Timing-aware routing in the RapidWright framework

Leo Liu, Nachiket Kapre
leo.liu@uwaterloo.ca, nachiket@uwaterloo.ca

Background

RapidWright: Open source project for accessing low-level resources for Xilinx FPGAs

Advantage: design generation without FPGA CAD tools

Weakness: no timing knowledge of FPGA resources; hard to build timing-driven tools

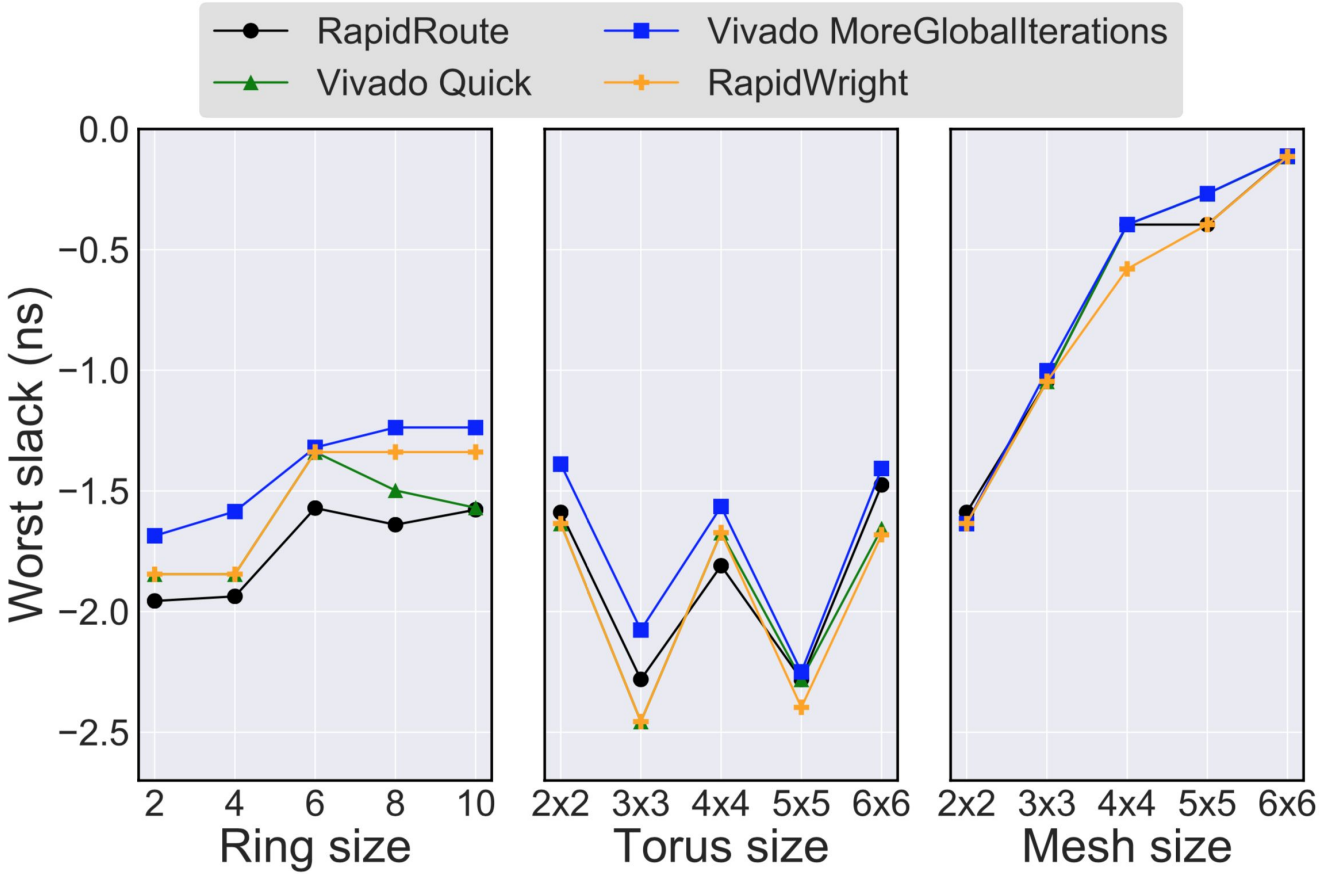
Problem Statement

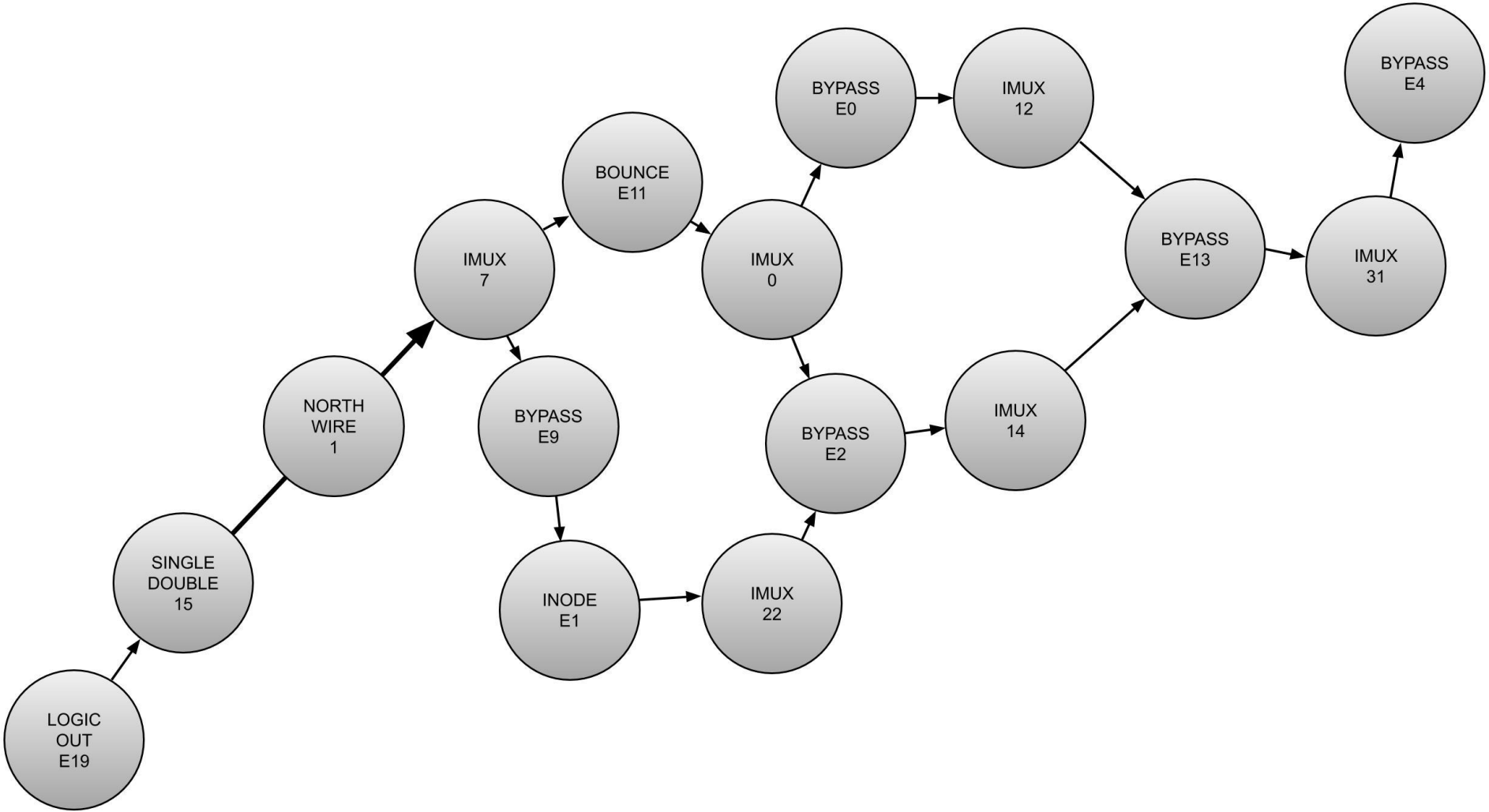
We used RapidWright to design **RapidRoute**, a fast router for building communication networks

