Karlsruhe Institute of Technology

Workshop on Realtime Machine Learning
Gießen 2024

Belle II

Real-time Graph Building on FPGAs for Machine Learning Trigger Applications in Particle Physics

Marc Neu, Jürgen Becker, Philipp Dorwarth, Torben Ferber, Lea Reuter, Slavomira Stefkova, Kai Unger

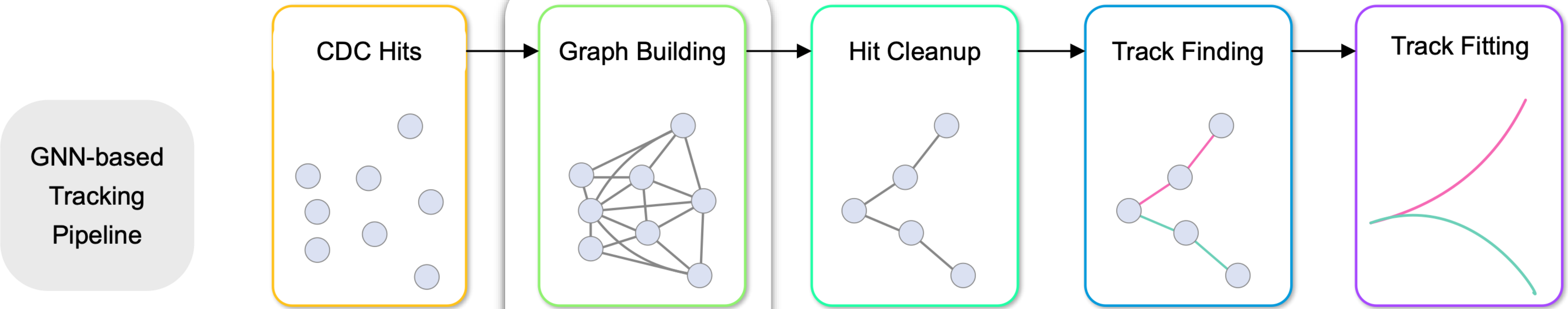
AI Trigger Group at Belle II

KIT ITIV
- Marc Neu
- Kai Unger
- Jürgen Becker

KIT ETP
- Isabel Haide
- Greta Heine
- Lea Reuter
- Slavomira Stefkova
- Torben Ferber

MPI & TUM
- Timo Forsthofer
- Simon Hiesl
- Christian Kiesling
- Alois Knoll

Motivation



See also previous talk Greta Heine

FPGA

How to build graphs on FPGAs in real-time?

