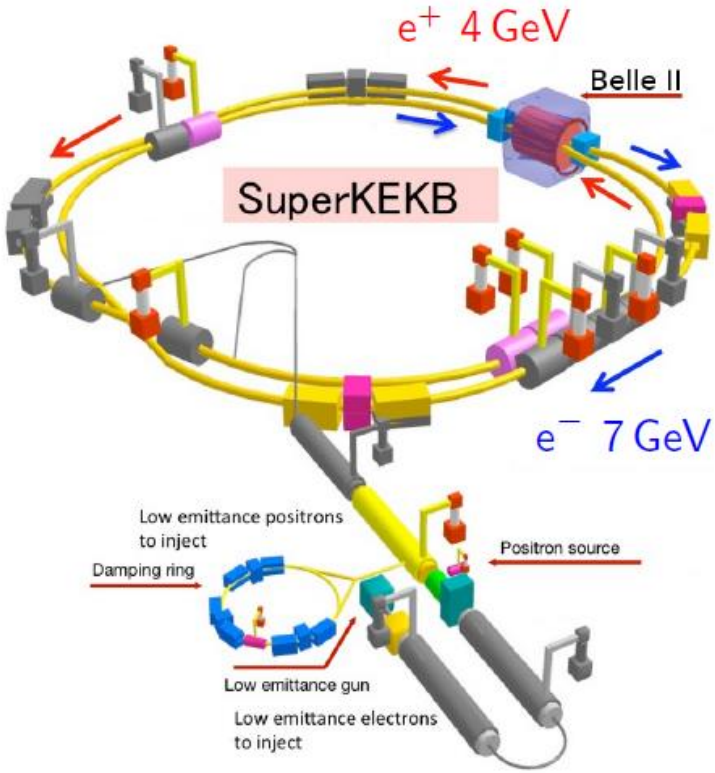
Karlsruhe Institute of Technology

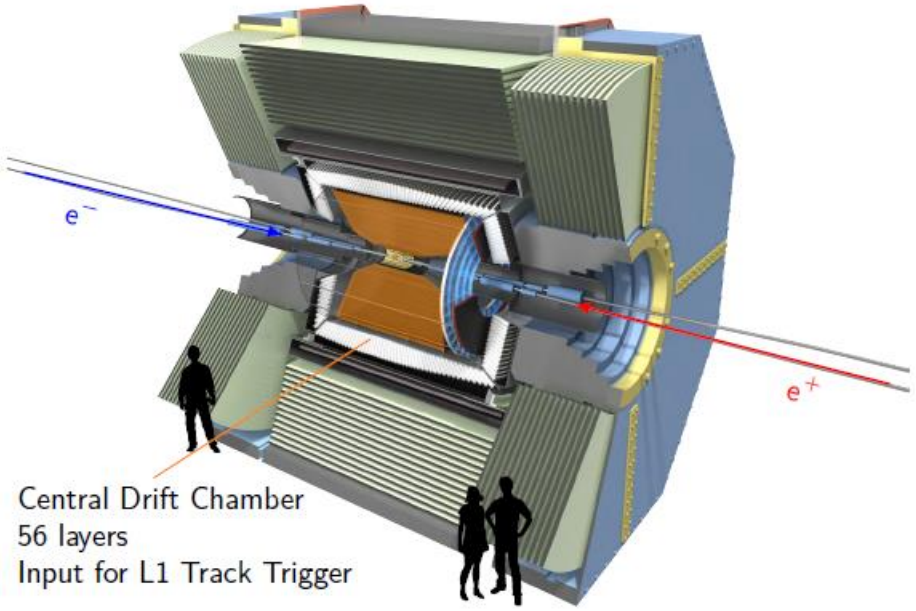
Institute for Information Processing Technologies

Towards Redesigning Track Segment Identification at the Belle II Trigger System

Kai Unger, S. Bähr, J. Becker, F. Meggendorfer, C. Kiesling, S. Skambraks, T. Voßhenrich