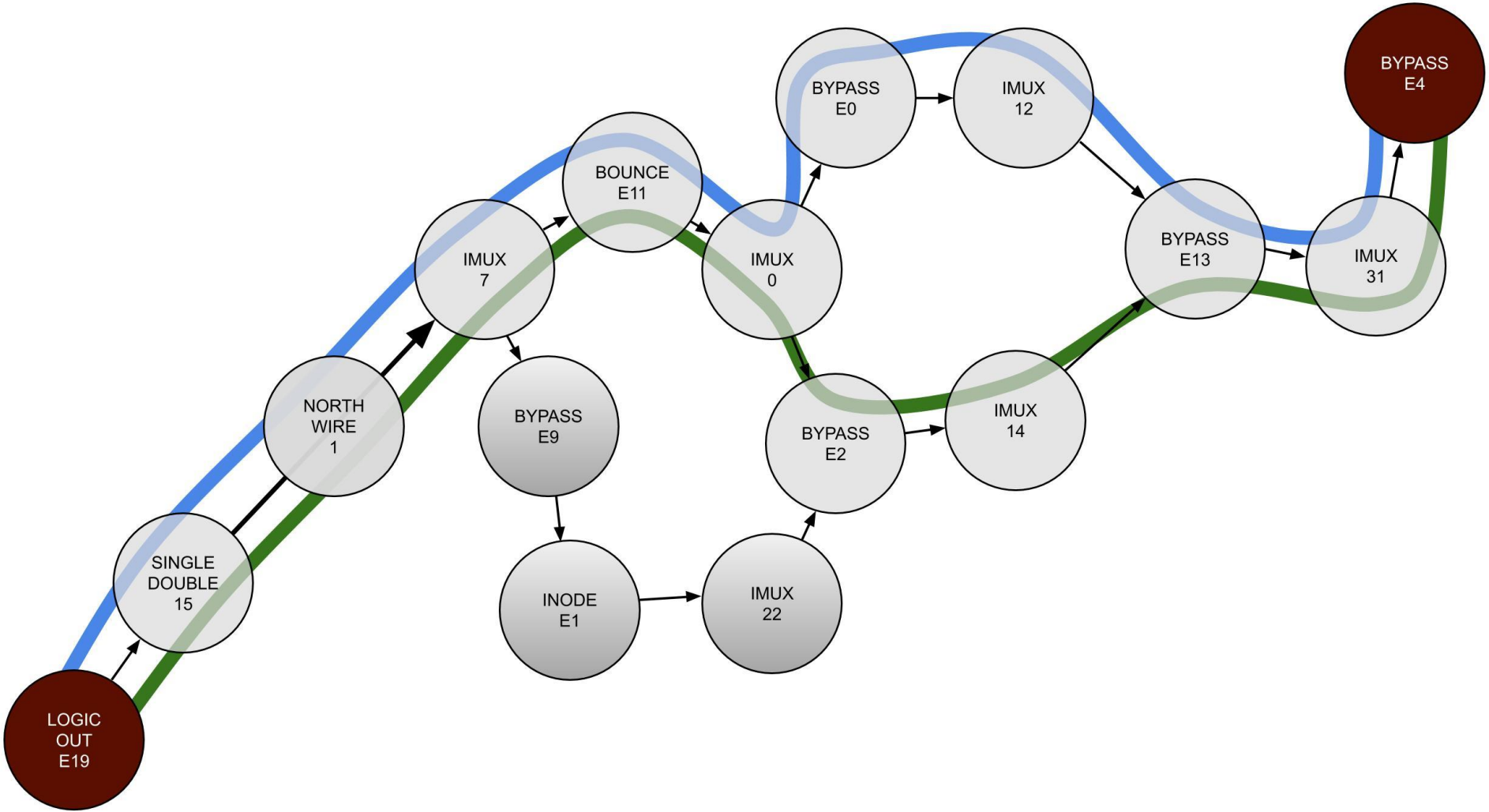
Problem: RapidWright does not allow RapidRoute to be timing-driven:

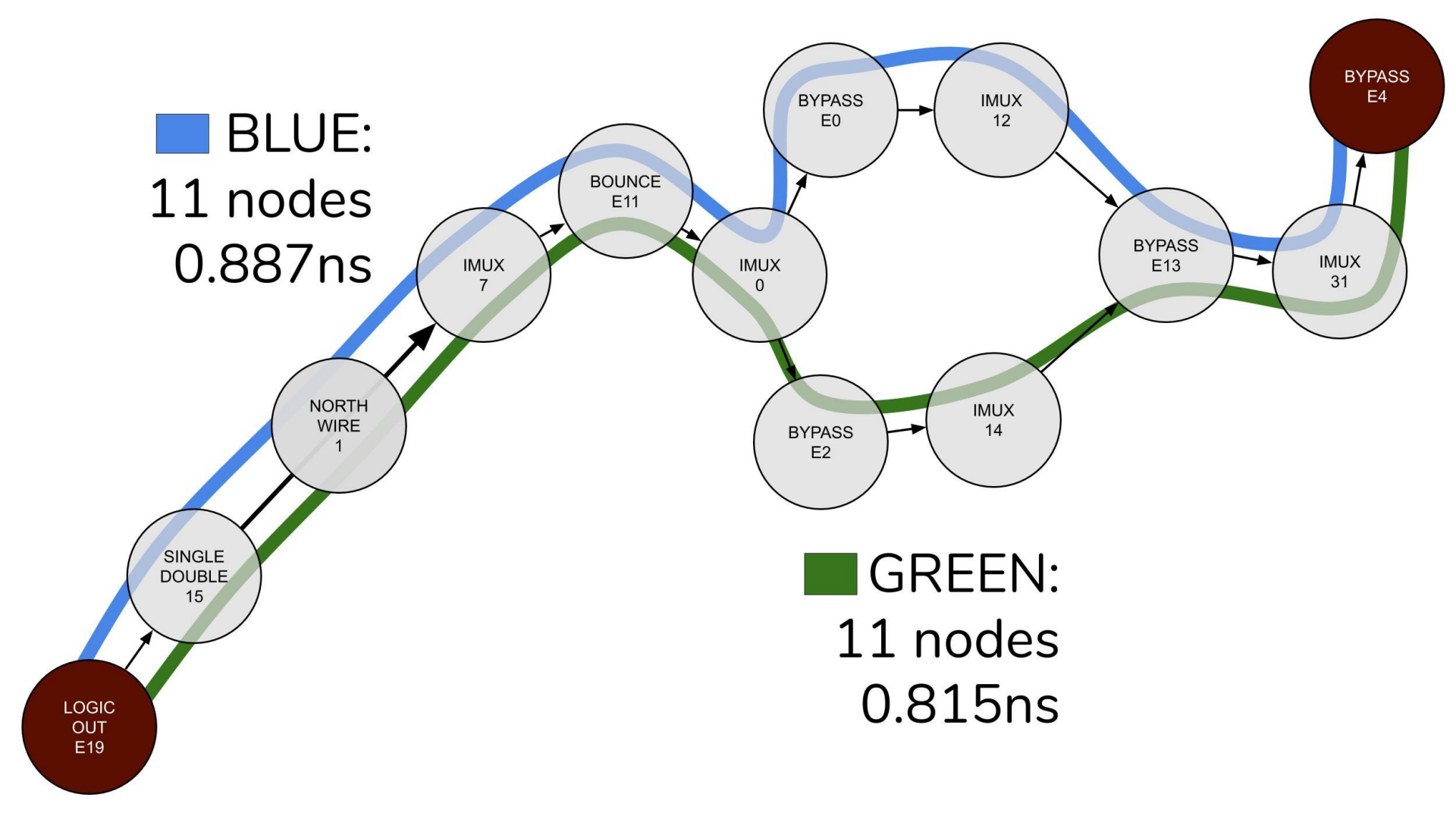
- Routing algorithms cannot optimize for shortest path
- Consistently loses to Vivado

Timing Slack




LOGIC
OUT
E19
SINGLE
DOUBLE
15
NORTH
WIRE
1
IMUX
7
BOUNCE
E11
BYPASS
E9
INODE
E1
IMUX
22
IMUX
0
BYPASS
E0
BYPASS
E2
IMUX
12
IMUX
14
BYPASS
E13
BYPASS
E4
IMUX
31


LOGIC
OUT
E19
SINGLE
DOUBLE
15
NORTH
WIRE
1
IMUX
7
BOUNCE
E11
BYPASS
E9
INODE
E1
IMUX
22
IMUX
0
BYPASS
E0
IMUX
12
BYPASS
E2
IMUX
14
BYPASS
E13
IMUX
31
BYPASS
E4


BLUE:
11 nodes
0.887ns
GREEN:
11 nodes
0.815ns
LOGIC
OUT
E19
SINGLE
DOUBLE
15
NORTH
WIRE
1
IMUX
7
BOUNCE
E11
IMUX
0
BYPASS
E0
IMUX
12
BYPASS
E2
IMUX
14
BYPASS
E13
IMUX
31
BYPASS
E4

Solution

Build our own timing library

Main ideas:

- Extract relevant timing data using both RapidWright and Vivado
- Integrate extracted data into RapidRoute algorithm

