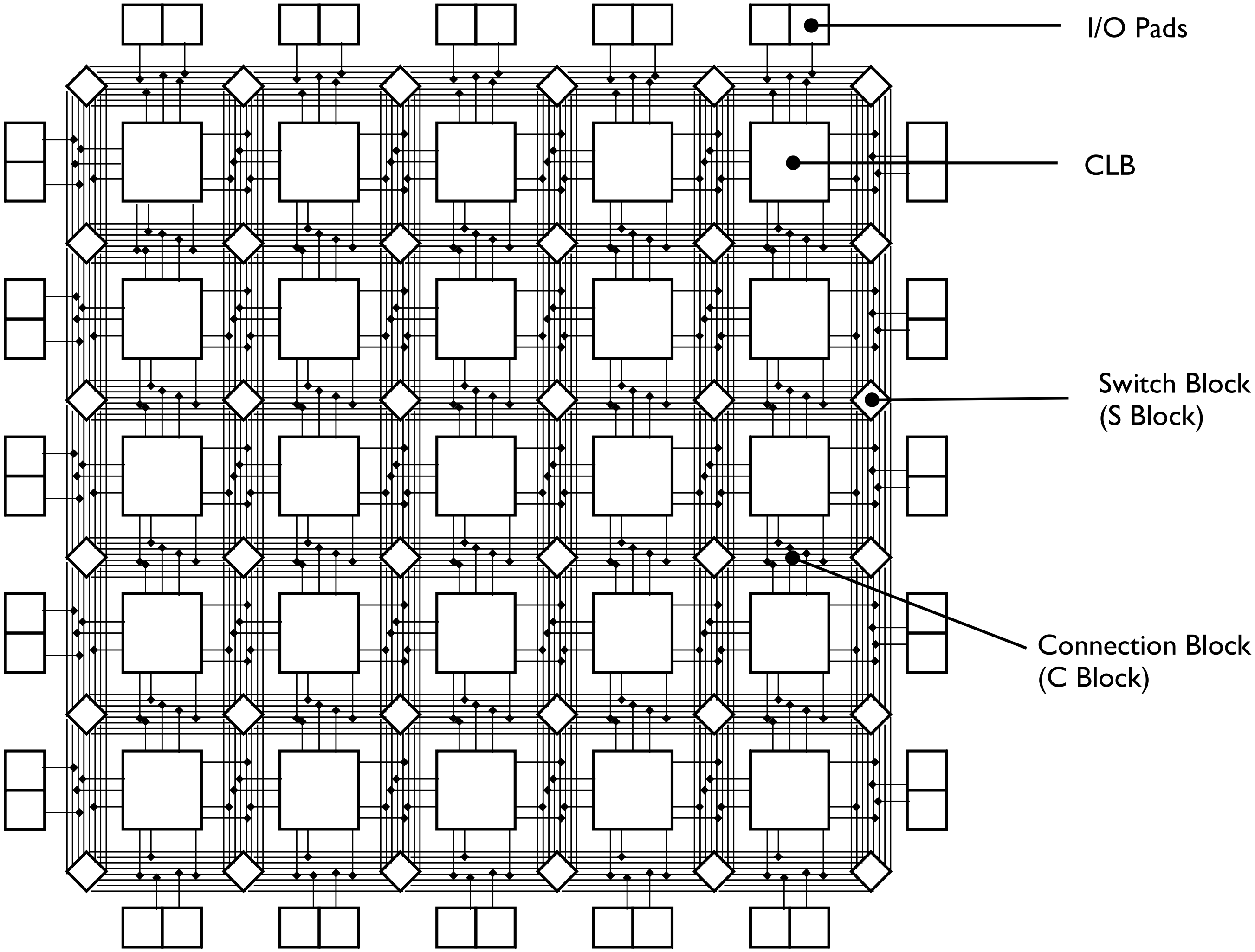
Basic Logic Element (BLE)

FPGA Architecture (2/3)

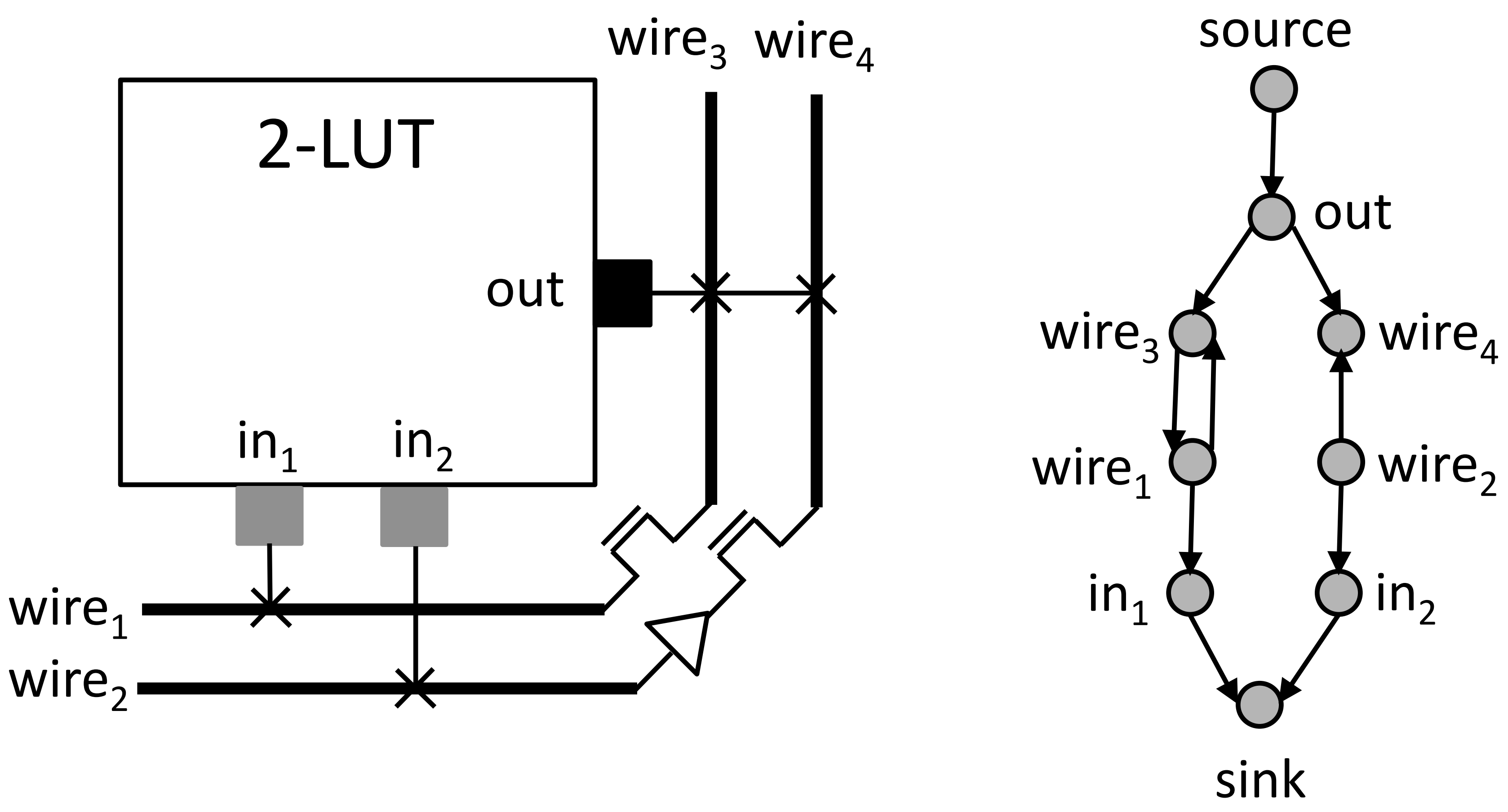

W×Fc$_{in}$:1 multiplexer
Each CLB has N BLEs (K-LUTs)
Each BLE connects to W×Fc$_{out}$ segments in the routing channel
Isolation Buffers
Configurable Logic Block (CLB)
Intra-Cluster Routing
K
BLE
...
K
BLE
...
...
C Block (inputs)
...
...
W routing segments
N local feedbacks
W routing segments
I = Number of of CLB inputs
C Block (outputs)

Versatile Place and Route (VPR) CLB Architecture

FPGA Architecture (3/3)



Routing Resource Graph (RRG)



www.eecg.toronto.edu/~aling/ece1718/project/fang/route_rr_graph.png

FPGA Routing

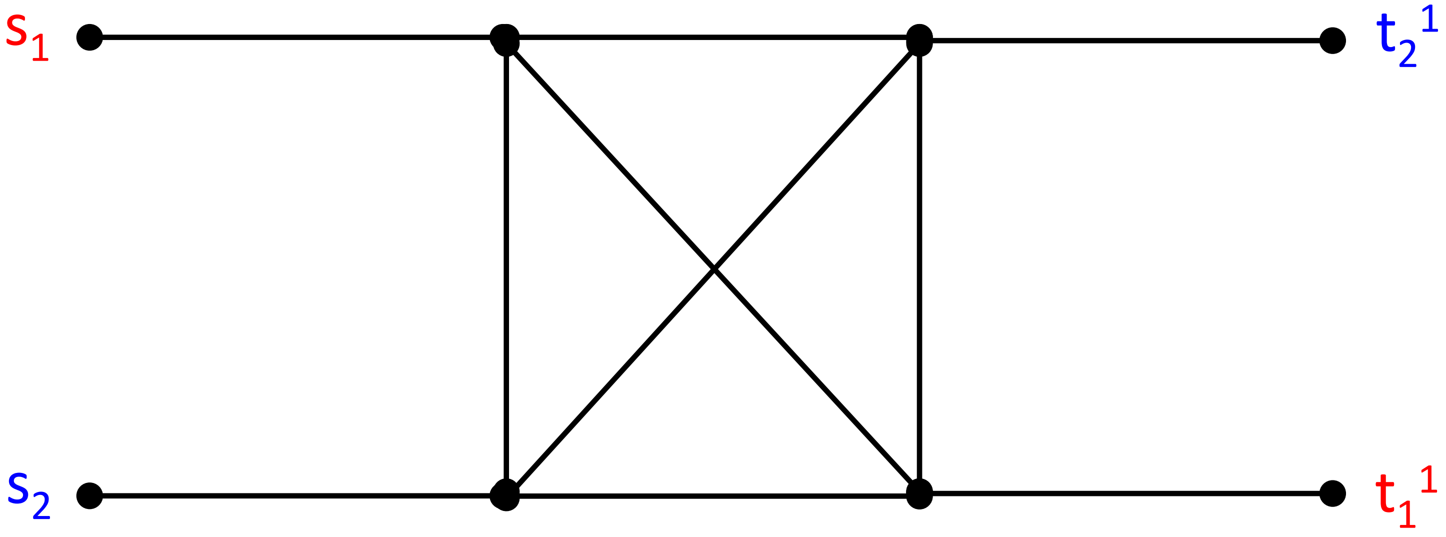
- Disjoint-path problem (on the RRG); NP-complete

- Input:
  - Graph G(V, E)
  - Set of sources S = {s_1, s_2, …, s_m}
  - Set of sets of sinks T = {T_1, T_2, …, T_n}, T_i = {t_i^1, t_i^2, …, t_i^k}

- Solution
  - Finds paths from each source s_i to all sinks in T_i
  - Paths emanating from different sinks must be disjoint (cannot shared any vertices or edges)

- Objective(s)
  - Minimize delay, wirelength, etc.

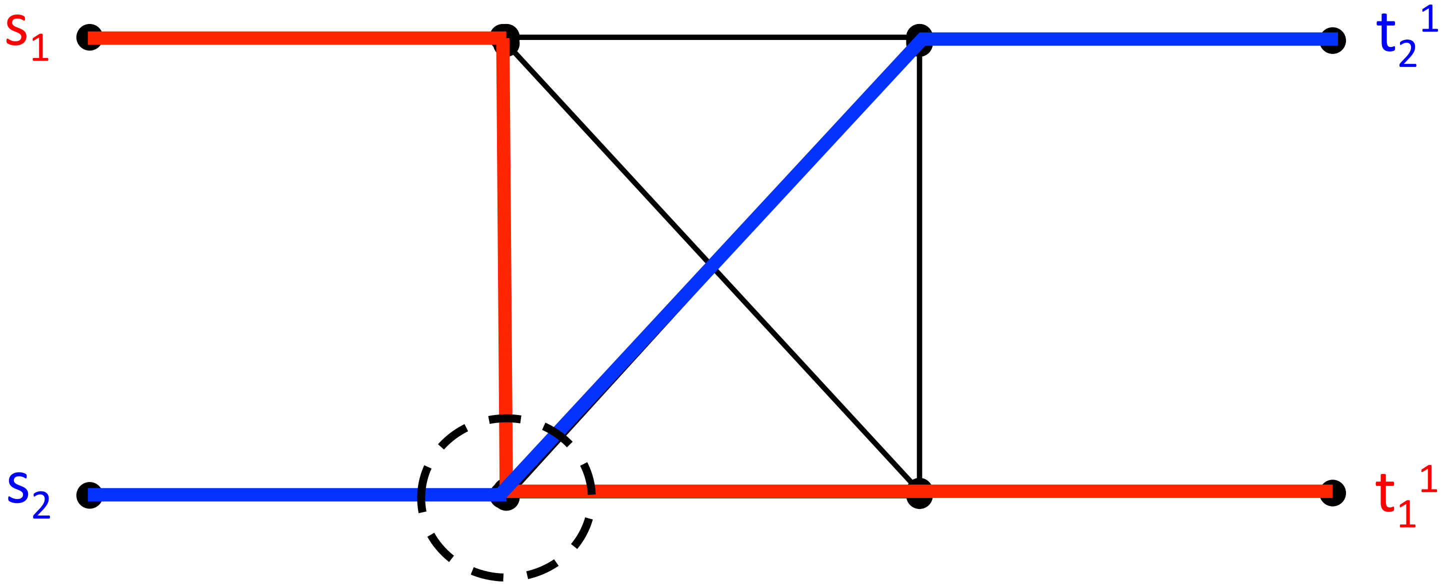
Disjoint and Non-disjoint Paths



Disjoint and Non-disjoint Paths

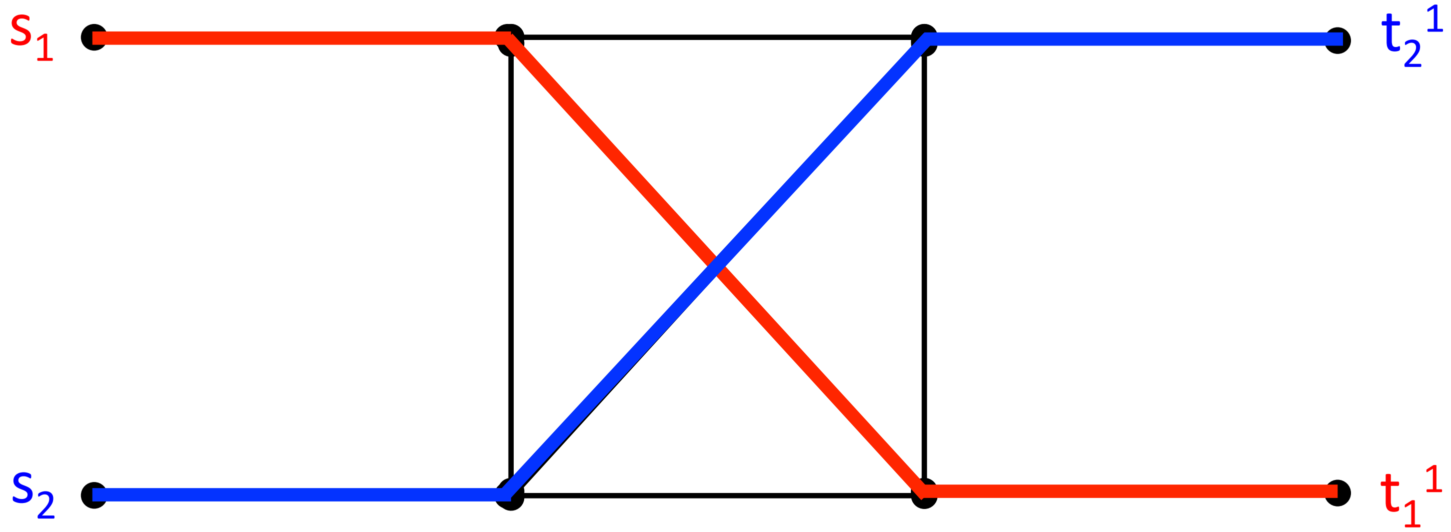
Two nets are routed on the same FPGA segment

(remember, RRG vertices represent wires and CLB I/Os)



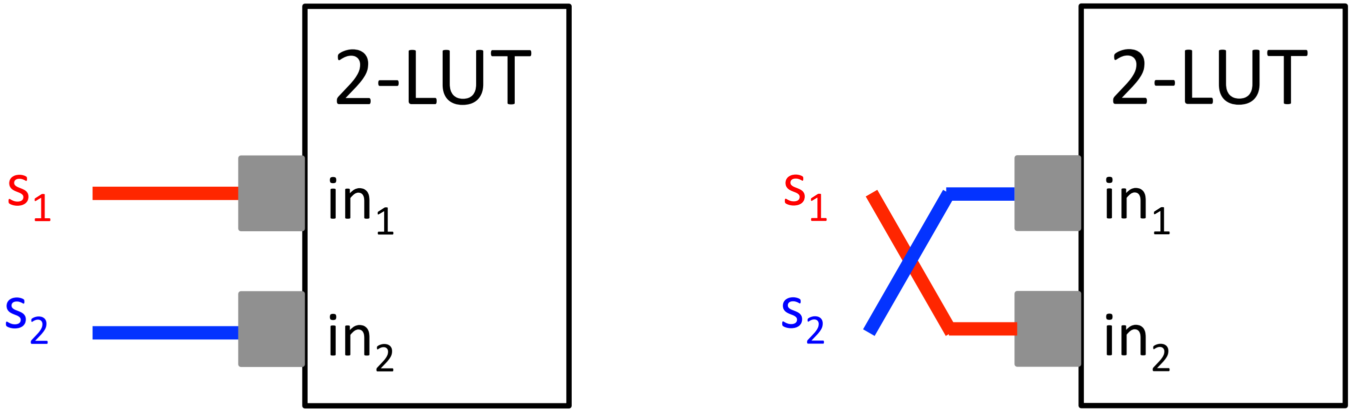
This route is illegal!

