EPFL

A Network Tour of Data Science | EE-558

Understanding Negotiated-Congestion Routing for FPGAs

Group 49

Mikhail Asiatici
Samuel Bosch
Zander Harteveld
Stefan Nikolić

January 21th, 2020

What Is an FPGA?

FPGA is a chip that can change behavior after power on

What Is an FPGA?

FPGA is a chip that can change behavior after power on

But, hardware can not magically appear or disappear as needed

What Is an FPGA?

FPGA is a chip that can change behavior after power on

But, hardware can not magically appear or disappear as needed

We can only decide which part of the existing hardware we are going to use and for what, and which part will be left unused

What Is an FPGA?

FPGA is a chip that can change behavior after power on

But, hardware can not magically appear or disappear as needed

We can only decide which part of the existing hardware we are going to use and for what, and which part will be left unused

This holds both for gates and connections

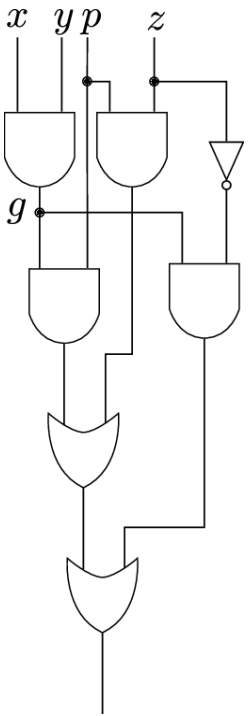
How Is an FPGA Used?

$$f(x, y, z, p) = (xy + z)p + xy\overline{z}$$

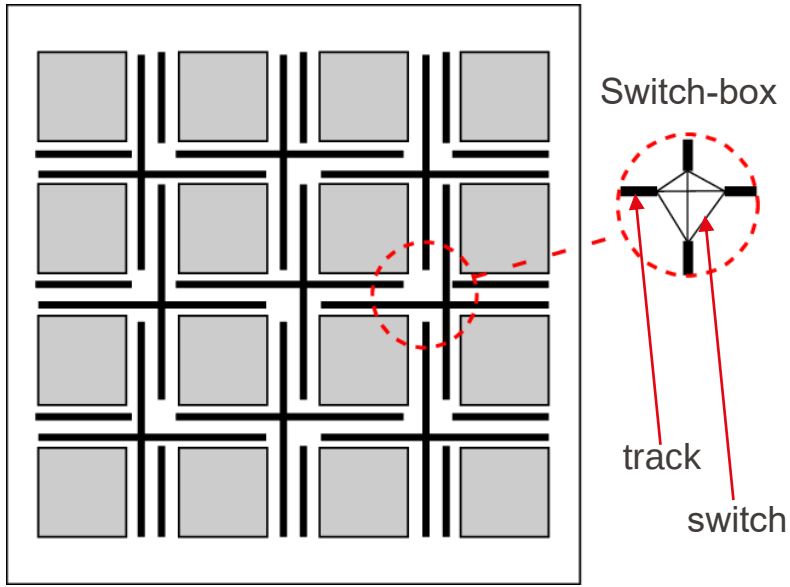
1) Specify the desired behavior
   (a circuit graph)

2) Determine location of each gate
   (placement)

3) Determine how their terminals are connected
   (routing)

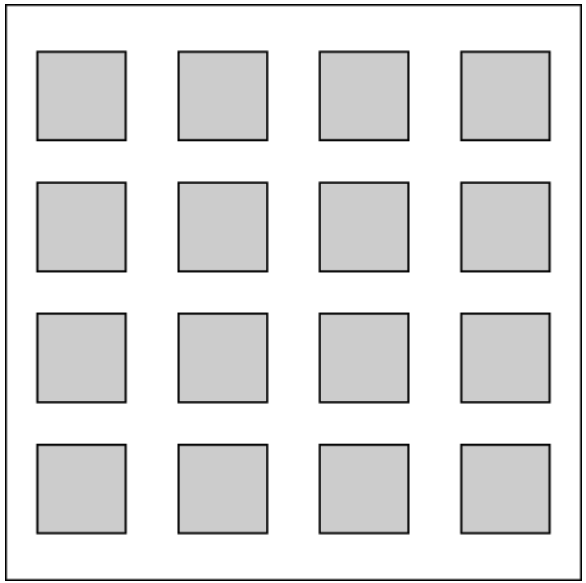


What Is an FPGA?



Turn switches "on" or "off"
to connect existing tracks

Placement



Task: Assign each node of the circuit graph a unique location on the FPGA grid s.t:

1) Total wire length is minimized

2) Delay of the slowest (critical) path is minimized

A. Marquardt, V. Betz, and J. Rose, "Timing-driven placement for FPGAs", FPGA'00

Placement: Timing Optimization

We need to know t_d for each connection (u, v)

Placement: Timing Optimization

We need to know t_d for each connection (u, v)

This depends on locations of u and v, and the path through tracks and switches chosen to connect them

Placement: Timing Optimization

We need to know t_d for each connection (u, v)

This depends on locations of u and v, and the path through tracks and switches chosen to connect them

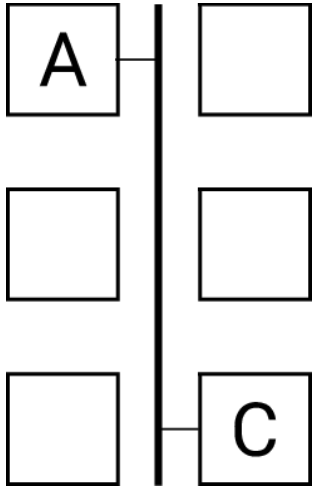
Routing after each place modification is not feasible!

Placement: Timing Optimization

Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement

Placement: Timing Optimization

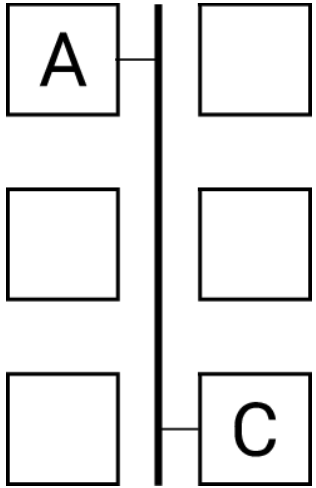
Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement

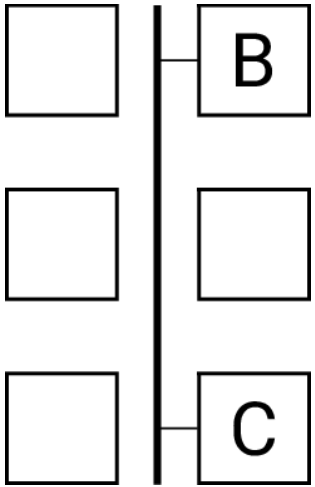


t_d(A, C) = 126 ps

Placement: Timing Optimization

Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement



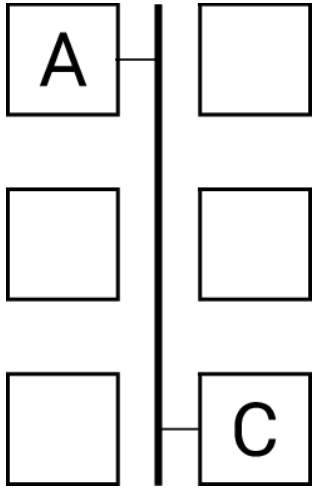


t_d(A, C) = 126 ps

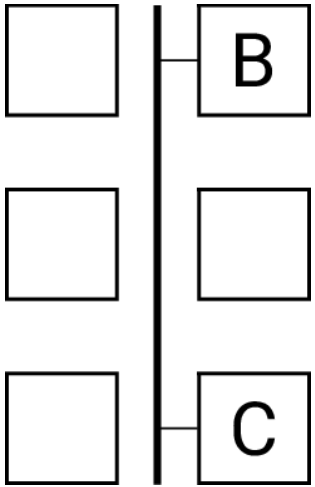
t_d(B, C) = 126 ps

Placement: Timing Optimization

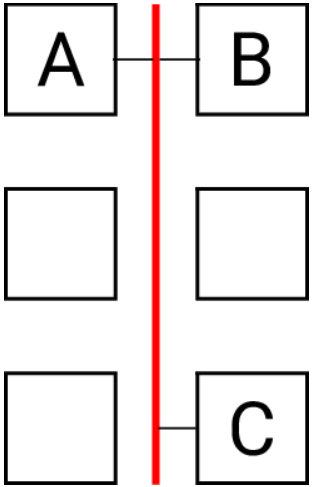
Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement