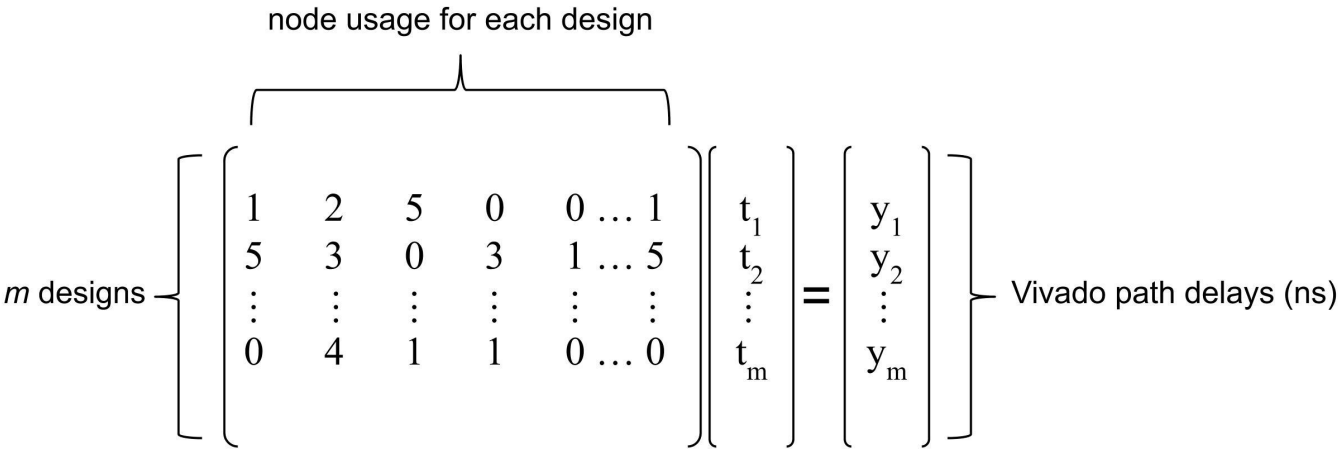
Key Claim

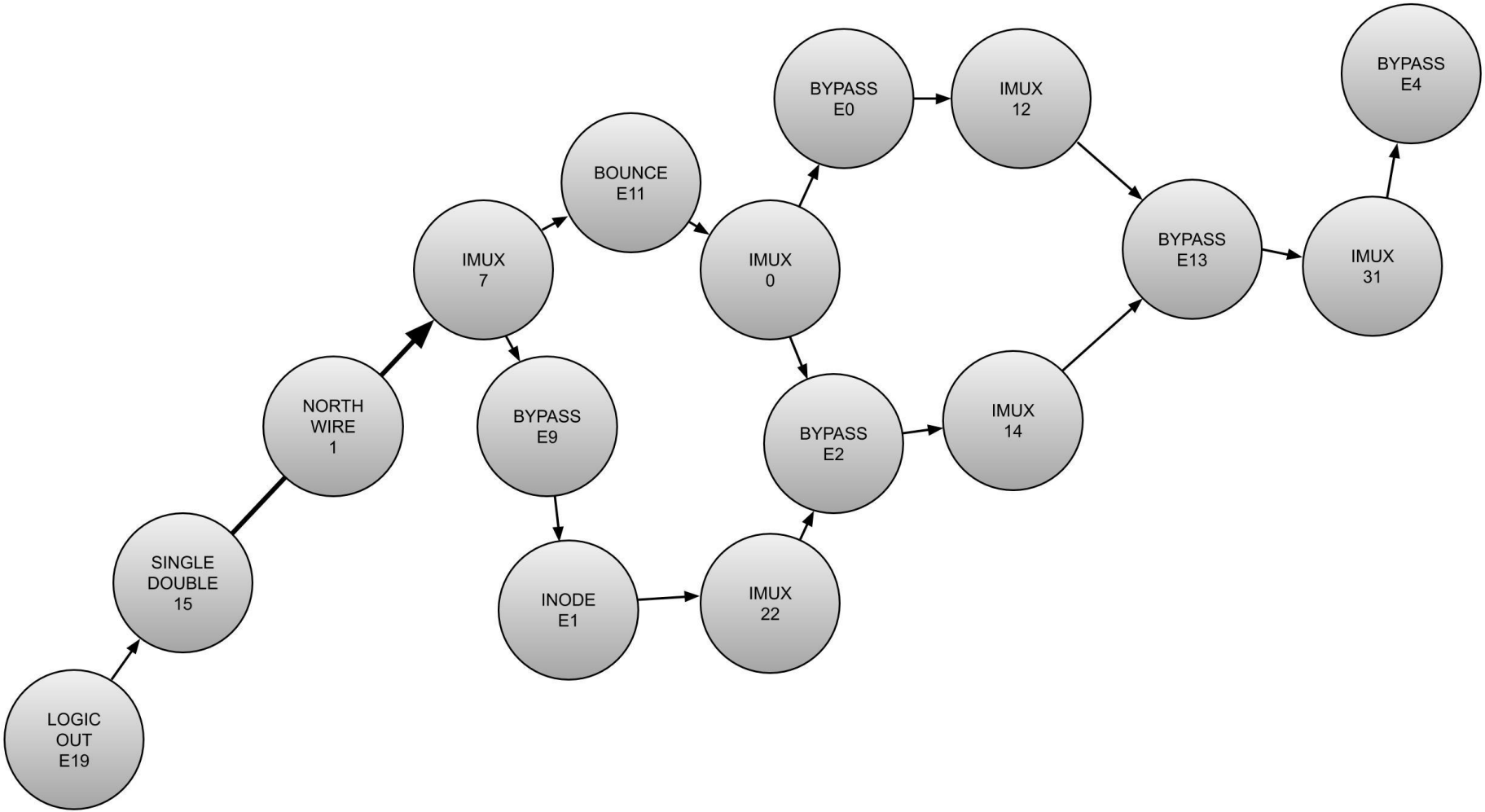
We can extract fine-grained timing information of Xilinx FPGA routing resources

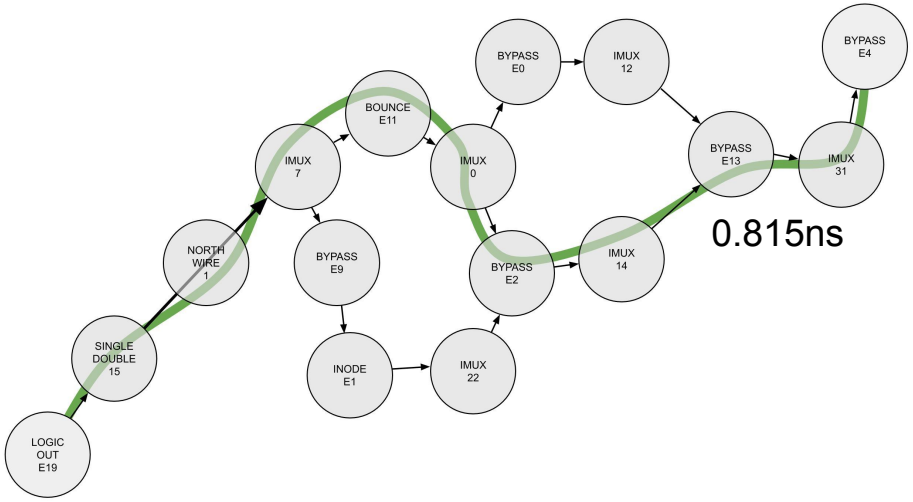
- Library allows RapidRoute to **match Vivado performance**
- **Less than 10 mins** of one-time analysis
- **Extremely lightweight**
  - RapidRoute retains its routing speed
  - Low memory overhead

Main Approach

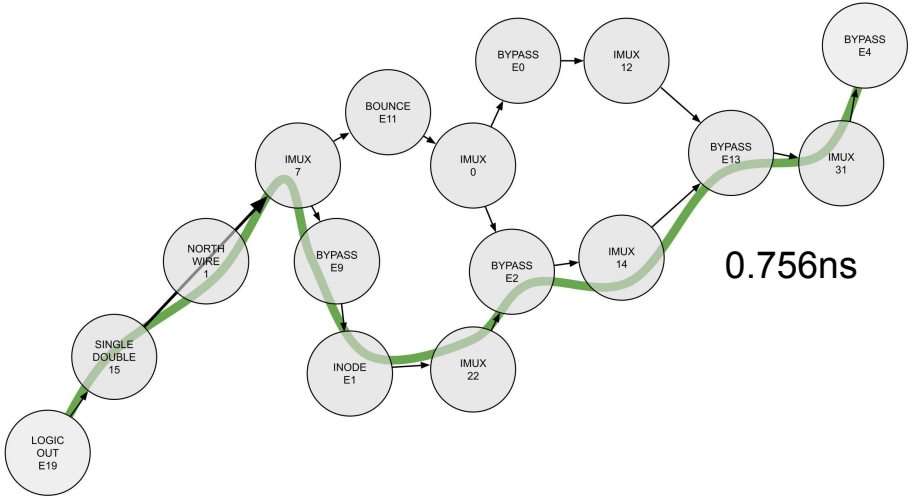
1. Build many calibration designs with RapidWright
2. Load designs in Vivado for timing feedback
3. Organize into a linear system



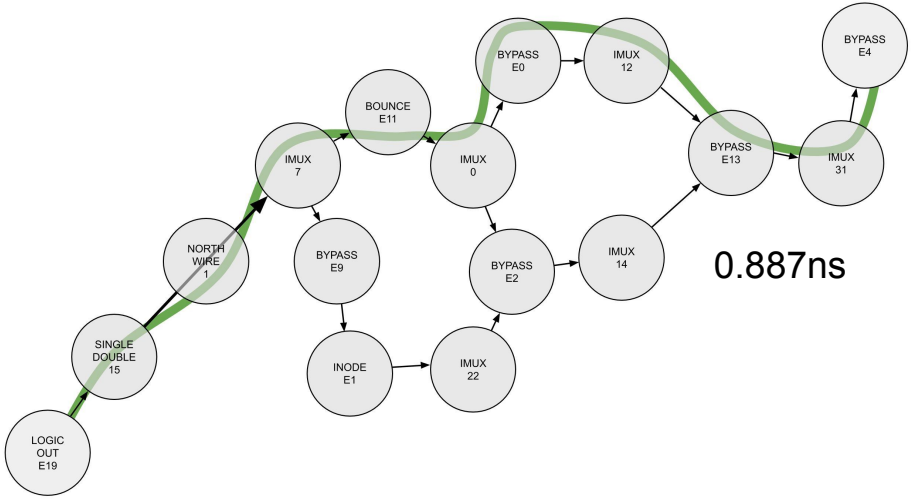

LOGIC
OUT
E19
SINGLE
DOUBLE
15
NORTH
WIRE
1
IMUX
7
BOUNCE
E11
BYPASS
E9
INODE
E1
IMUX
22
IMUX
0
BYPASS
E0
BYPASS
E2
IMUX
12
IMUX
14
BYPASS
E13
BYPASS
E4
IMUX
31



