Karlsruhe Institute of Technology

Institut fuer Technik der Informationsverarbeitung

Belle 2 General Meeting
02.06.2023

Track segments for the Displaced Vertex trigger and Graph Building for GNN tracking on FPGAs

Marc Neu – neu@kit.edu

AI Trigger Group at Belle II

KIT ITIV
- Marc Neu
- Kai Unger
- Jürgen Becker

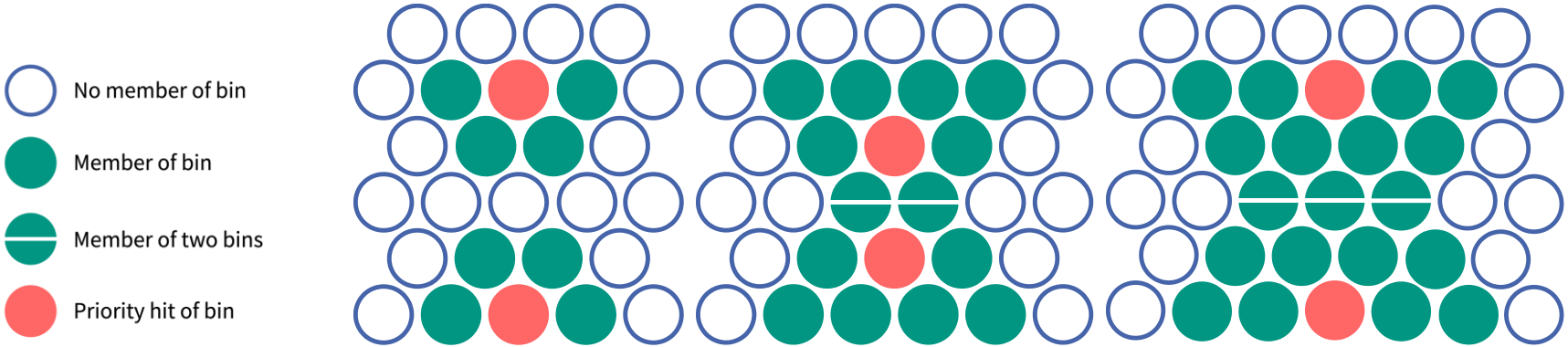
KIT ETP
- Lea Reuter
- Greta Heine
- Stefkova Slavomira
- Torben Ferber

MPI & TUM
- Felix Meggendorfer
- Elia Schmidt
- Christian Kiesling
- Alois Knoll

Agenda

- Displaced Vertex Track Segment Finder – Implementation for the upcoming Trigger Upgrade
  - Preprocessing for the upcoming 3D Hough Upgrade
  - See also Talk Kai Unger
- Graph Building – A case study on the Belle II Trigger System
  - Towards online GNN tracking on FPGAs
  - See also Talk Lea Reuter

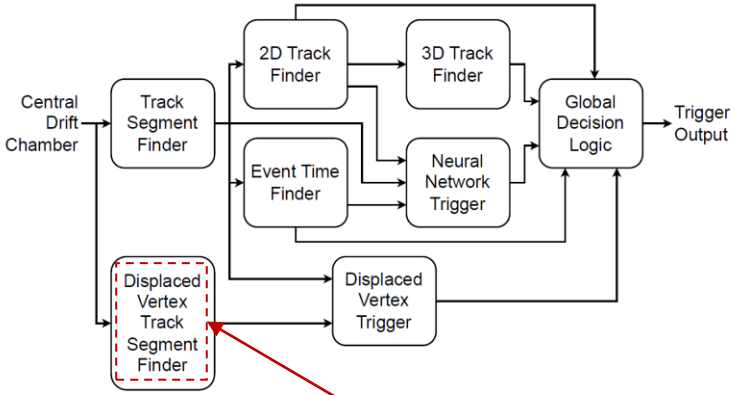
Lookup table based TSF classifiers



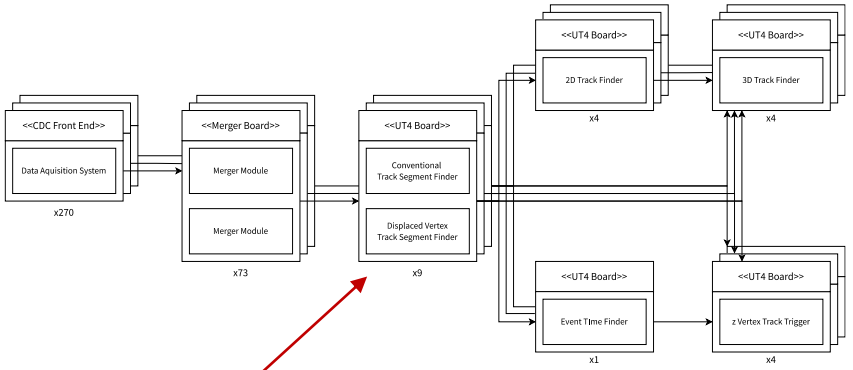
- Split up hourglass into two parts
- Perform pattern matching for each hitmap