t_d(A, C) = 126 ps



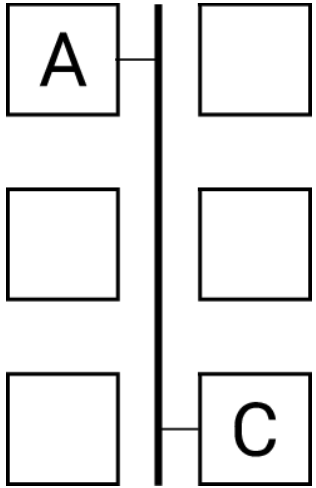
t_d(B, C) = 126 ps



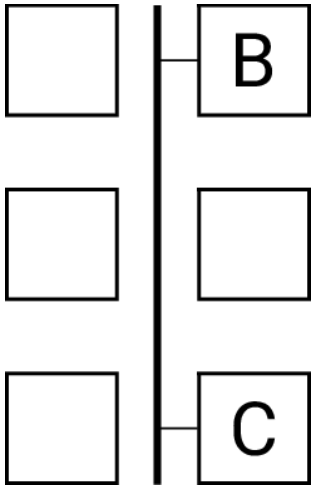
t_d(B, C) = 126 ps
t_d(A, C) = 126 ps

Placement: Timing Optimization

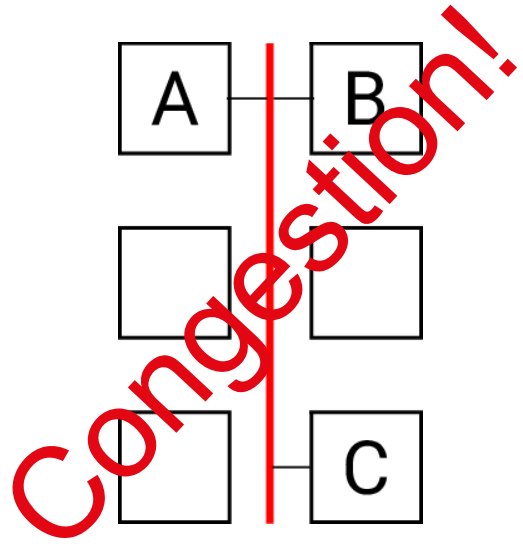
Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement



t_d(A, C) = 126 ps



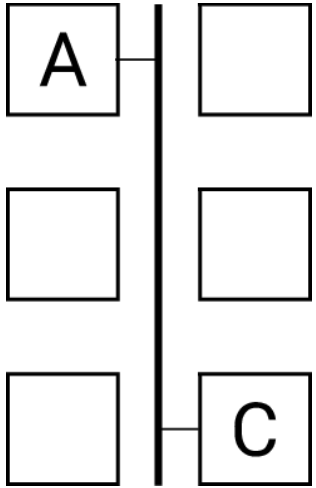
t_d(B, C) = 126 ps



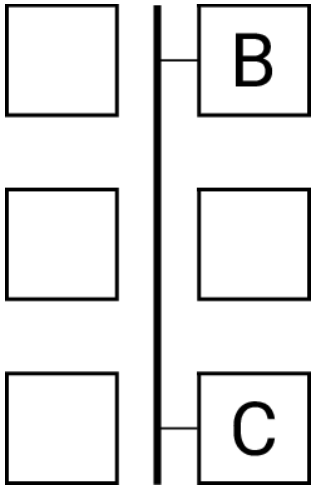
t_d(B, C) = 126 ps
t_d(A, C) = 126 ps

Placement: Timing Optimization

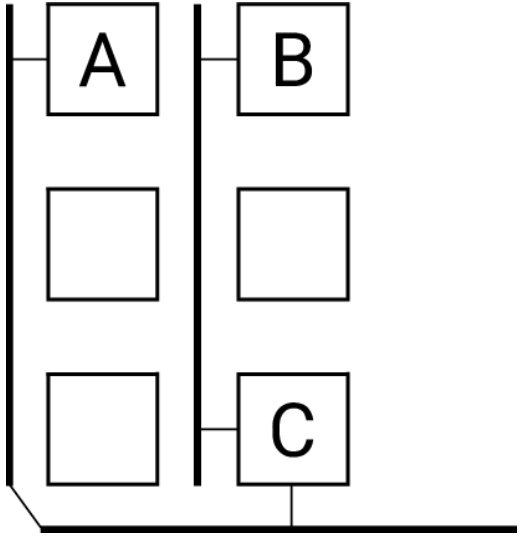
Solution: Compute the fastest path delay between each pair of locations (A, B), store the delays in a look-up table, and use this table during placement



t_d(A, C) = 126 ps



t_d(B, C) = 126 ps



t_d(B, C) = 126 ps
t_d(A, C) = 218 ps

When Does Congestion Influence Delay?

Goals:

- Understand the phenomenon
- Predict it from placement
- Is the placer causing it, or the inherent structure of the circuit?

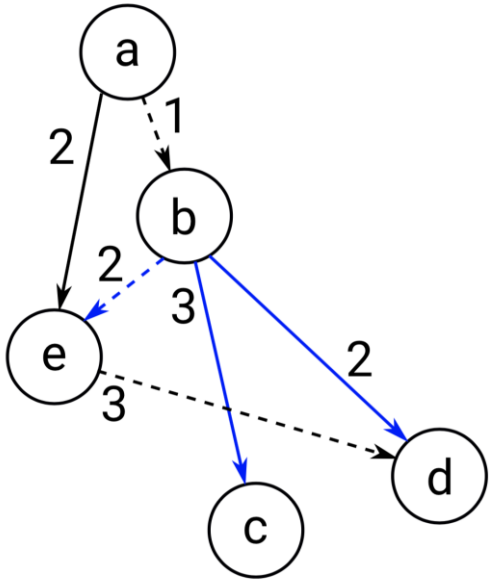
Potential benefits:

- Understand which algorithms need to be improved and get inspiration for doing that
- Get inspiration for FPGA architecture improvement by better appreciating the circuit structure
- Enable wider architectural space exploration by avoiding routing (if we final delay can be more accurately predicted from placement)

The basic model design

Net:

- The set of connections between an output of a particular gate and inputs of all the gates that it drives.
- How can we weight the nets? Criticality!