Upgrade of the Belle II CDC Trigger System

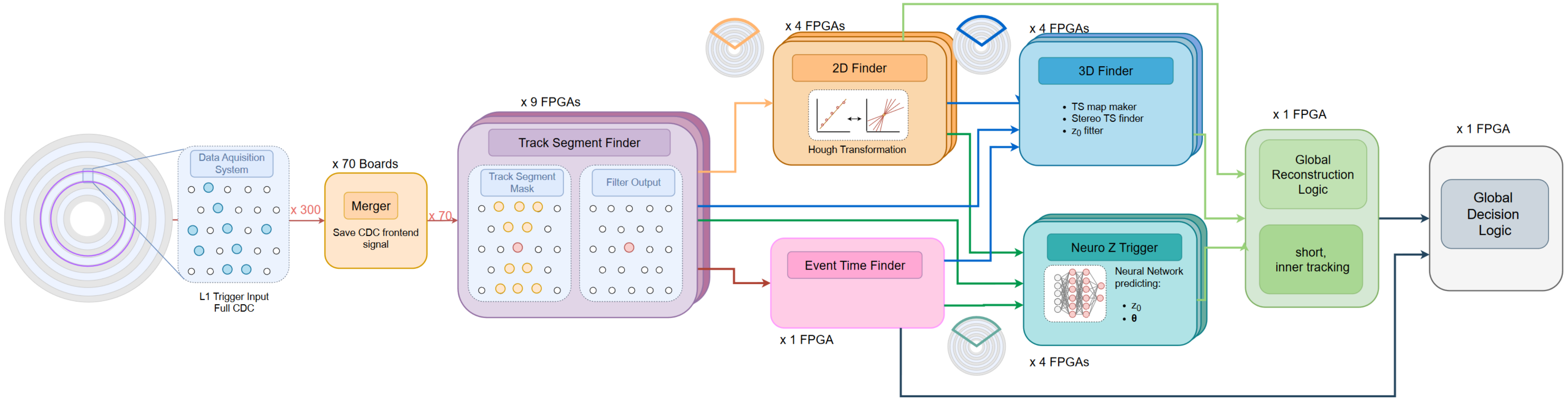


Figure from Lea Reuter

Upgrade of the Belle II CDC Trigger System

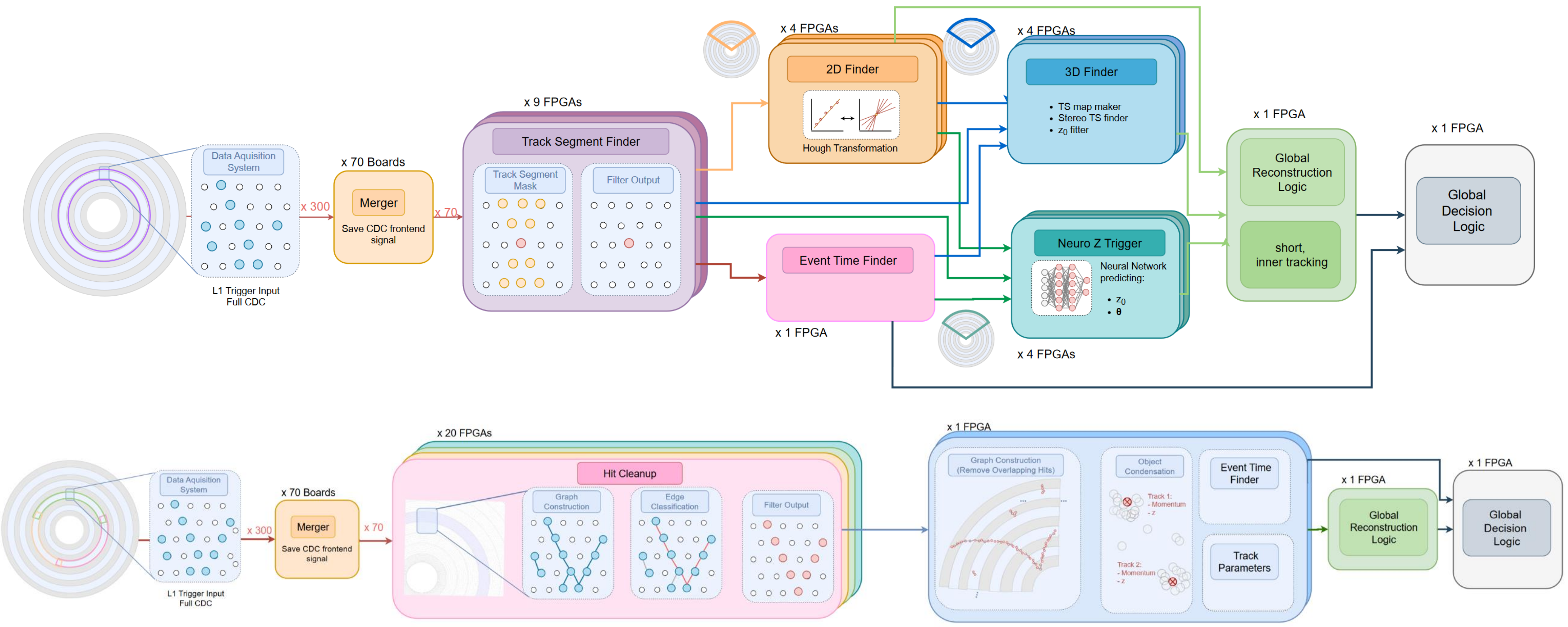


Figure from Lea Reuter

Upgrade of the Belle II CDC Trigger System

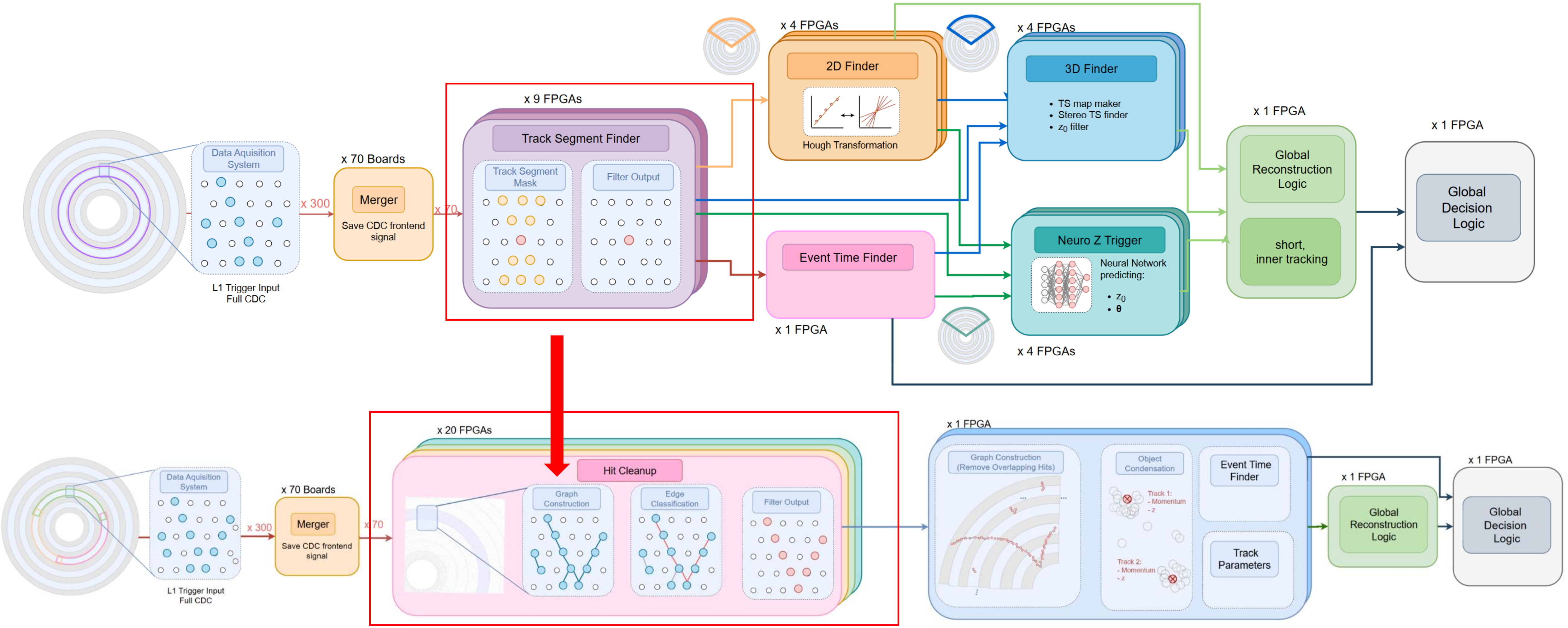
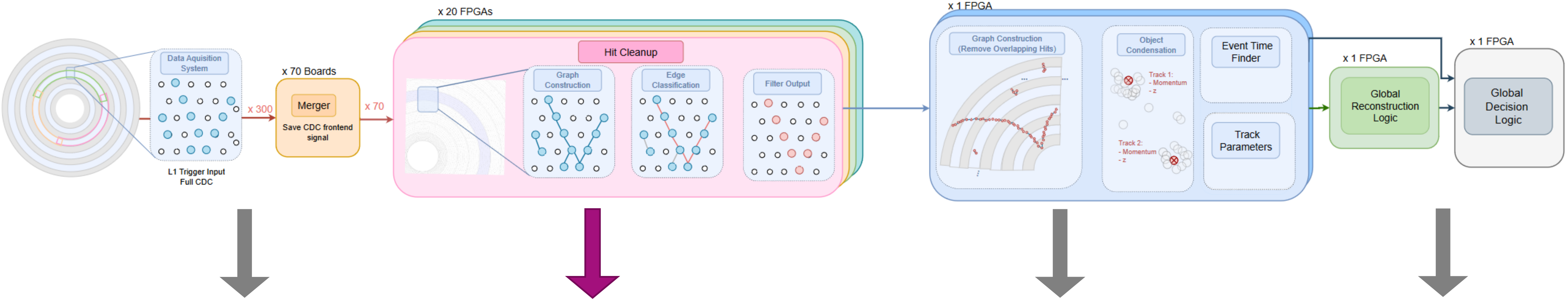


Figure from Lea Reuter

Belle II Central Drift Chamber : First-Level Trigger

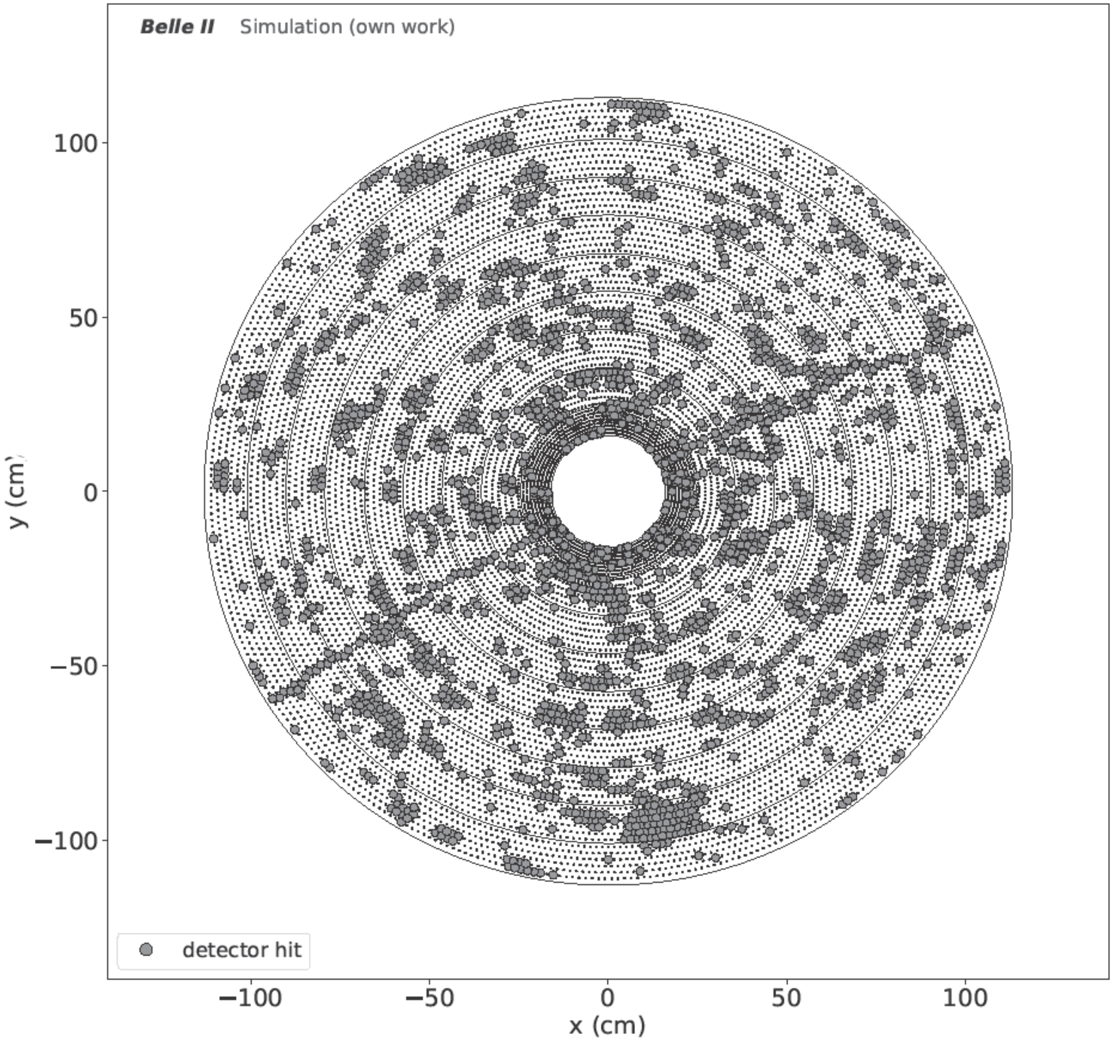
- 14336 sense wires at 32 MHz trigger input rate
- 5 us first-level trigger budget
- Apprxomately $1us$ for graph building **and** GNN inference
- Minimizing latency is crucial