Including the Displaced Vertex Trigger

System Level View



Deployment Level View



Proposed Extension

Implementation

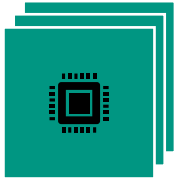
- Implementation on XCVU160, UT4 Board
- Configuration for Superlayer 9

	Available		TSF LUT 5		TSF LUT 9		TSF LUT 12	
CLB LUTs as Logic	926400	100%	88096	9,51%	87039	9,40%	90943	9,82%
CLB LUTs as Memory	219840	100%	748	0,34%	12524	5,70%	780	0,34%
CLB Registers	1852800	100%	39124	2,11%	42814	2,31%	41442	2,24%
CARRY8	125760	100%	4983	3,96%	4983	3,96%	5336	4,24%
Block RAM	3276	100%	0	0%	0	0%	384	11,7%

Summary

b2tsf

- ✓ Model
- ✓ Data Structure
- ✓ Visualization
- ✓ **Simulation**



LUT-5
LUT-9
LUT-12

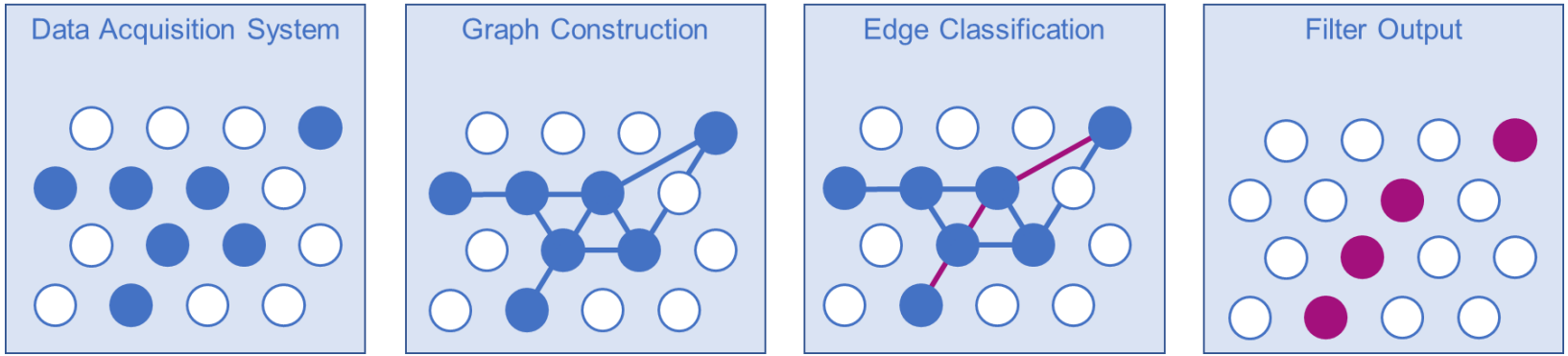
- ✓ Latency Optimized Implementation Stream Compaction
- ✓ Functional Tests
- ✓ Timing & Utilization
- ✓ Synthesis
- ✓ **Ready for Deployment**

belle2-dv-tsf-sw
belle2-dv-tsf-hw

- ✓ Automated Pattern Evaluation
- ✓ Design Space Exploration
- ✓ HW/SW Codesign
- ✓ **Full Development Pipeline from Dataset to Firmware**

Motivation

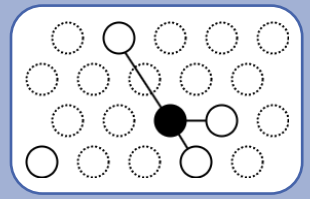
- GNN edge classification has shown promising performance for background rejection[1]
- Hardware efficient graph construction remains an open challenge [2]
- Full GNN-based track finding and fitting solution



[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).