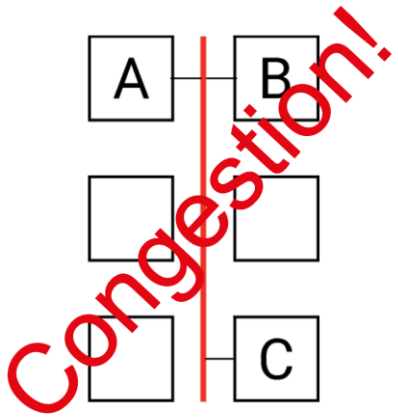


Critical net: Criticality = 1

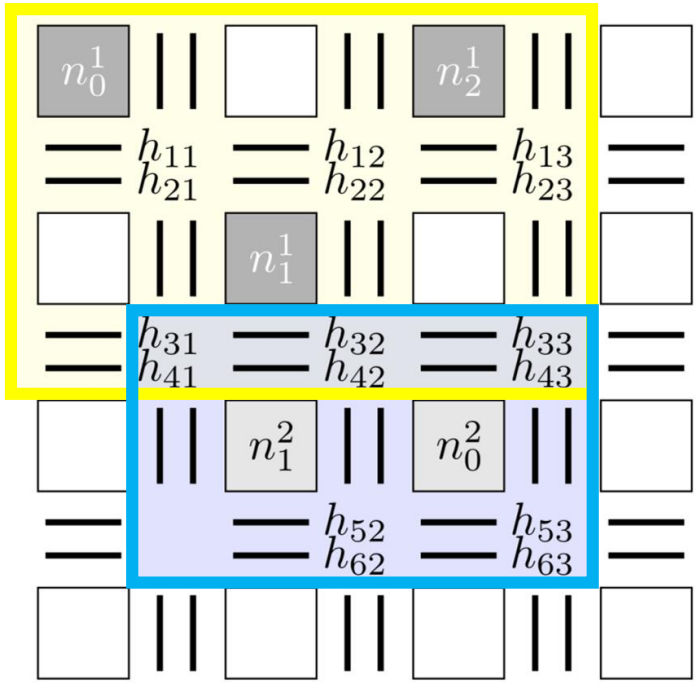
We may run into a problem...

Congestion

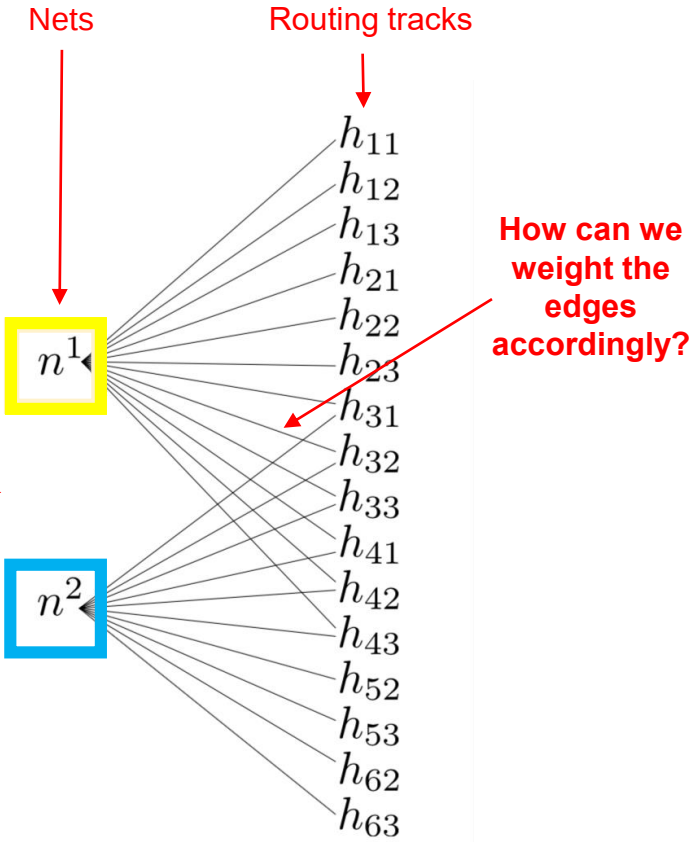
- problematic if, for the given placement of a given circuit, two nets that are (close to) critical must be routed using the same track in order to achieve the shortest route



Design bipartite graph



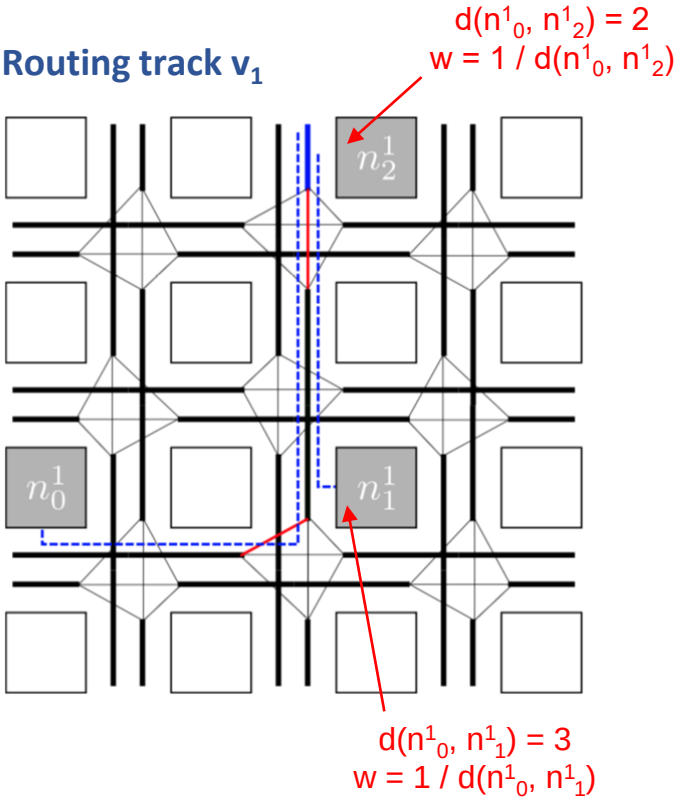
Bipartite graph

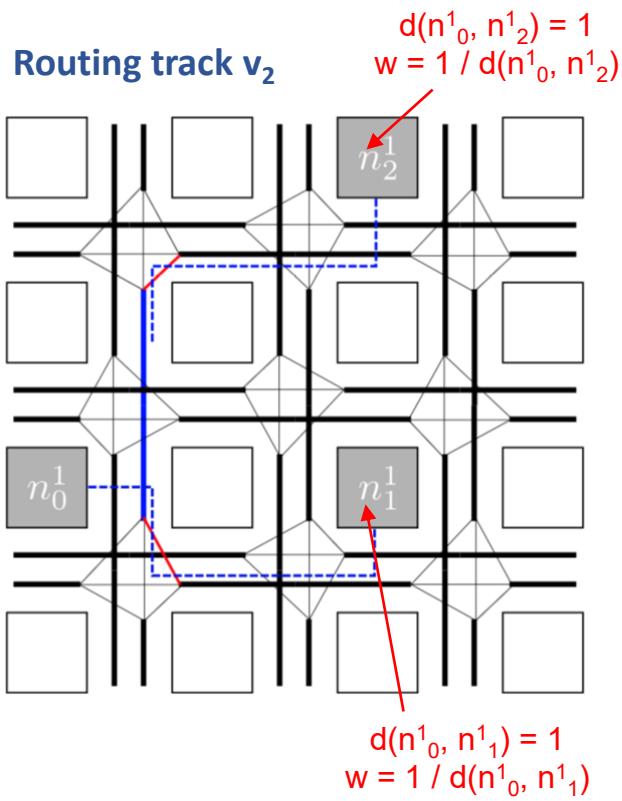


How can we weight the edges accordingly?