Disjoint and Non-disjoint Paths



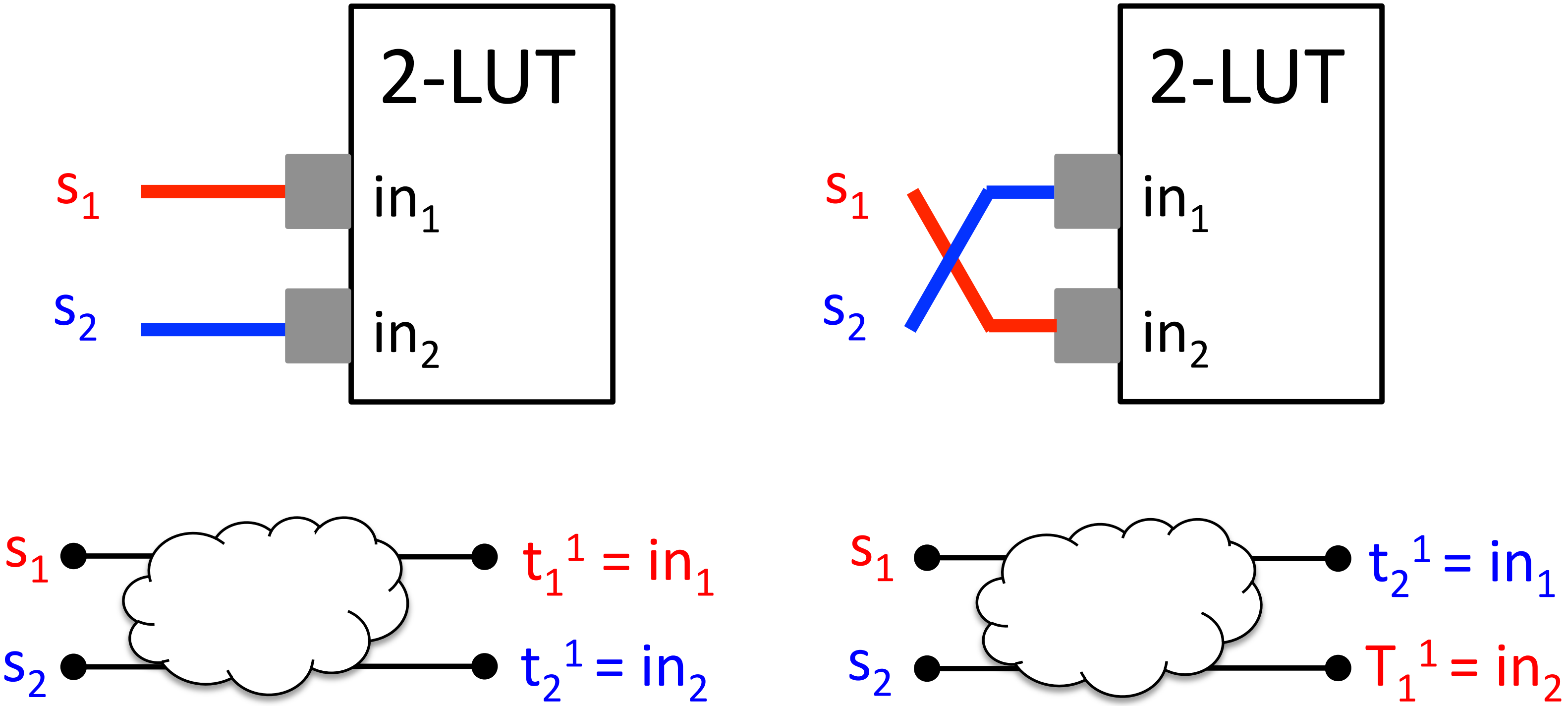
This route is legal!

LUT Input Equivalence (1/3)



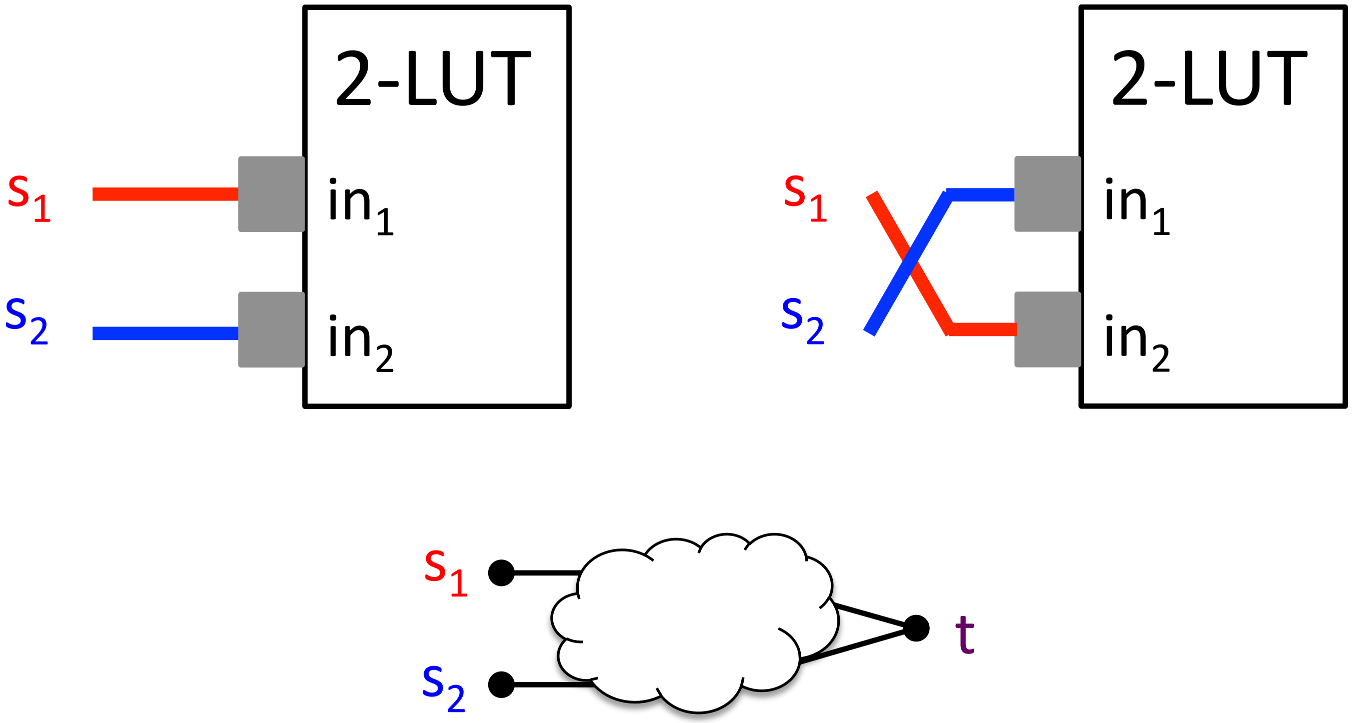
LUT inputs are interchangeable

LUT Input Equivalence (2/3)



Overly restrictive disjoint-path problem formulation

LUT Input Equivalence (3/3)

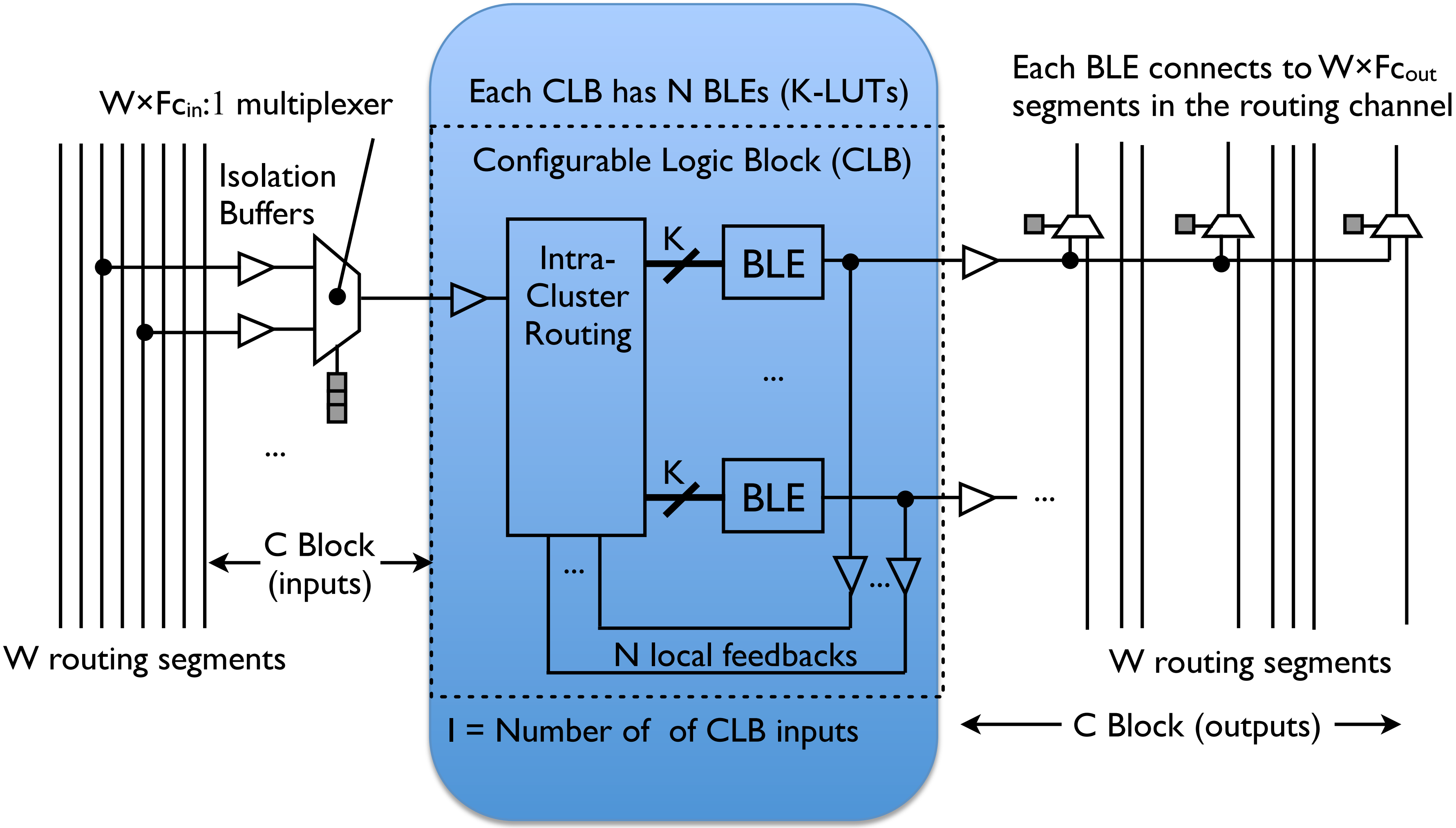


Represent all inputs of a LUT as one RRG sink, t

Outline

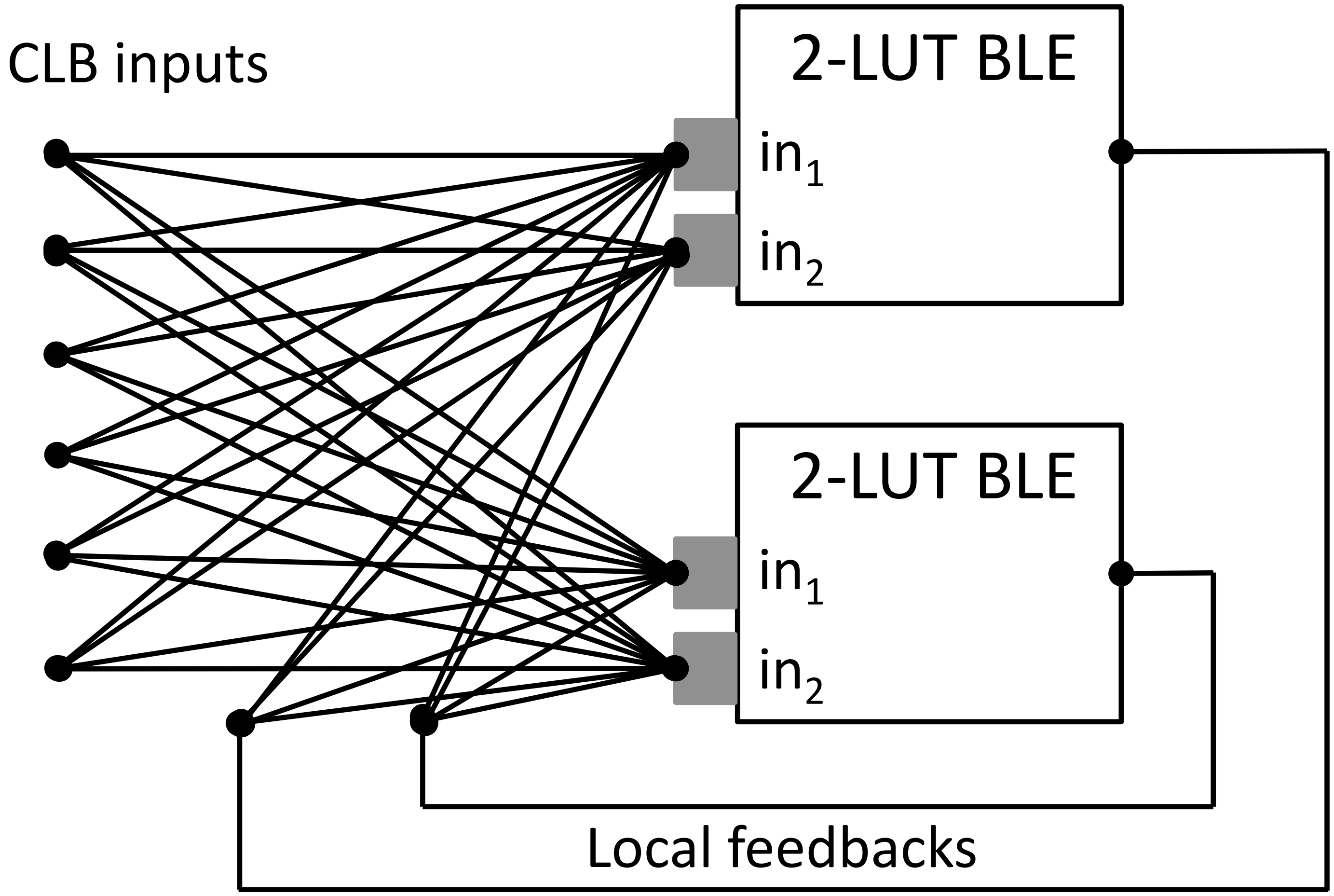
- Background
  - FPGA architecture
  - FPGA routing
- FPGAs w/full intra-cluster routing crossbars
  - Implications for routing
- FPGAs w/sparse intra-cluster routing crossbars
  - Routing challenges
  - New routing algorithms
- Experimental Results
- Conclusion

CLB Architecture (Recap)


W×Fc$_{in}$:1 multiplexer
Isolation Buffers
…
C Block (inputs)
W routing segments
Each CLB has N BLEs (K-LUTs)
Configurable Logic Block (CLB)
Intra-Cluster Routing
K
BLE
…
K
BLE
…
…
N local feedbacks
I = Number of of CLB inputs
Each BLE connects to W×Fc$_{out}$ segments in the routing channel
…
W routing segments
C Block (outputs)

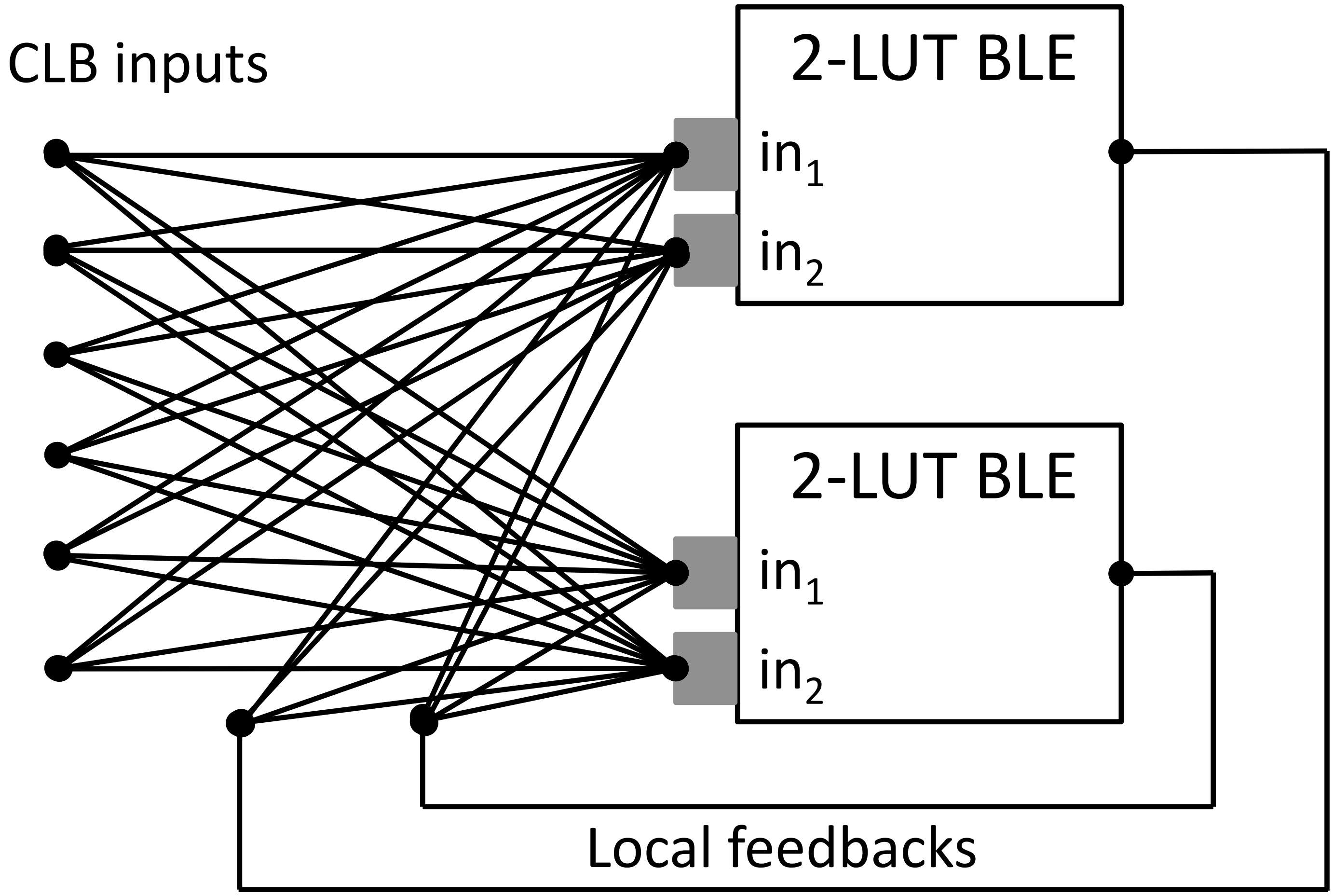
Versatile Place and Route (VPR) CLB Architecture

Full Crossbar Intra-cluster Routing (1/6)