Belle II



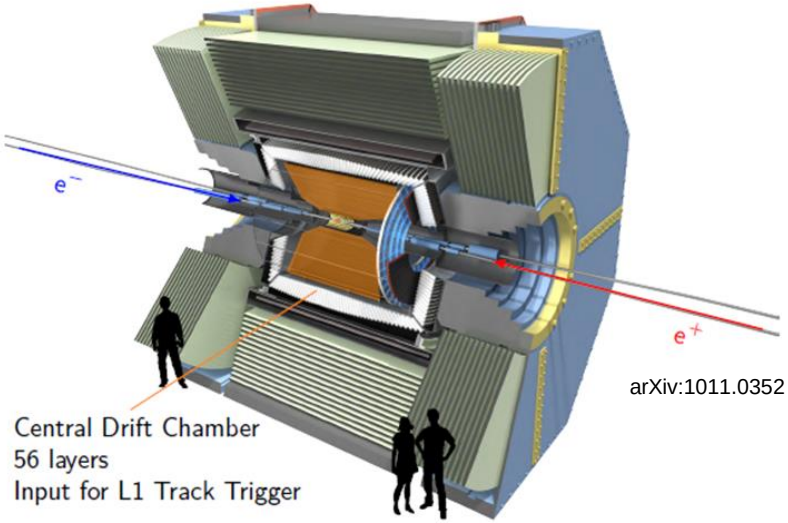


arXiv:1011.0352

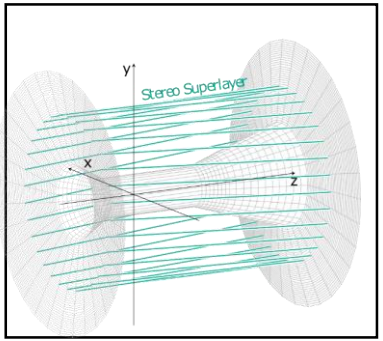
- Located at KEK, Tsukuba, Japan
- Asymmetric e^+e^- collider: “B-factory”
- June 2021: luminosity world record with 3.1×10^{34} $cm^{-2}s^{-1}$

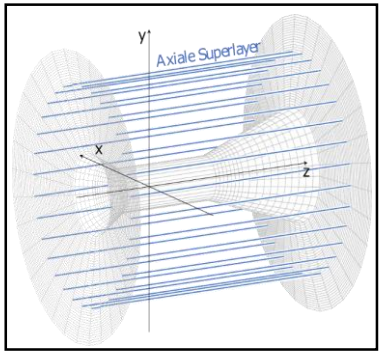
Belle II Central Drift Camber (CDC)

- Central Drift Camber track reconstruction
  - 14336 Drift Cells
    - Sense wires 14 336
    - Field wires 42 240
  - 56 layers
- 9 Superlayers (SL) trigger
  - 4 Stereo Superlayers
  - 5 Axial Superlayers
  - 2336 Track Segments (TS)