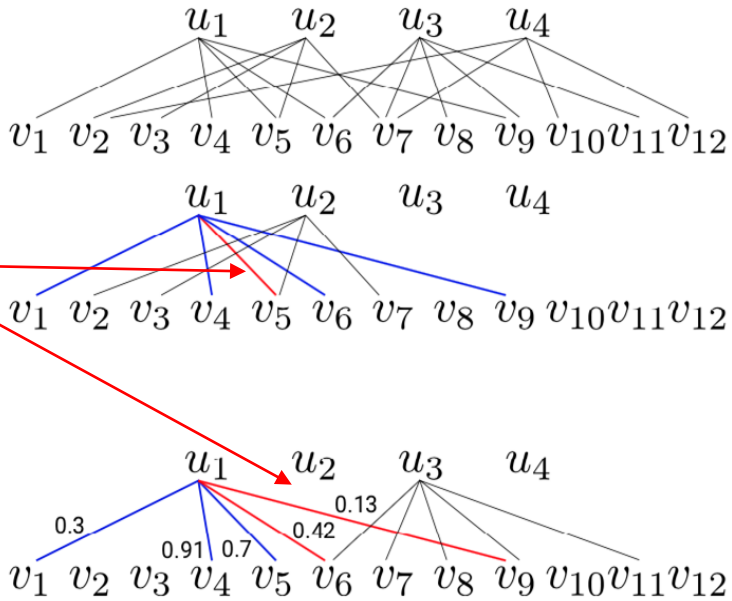
Mikhail Asiatici, Samuel Bosch, Zander Harteveld and Stefan Nikolić

Bipartite graph edge weights



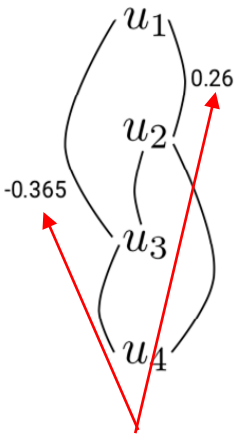


Projection of the bipartite graph



How can we describe these conflicting regions?

Projection



gives an idea about how much a net would lose by giving up its preferred track to its opponent net!

$$\langle w_c(u_1) \rangle = \frac{\sum_{v \in \{N_1 \cap N_2\}} w(u_1, v)}{|N_1 \cap N_2|}$$

$$\langle w_{uc}(u_1) \rangle = \frac{\sum_{v \in \{N_1 \setminus N_2\}} w(u_1, v)}{|N_1 \setminus N_2|}$$

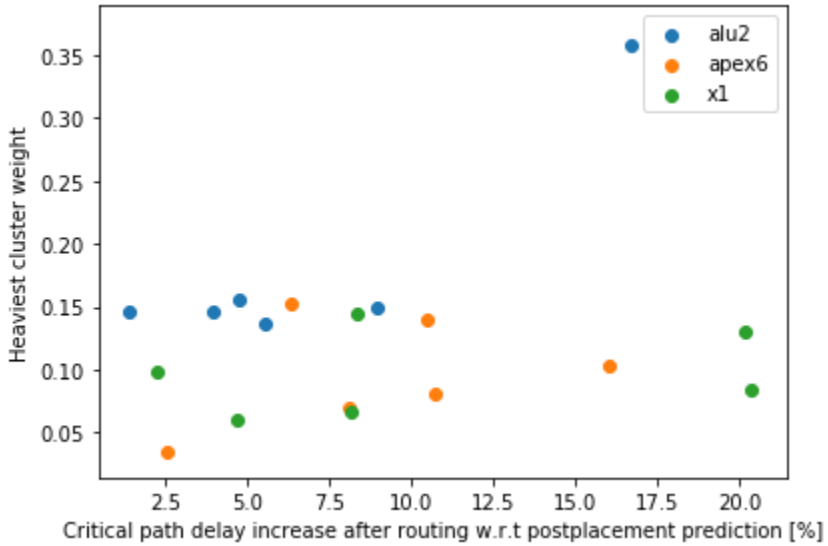
$$w_p(u_1, u_2) = min(\{\langle w_c(u_1) \rangle - \langle w_{uc}(u_1) \rangle, \langle w_c(u_2) \rangle - \langle w_{uc}(u_2) \rangle\})$$

Mikhail Asiatici, Samuel Bosch, Zander Harteveld and Stefan Nikolić

Testing the Main Conjecture

CONJECTURE 1. *The routed delay will be noticeably larger than the post-placement prediction iff there are heavy clusters in the projection of the net-representing partition.*

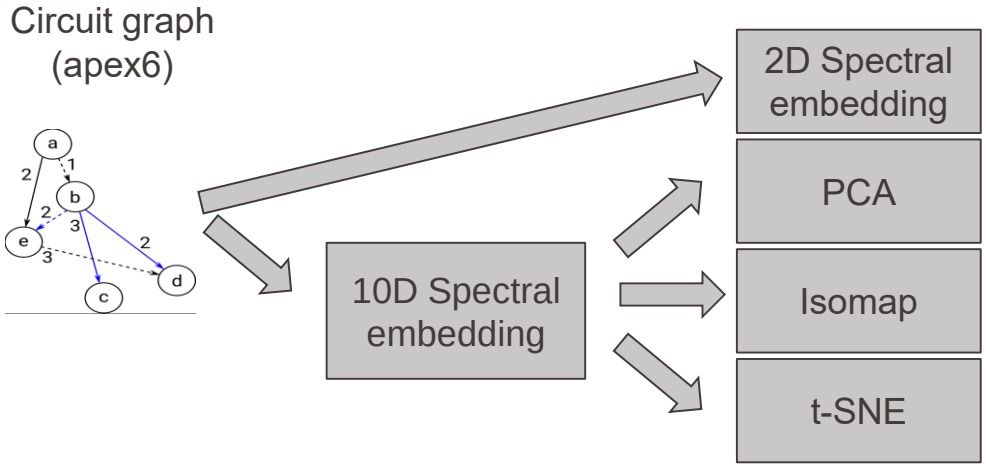
1. Identify 16 clusters using spectral clustering on projection graph
2. Find heaviest clusters
3. Plot heaviest cluster weight vs routed critical path delay increase w.r.t. post-placement prediction

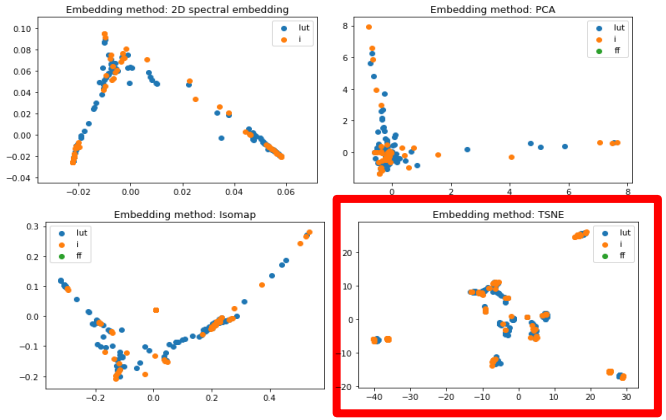


Conjecture rejected

How Good is the Placer?

- Is the structure of the original circuit graph preserved in the 2D embedding produced by the placer?