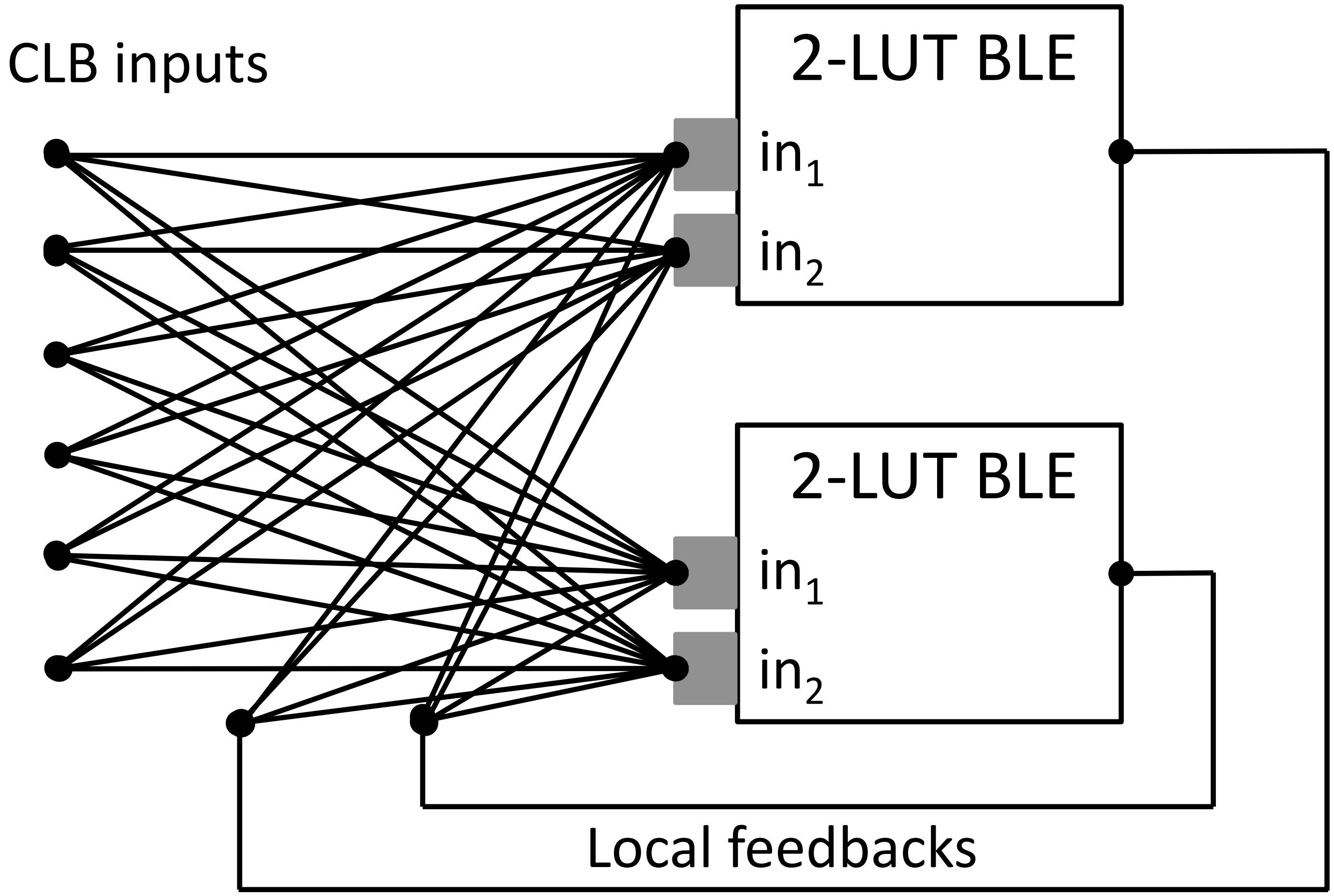
- Typical of FPGAs 10+ years ago

Full Crossbar Intra-cluster Routing (2/6)



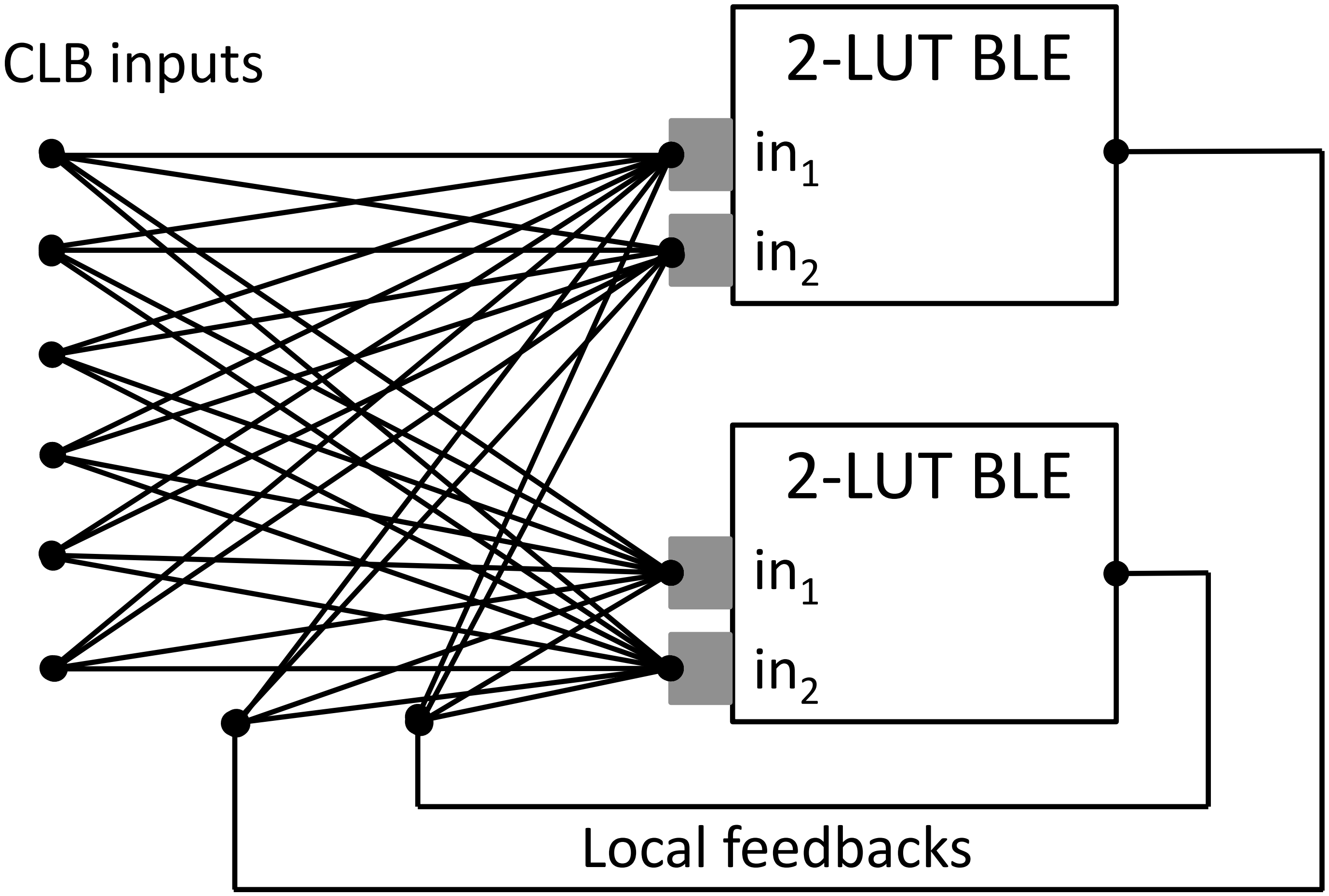
- Each CLB input connects to each BLE input

Full Crossbar Intra-cluster Routing (3/6)



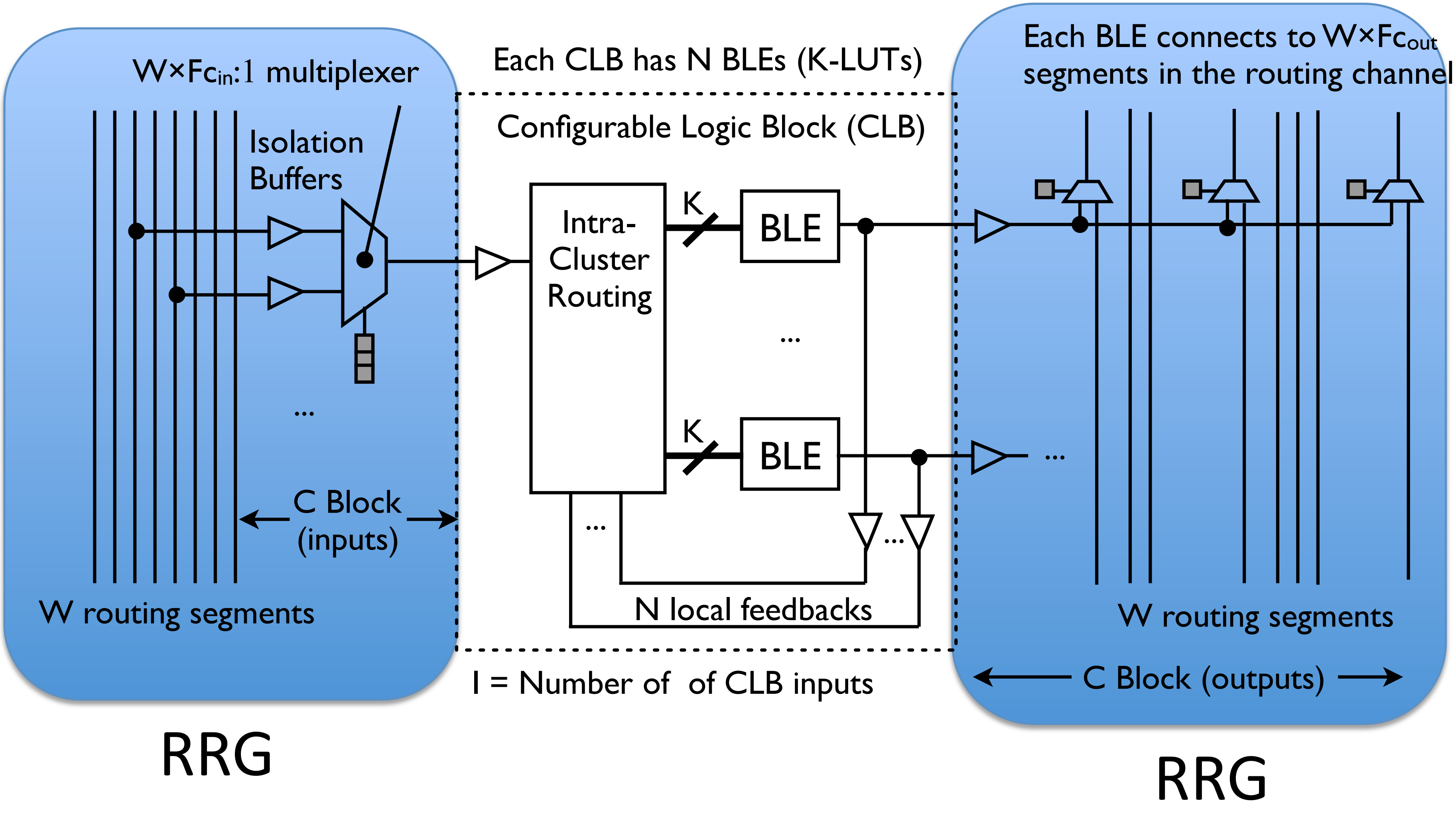
- CLB inputs are equivalent

Full Crossbar Intra-cluster Routing (4/6)

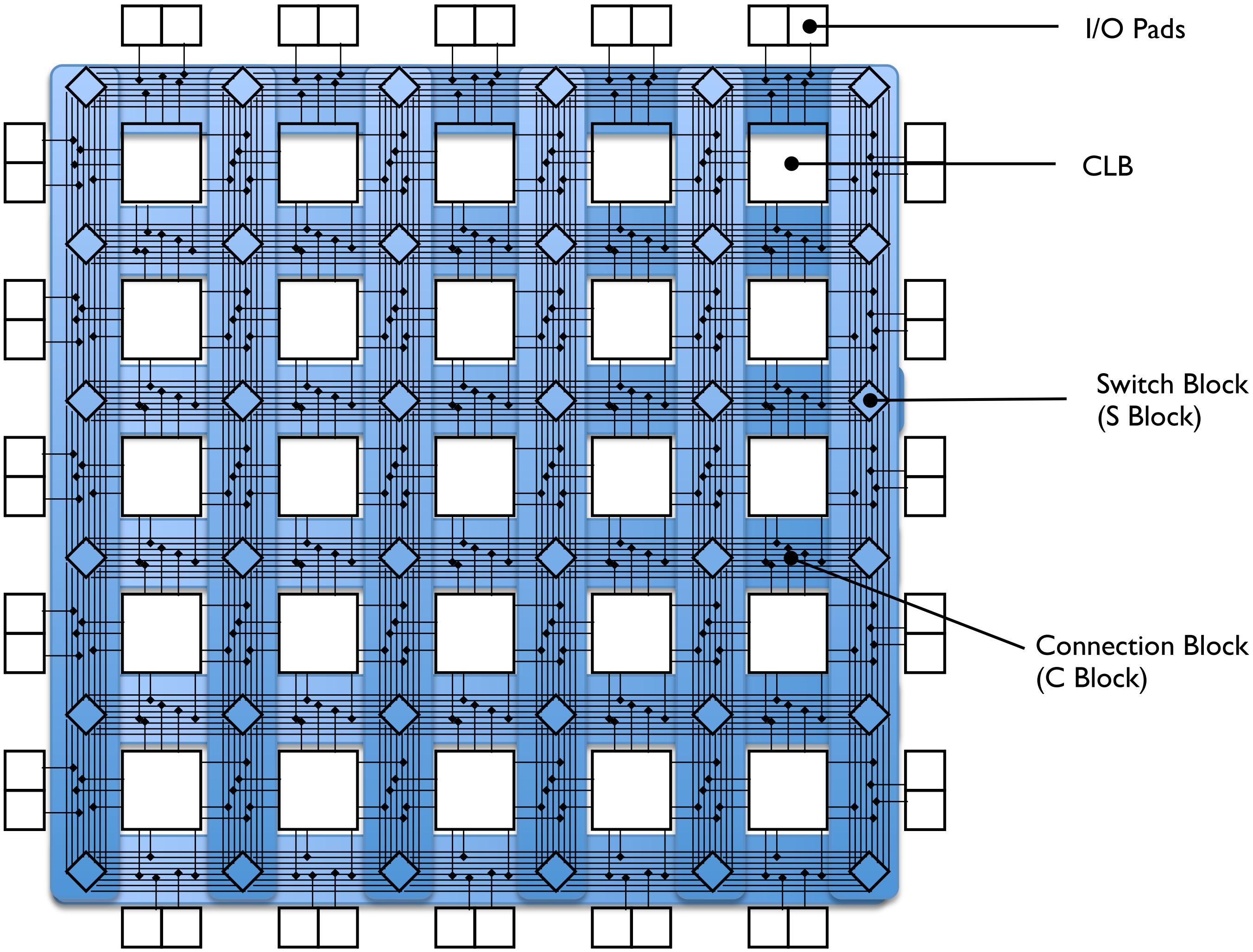


- No need to model intra-cluster routing in the RRG

Full Crossbar Intra-cluster Routing (5/6)


W×Fc$_{in}$:1 multiplexer
Isolation Buffers
…
C Block (inputs)
W routing segments
RRG
Each CLB has N BLEs (K-LUTs)
Configurable Logic Block (CLB)
Intra-Cluster Routing
K
BLE
…
K
BLE
…
…
N local feedbacks
I = Number of of CLB inputs
Each BLE connects to W×Fc$_{out}$ segments in the routing channel
…
W routing segments
C Block (outputs)
RRG

Full Crossbar Intra-cluster Routing (6/6)

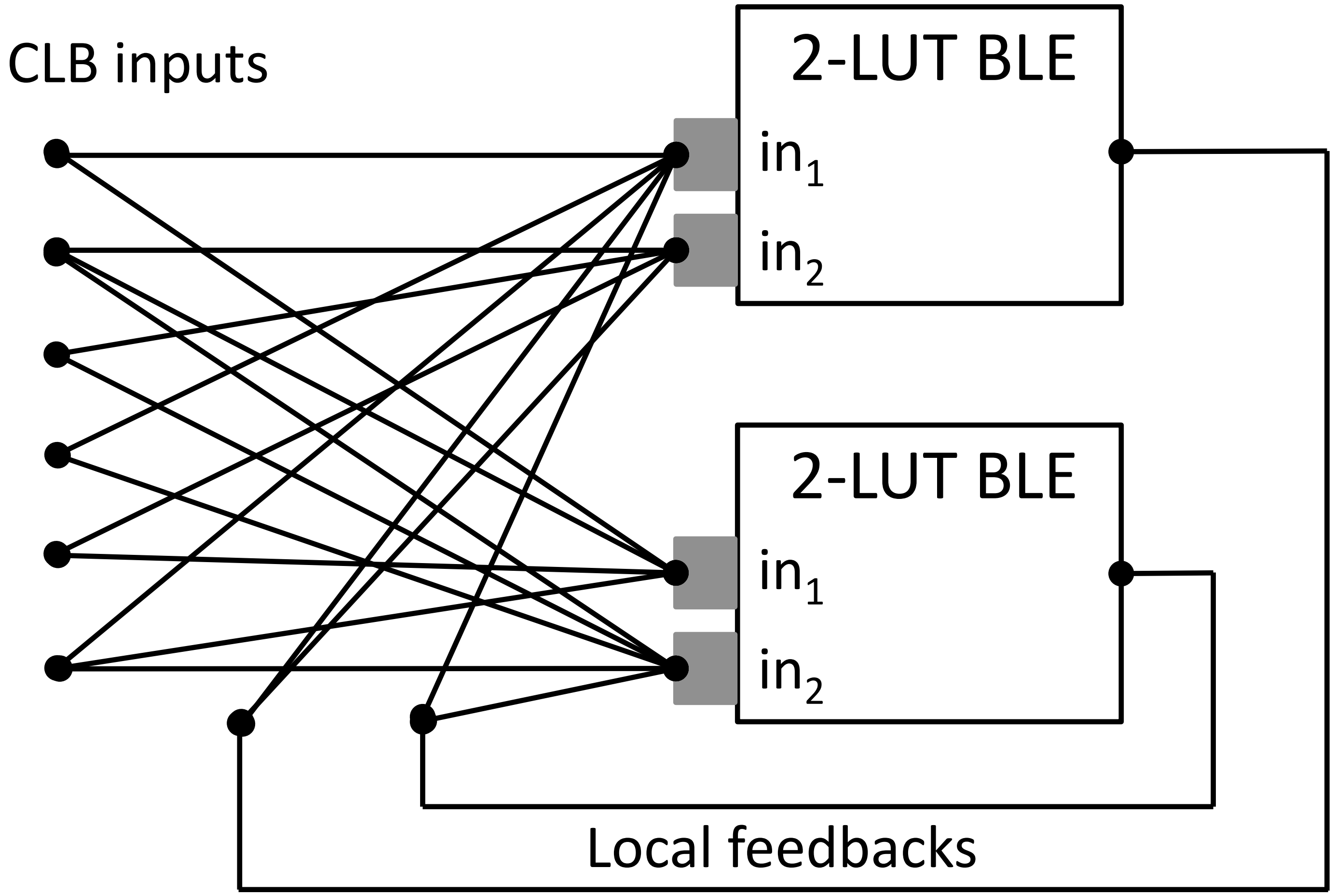


Don’t model the RRG inside each CLB!