$\approx 1\,us$ | $\approx 1\,us$ | $\approx 2\,us$ | $\approx 1\,us$

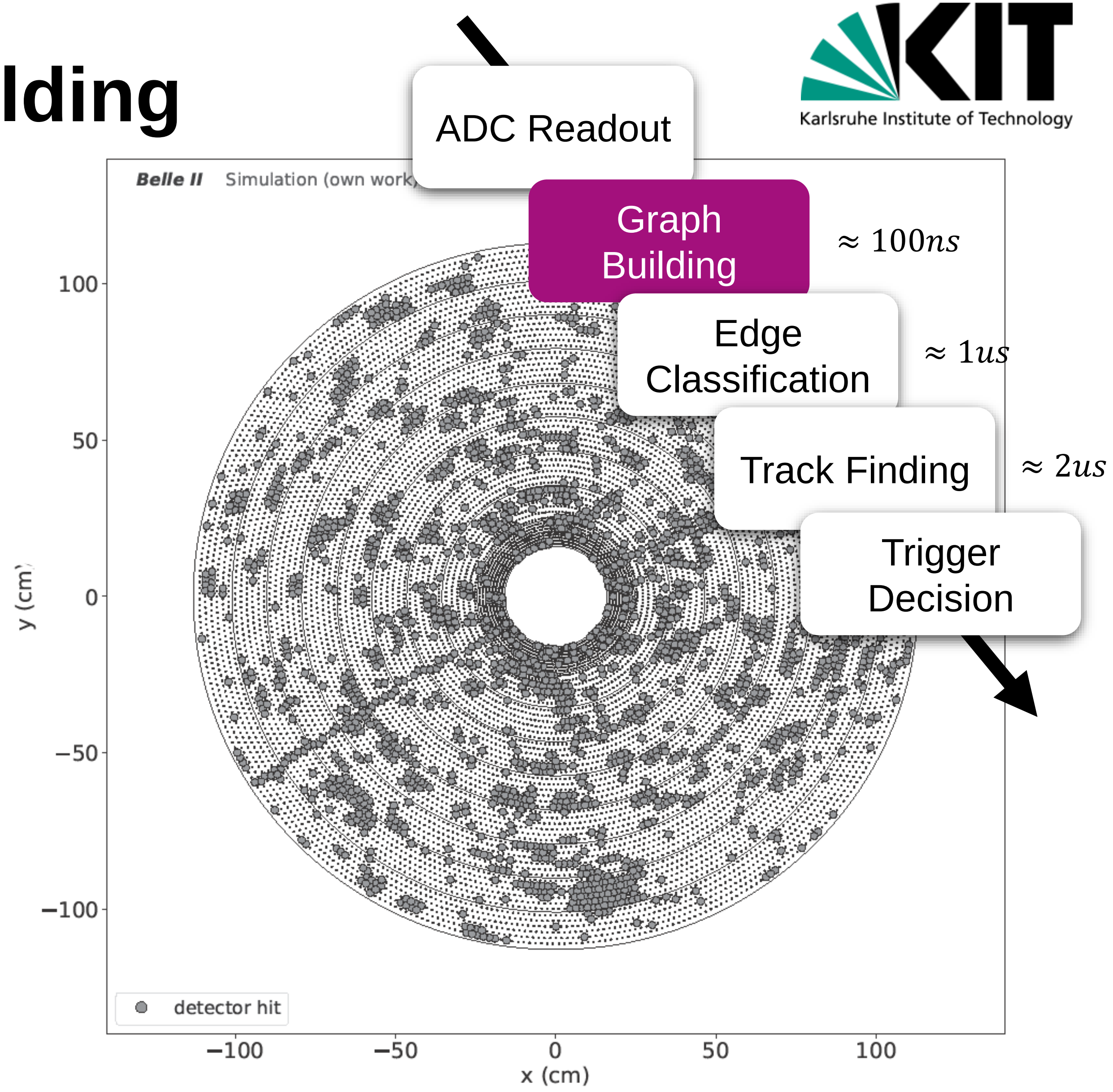
Requirements for Graph Building

- Hard real-time constraints
- Latency in the order of $O(100ns)$
- Up to 14336 inputs with undefined degree of sparsity
- Throughput of $32 \cdot 10^6$ events per seconds