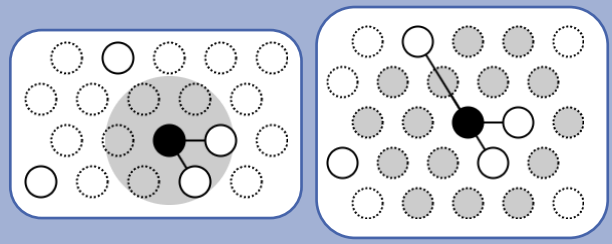
Graph Building

- State of the Art
  - K-Nearest Neighbour Graph [3]
  - Approximate k-NN Graphs [4]



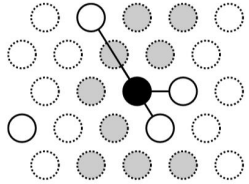
- Not suitable for FPGA implementation
- NP complex
- Intrinsic sequential algorithms

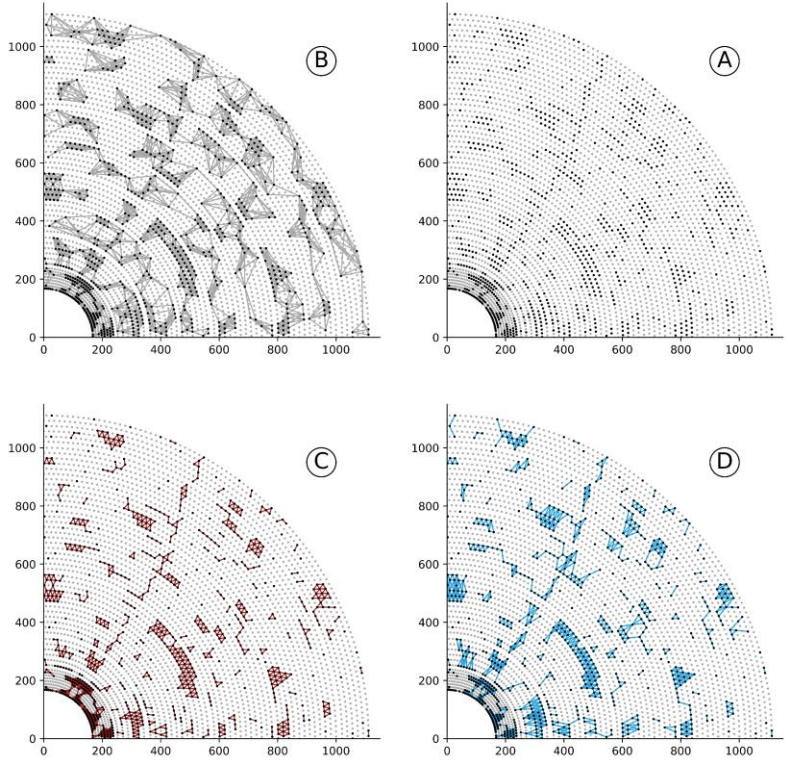
- Graph Building under Local Constraints (See Talk Lea Reuter)
  - Construction of local optimal graphs
  - Use information at design-time to identify edge candidates
  - Segmentation of CDC



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

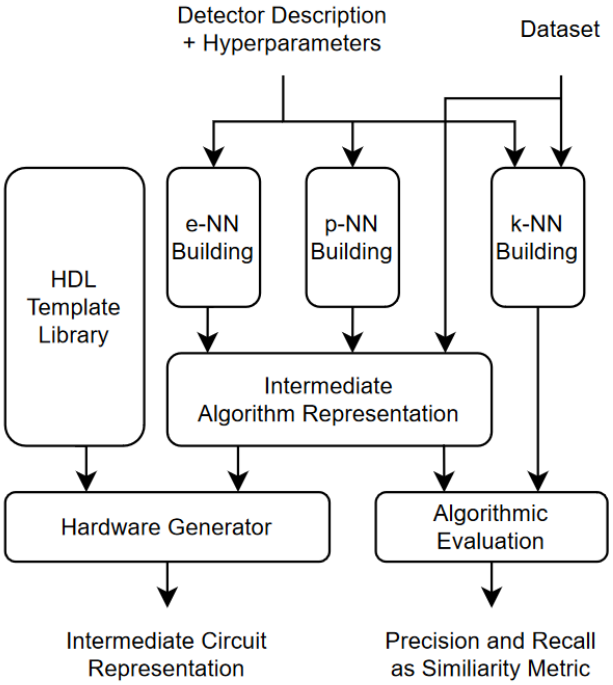
Three Ways to Building Graphs

Sector	Type	Parameter
A	Raw Trigger Input	Nominal Background
B	k-Nearest Neighbour Graph	$k = 6$
C	e-Nearest Neighbour Graph	$\epsilon = 22mm$
D	p-Nearest Neighbour Graph	