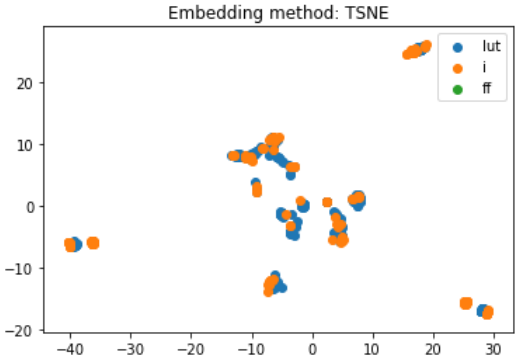




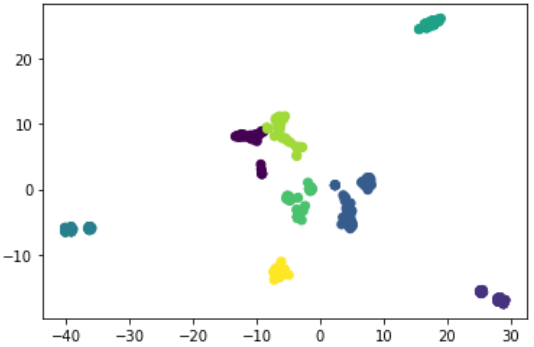
Most clearly distinguishable clusters

Mikhail Asiatici, Samuel Bosch, Zander Harteveld and Stefan Nikolić

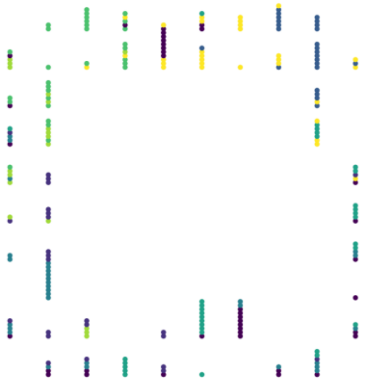
How Good is the Placer?



KMeans



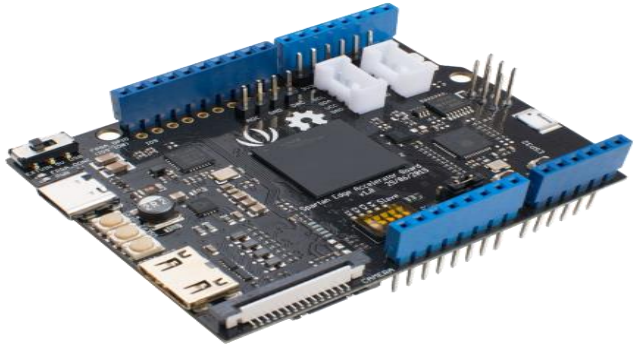
Using the same labels for the placed circuit



Most of the clusters in the original circuit graph seem to have been preserved by the placer

What comes next?

- Industry seem to have optimized the hell out of FPGAs
- It might be very difficult to make significant improvements



What comes next?

2018 International Conference on Field-Programmable Logic and Applications

Machine-Learning Based Congestion Estimation for Modern FPGAs

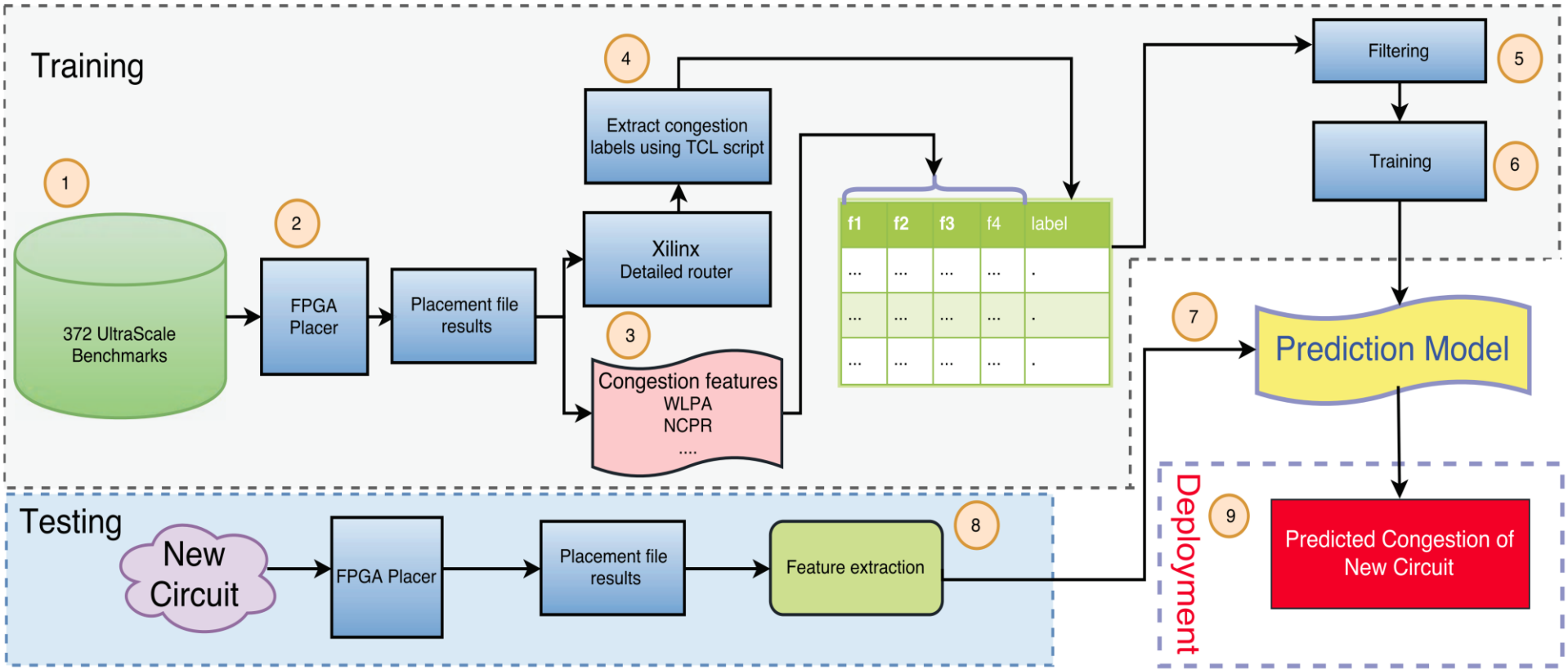
D. Maarouf, A. Alhyari, Z. Abuowaimer, T. Martin, A. Gunter, G. Grewal, S. Areibi, A. Vannelli
School of Engineering/School of Computer Science, University of Guelph
Guelph, Ontario, Canada