Outline

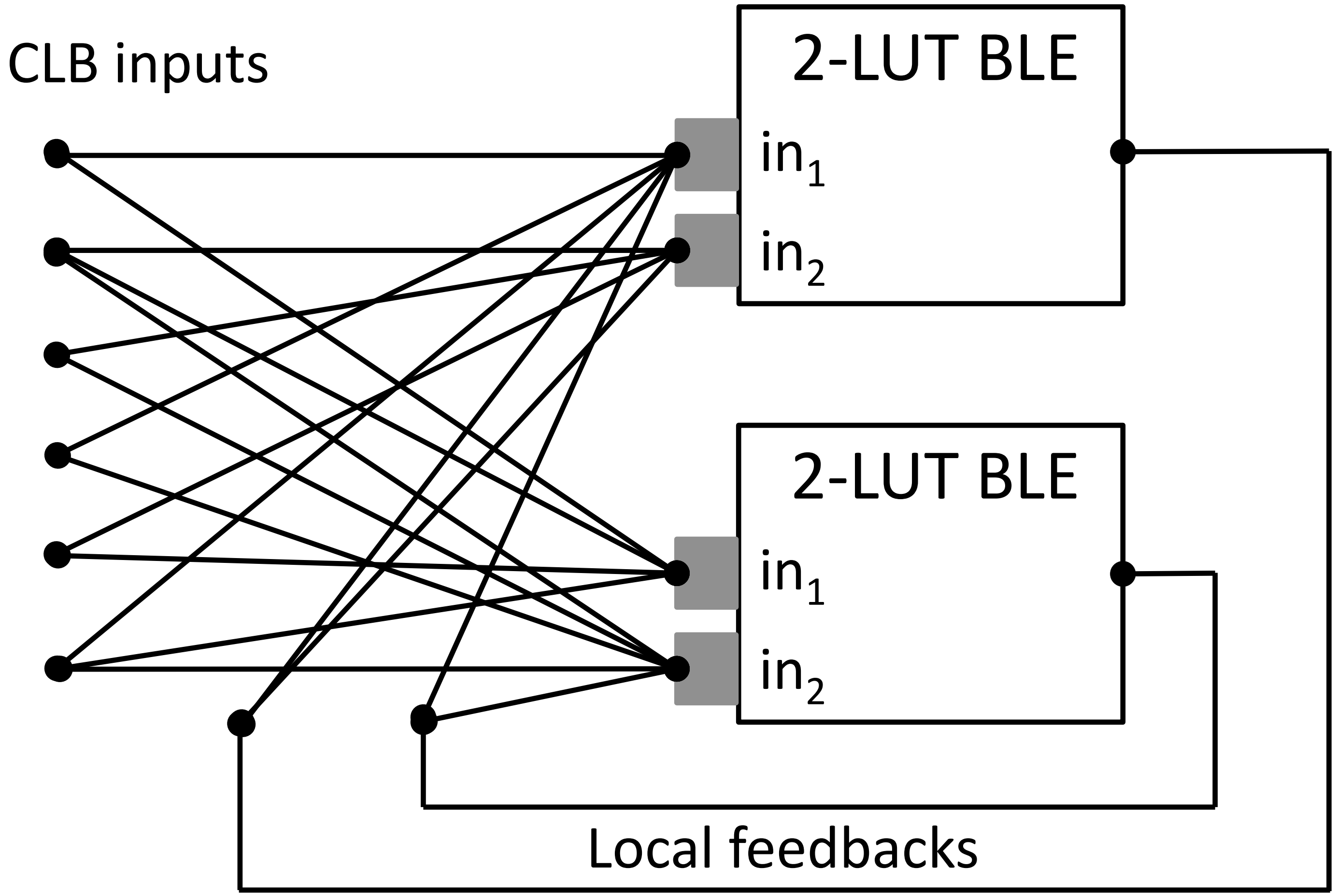
- Background
  - FPGA architecture
  - FPGA routing
- FPGAs w/full intra-cluster routing crossbars
  - Implications for routing
- FPGAs w/sparse Intra-cluster routing crossbars
  - Routing challenges
  - New routing algorithms
- Experimental Results
- Conclusion

Sparse Crossbar Intra-cluster Routing (1/6)



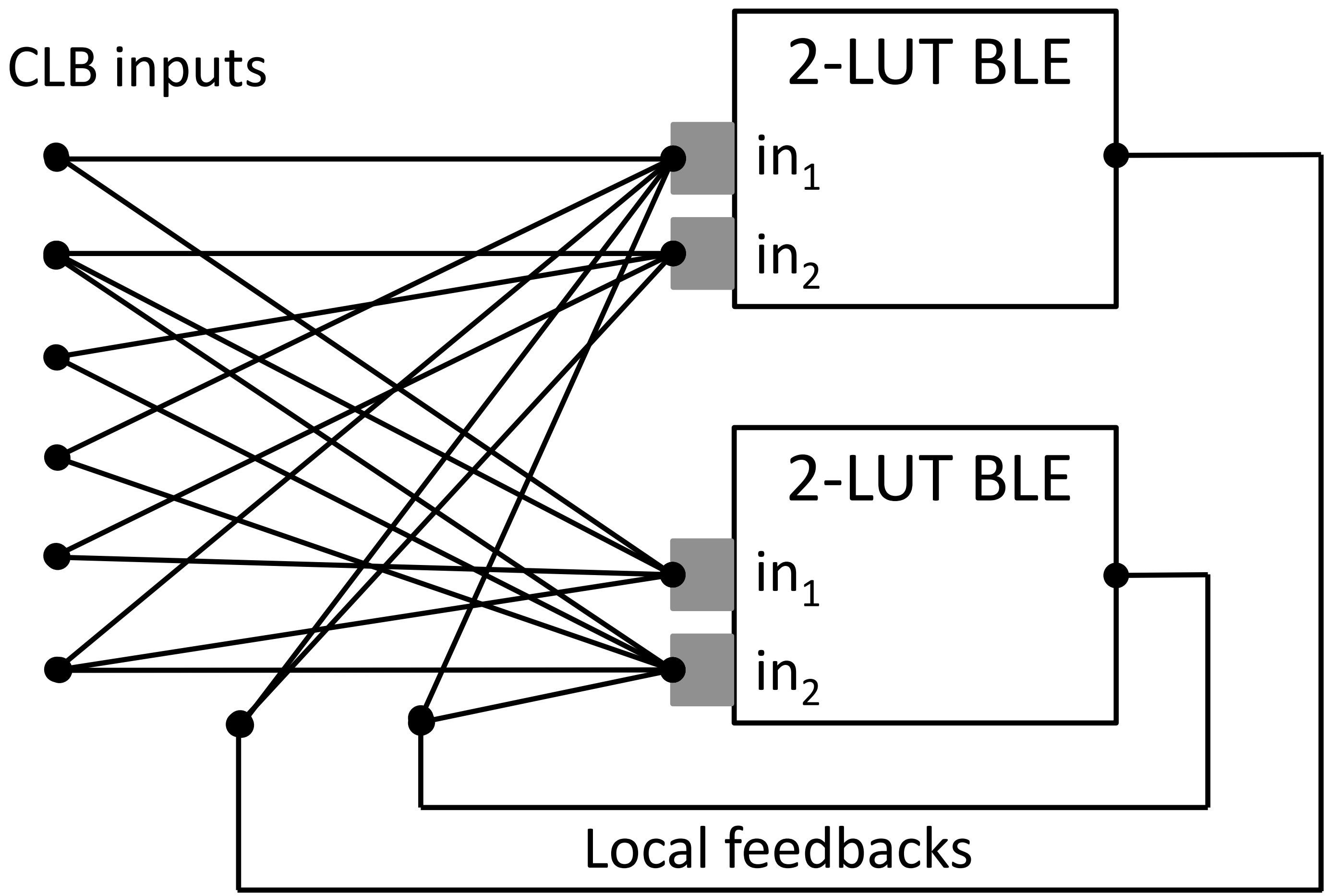
- Most FPGAs today

Sparse Crossbar Intra-cluster Routing (2/6)



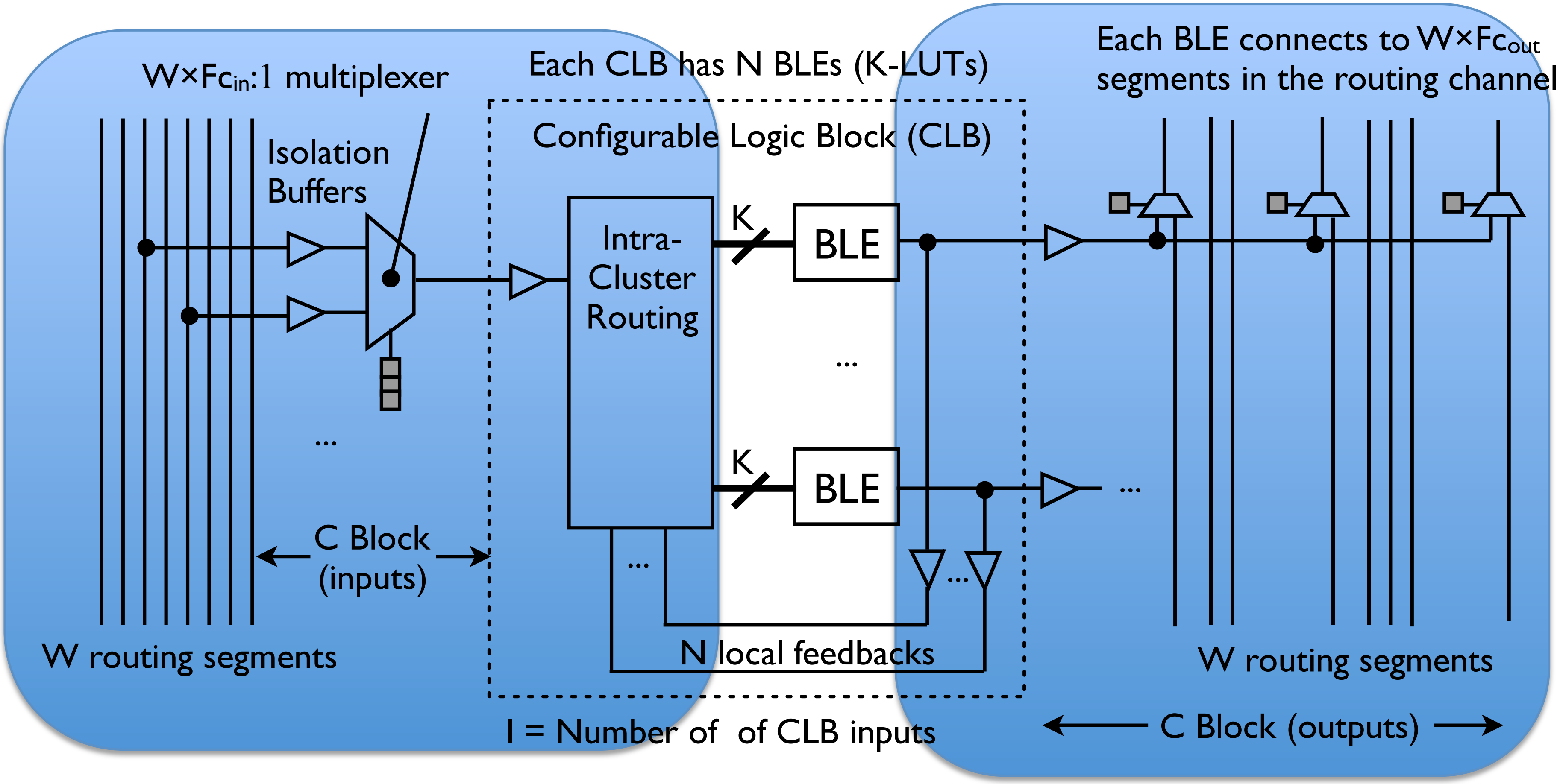
- CLB inputs are not equivalent!

Sparse Crossbar Intra-cluster Routing (3/6)



- RRG must include the intra-cluster routing

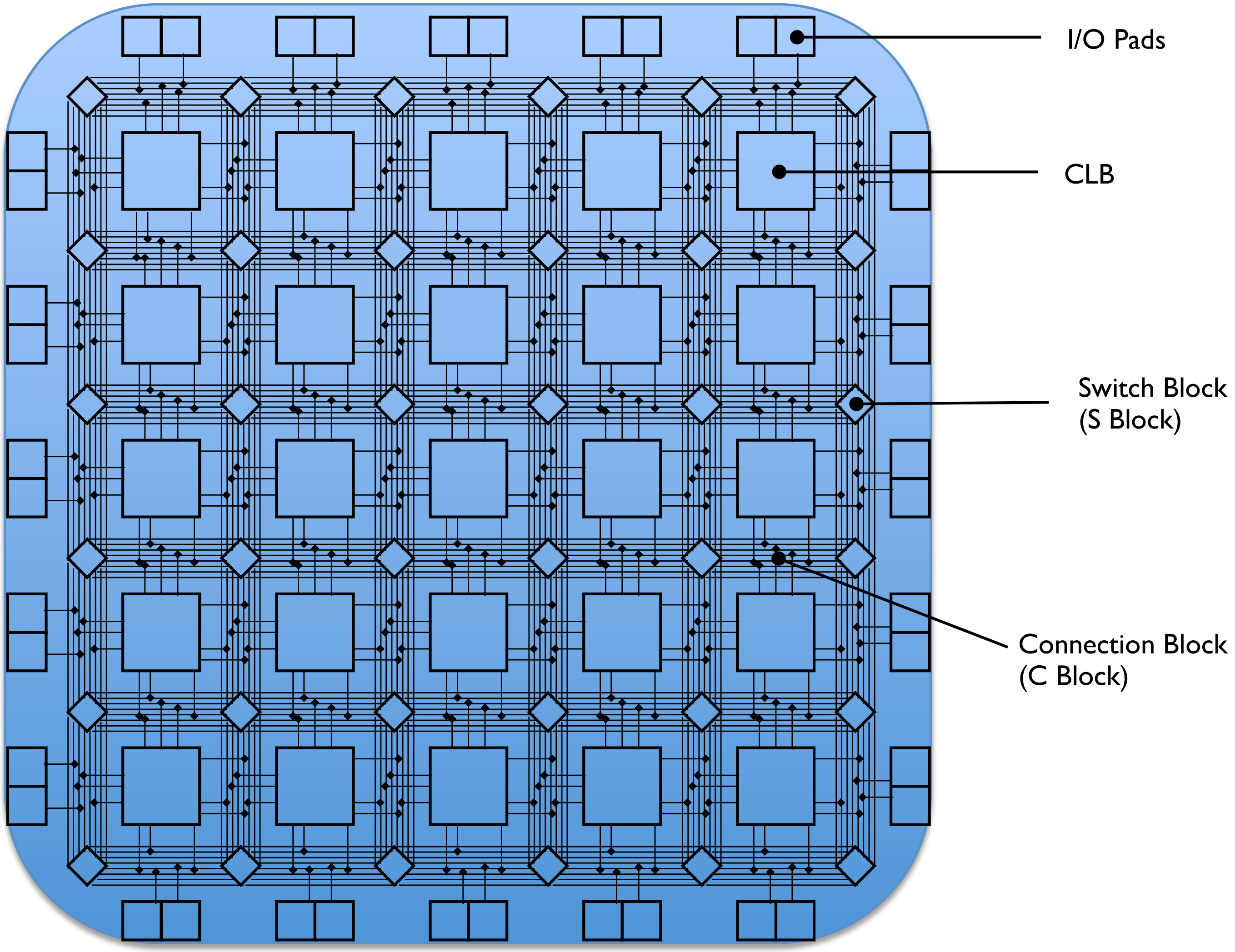
Sparse Crossbar Intra-cluster Routing (4/6)



W×Fc$_{in}$:1 multiplexer
Each CLB has N BLEs (K-LUTs)
Each BLE connects to W×Fc$_{out}$ segments in the routing channel
Configurable Logic Block (CLB)
Isolation Buffers
Intra-Cluster Routing
K
BLE
...
K
BLE
...
C Block (inputs)
...
...
...
W routing segments
N local feedbacks
W routing segments
I = Number of of CLB inputs
C Block (outputs)

RRG

Sparse Crossbar Intra-cluster Routing (5/6)



RRG encompasses the entire FPGA

Sparse Crossbar Intra-cluster Routing (6/6)

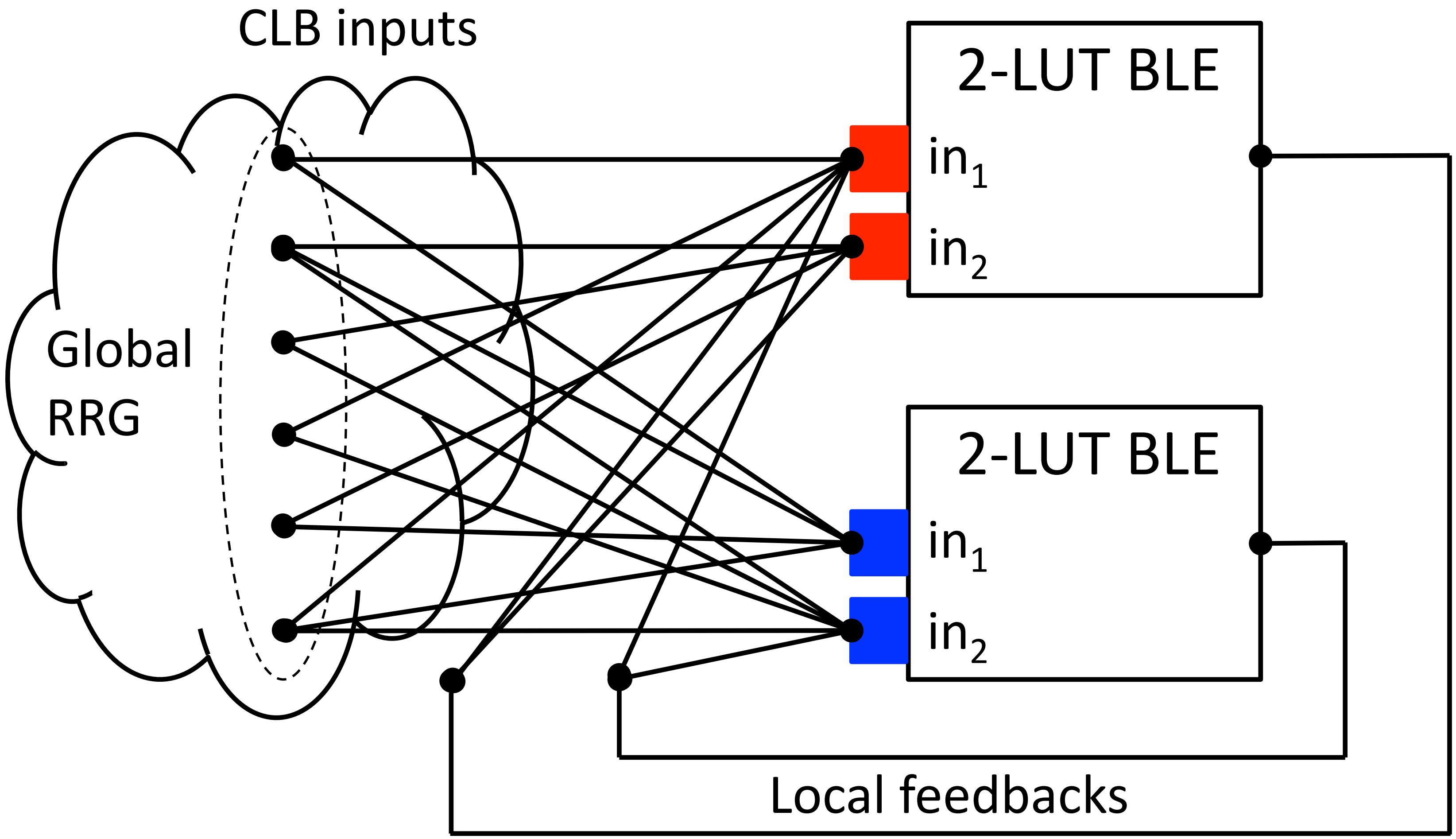
- Key Issues relating to RRG size
  - May run out of memory
    - If you are <u>not</u> routing in the cloud
  - Long router runtimes
    - Large search space leads to slow convergence
    - Thrashing