0.815ns



0.756ns



0.887ns

LOGIC_OUT_E19 + SNG_DBL_15 + NN1 + IMUX_7
+ BNCE_E11 + … = 0.815ns

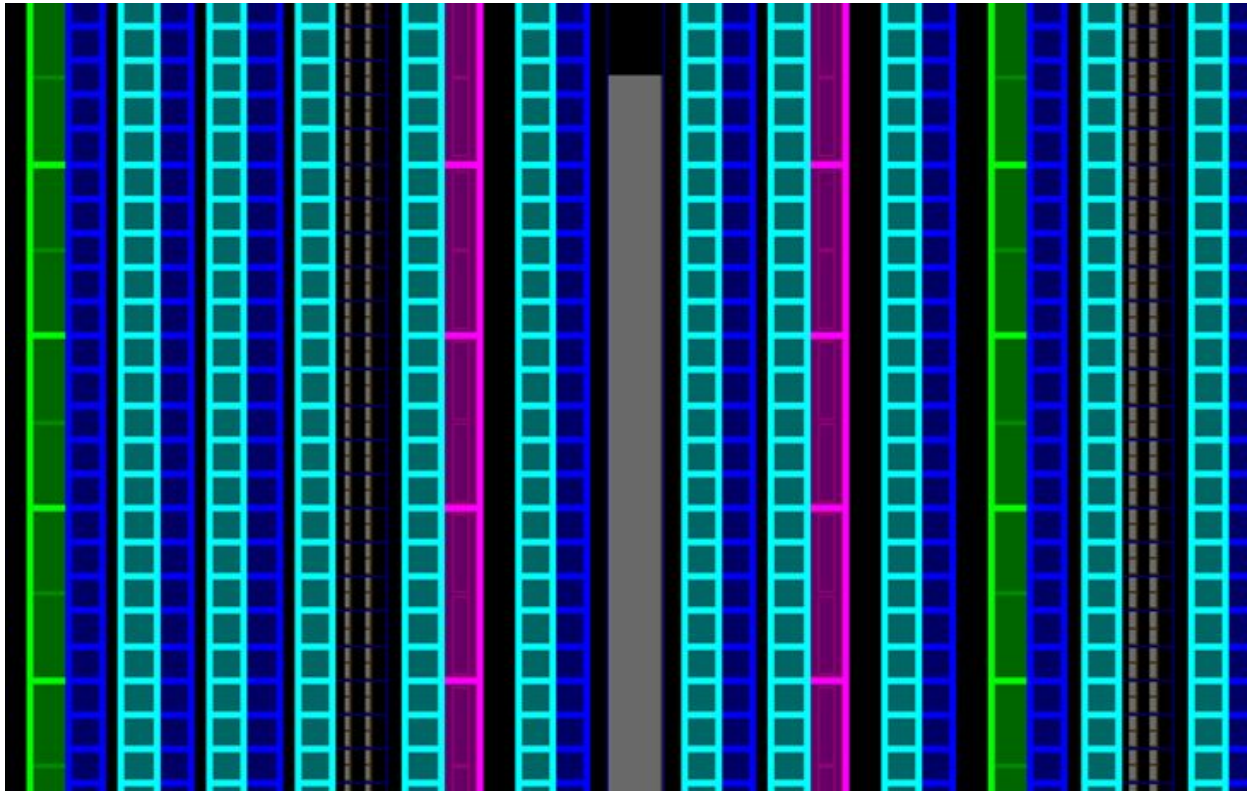
LOGIC_OUT_E19 + SNG_DBL_15 + NN1 + IMUX_7
+ BNCE_E11 + … = 0.887ns

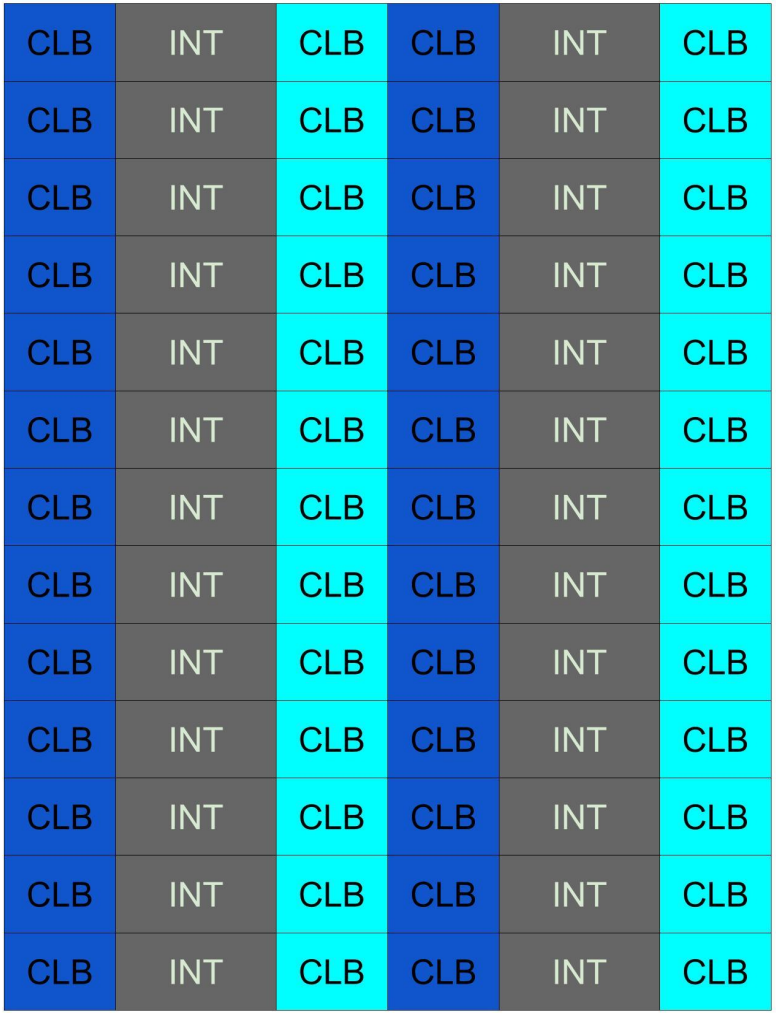
LOGIC_OUT_E19 + SNG_DBL_15 + NN1 + IMUX_7
+ BYPASS_E9 + … = 0.756ns