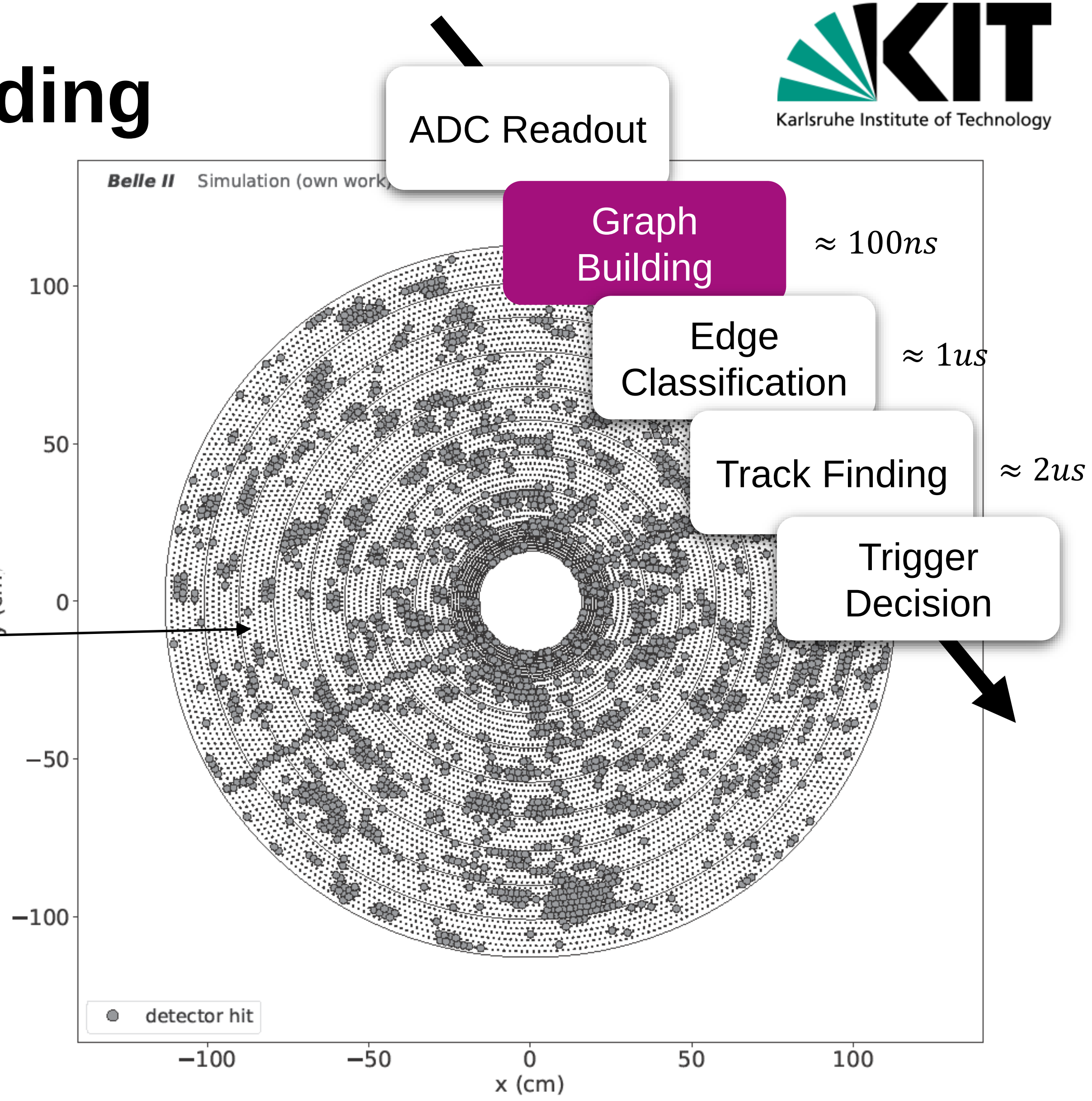
Requirements for Graph Building

- Hard real-time constraints
- Latency in the order of $O(100ns)$
- Up to 14336 inputs with undefined degree of sparsity
- Throughput of $32 \cdot 10^6$ events per seconds



Requirements for Graph Building

- Hard real-time constraints
- Latency in the order of $O(100ns)$
- Up to 14336 inputs with undefined degree of sparsity
- Throughput of $32 \cdot 10^6$ events per seconds



Objective

GNN edge classification has shown promising performance for background rejection [1] (See previous Talk Greta Heine)

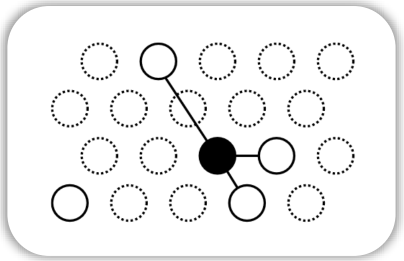
Hardware-efficient graph building remains an open challenge [2]

How to build graphs on FPGAs under latency constraints in high-throughput particle physics applications based on an algorithmic description

[1] DeZoort et al. Charged Particle Tracking via Edge-Classifying Interaction Networks. In Comput Softw Big Sci 5, 26 (2021).
[2] A. Elabd et. Al, Graph Neural Networks for Charged Particle Tracking on FPGAs. In Front. Big Data (2022).

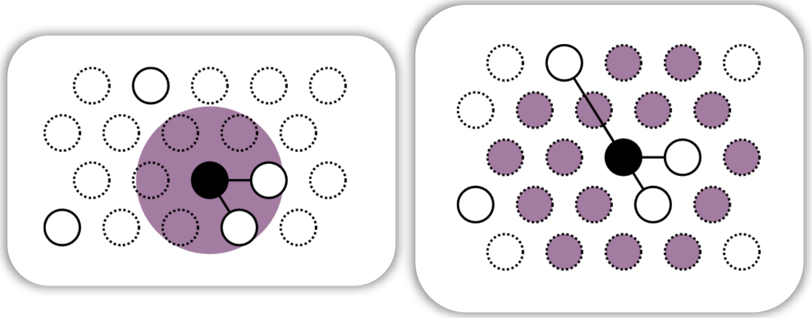
Graph Building

- State of the Art
  - K-Nearest Neighbour Graphs [1]
  - Approximate k-NN Graphs [2]



- Not suitable for implementation in $O(1us)$
- Intrinsic sequential algorithms
- Time complexity $O(k|V| \cdot \log(|V|))$

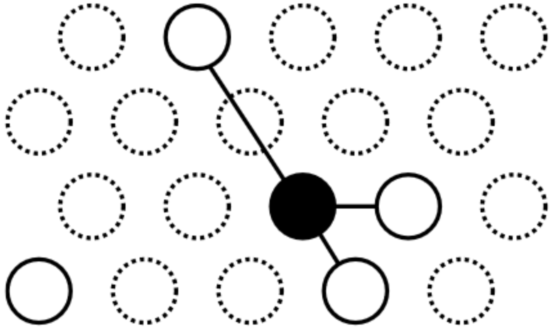
- Graph Building under Local Constraints
  - Use information at design-time to identify edge candidates
  - Intrinsic parallel algorithms
  - Similarity to pattern-based track reconstruction

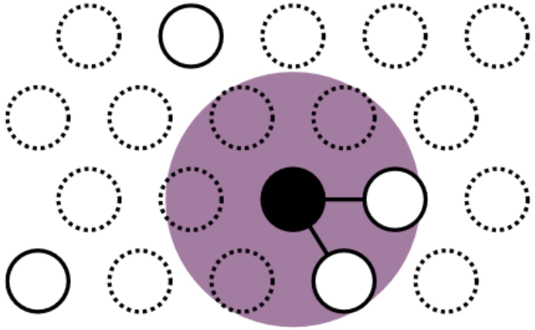


[1] Data Algorithms. O'Reilly Media, Inc. (2015).
[2] Zhang et. Efficient Large-Scale Approximate Nearest Neighbor Search on OpenCL FPGA. IEEE/CVF (2018).

k-NN Graphs and ε-NN Graphs

For uniformly distributed datasets, connecting an element x_i to a given number of nearest neighbours is essentially equivalent to connecting it to all such nodes $d(x_i, x_j) < \epsilon^*$ with some appropriate ϵ^* [1]





A parameter ϵ^* should exists,
such that the constructed ε-NN graph resembles a k-NN graph [1]

[1] Prokhorenkova, L., Shekhovtsov, A.: Graph-based nearest neighbor search: From practice to theory. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp.7803–7813. PMLR

Similiarity Metrics

- For each event, we define
  - E_k as the set edges for the k-NN graph
  - E_o as the set of edges for the locally constrained graph, e.g. ε-NN or p-NN