- Objective
  - Route without expanding the whole RRG

Routing Algorithms

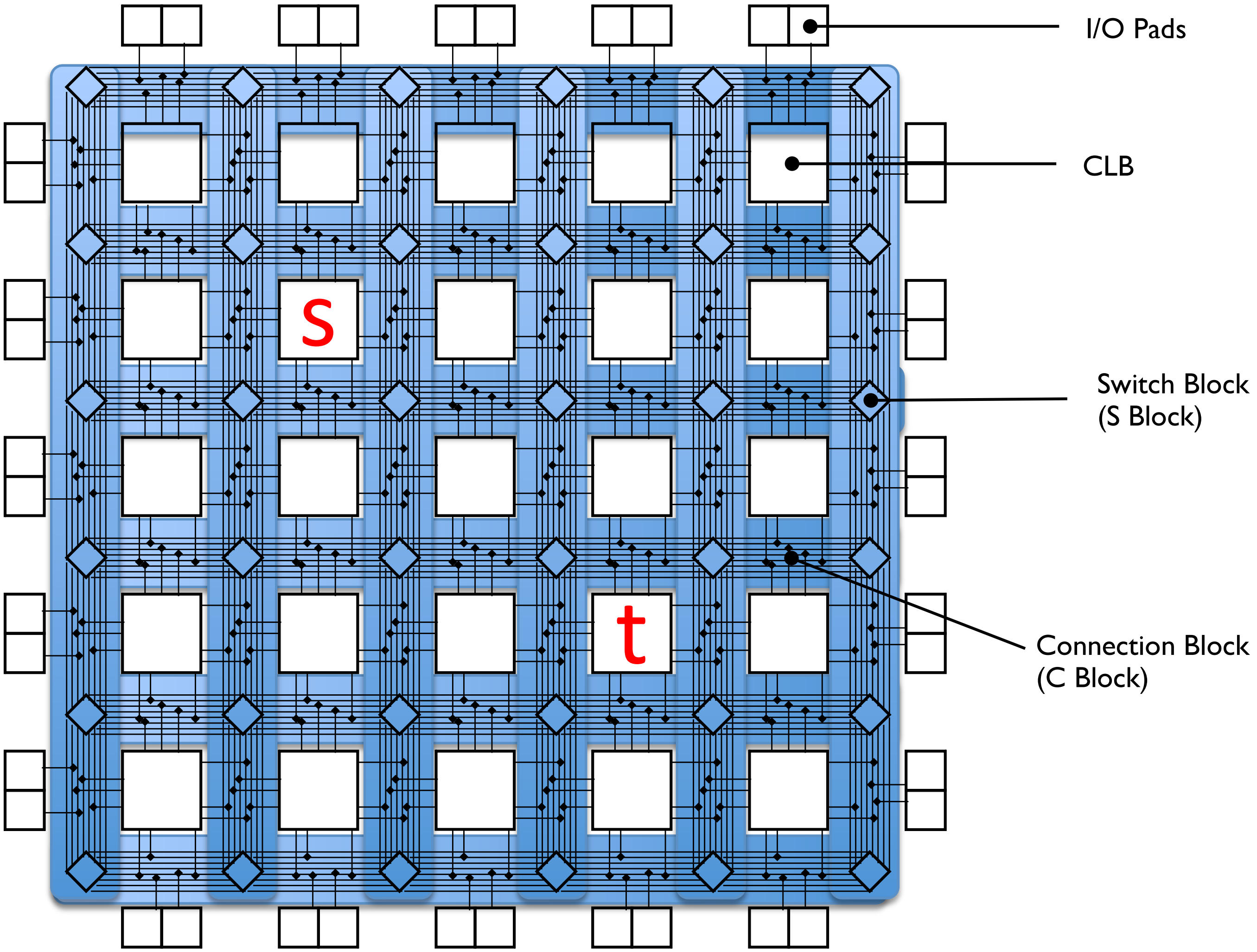
- Extensions to PathFinder routing algorithm
  - Routes one net at a time via wavefront expansion
- Baseline
  - Extend the RRG with intra-cluster routing at each CLB
- Selective RRG Expansion (SERRGE)
  - Route nets to CLB inputs
  - Dynamically expand the RRG to finish the route
- Partial Pre-Routing (PPR)
  - Pre-route all nets within CLBs
  - Complete global routes to CLB inputs

Baseline Router



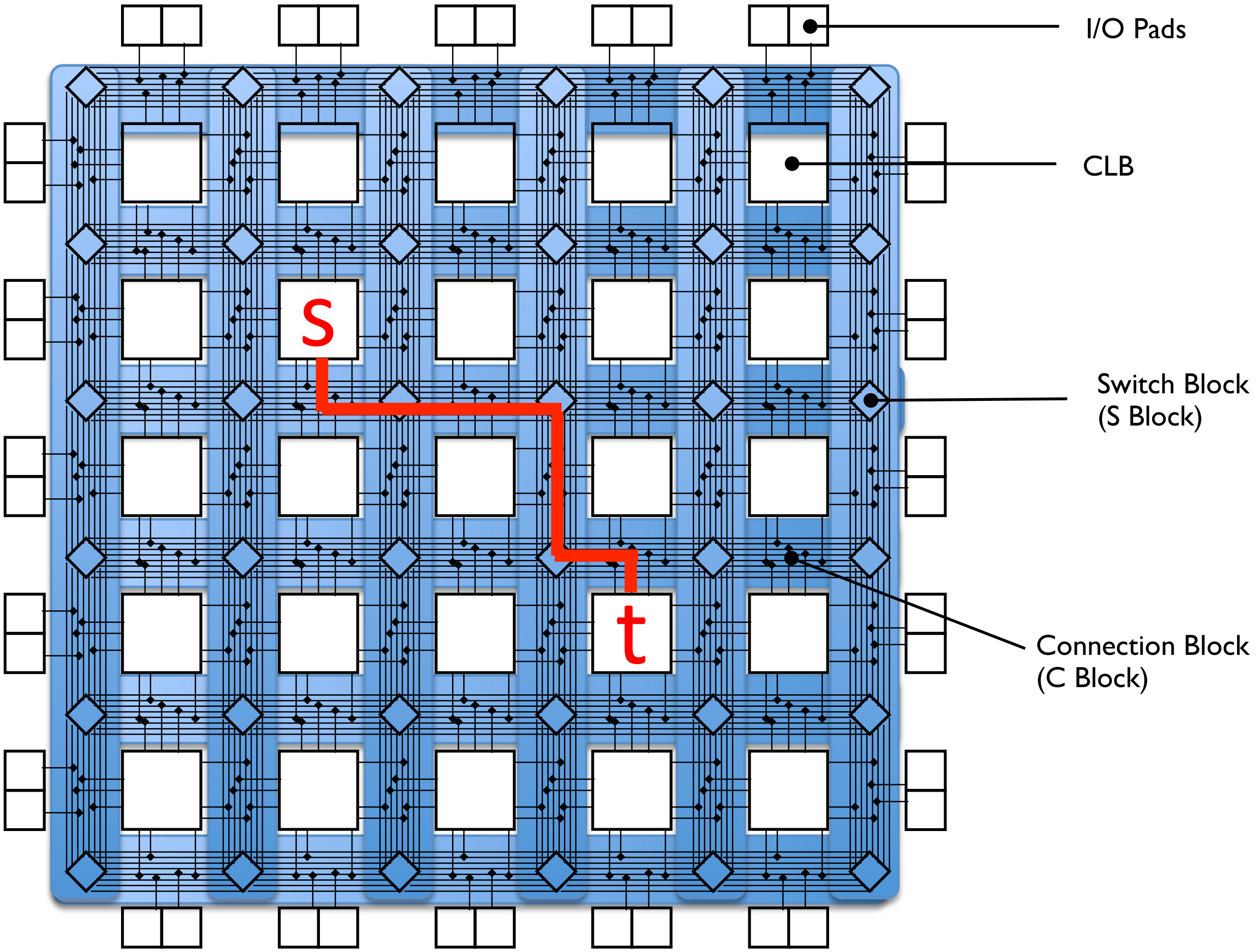
- Inputs of the same LUT are equivalent!

SERRGE in Action (1/8)



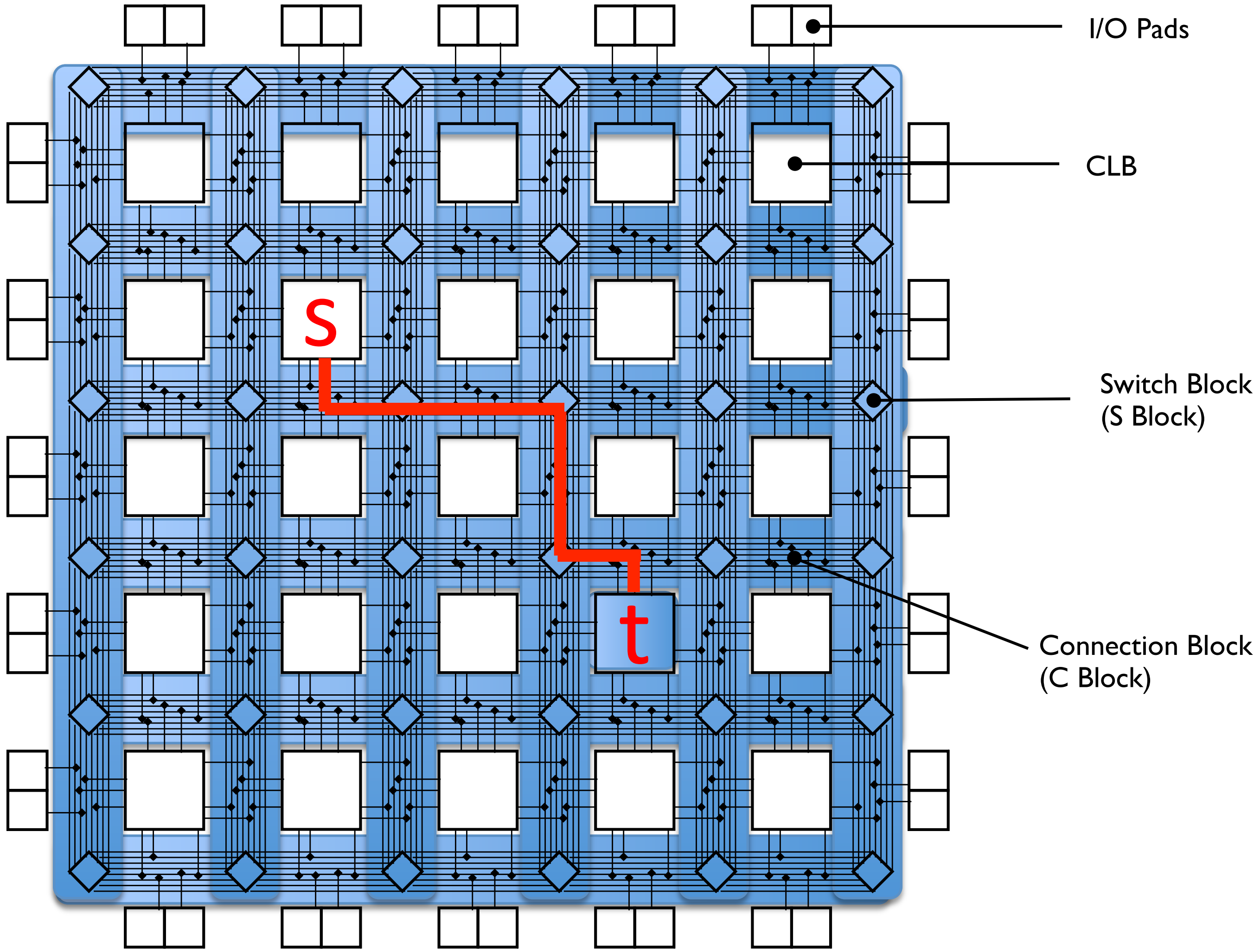
Start with a global RRG

SERRGE in Action (2/8)



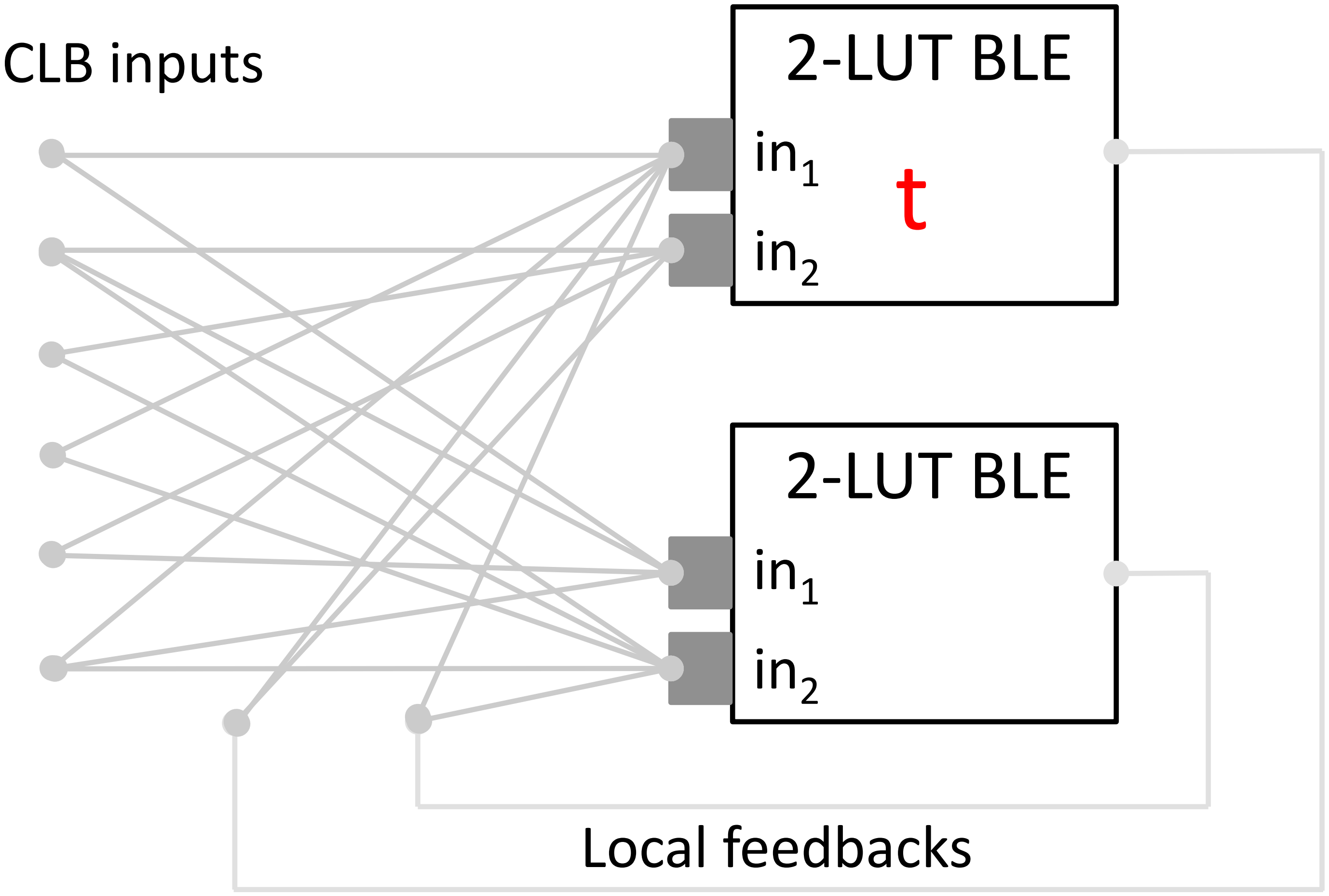
Route from s's CLB to t's CLB

SERRGE in Action (3/8)



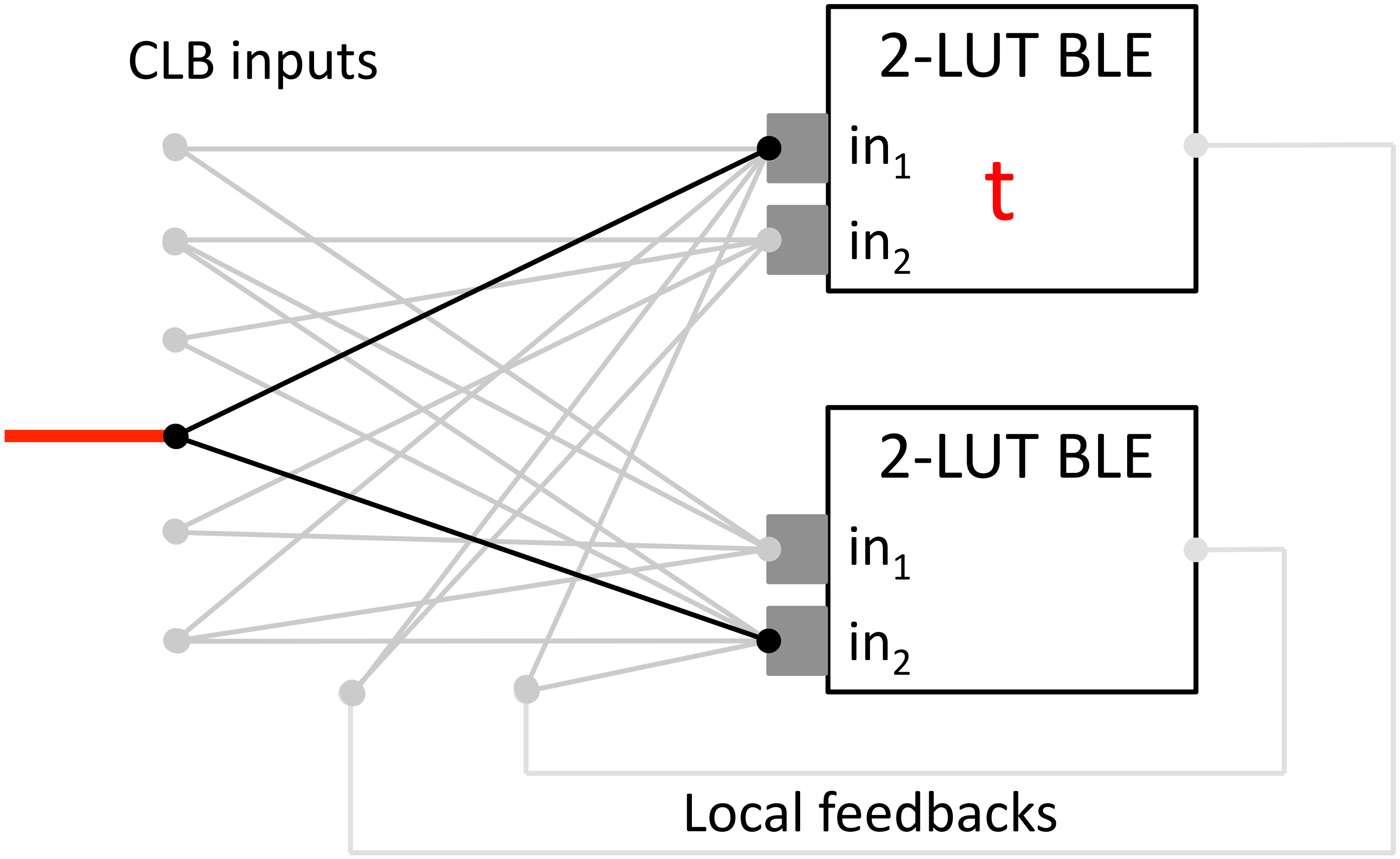
Locally expand the RRG with part of t's CLB