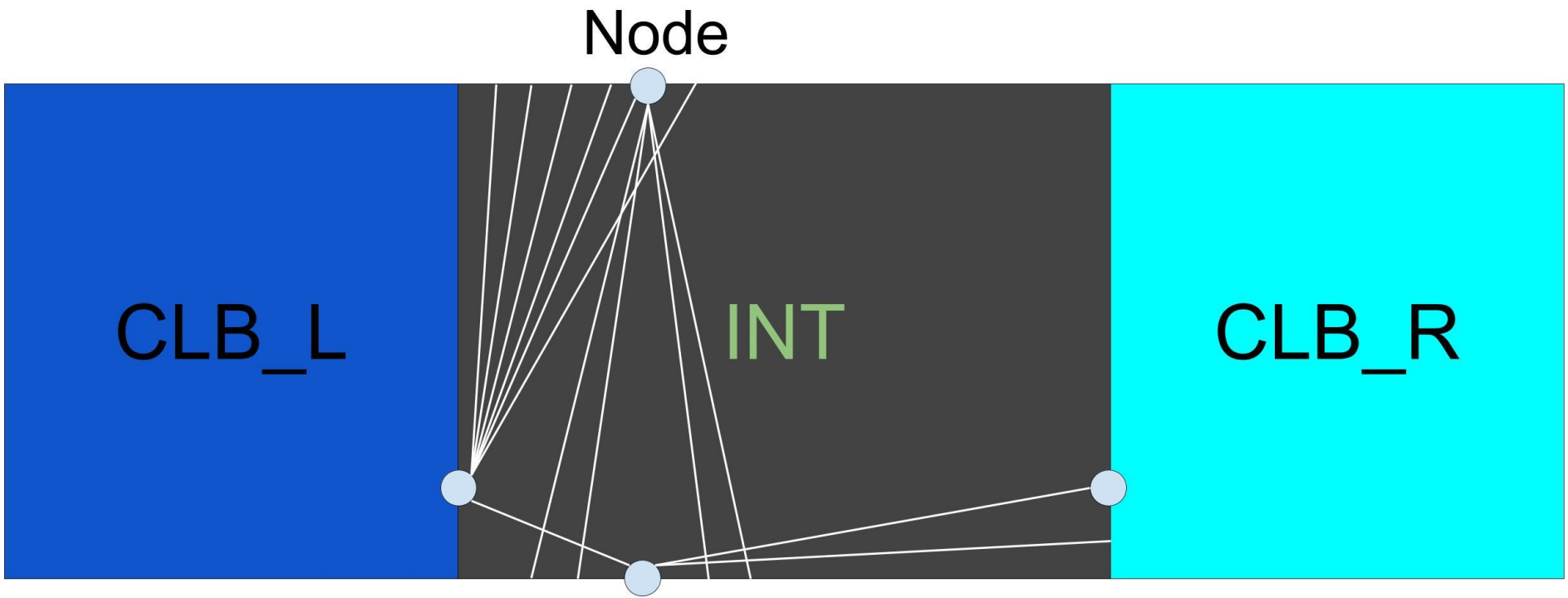
Opportunities in Timing Extraction

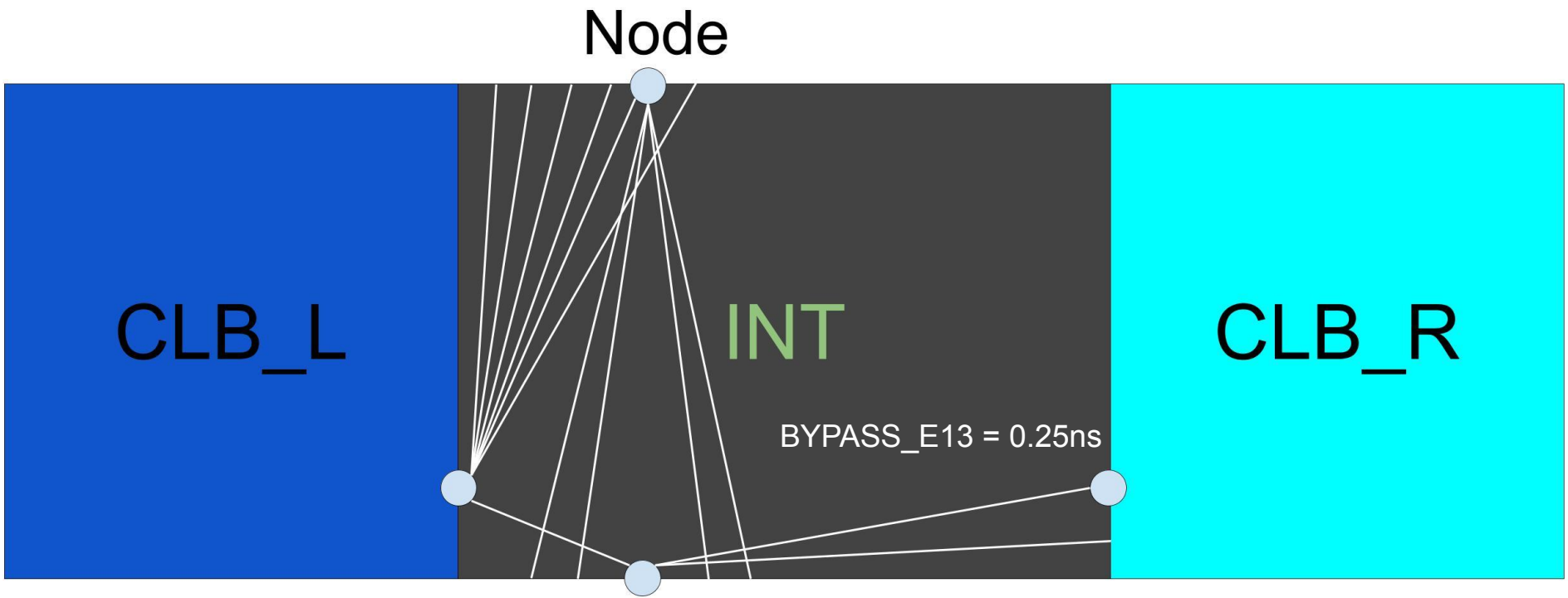
Symmetry: symmetrical elements on FPGA have similar timing characteristics

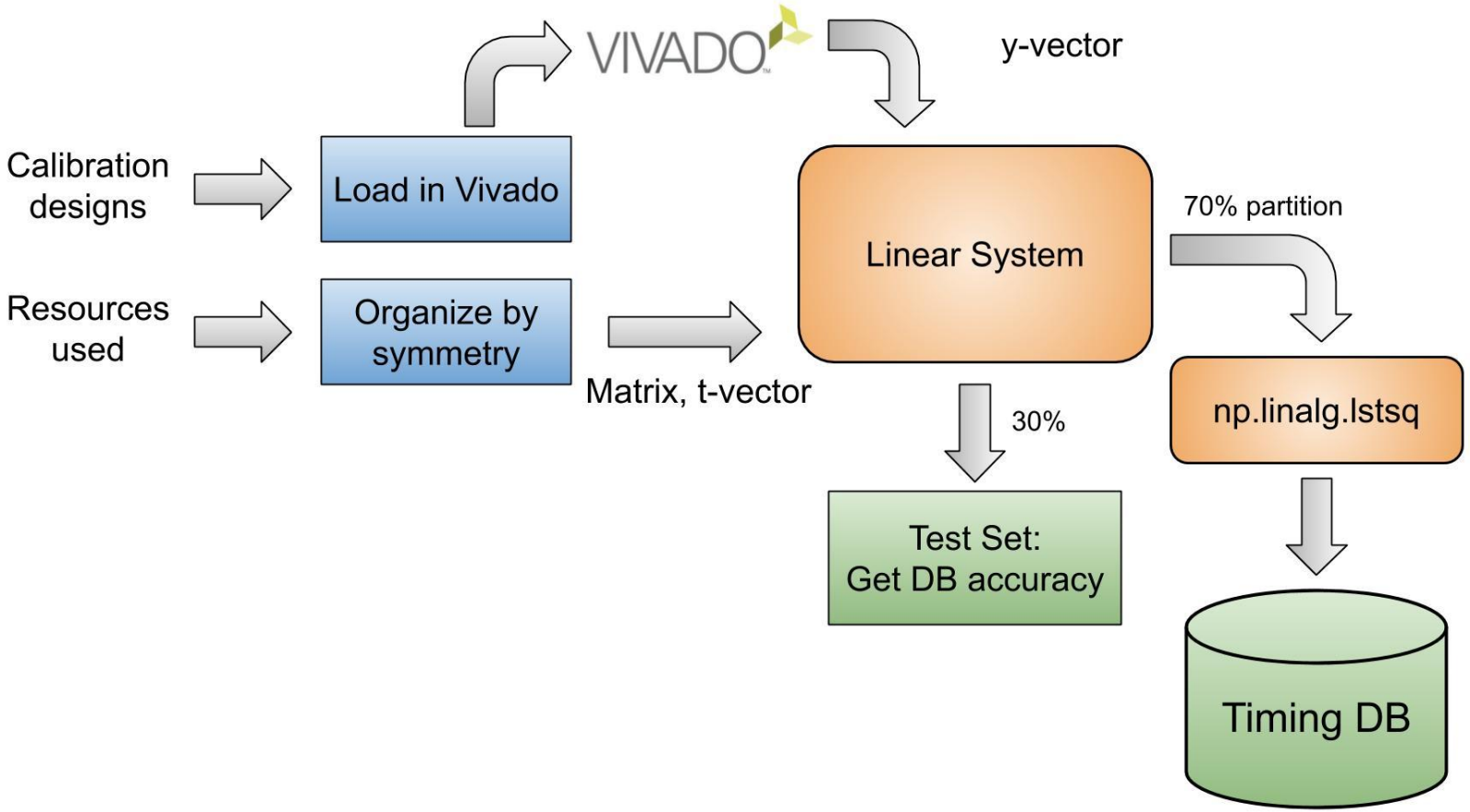
Narrow usage: RapidRoute only targets communication overlays




CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB
CLB INT CLB CLB INT CLB


Node
CLB_L
INT
CLB_R

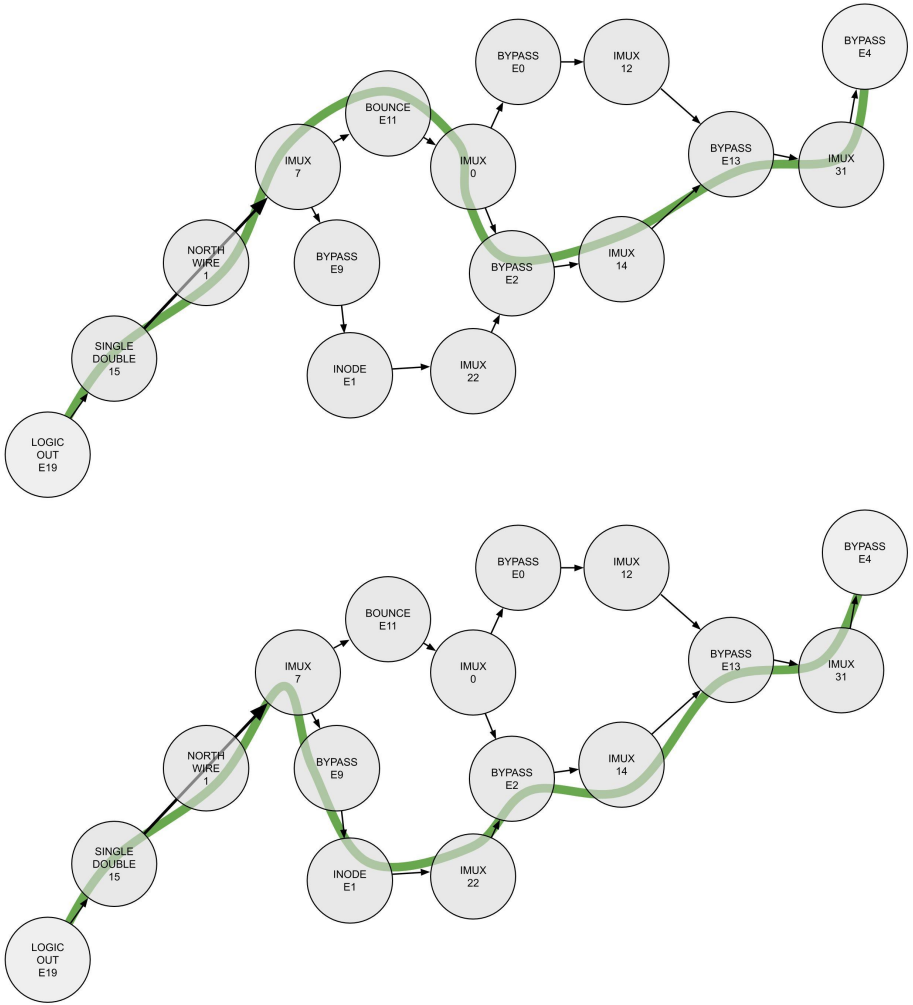

Node
CLB_L
INT
CLB_R
BYPASS_E13 = 0.25ns

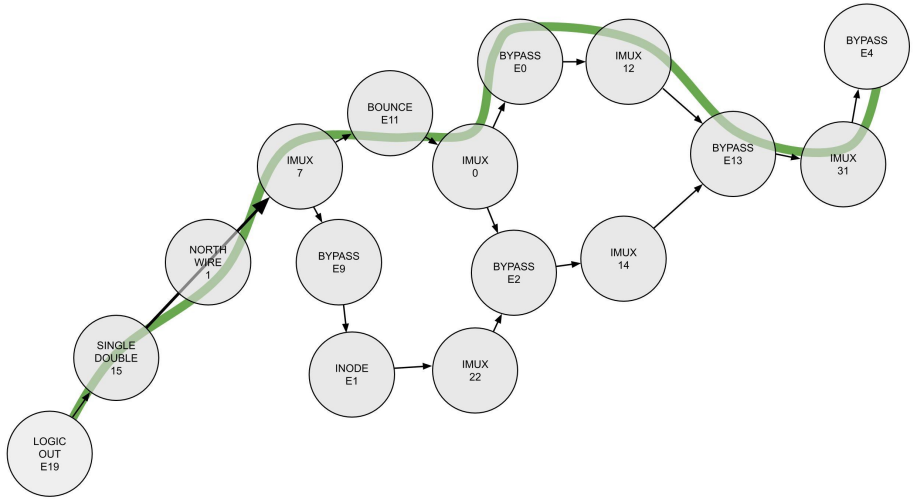

VIVADO
y-vector
Calibration designs
Load in Vivado
Resources used
Organize by symmetry
Matrix, t-vector
Linear System
70% partition
30%
np.linalg.lstsq
Test Set:
Get DB accuracy
Timing DB

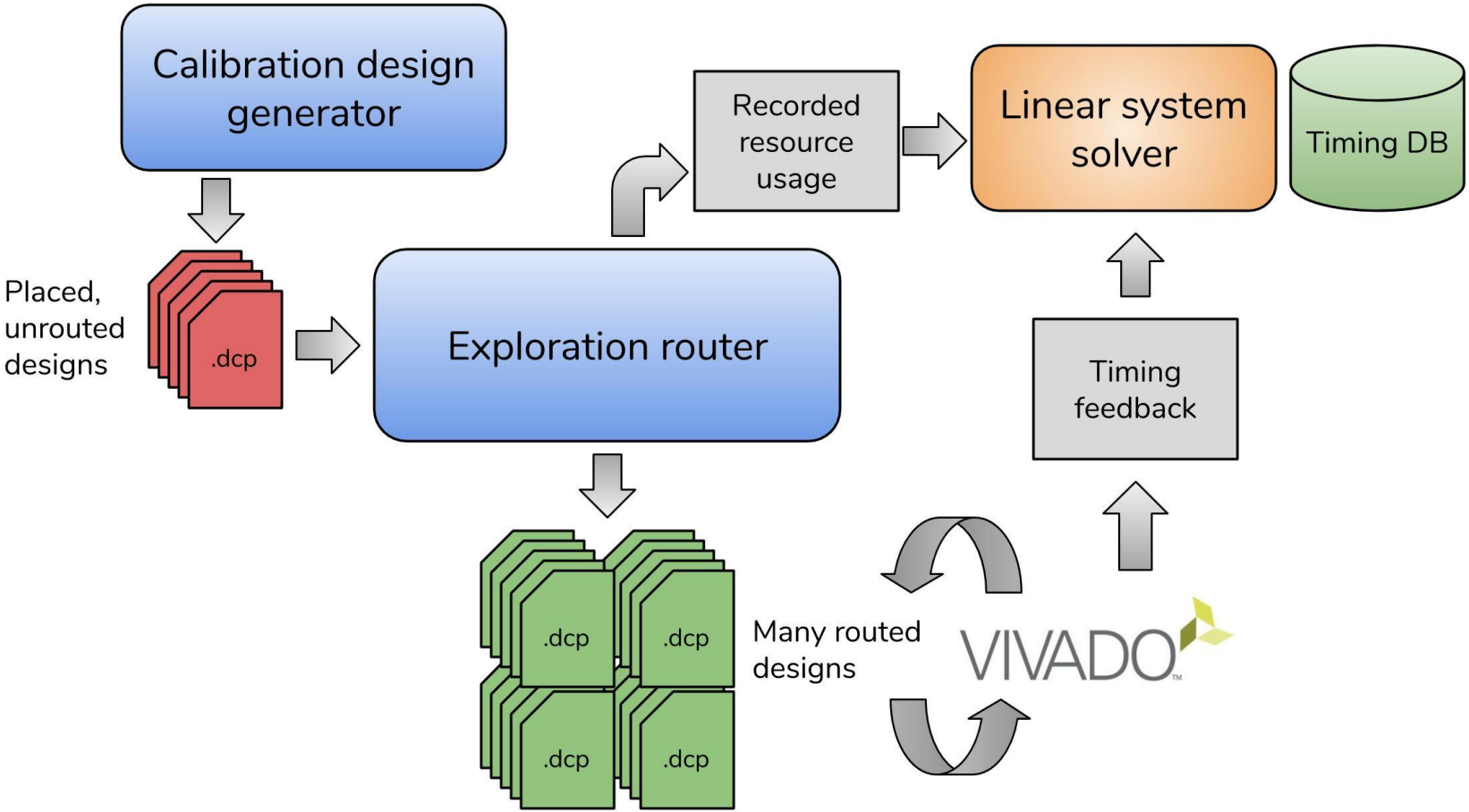
Calibration Designs

Designs: **1-bit signal**, with **changing start and end nodes**

1. For each design, **route in different ways**
2. Write out each routing result into DCP
3. Track which nodes are used for each route


LOGIC OUT E19
SINGLE DOUBLE 15
NORTH WIRE 1
IMUX 7
BOUNCE E11
IMUX 0
BYPASS E0
IMUX 12
BYPASS E4
BYPASS E13
IMUX 31
BYPASS E9
BYPASS E2
IMUX 14
INODE E1
IMUX 22


LOGIC OUT E19
SINGLE DOUBLE 15
NORTH WIRE 1
IMUX 7
BOUNCE E11
IMUX 0
BYPASS E0
IMUX 12
BYPASS E4
BYPASS E13
IMUX 31
BYPASS E9
BYPASS E2
IMUX 14
INODE E1
IMUX 22


Calibration design generator
Recorded resource usage
Linear system solver
Timing DB
Placed, unrouted designs
.dcp
Exploration router
Timing feedback
.dcp
.dcp
.dcp
.dcp
Many routed designs
VIVADO

Experimental Setup

- Metrics:
  - **Timing prediction accuracy**
- Calibration designs
  - **Single-bit routes of arbitrary displacement**
  - **Various devices and speed grades**
- Compare methods:
  - Partition **70% training**, **30% testing** of all calibration runs

Experimental Setup

- Devices:
  - Ultrascale XCKU115 (-3, -2, -1 speed grades)
  - Ultrascale+ XCKU5P (-3, -2, -1 speed grades)
- Vivado: 2018.3
- RapidWright: 2018.3.3-beta
- Hardware: Intel Xeon E5-1630