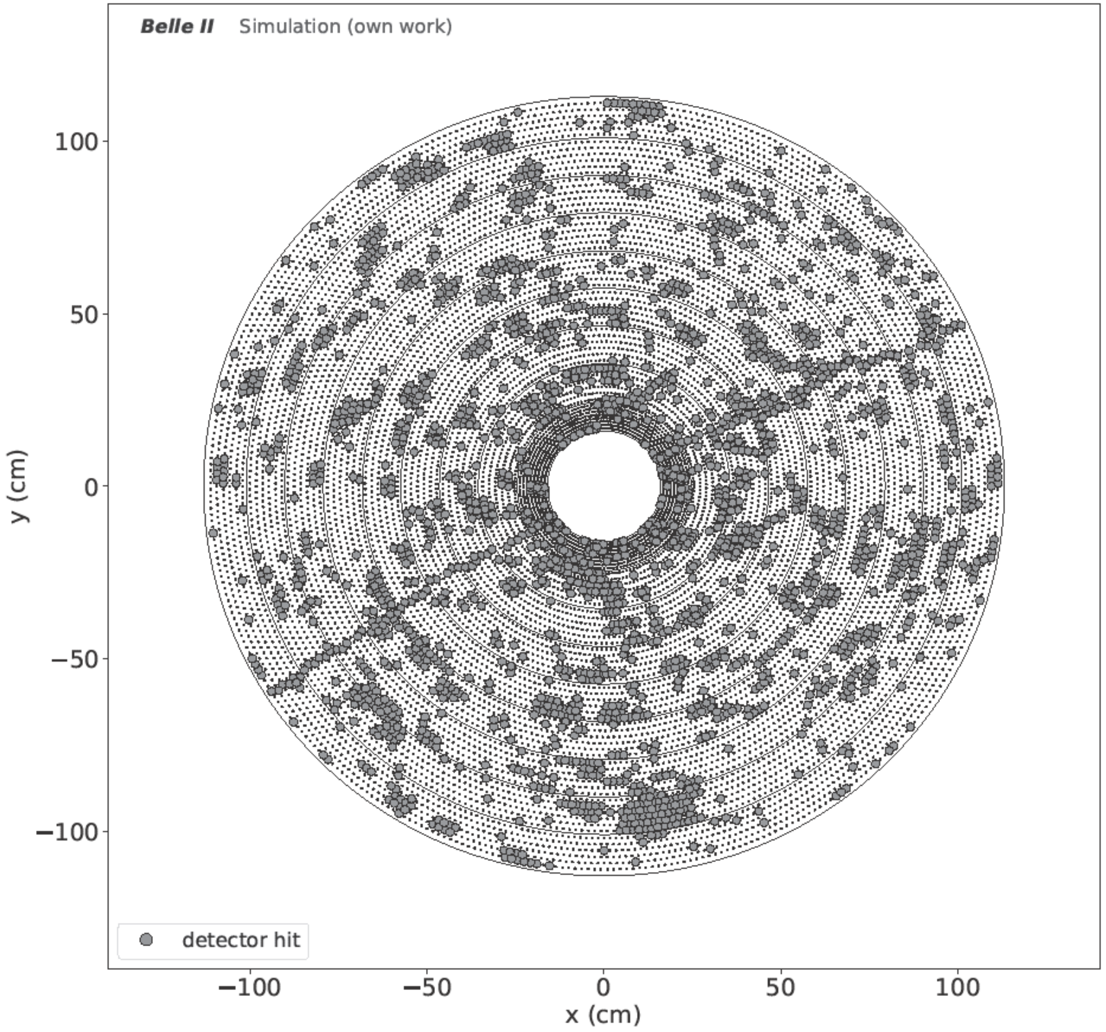
- Without considering difference in track or noise hits we define

$$Precision = \frac{|E_k \cap E_o|}{|E_o|} \text{ and } Recall = \frac{|E_k \cap E_o|}{|E_k|}$$

If $Precision \approx 1$ and $Recall \approx 1$, two graphs are considered similar

Case Study: Evaluation Setup

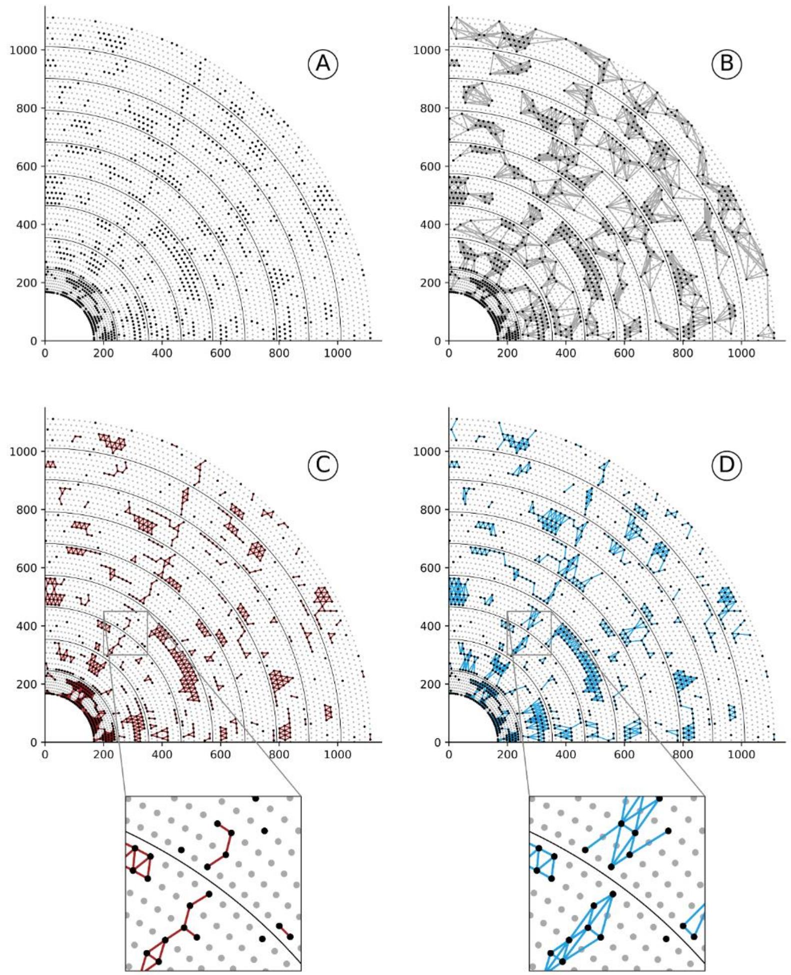
- Benchmark on simulated 2000 simulated events
- We consider
$$e^+e^- \rightarrow \mu^+\mu^-(\gamma)$$
- Simulated beam background corresponding to
$$\mathcal{L}_{beam} = 6.5 \cdot 10^{35} cm^{-2} s^{-1}$$
- Assuming highest background levels for runs in the next years
- Overall hit distribution is dominated by the beam background signal