Dataset: displaced_processed_simulated_2_tracks_0_nominal-phase3

Methodology

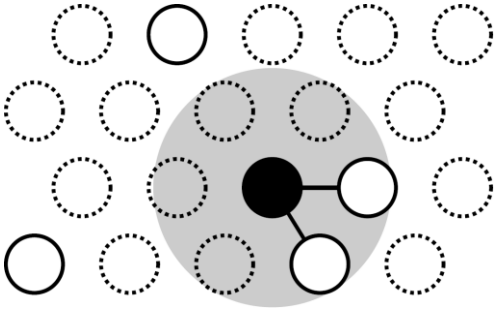


1) Generate Algroithm Description according to Specification
2) Perform Similarity Evaluation
   a. Build ε-NN and p-NN graphs
   b. Build k-NN graphs
3) Generate Hardware
4) Synthesize SystemVerilog using Xilinx Vivado 2022.1

k-NN Graphs and ε-NN Graphs

- For uniformly distributed datasets, connecting an element x_i to a given number of nearest neighbors is essentially equivalent to connecting it to all such nodes $d(x_i, x_j) < \epsilon^*$ with some appropriate ϵ^* [5]

- Assumptions
  - Uniformly distributed dataset may be fullfilled in nominal background conditions
  - All nodes belong to a unit euclidean sphere



Based on Prokhorenkova et. al , a parameter ρ_l should exists, such that the constructed ε-NN graph resembles a k-NN graph

[5] 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

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

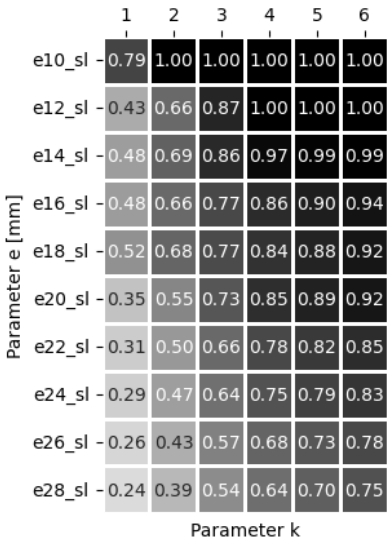
If $Precision \approx 1$ and $Recall \approx 1$, the two graphs can be considered similary

Similiarity ε-NN and k-NN

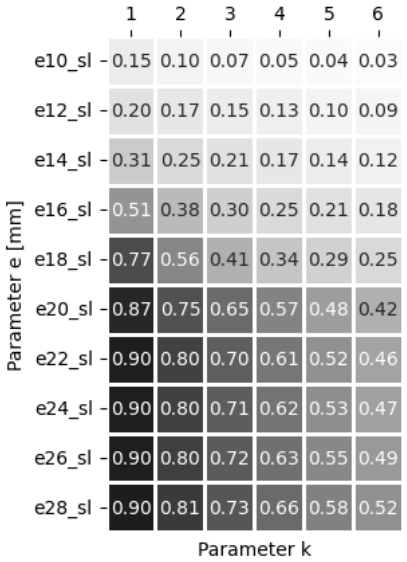
- Evaluation on 2000 events for Nominal Background
- Mean of Precision and Recall Score is reported
- Hyperparameter Search
  - $k \in [1,6]$
  - $\epsilon \in [10mm, 28mm]$

k-NN and ε-NN graphs differ in nominal background conditions,

$$Precision = \frac{|E_k \cap E_o|}{|E_o|}$$

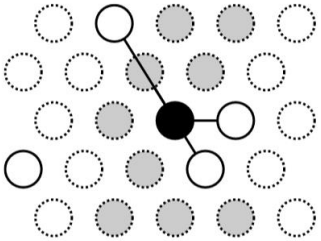


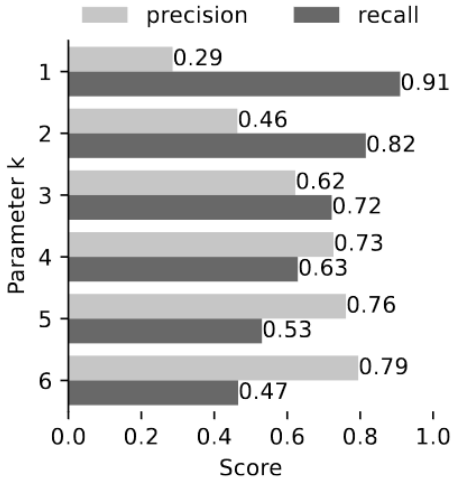
$$Recall = \frac{|E_k \cap E_o|}{|E_k|}$$