SERRGE in Action (4/8)



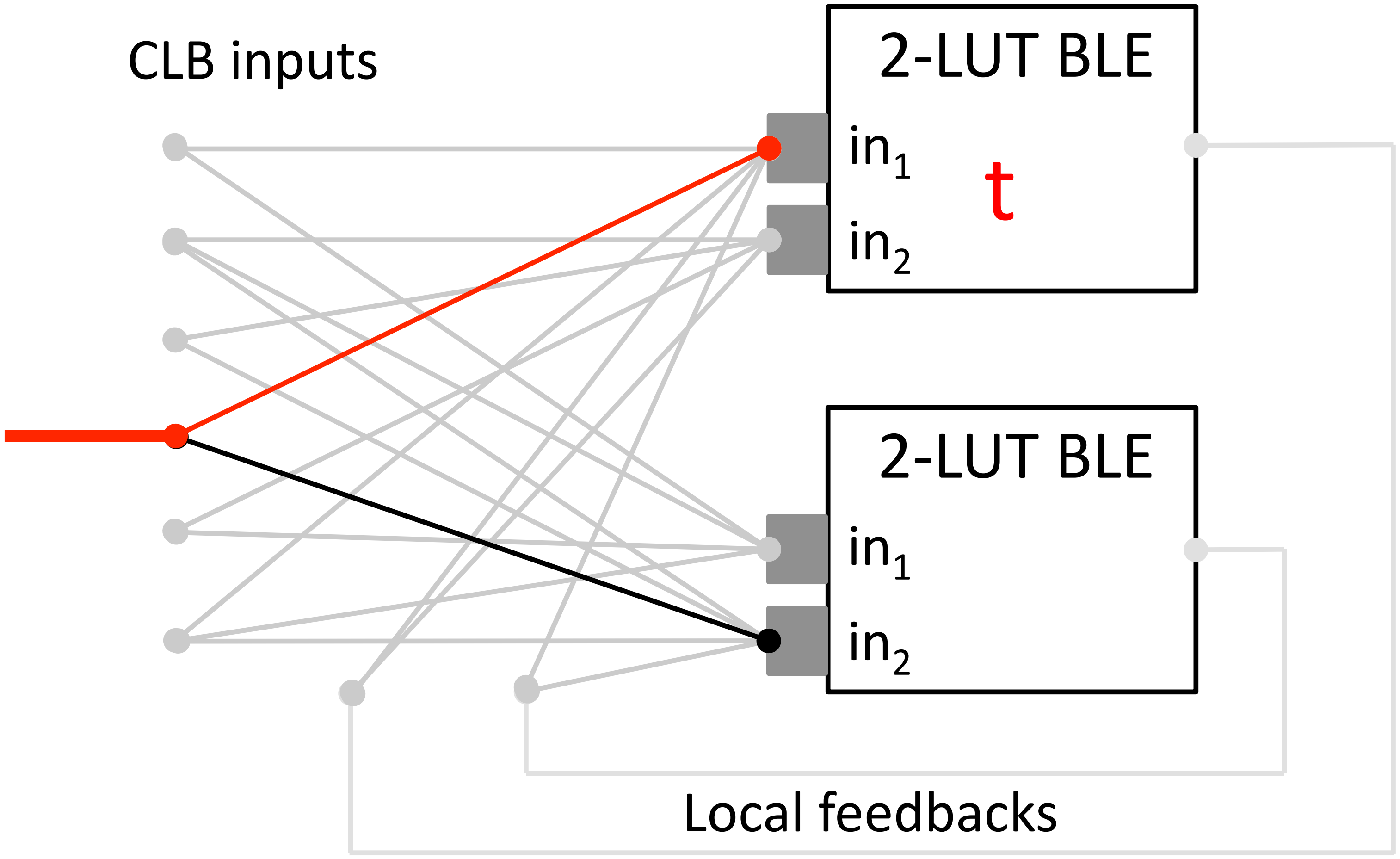
Maintain one copy of the intra-cluster topology

SERRGE in Action (5/8)



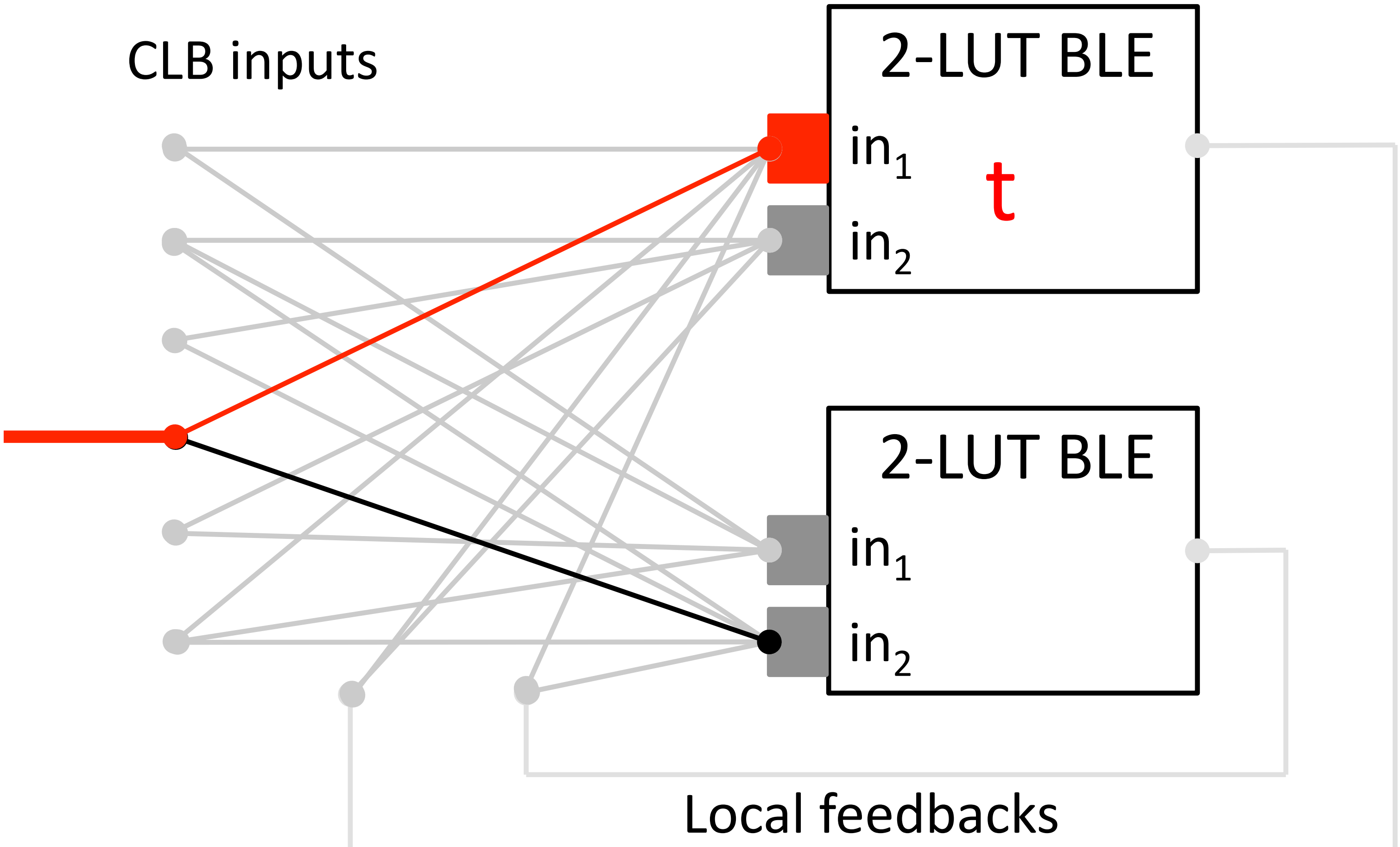
Expand the fanout of the CLB input being routed

SERRGE in Action (6/8)



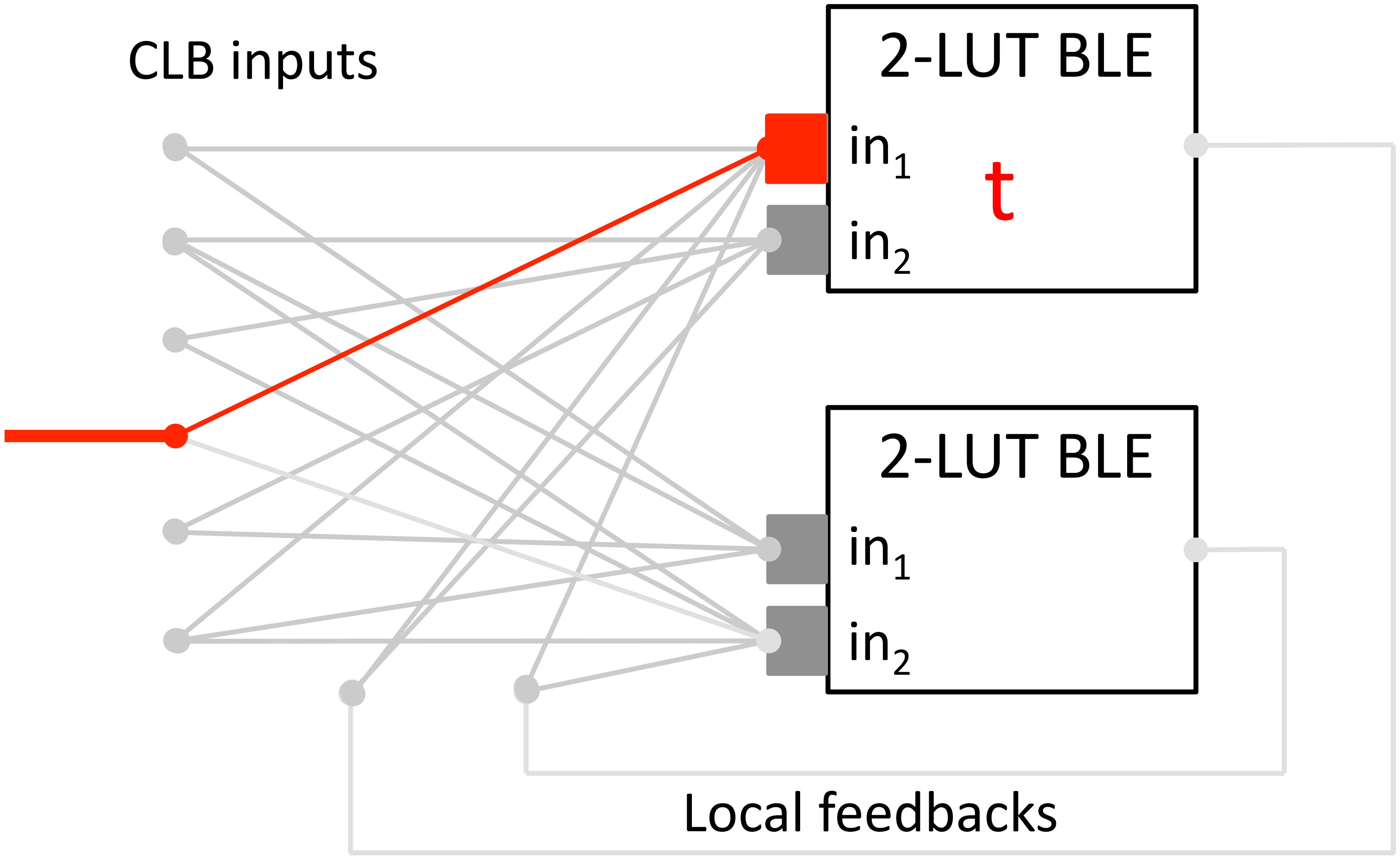
Complete the route if possible; if not, try again

SERRGE in Action (7/8)



Track used CLB and BLE inputs; remember the route

SERRGE in Action (8/8)

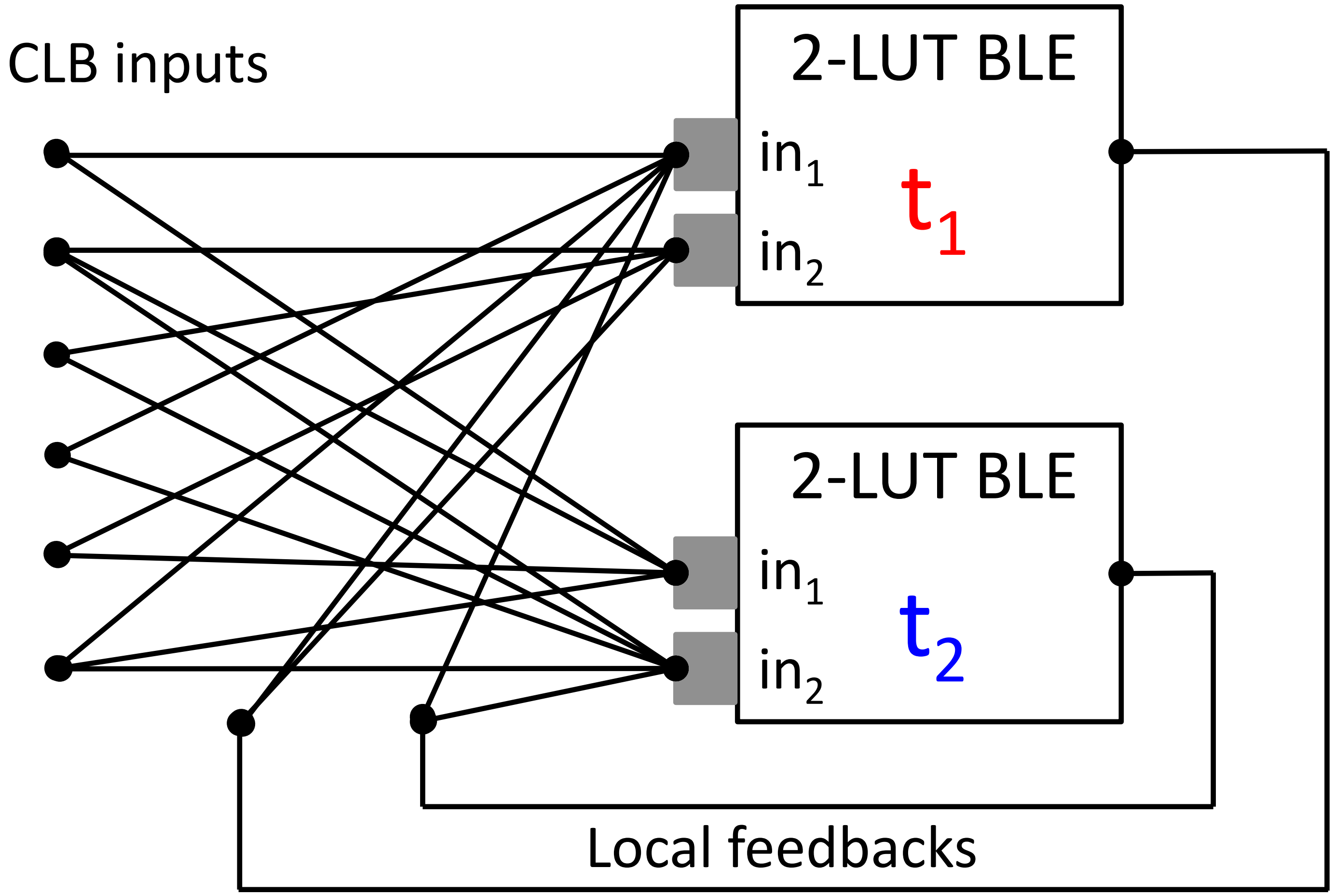


Deallocate unused RRG resources

SERRGE Summary

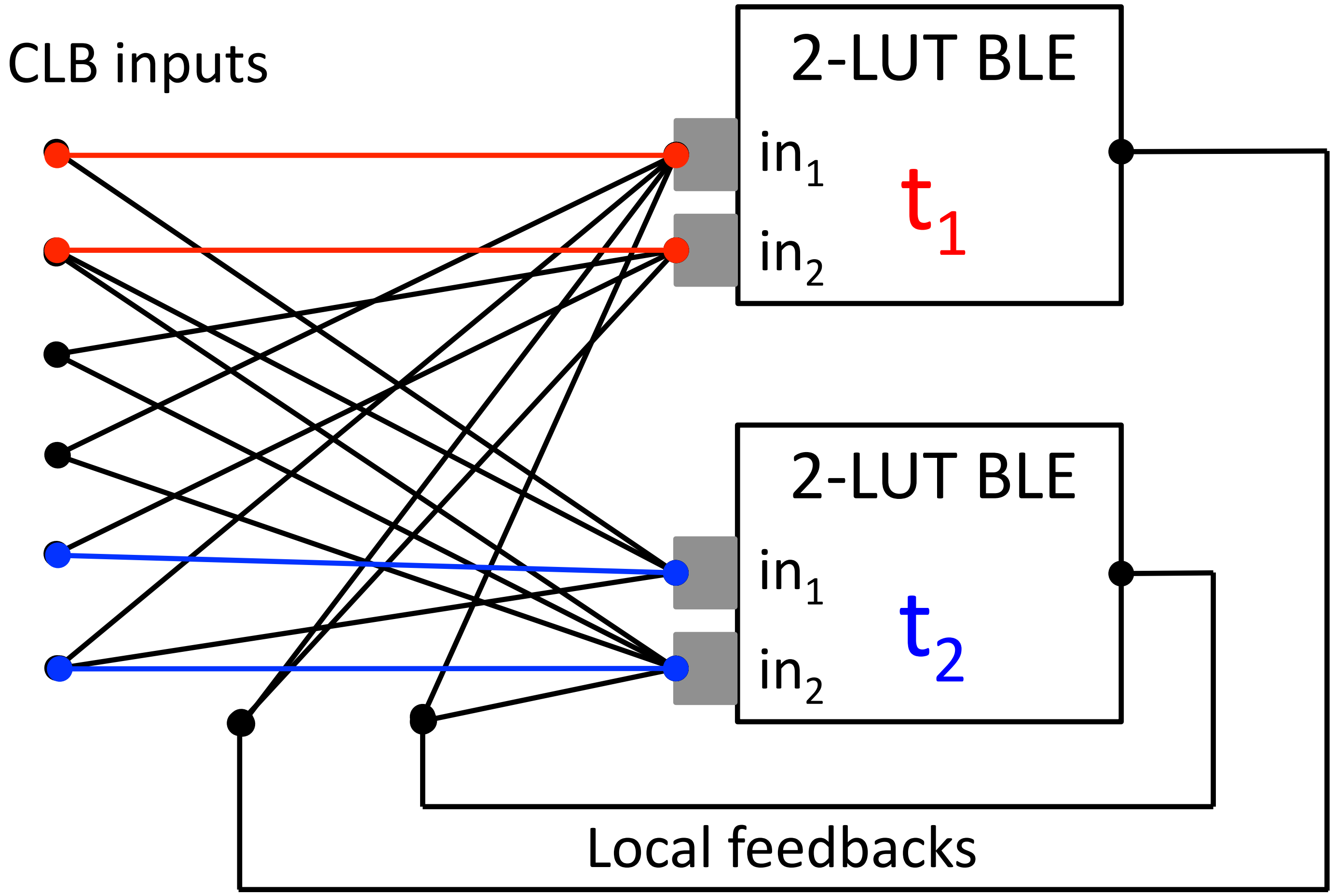
- Global routing uses standard PathFinder
- Selectively expand portions of the RRG to complete routes from CLB inputs to BLE inputs
- Storage Requirement
  - Global RRG (standard PathFinder requirement)
  - One copy of the intra-cluster routing topology
  - All intra-cluster routes computed thus far
- Fairly challenging to implement!