Case Study:
Graph Building

Input Event as received by the first-level trigger

k-NN graph building for k = 6



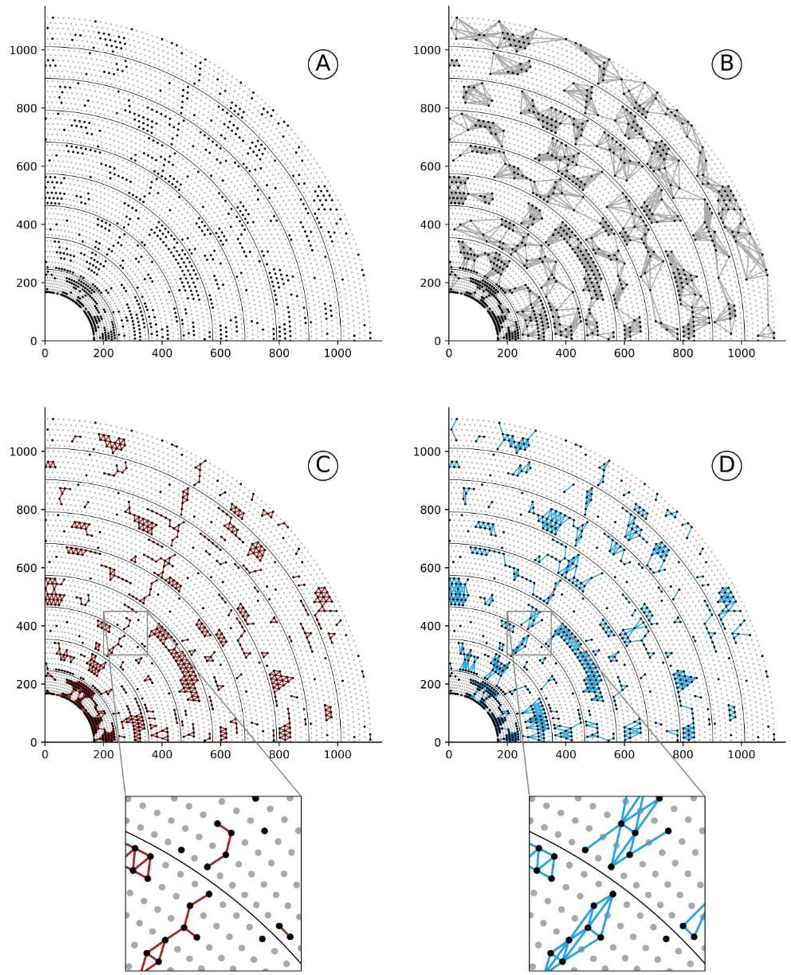
ε-NN graph building for ε = 22mm

p-NN graph building

Case Study:
Graph Building

Input Event as received
by the first-level trigger

k-NN graph building
for k = 6



ε-NN graph building for
ε = 22mm

p-NN graph building

Corresponds approximately
to the distances of adjacent wires
in the outermost layers

Case Study:
Similarity for k-NN Graphs and ε-NN Graphs

Hyperparameter Search $k \in [1{,}6]$ and $\epsilon \in [10mm, 28mm]$

Recall

Parameter e [mm] \ Parameter k	1	2	3	4	5	6
14	0.31	0.25	0.21	0.17	0.14	0.12
16	0.51	0.38	0.30	0.25	0.21	0.18
18	0.77	0.56	0.41	0.34	0.29	0.25
20	0.87	0.75	0.65	0.57	0.48	0.42
22	0.90	0.80	0.70	0.61	0.52	0.46
24	0.90	0.80	0.71	0.62	0.53	0.47
26	0.90	0.80	0.72	0.63	0.55	0.49
28	0.90	0.81	0.73	0.66	0.58	0.52

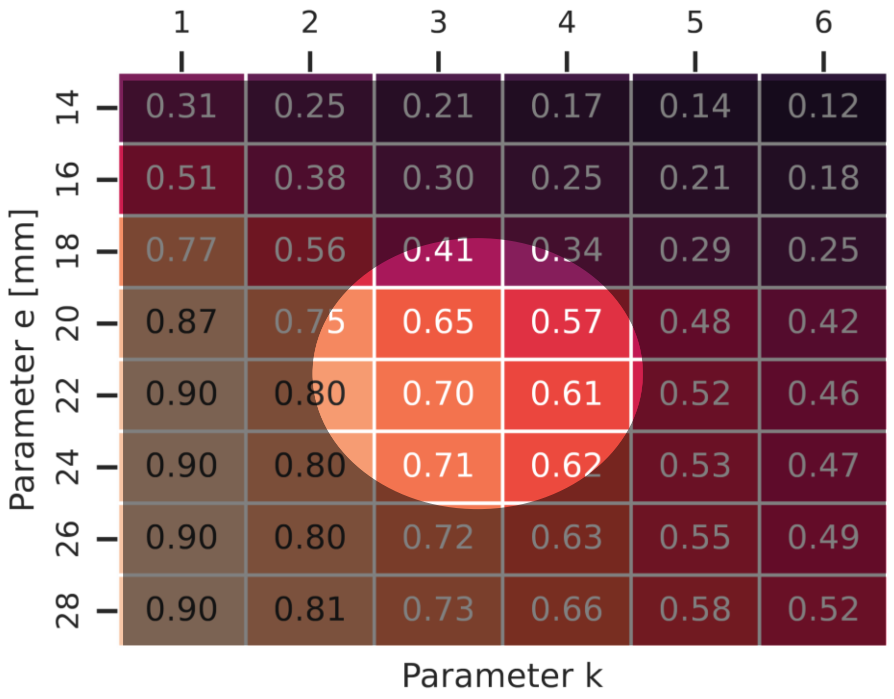
Precision

Parameter e [mm] \ Parameter k	1	2	3	4	5	6
14	0.48	0.69	0.86	0.97	0.99	0.99
16	0.48	0.66	0.77	0.86	0.90	0.94
18	0.52	0.68	0.77	0.84	0.88	0.92
20	0.35	0.55	0.73	0.85	0.89	0.92
22	0.31	0.50	0.66	0.78	0.82	0.85
24	0.29	0.47	0.64	0.75	0.79	0.83
26	0.26	0.43	0.57	0.68	0.73	0.78
28	0.24	0.39	0.54	0.64	0.70	0.75

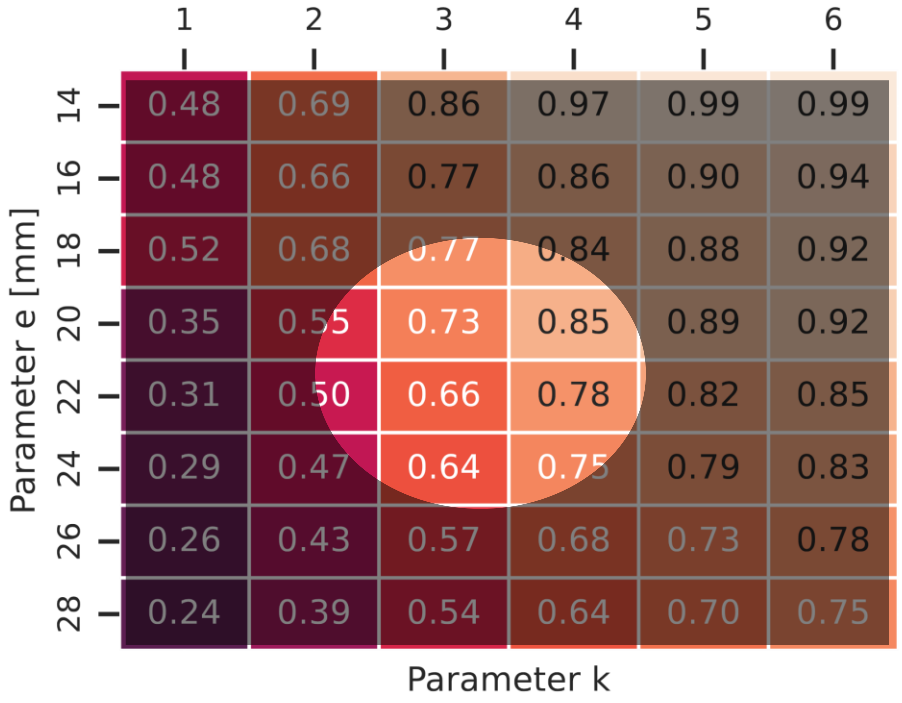
Case Study:
Similarity for k-NN Graphs and ε-NN Graphs