Similiarty p-NN vs k-NN

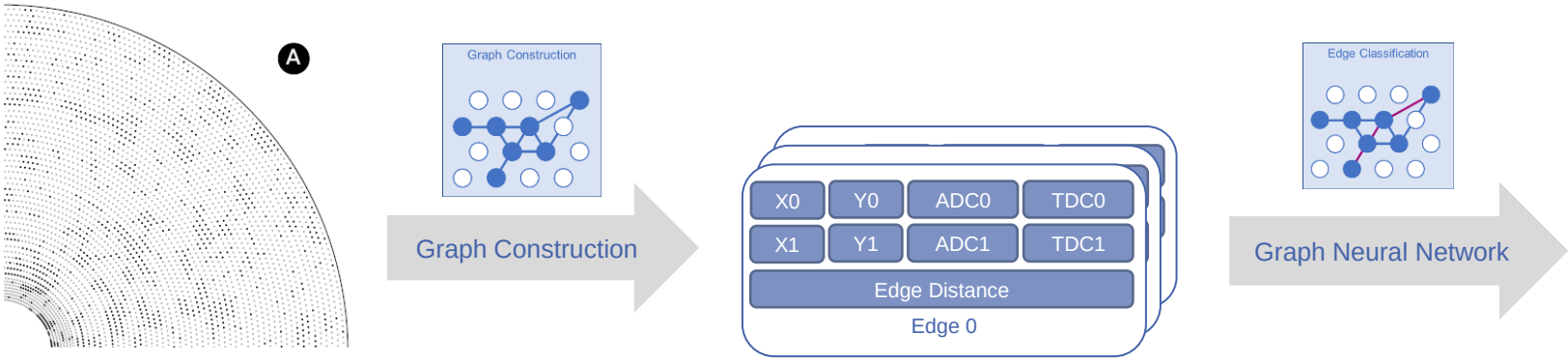
- Evaluation on 2000 events for Nominal Background
- Mean of Precision and Recall Score is reported
- We propose a pattern similar to the LUT-based Track Segment Finder





k-NN and p-NN graphs differ in nominal background conditions,

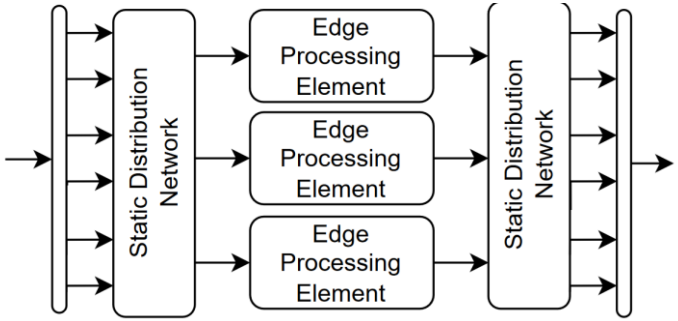
Challenges in Low Latency Graph Construction

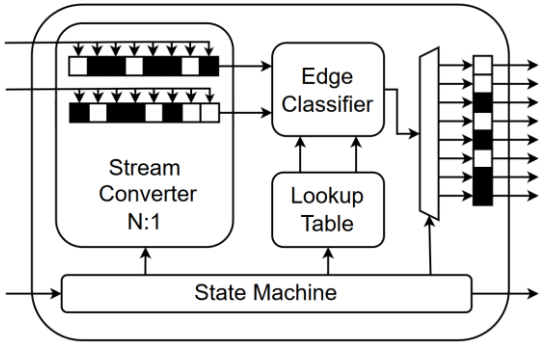


- Combined Latency for Graph Construction and GNN Inference $\approx 1.5\ \mu s$
- Process one sector of the full event on every FPGA
- First stage filtering of relevant edges
- Static information for each edge must be retrieved from Registers
- Provide an Interface to GNN Frameworks