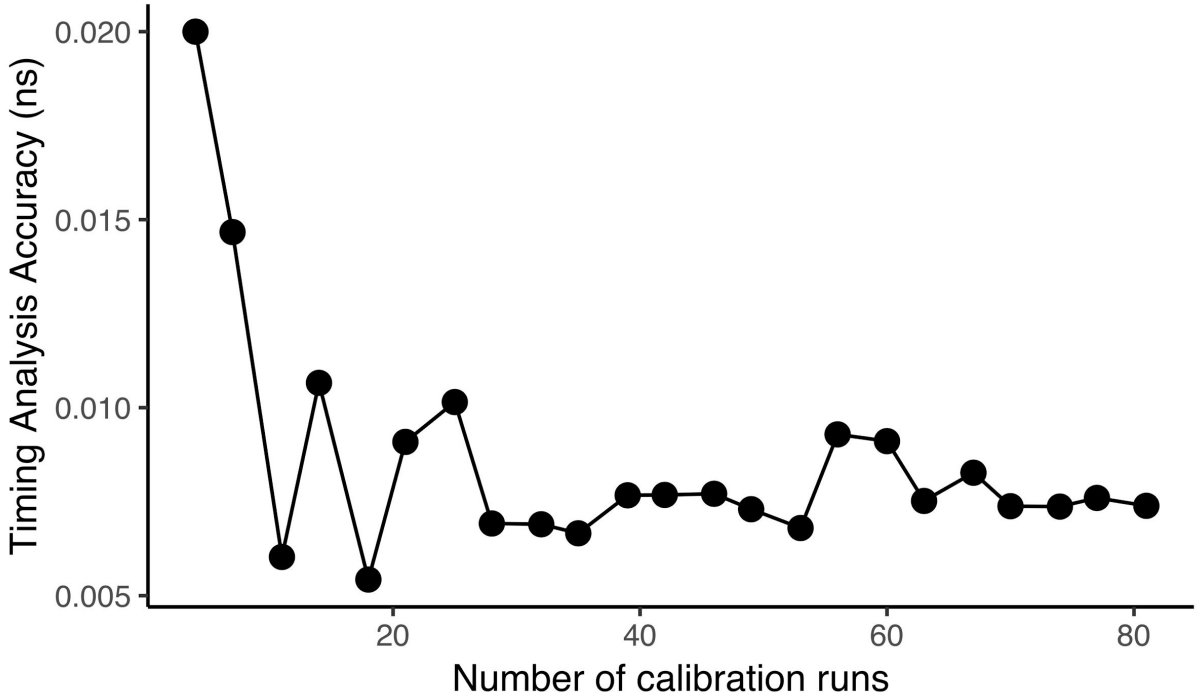
Timing Accuracy

Measuring accuracy:

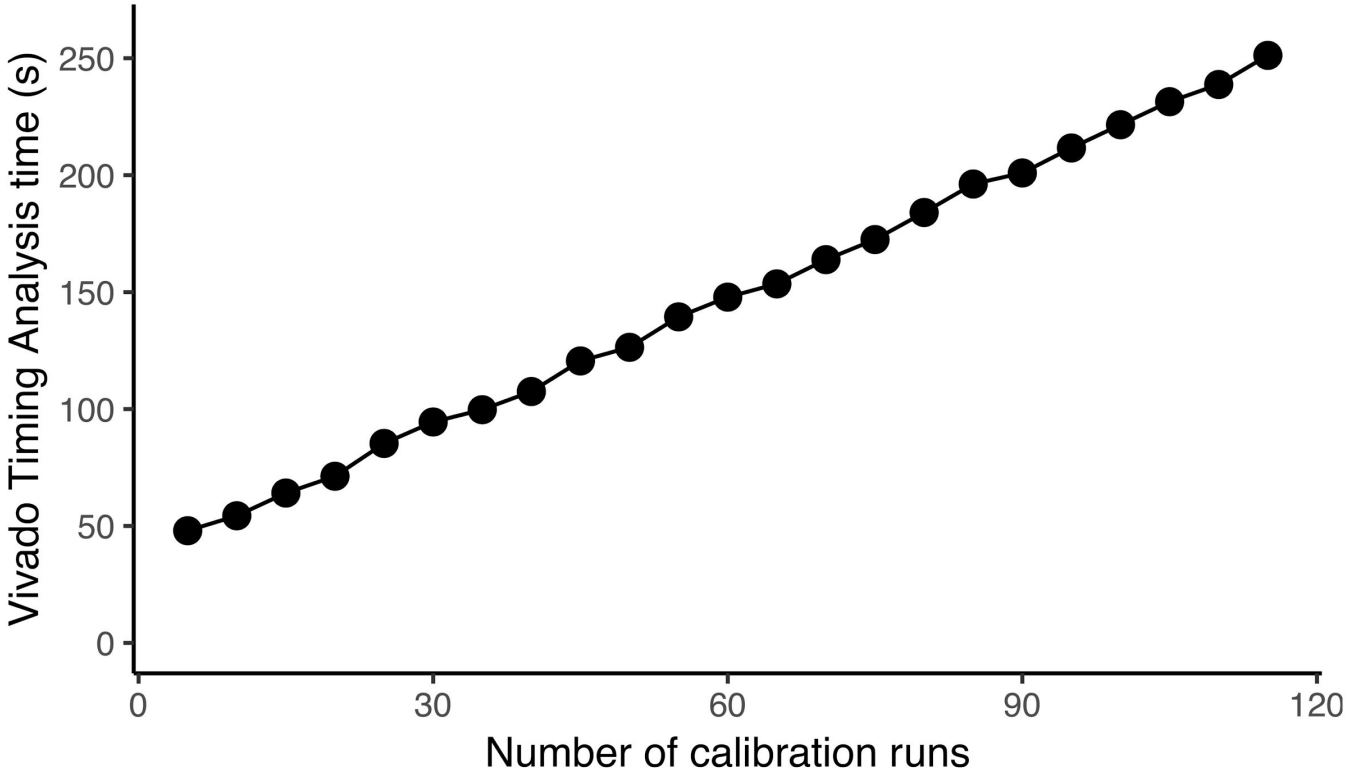
We check datapath prediction errors of 30% partition.

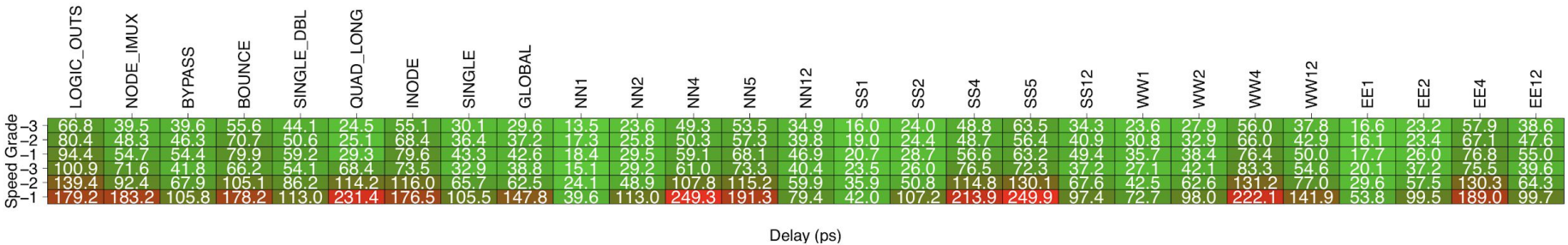
X-axis: size of 70% partition

Y-axis: average prediction error



Vivado Runtime




LOGIC_OUTS
NODE_IMUX
BYPASS
BOUNCE
SINGLE_DBL
QUAD_LONG
INODE
SINGLE
GLOBAL
NN1
NN2
NN4
NN5
NN12
SS1
SS2
SS4
SS5
SS12
WW1
WW2
WW4
WW12
EE1
EE2
EE4
EE12
Speed Grade
–3 66.8 39.5 39.6 55.6 44.1 24.5 55.1 30.1 29.6 13.5 23.6 49.3 53.5 34.9 16.0 24.0 48.8 63.5 34.3 23.6 27.9 56.0 37.8 16.6 23.2 57.9 38.6
–2 80.4 48.3 46.3 70.7 50.6 25.1 68.4 36.4 37.2 17.3 25.8 50.3 57.3 39.8 19.0 24.4 48.7 56.4 40.9 30.8 32.9 66.0 42.9 16.1 23.4 67.1 47.6
–1 94.4 54.7 54.4 79.9 59.2 29.3 79.6 43.3 42.6 18.4 29.5 59.1 68.1 46.9 20.7 28.7 56.6 63.2 49.4 35.7 38.4 76.4 50.0 17.7 26.0 76.8 55.0
–3 100.9 71.6 41.8 66.2 54.1 68.4 73.5 32.9 38.8 15.1 29.2 61.0 73.3 40.4 23.5 26.0 76.5 72.3 37.2 27.1 42.1 83.3 54.6 20.1 37.2 75.5 39.6
–2 139.4 92.4 67.9 105.1 86.2 114.2 116.0 65.7 62.5 24.1 48.9 107.8 115.2 59.9 35.9 50.8 114.8 130.1 67.6 42.5 62.6 131.2 77.0 29.6 57.5 130.3 64.3
–1 179.2 183.2 105.8 178.2 113.0 231.4 176.5 105.5 147.8 39.6 113.0 249.3 191.3 79.4 42.0 107.2 213.9 249.9 97.4 72.7 98.0 222.1 141.9 53.8 99.5 189.0 99.7
Delay (ps)