Hyperparameter Search $k \in [1,6]$ and $\epsilon \in [10mm, 28mm]$

Recall

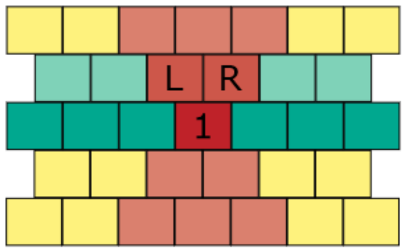


Precision

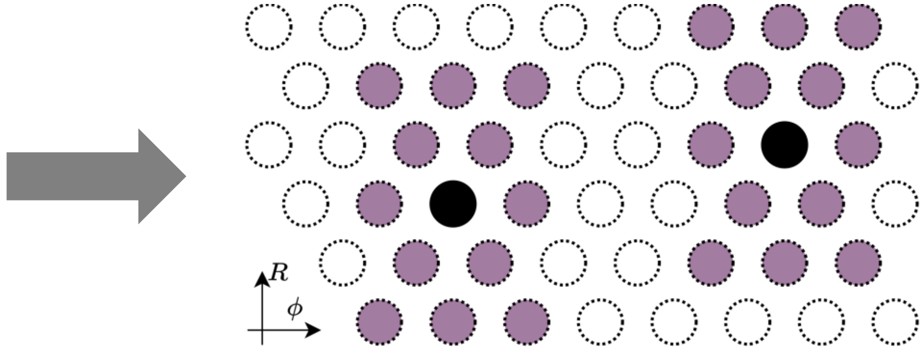


Case Study:
Similarity for k-NN Graphs and p-NN Graphs

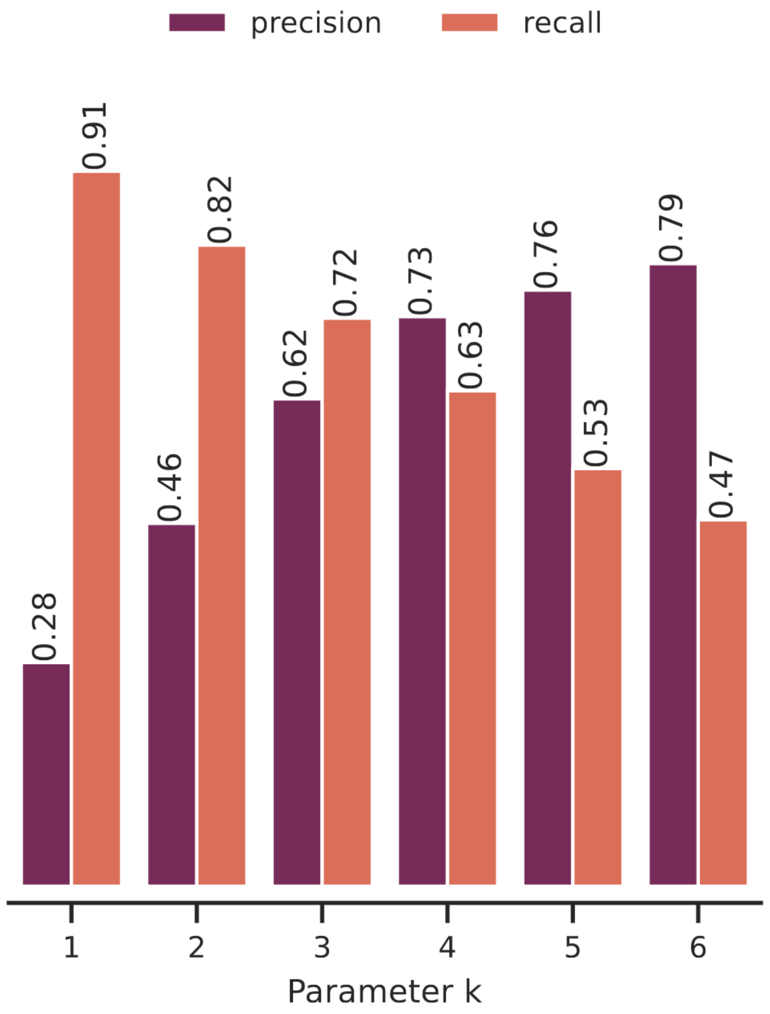
- Similiar pattern to the exisiting Track Segment Finder
- Slight differences to the pattern presented by Greta
- Hyperparameters $k \in [1,6]$



Current Patterns



Proposed Patterns



Thoughts to Go

Karlsruhe Institute of Technology

- Comparing k-NN and ε-NN or p-NN graph building,
  we find a **limited** similarity.

- Our analysis reveals that even high **beam background can not be considered uniform** inside the CDC.

- We follow, that graph building approaches must be evaluated with GNNs for hit cleanup in an **algroithm co-design approach.**

- Thus, we require an **semi-automated methodology** to evaluate graph building approachs that map well onto FPGAs.

Methodology

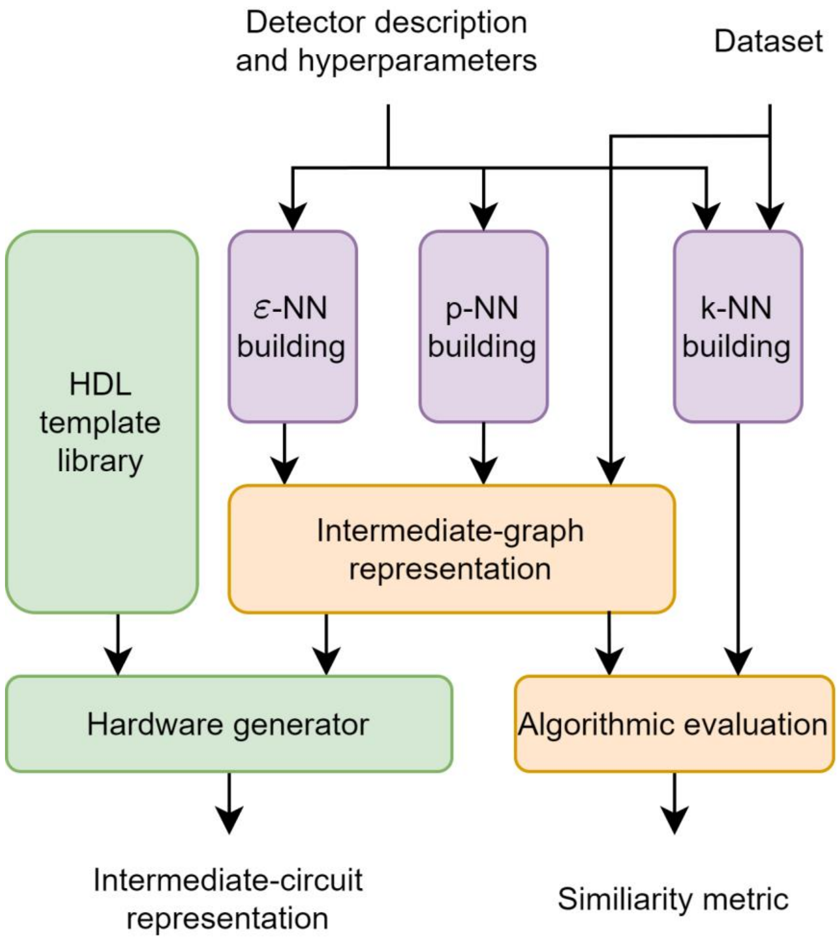
1. We describe our detector in a database