- **25% higher prediction accuracy**

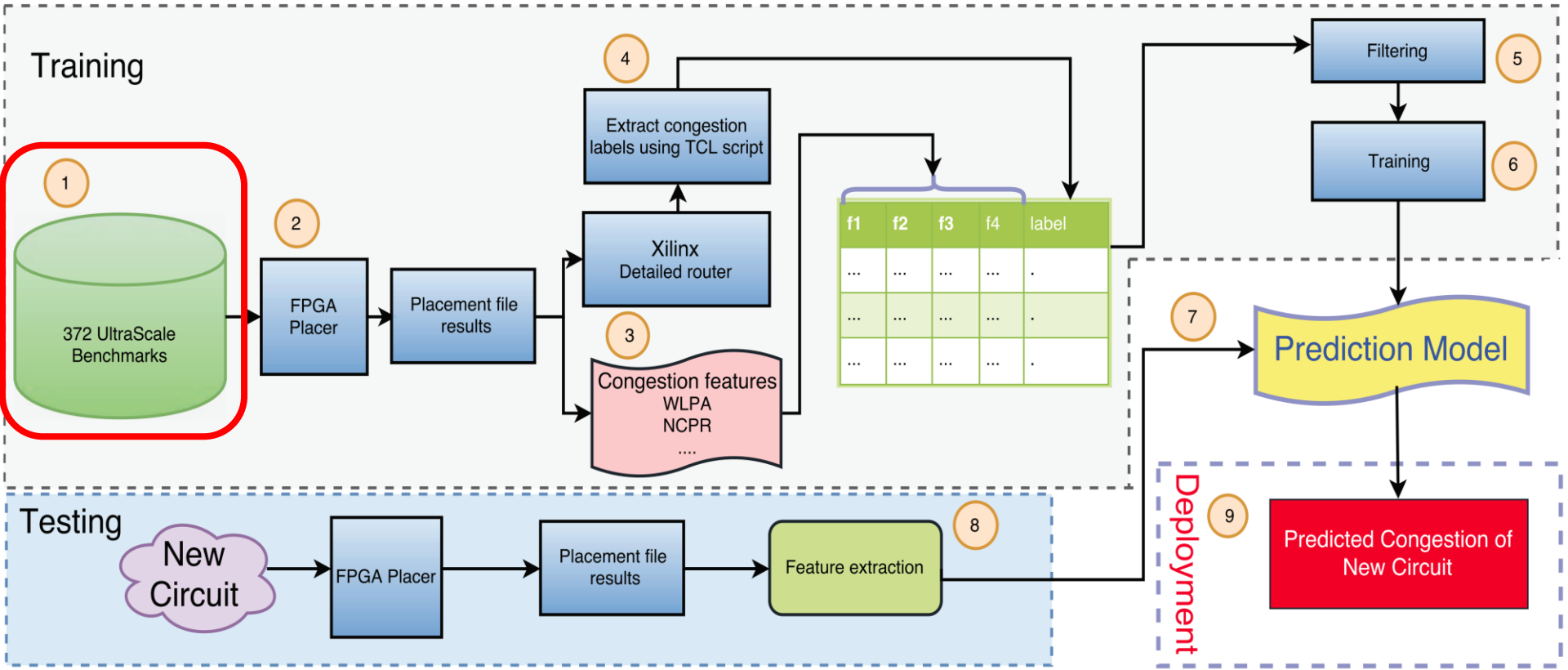
  (compared with the state-of-the-art ML congestion-estimation models)

- Same accuracy as congestion estimates using **global router**, but much faster.

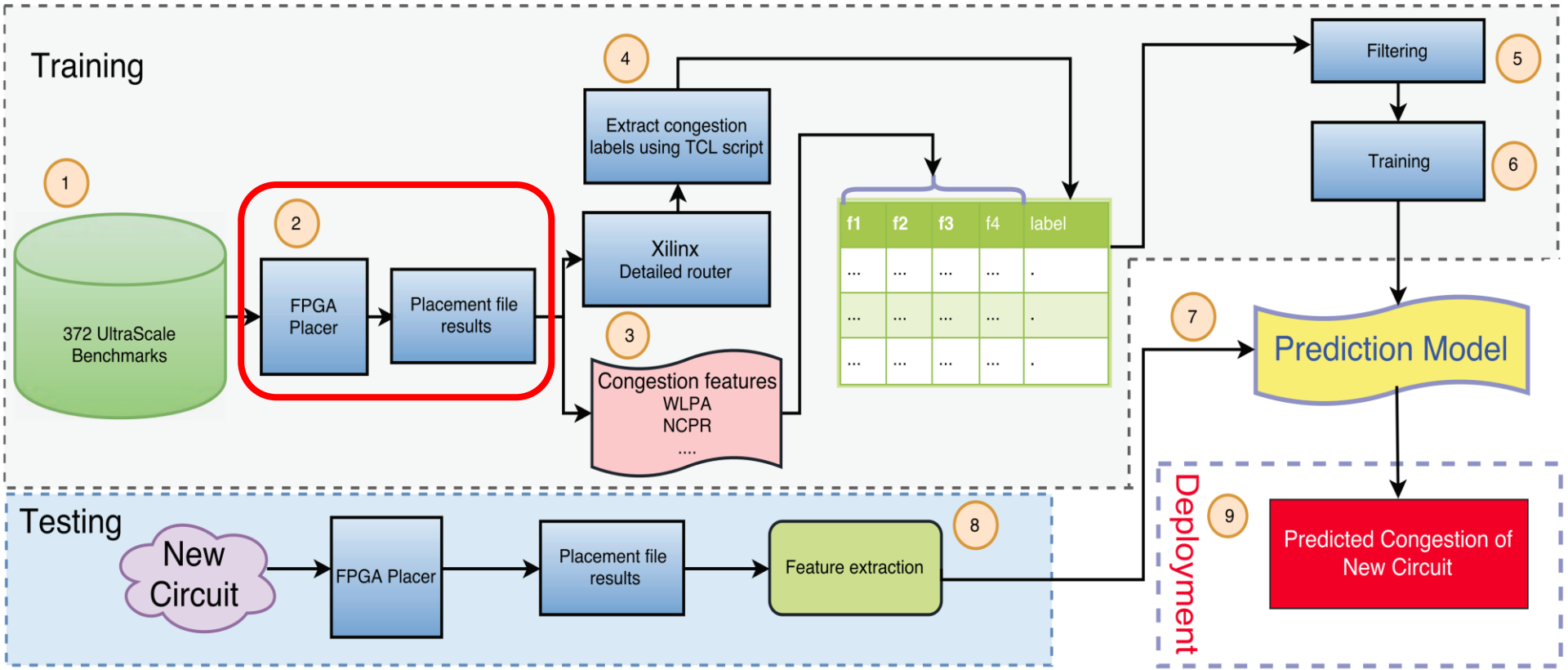
What comes next?