PPR in Action (1/4)



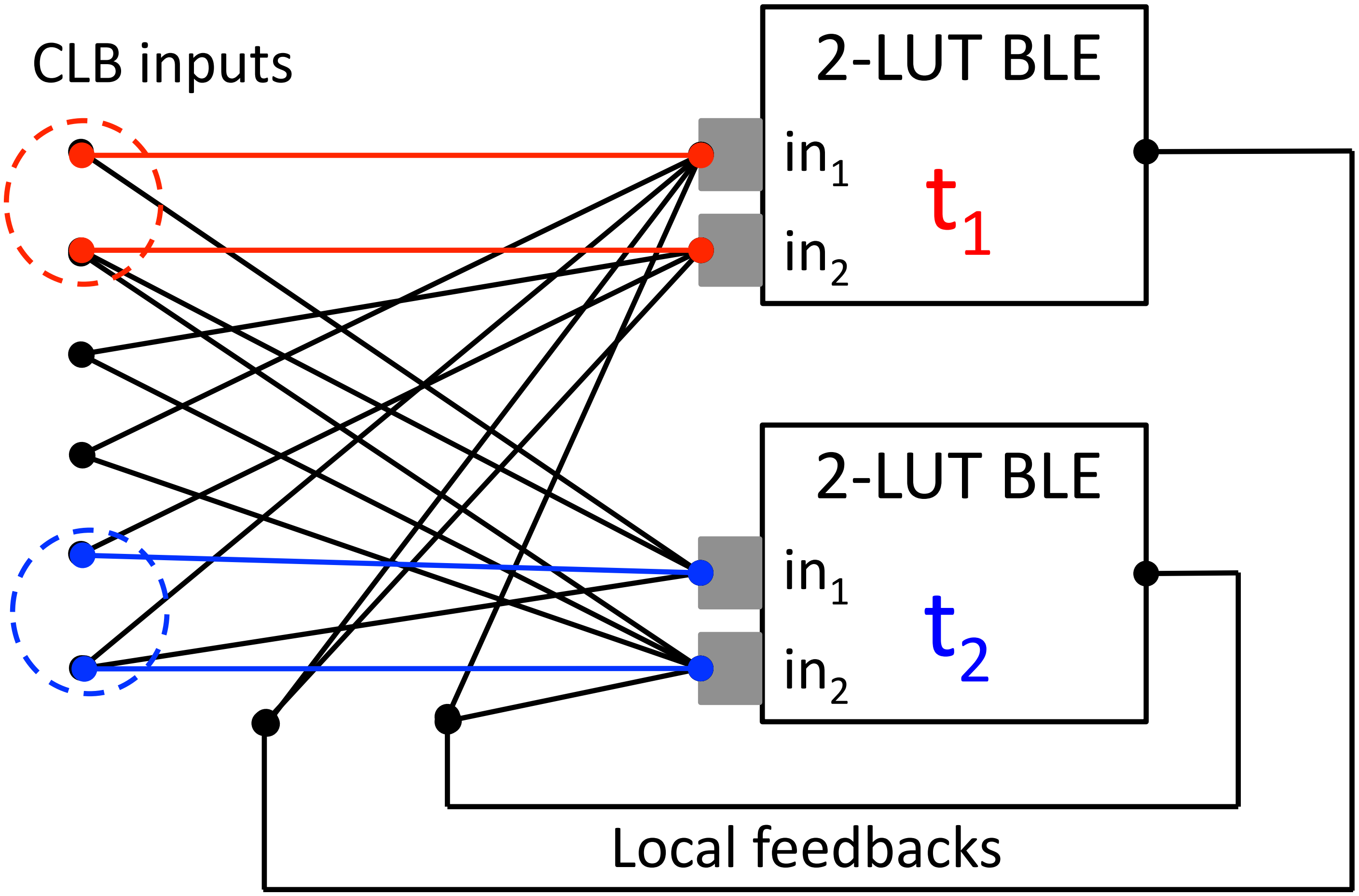
Process CLBs one at a time

PPR in Action (2/4)



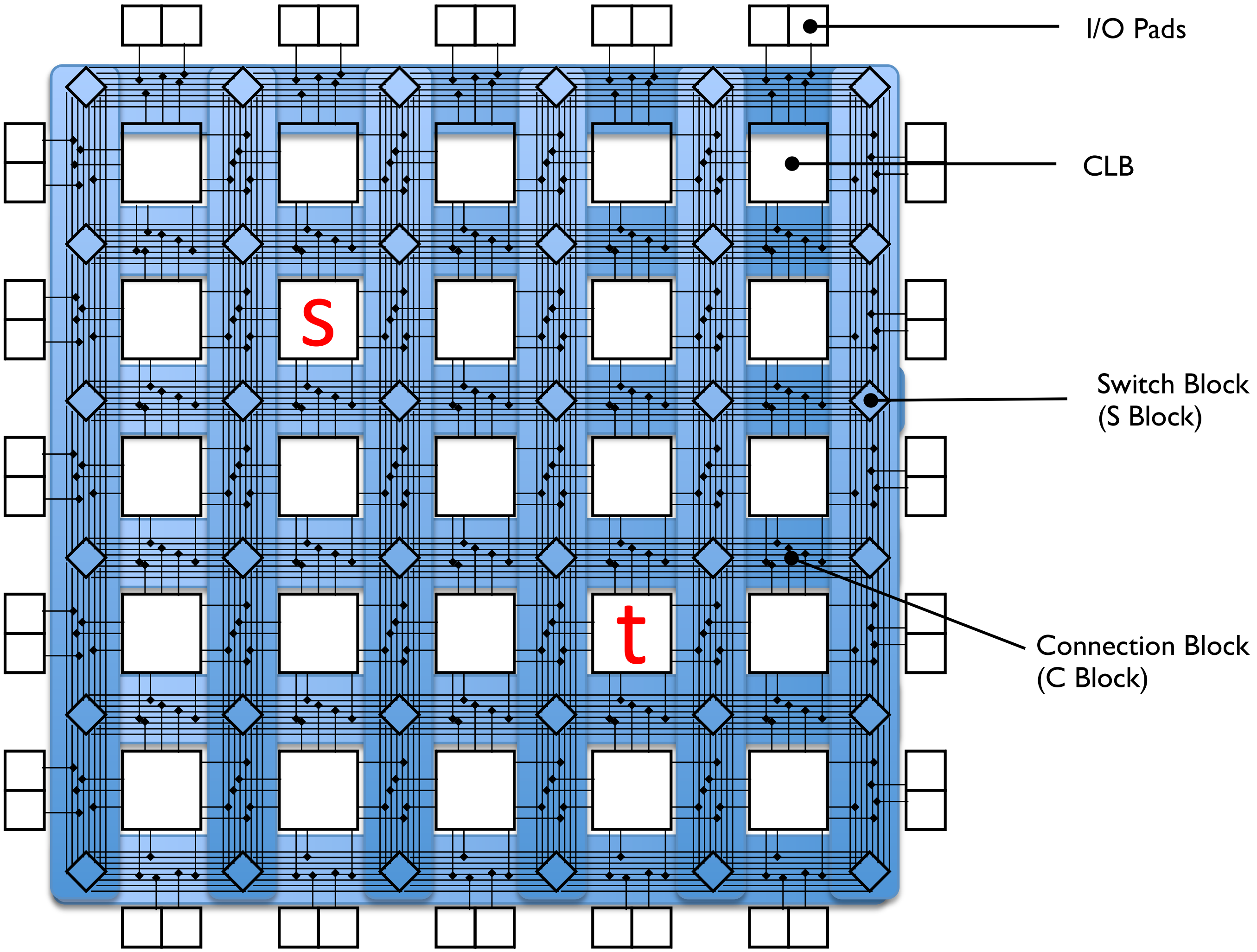
Compute local routes for each BLE

PPR in Action (3/4)



CLB inputs now form equivalence classes.

PPR in Action (4/4)



Perform global routing w/equivalence class constraints on CLB inputs

PPR Summary

- Pre-route each CLB
- Propagate LUT equivalences to CLB inputs
  - A baseline router on the full-blown RRG would still need to satisfy BLE input-pin equivalence constraints
- Route as normal
- Much easier implementation than SERRGE

Outline

- Background
  - FPGA architecture
  - FPGA routing
- FPGAs w/full intra-cluster routing crossbars
  - Implications for routing
- FPGAs w/sparse Intra-cluster routing crossbars
  - Routing challenges
  - New routing algorithms
- Experimental Results
- Conclusion

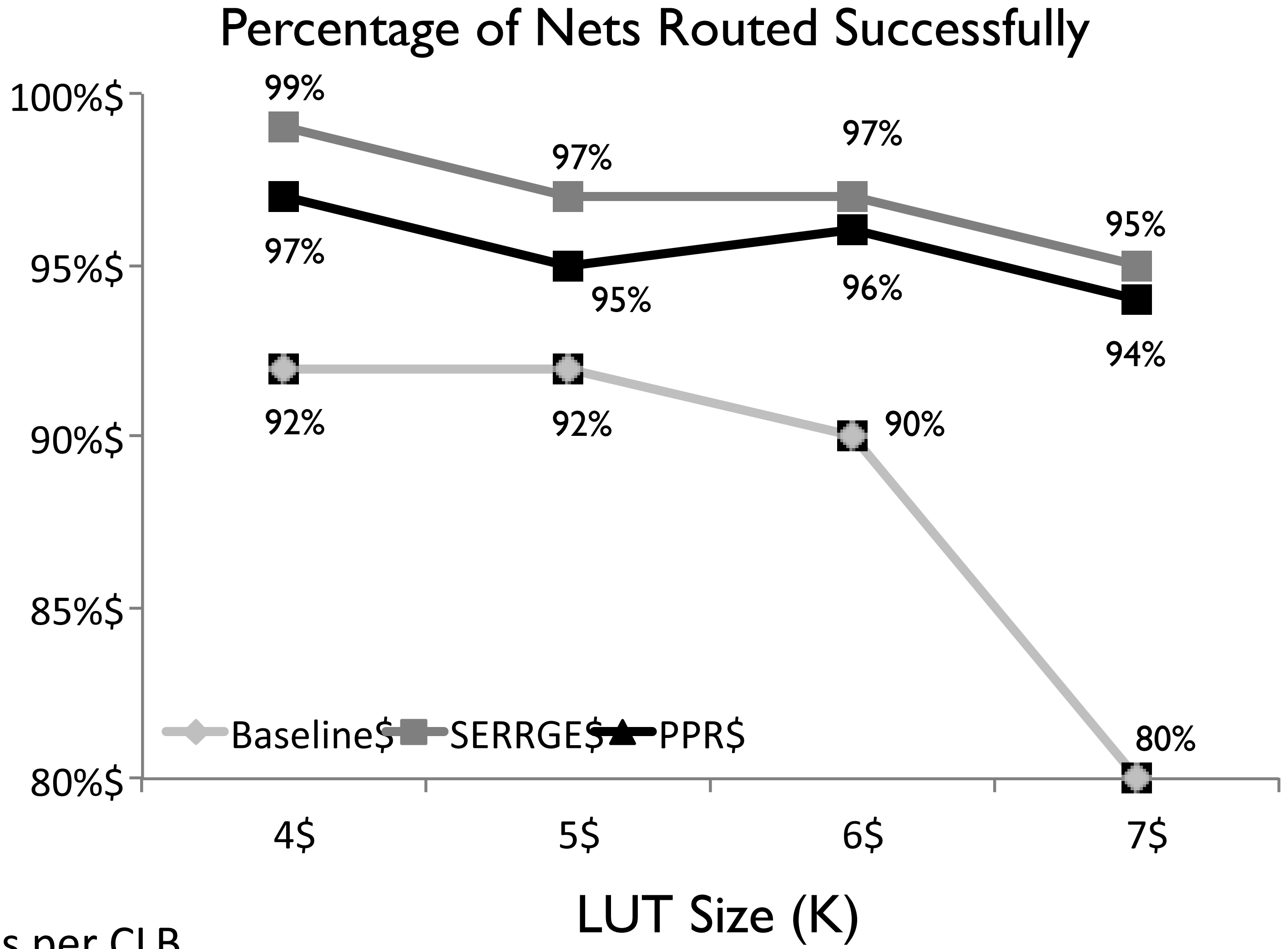
Experimental Setup

- VPR 5.0, iFAR FPGA architecture files
  - Crossbar population density, p = {25%, 50%, 75%, 100%}
  - LUT size = K = {4, 5, 6, 7}
  - BLEs per cluster, N = {4, 6, 8, 10}

- 10 largest IWLS benchmarks
  - Results are averaged

- VPR-PathFinder runs for 100 iterations, then stops
  - Baseline (full-blown RRG)
  - SERRGE
  - PPR
  - VPR 5.0 (when crossbar population density is 100%)