2. We define our graph building approach

3. We perform an offline similiarity evaluation

4. We automatically generate a intermediate representation

5. We map our intermediate representation to predefined HDL templates in our library

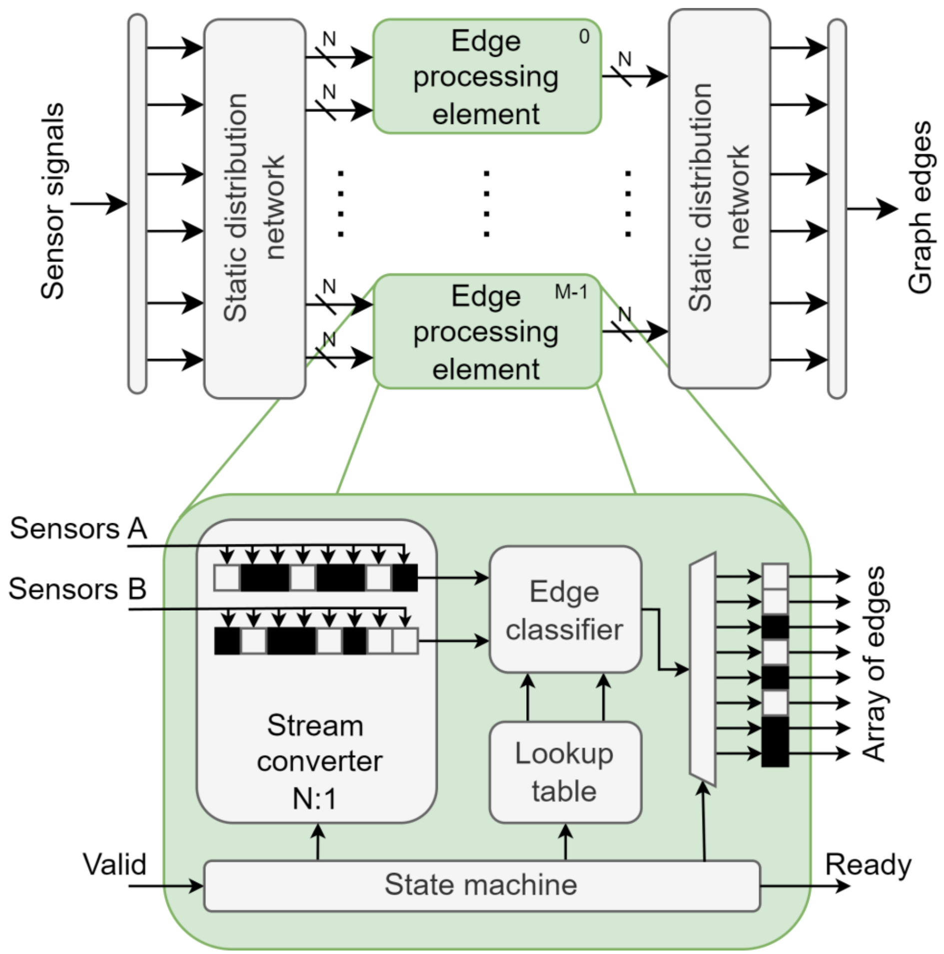


Hardware Architecture

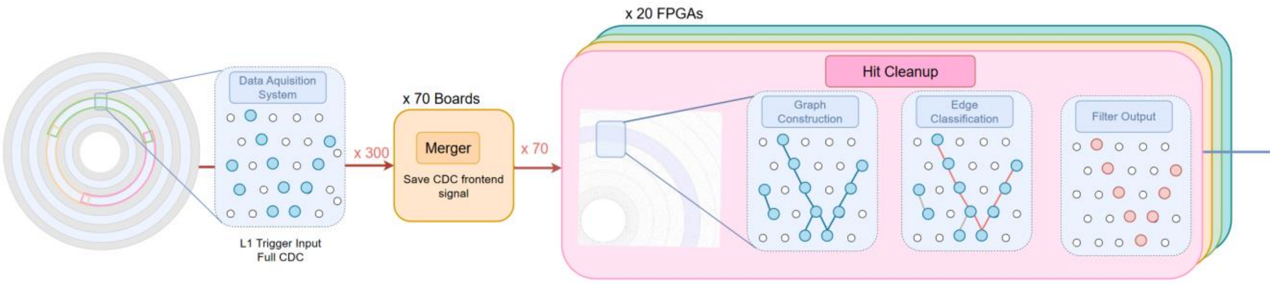
- HDL templates written in Chisel3

- Sensor signals from CDC frontend
- Static sensor features from lookup table
- Generates graph edges for successive GNN Inference

- Throughput $T = \frac{f_{sys}}{f_{data}} \cdot \frac{1}{N}$



Case Study:
First-level Trigger System Design

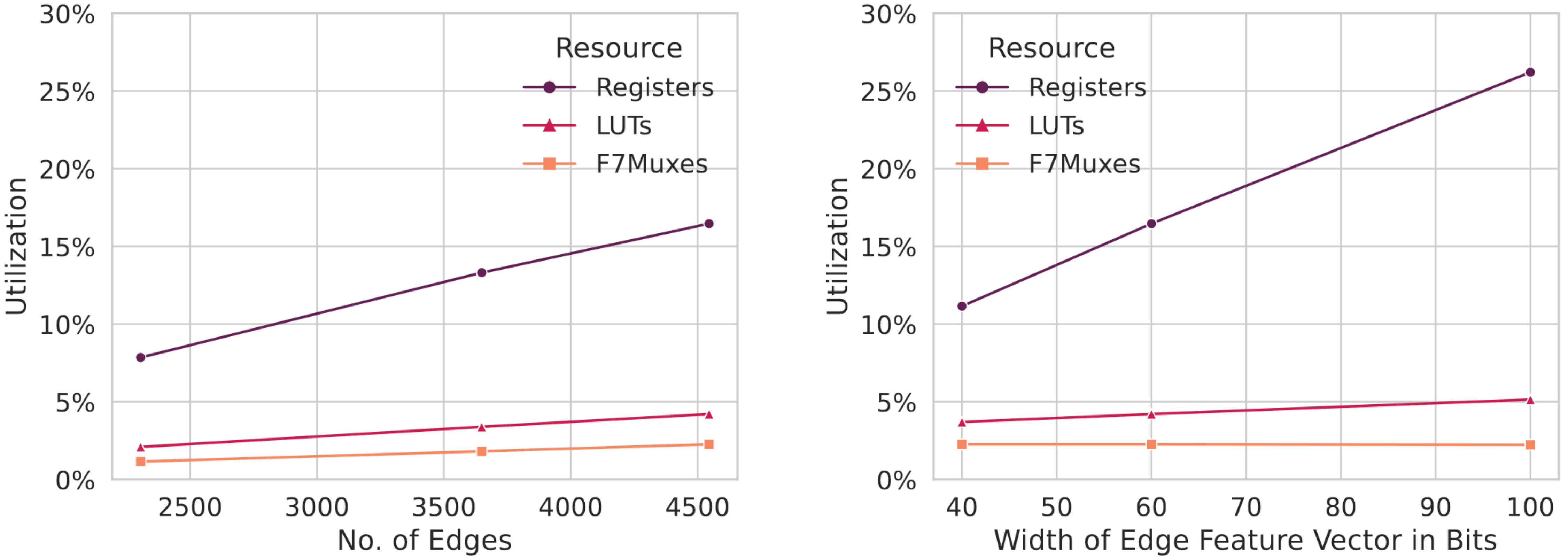


Superlayer	FPGAs	Sensors	Edges
0	2	664	3293
1	2	498	2305
2	2	588	2725
3	2	691	3201
4	2	786	3649
5	2	882	4097
6	2	978	4545
7	3	730	3372
8	3	786	3649

XCVU160	LUTs	Registers	BRAM	DSPs	CARRY
Ressources	926k	1.85M	3726	1560	11k

For 20 overlapping sectors we receive $2305 < |E| < 4545$ and $498 < |V| < 978$.

Case Study: Implementation on the Xilinx Ultrascale XCVU160



Total latency $\mathrm{L} = 39.06ns$ for $f_{clk} = 256\ MHz$, corresponding to 10 clock cycles

Conclusion

- We have investigated k-NN, ε-NN and p-NN graph building in the first-level trigger at Belle II
- We have proposed an methodology to automatically generate graph building hardware modules on FPGAs
- We have implemented a proof-of-concept of our graph building approach on the Universal Trigger Board 4

Questions?

Karlsruhe Institute of Technology

For more information, check out our publication in the Computing and Software for Big Science Journal and our Source Code on Github.

Publication

https://link.springer.com/article/10.1007/s41781-024-00117-0

Source Code

https://github.com/realtime-tracking/graphbuilding