Additional Notes

- Output timing database is extremely small (< 100KB)
- Majority of timing extraction solver runtime is due to Vivado query wait times

Integrating with RapidRoute

- RapidRoute accepts timing database file(s) as input, overwriting default heuristic
- Heuristic has nearly identical computing cost as default heuristic

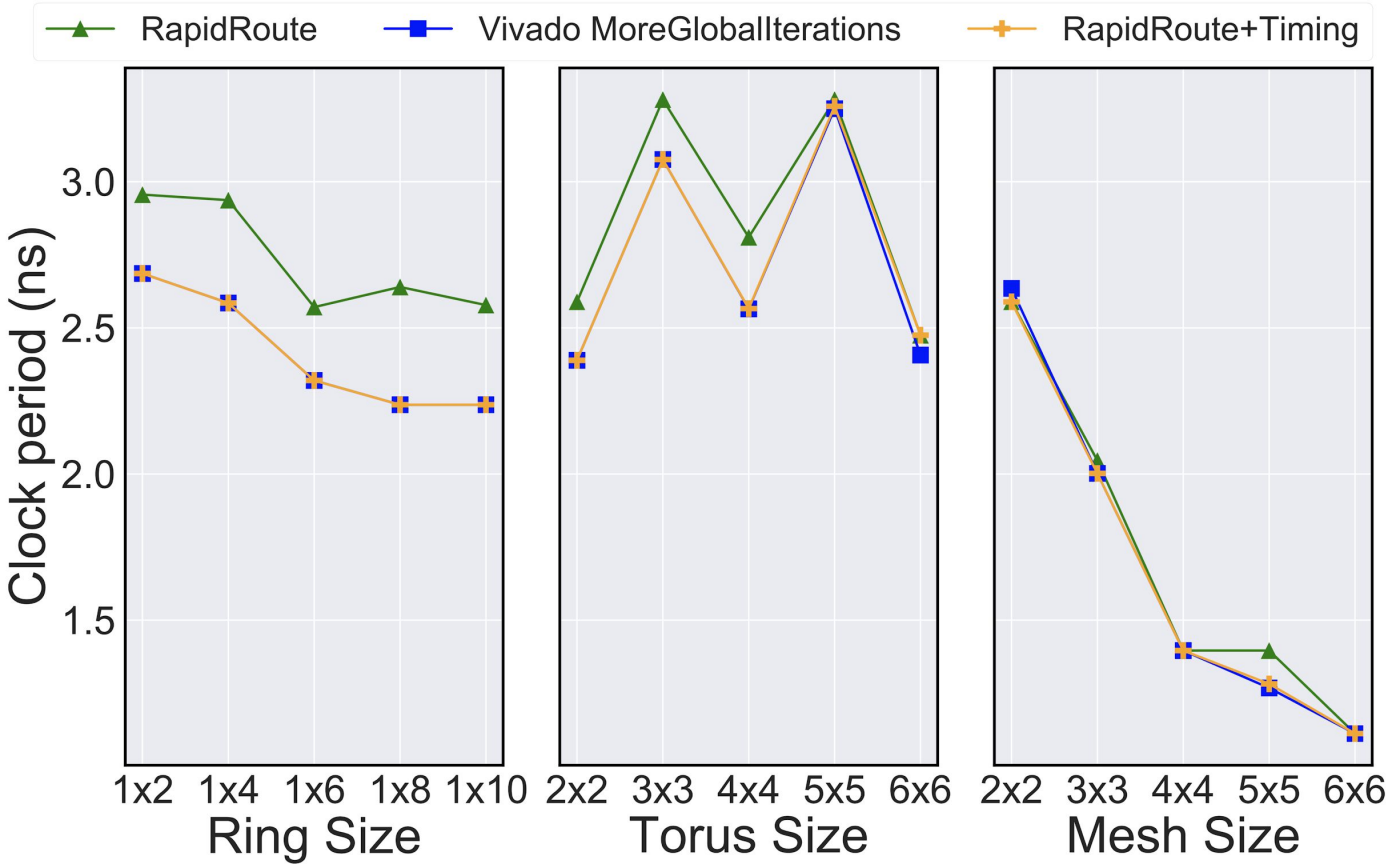
Experimental Setup

- Metrics:
  - **Timing performance**
  - **Routing runtimes**
- Communication structures:
  - **1D rings, 2D torii, 2D meshes**
- Compare methods:
  - **RapidRoute default, RapidRoute+Timing**, **Vivado**

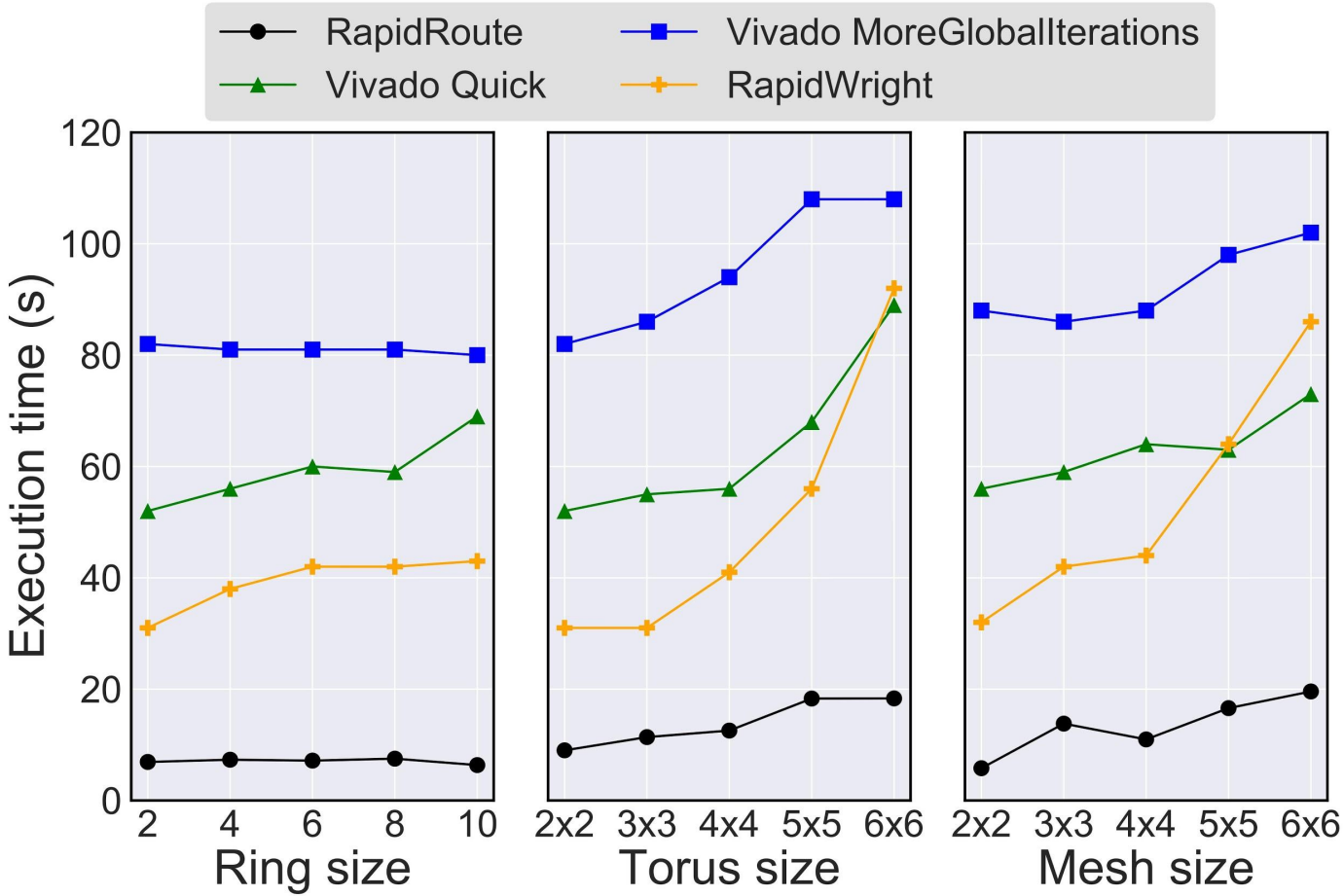
Experimental Setup

- Device: Ultrascale XCKU115 `xcku115-flva1517-3-e`
- Vivado: 2018.3
- RapidWright: 2018.3.3-beta
- Hardware: 32-core 2.6GHz Intel Xeon

Timing Results



Routing Runtime



Conclusion

- We developed a **timing extraction tool**, which is **light-weight** and **highly-accurate**
- Timing results expected to be within 1% error margin
- Total calibration phase takes minutes
- Extremely lightweight output and usage

Improved RapidRoute

- RapidRoute retains a **5-8x** routing speed advantage over Vivado
- RapidRoute now gains **competitive timing performance** on communication overlay designs