Hardware Implementation

- Time Complexity $O(1)$
- Space Complexity $O(|E|)$
- Variable Number of Output Queues (depends on successive GNN)





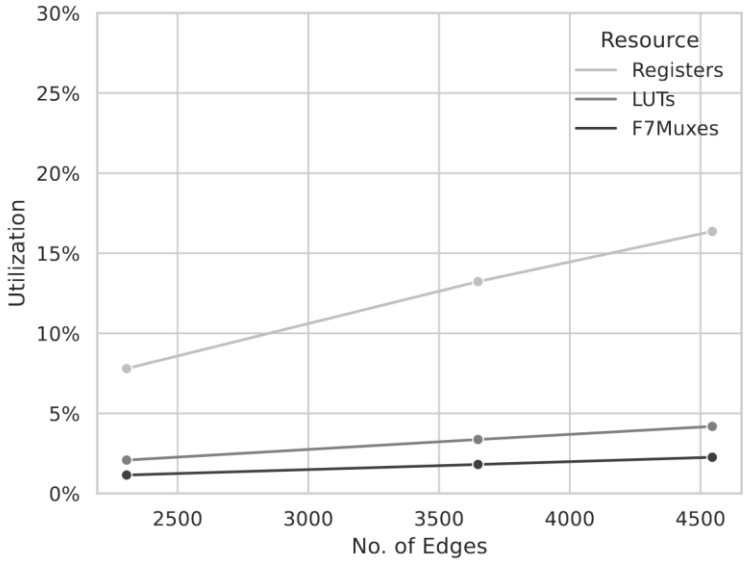
Idea: Hardware Implementation

Examined CDC Sectors
(See Talk Lea Reuter)

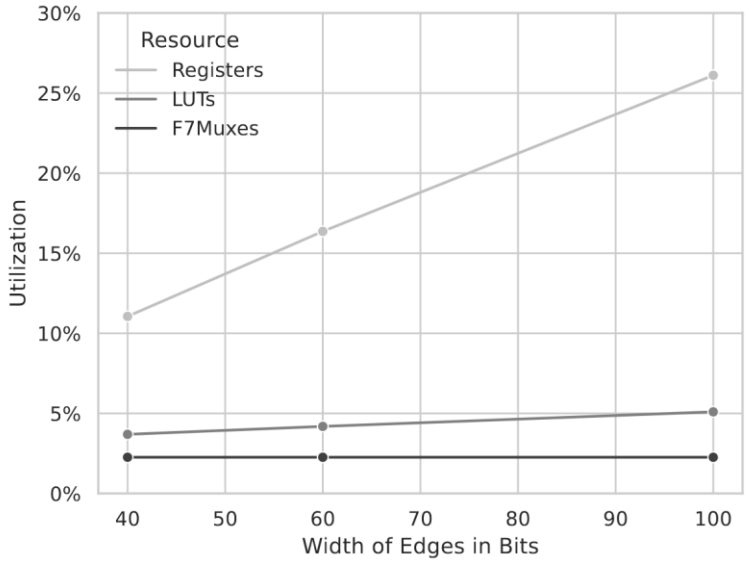
Superlayer	FPGAs	Vertices	Edges
0	2	664	3293
1	2	498	2305
2	2	588	2725
3	2	691	3201
4	2	786	3649
5	2	882	4097
6	2	978	4545
7	3	730	3372
8	3	786	3649

- CDC partitioned into 20 sectors
- 2-3 FPGAs per superlayer
- Lower and upper bound of problem size determined
  $$2305 < |E| < 4545$$
- System Frequency
  $$f_{sys} = 254\ MHz$$
- Event processing rate
  $$f_{event} = 31.75\ MHz$$

Hardware Implementation



(a) Utilization for a variable graph size $|E|$. The queue length parameter is set to eight, each edge is composed of 60 bits.



(b) Utilization for a variable edge width. The queue length parameter is set to eight, the input graph is composed of 4545 edges.

Conclusion

- ✓ We implemented a Database to Firmware Toolchain for Online Graph Building
- ✓ We performed Functional Tests
- ✓ We synthesized our module using out-of-context
- ✓ We explored Timing & Utilization on the Universal Trigger Board 4
- ✓ **Our System is ready for further Tests & Deployment**

Future Work

- We plan to publish our work on Graph Building, paper draft is in preparation
- We plan to implement a hardware accelerated GNN tracking solution (See Talk Lea Reuter)
- We consider co-optimization of Graph Building and GNN Inference