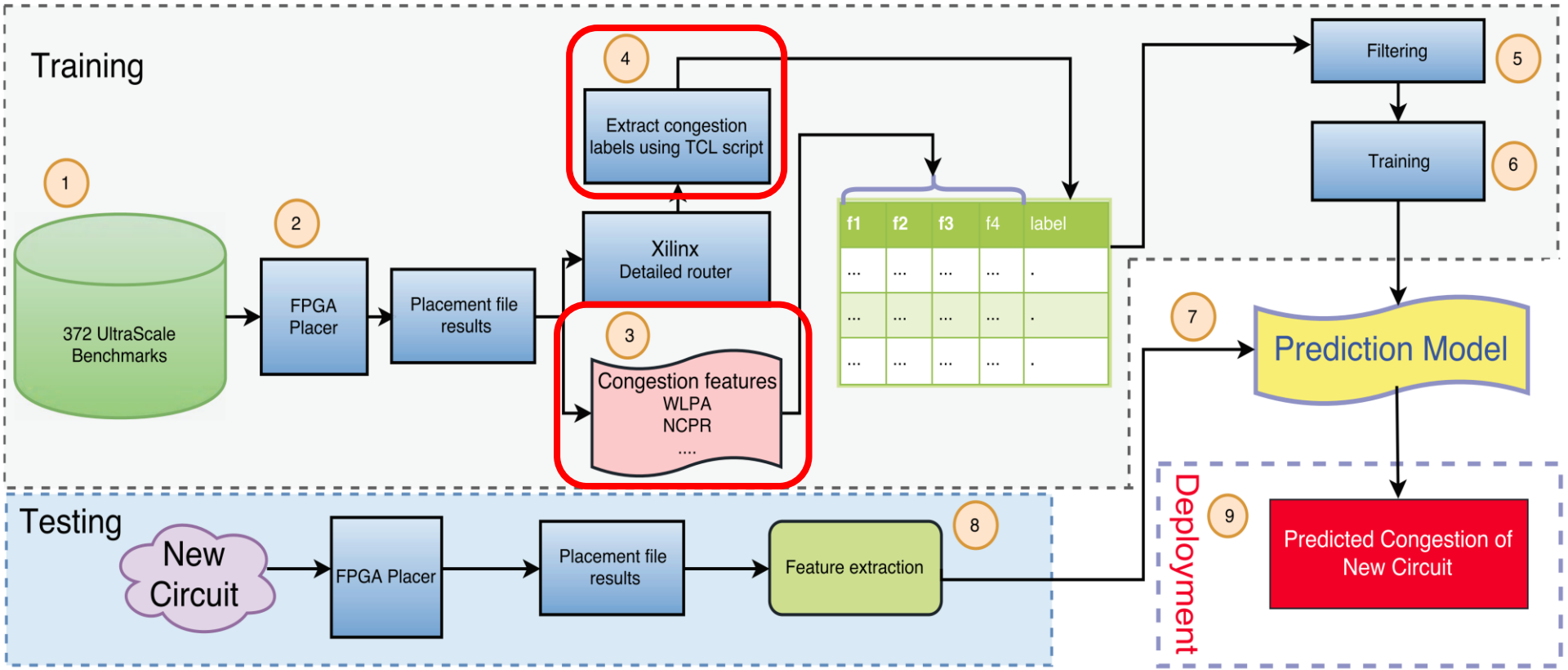
What comes next?



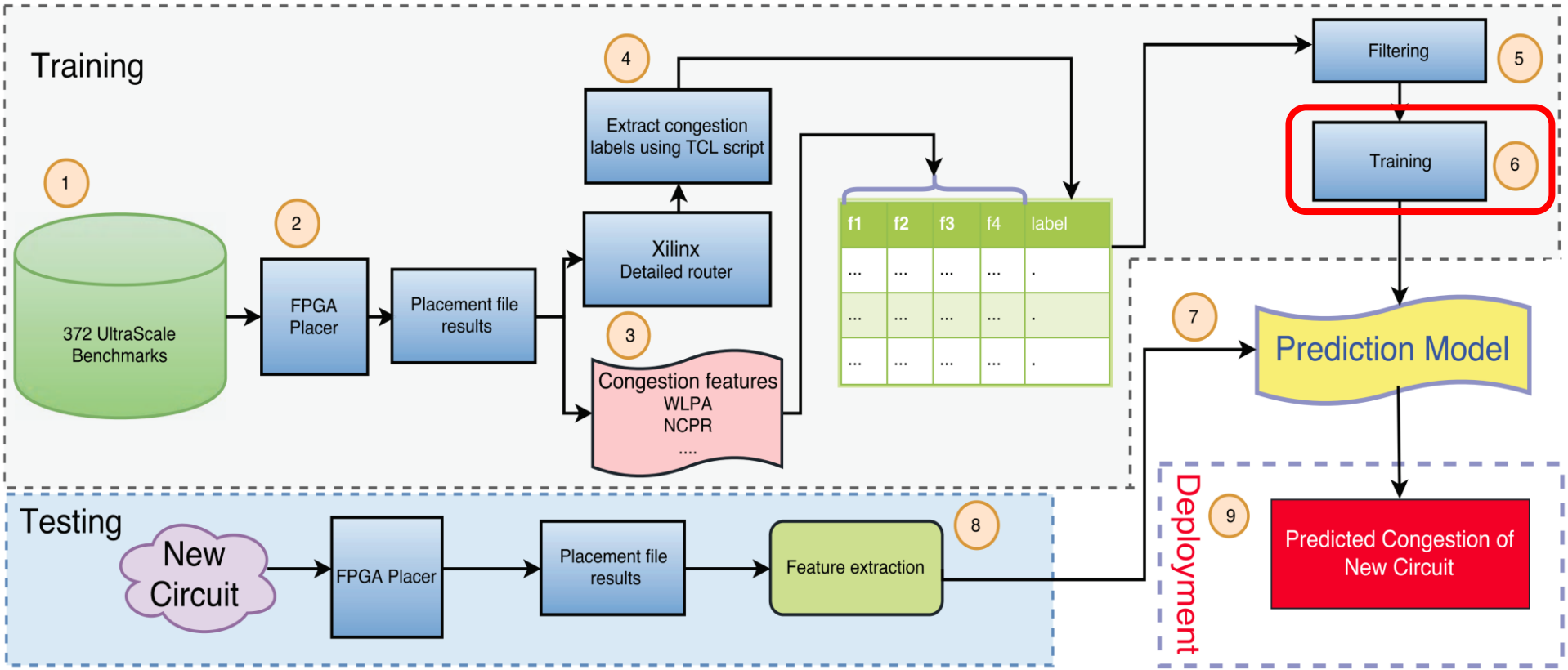
What comes next?



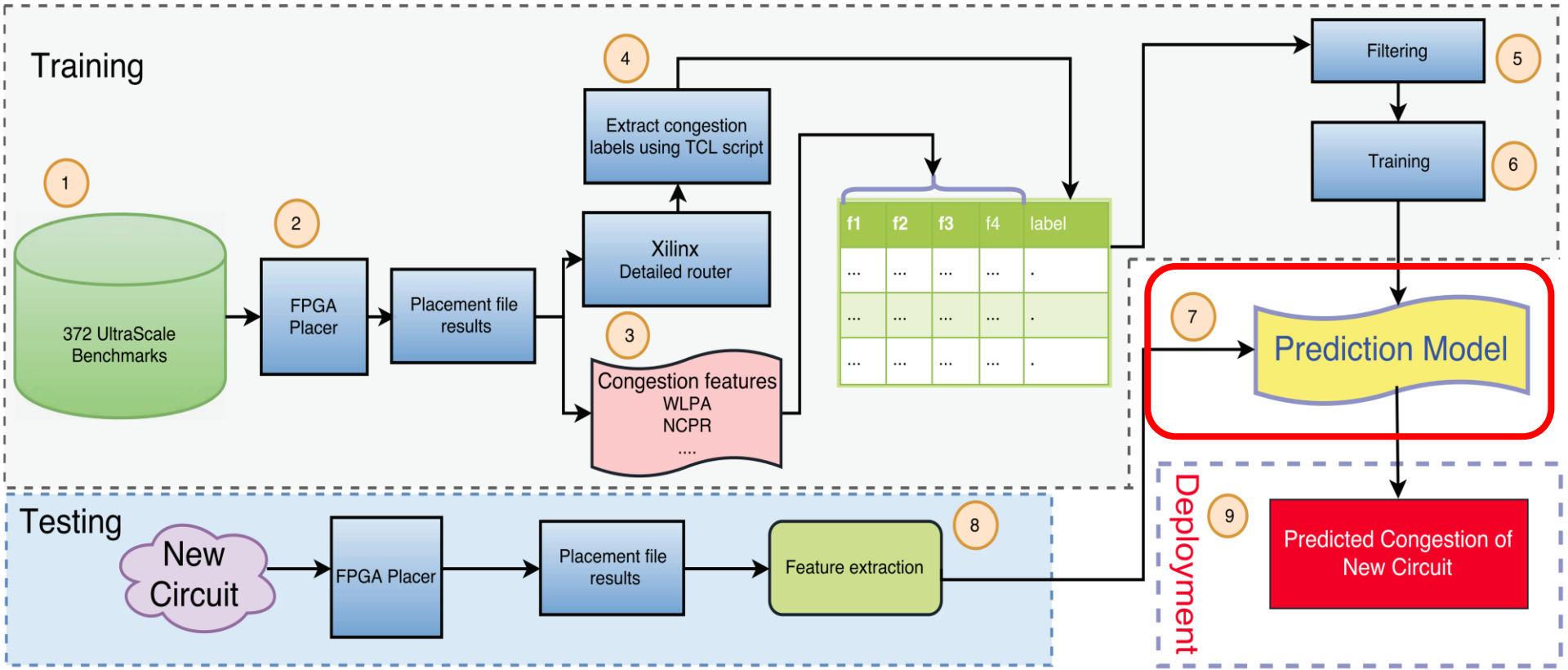
What comes next?