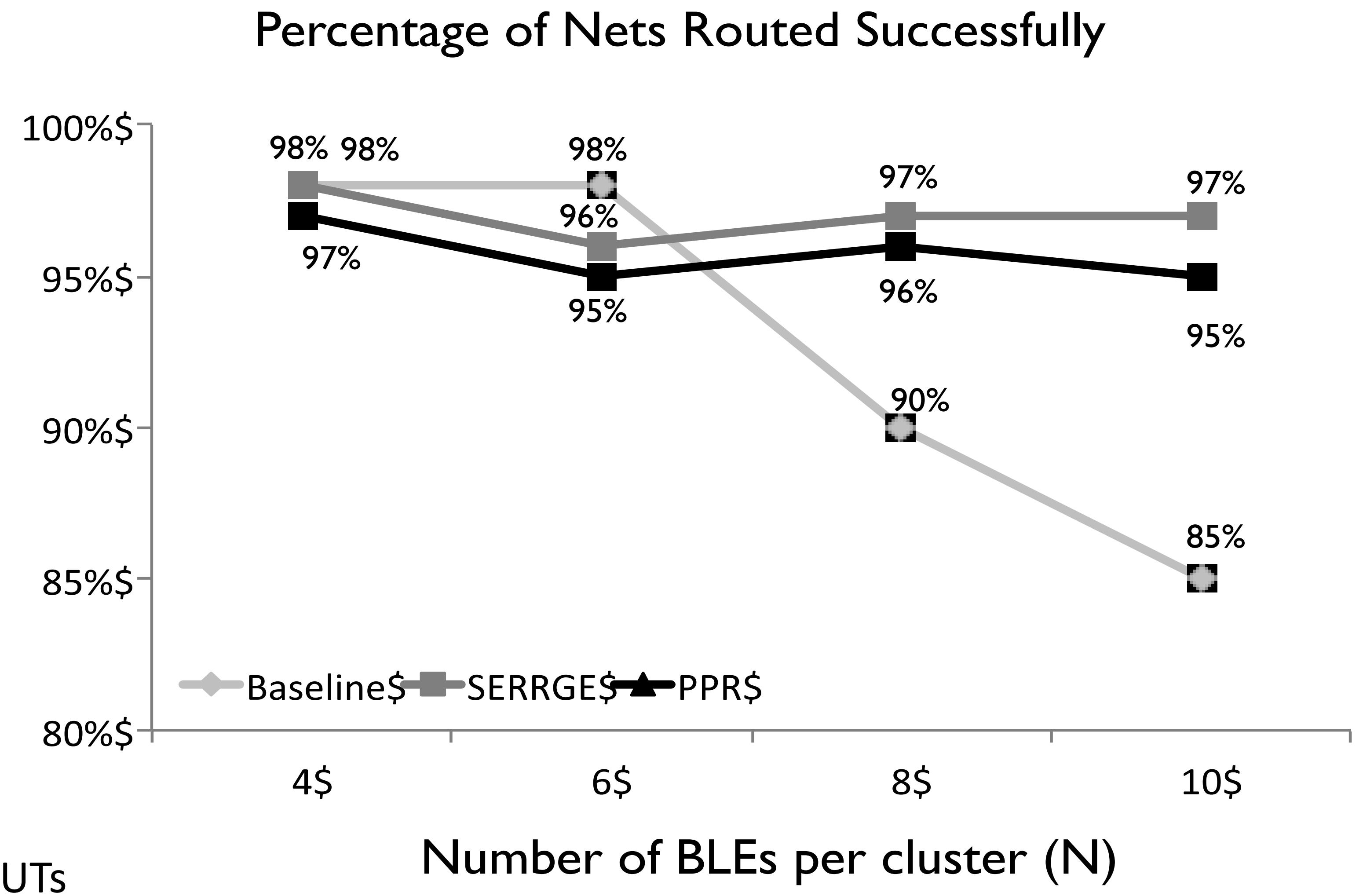
Routability as a Function of LUT Size



N = 8 BLEs per CLB

p = 50% population density

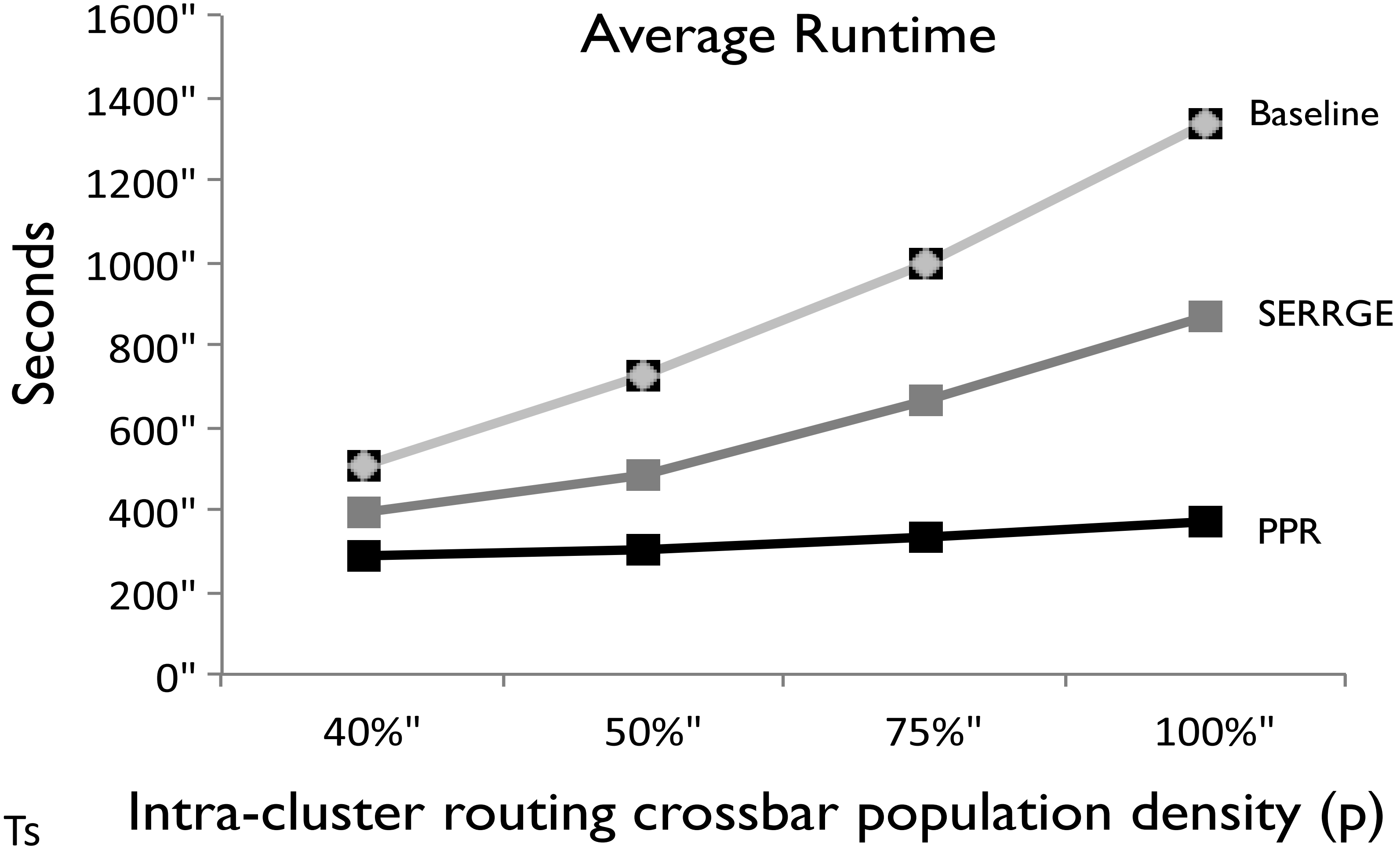
Routability as a Function of CLB Size



K = 6-LUTs

p = 50% population density

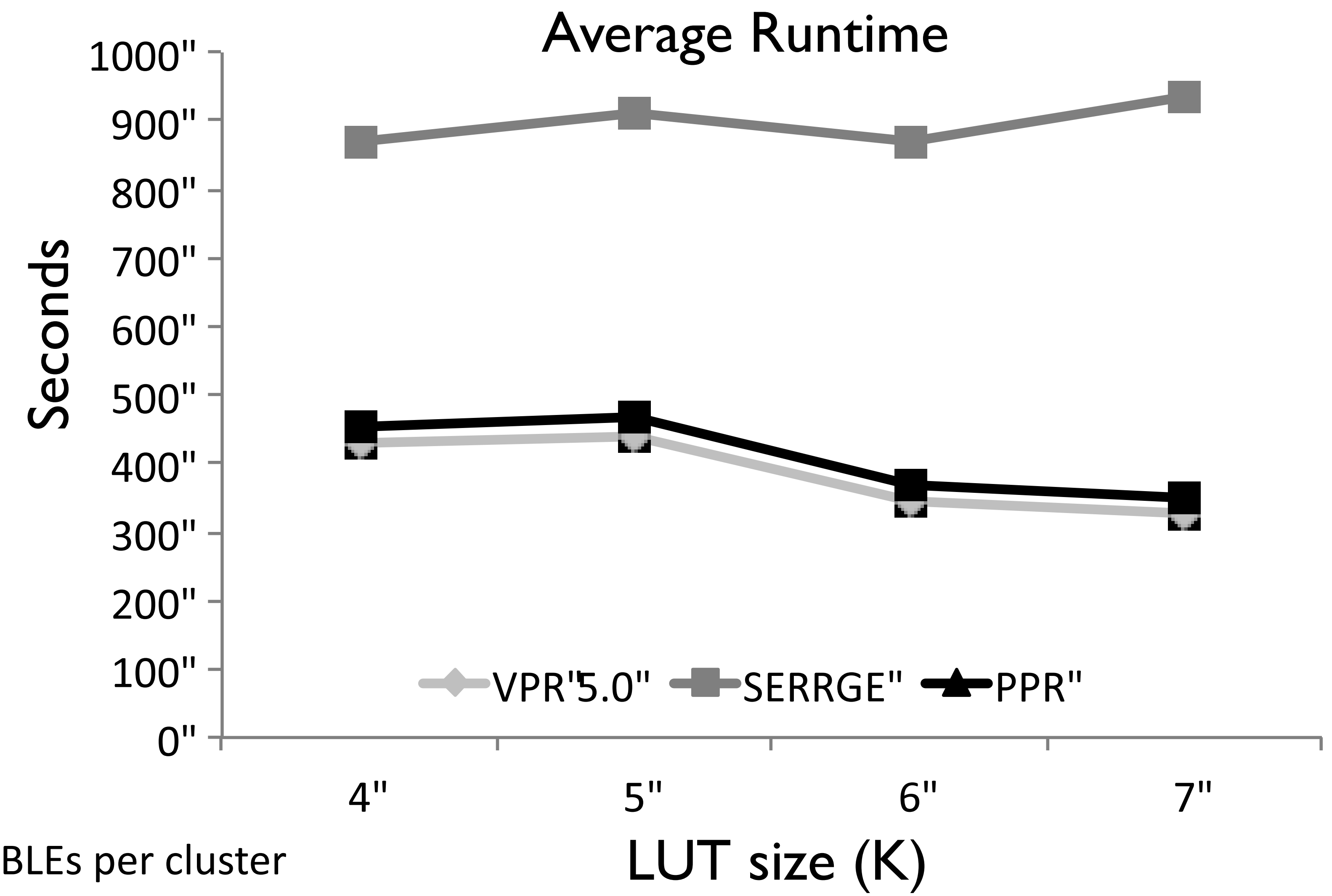
Runtime as a Function of Population Density



K = 6-LUTs

N=8 BLEs per cluster

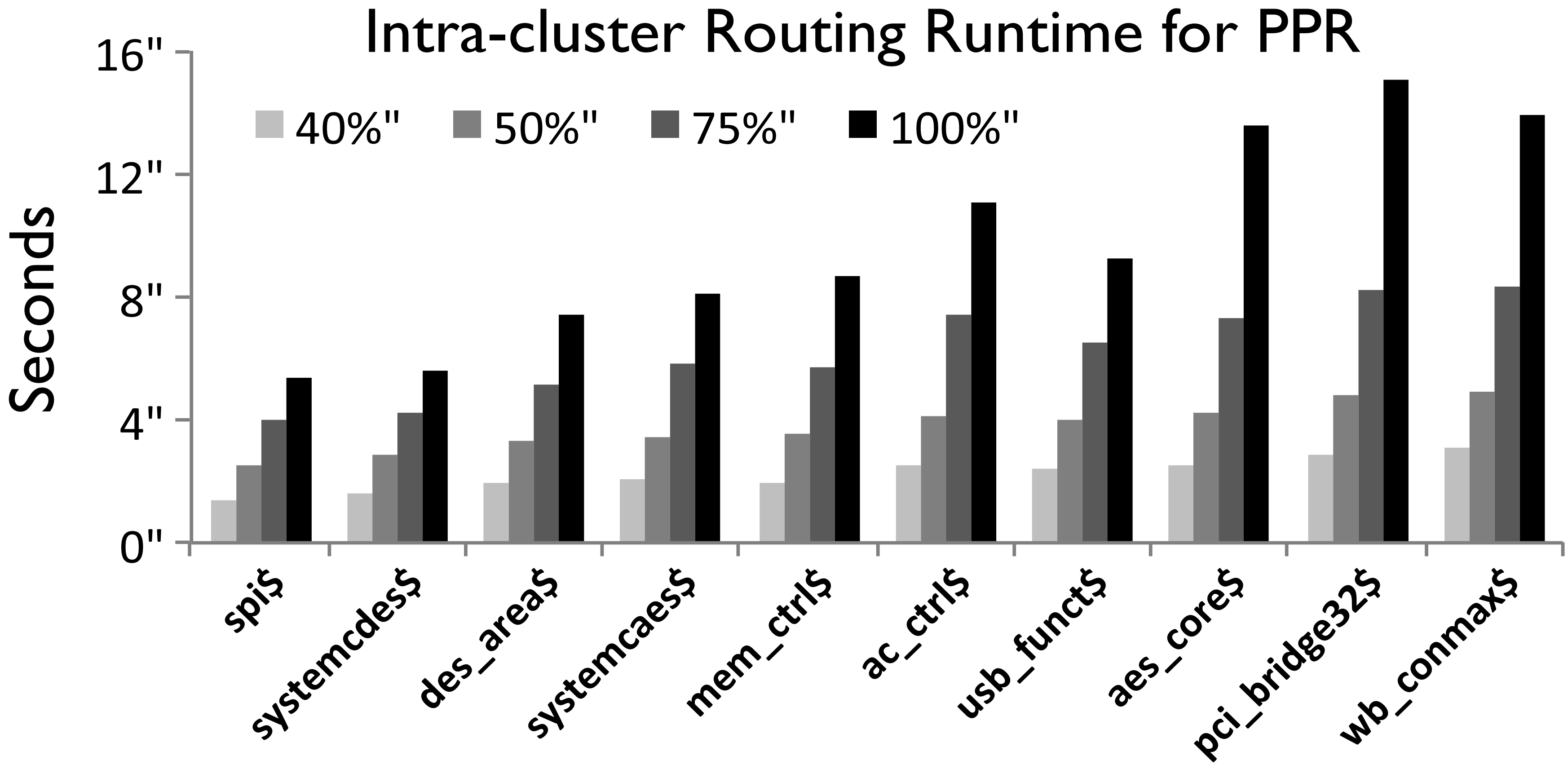
Runtime as a Function of LUT Size



N=8 BLEs per cluster

p = 100% population density

PPR Intra-cluster Routing Runtime

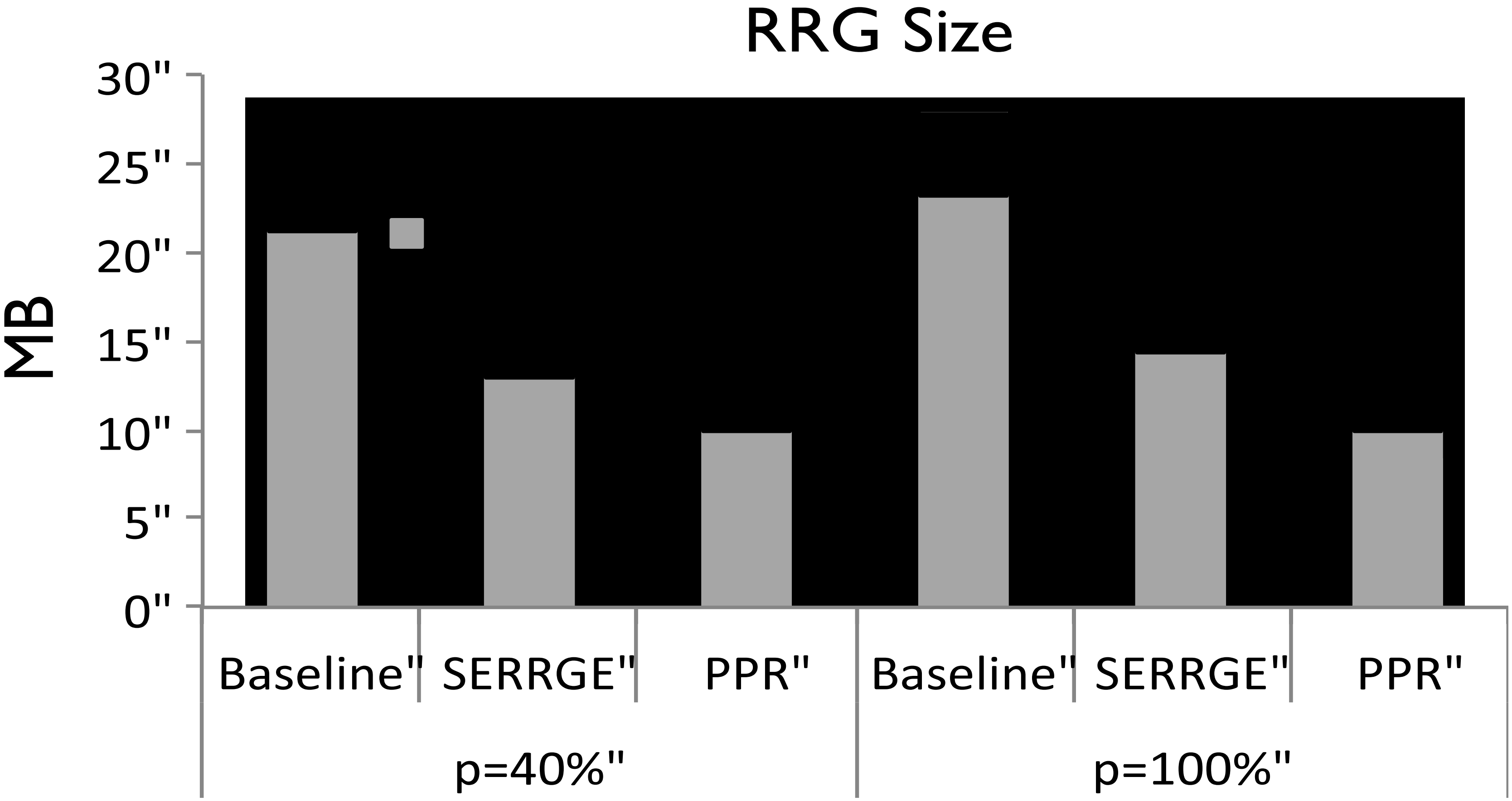


K=6-LUTs

N=8 BLEs per cluster

On average, total runtime was 100s of seconds!

Static and Dynamic RRG Size

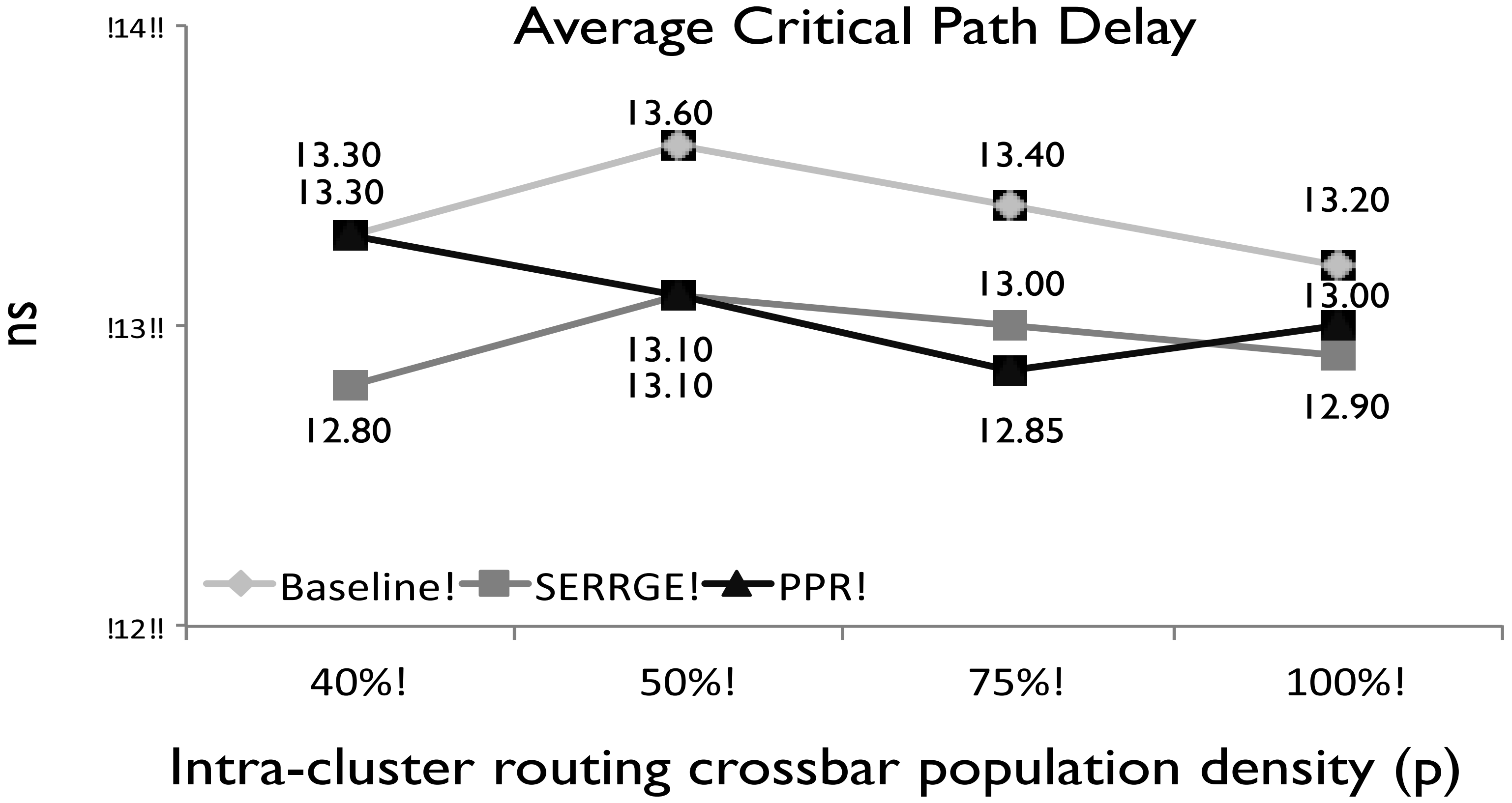


K=6-LUTs

N=8 BLEs per cluster

VPR generates FPGAs that are sized to the application, and are smaller than commercial products

Critical Path Delay



K=6-LUTs

N=8 BLEs per cluster

Outline

- Background
  - FPGA architecture
  - FPGA routing
- FPGAs w/full intra-cluster routing crossbars
  - Implications for routing
- FPGAs w/sparse Intra-cluster routing crossbars
  - Routing challenges
  - New routing algorithms
- Experimental Results
- Conclusion

Conclusion

- Addressed FPGA routing algorithms where CLBs have sparse intra-cluster routing crossbars

- Minimal modifications to PathFinder required

- SERRGE vs. PPR
  - SERRGE achieves better routability
  - PPR converges faster
  - PPR easier to implement