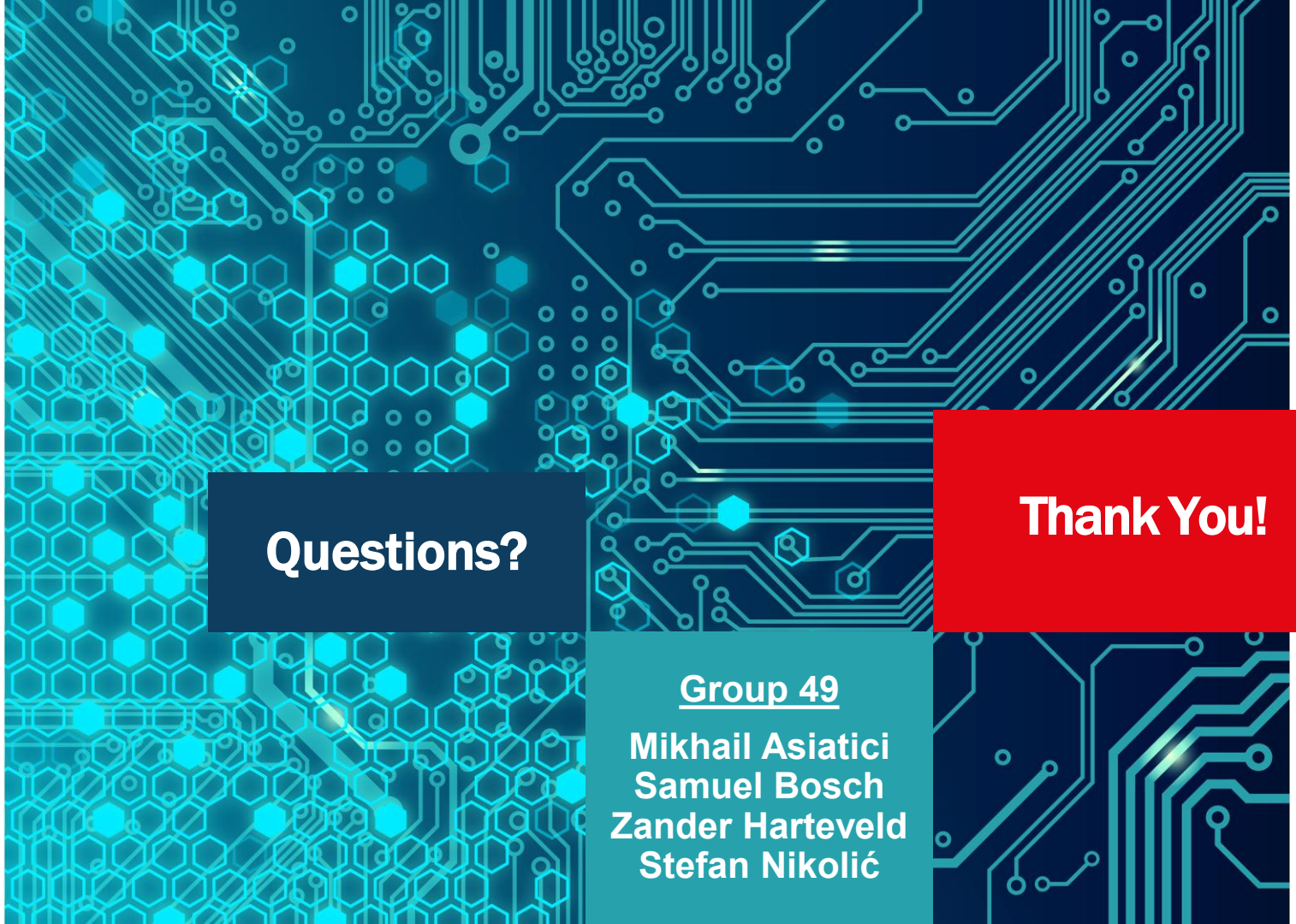


What comes next?



What comes next?





Questions?

Thank You!

Group 49

Mikhail Asiatici
Samuel Bosch
Zander Harteveld
Stefan Nikolić

References

- [1] M. Austin, W. L. Neto, L. Amaru, X. Tang, and P.-E. Gaillardon. Towards a novel logic synthesis framework supervised by convolutional neural network. In IWLS, 2019.
- [2] V. Betz and J. Rose. Automatic generation of FPGA routing architectures from high-level descriptions. In Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays, FPGA 2000, Monterey, CA, USA, February 10-11, 2000, pages 175–184, 2000.
- [3] V. Betz, J. Rose, and A. Marquardt. Architecture and CAD for deep-submicron FPGAs. Kluwer Academic, 1999.
- [4] P. Kannan, S. Balachandran, and D. Bhatia. FGREP - fast generic routing demand estimation for placed FPGA circuits. In Field-Programmable Logic and Applications, 11th International Conference, FPL 2001, Belfast, Northern Ireland, UK, August 27-29, 2001, Proceedings, pages 37–47, 2001.
- [5] J. Luu, J. Goeders, M. Wainberg, A. Somerville, T. Yu, K. Nasartschuk, M. Nasr, S. Wang, T. Liu, N. Ahmed, K. B. Kent, J. Anderson, J. Rose, and V. Betz. VTR 7.0: Next generation architecture and CAD system for FPGAs. ACM Transactions on Reconfigurable Technology and Systems (TRETS), 7(2):6:1–6:30, June 2014.
- [6] D. Maarouf, A. Alhyari, Z. Abuowaimer, T. Martin, A. Gunter, G. Grewal, S. Areibi, and A. Vannelli. Machine-learning based congestion estimation for modern FPGAs. In 2018 28th International Conference on Field Programmable Logic and Applications (FPL), pages 427–4277, 2018.
- [7] A. Marquardt, V. Betz, and J. Rose. Timing-driven placement for FPGAs. In Proceedings of the 2000 ACM/SIGDA Eighth International Symposium on Field Programmable Gate Arrays, FPGA ’00, pages 203–213, New York, NY, USA, 2000. ACM.
- [8] S. Yang. Logic synthesis and optimization benchmarks user guide, version 3.0. Technical report, Microelectronics Center of North Carolina, Research Triangle Park, N.C., Jan. 1991.
- [9] Manimegalai, R., E. Siva Soumya, V. Muralidharan, Balaraman Ravindran, V. Kamakoti, and D. Bhatia. "Placement and routing for 3D-FPGAs using reinforcement learning and support vector machines." In 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design, pp. 451-456. IEEE, 2005.