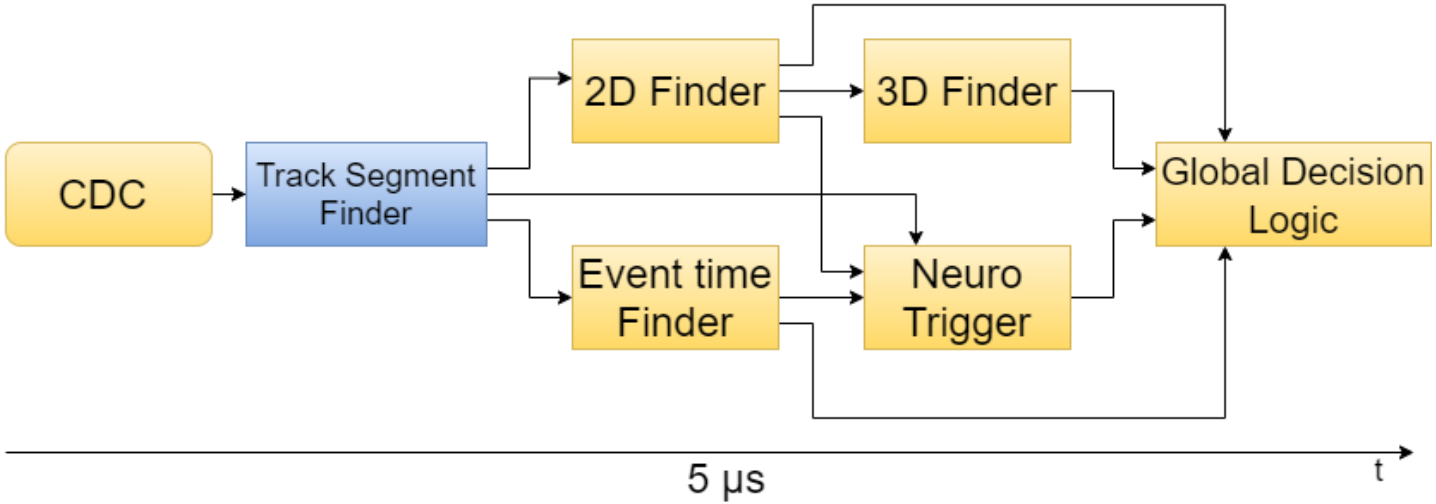


arXiv:1011.0352





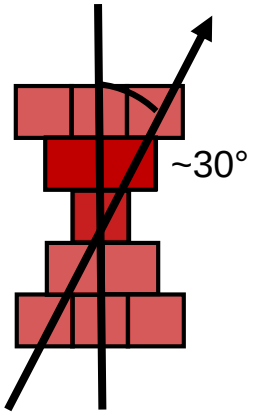
CDC Trigger



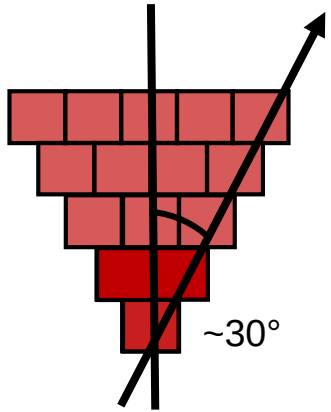
- Main System for online track reconstruction
- The Track Trigger System works with reduced data
- FPGA based Trigger
- 5 µs for the L1 Trigger System

Current Track Segment Finder TSF

- 2336 Track Segments in 9 Superlayers
  - 14336 Trigger Cells
  - Reduce the data
- Limits the tracks to tracks from the origin of the collision
  - (about 30° from the collision origin)
  - Reduce noise
- Parallel architecture implemented in 9 FPGAs
- Active since 2019



Super Layer 1-8

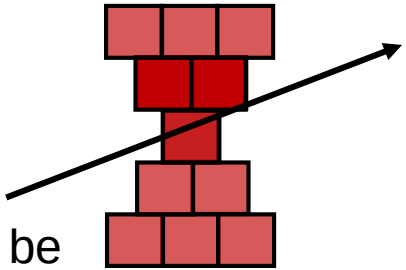


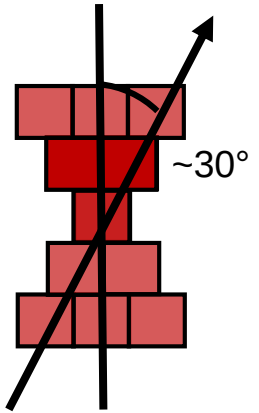
Super Layer 0

Super Layer	SL0	SL1	SL2	SL3	SL4	SL5	SL6	SL7	SL8
Track Segments	160	160	192	224	256	288	320	352	384

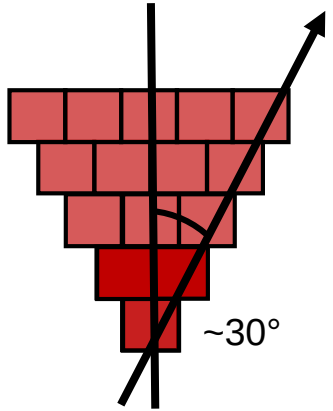
Problem: Low energy or displaced tracks

- Some tracks have a higher angle
  - Not enough hits for a valid TS
- Loss of TS means loss of information
  - Every information lost in the TSF cannot be reconstructed in the later stage
- Low energy track have a large angle due to the stronger distraction by the magnets
  - Lost in the current system
- Displaced tracks are often not pointing to the collision origin
  - Lost in the current system





Super Layer 1-8



Super Layer 0

- **An origin angle independent solution is needed**

Two approaches

Lookup Table TSF

- Use pattern stored in Lookup Tables to determ valid tracks
- Similar to the current approach
- Implementation in FPGA possible

Neural Network TSF

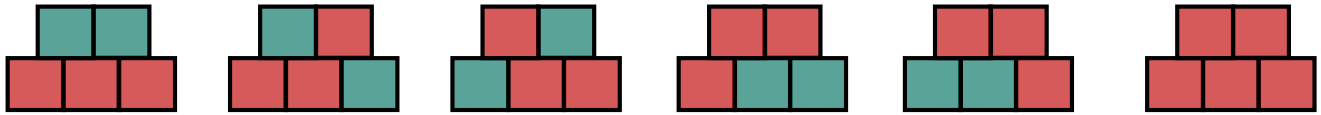
- Use neural network to determ valid tracks
- New approach
- Prototype Implementation needed

Lookup Table TSF

- Precalculated Pattern are stored in BRams
- Every wire in the Segment is a BRam address
- BRam result is valid track or no valid track (one bit output from 8 or 16 bit per BRam line)
- Address are limiting the Segment
- Different sizes are possible:

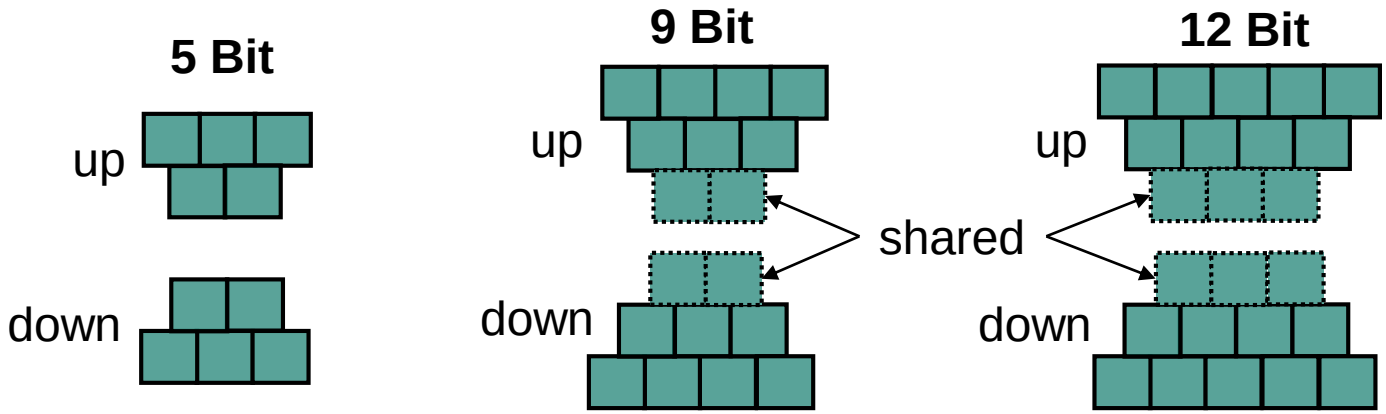
Bord*	Total BRAM	Max Pattern (SL8)
VX80	50 MB	17
VX160	115.2 MB	18

*FPGA boards are provided by the experiment



5bit Lut TSF

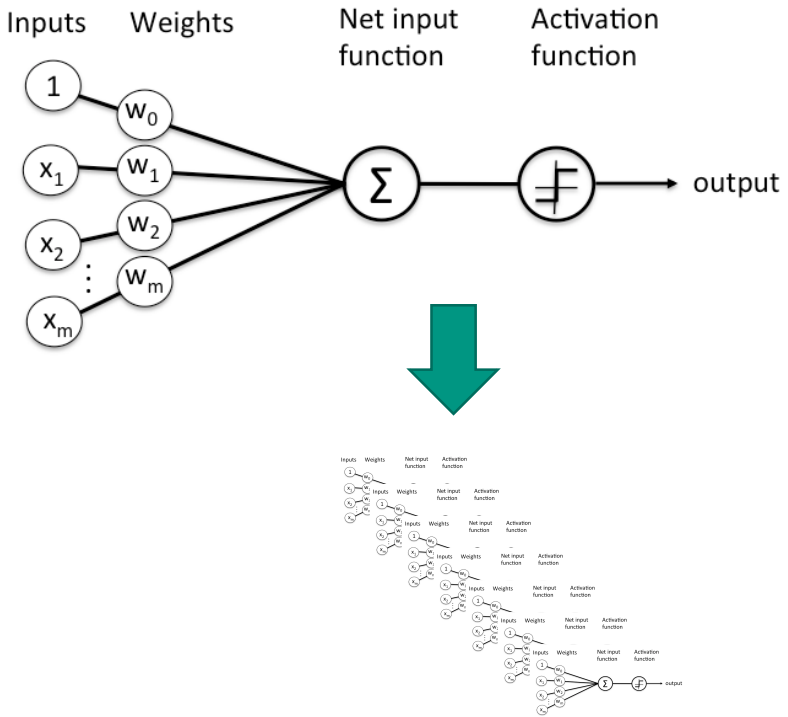
Lookup Table TSF shapes



- A 5x5 shape would be ideal, but is not possible due to the limited resources.
- 3 different shapes are shown
  - 5 bit lut is inefficient
  - Bigger lut give a better signal to noise ratio

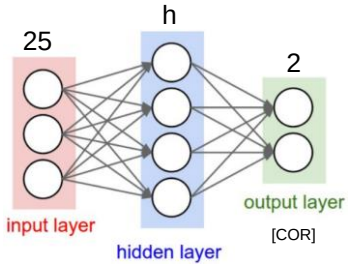
Neural Network TSF

- Predict tracks with Neural Networks
- Use many Neural Networks in parallel
- Make Neural Network so small as possible
- Networks should have a fixed bit length
- Due to the experiment data structure 5 bit high patterns are only possible in the wide all data are available
  - First tests with 5x5 inputs
  - 5x10 or 5x16 are also possible



Resources

- 3 binary Prototypes build and synthesized
  - Hard coded in VHDL
- Used smallest Experiment used FPGA
  - No optimization



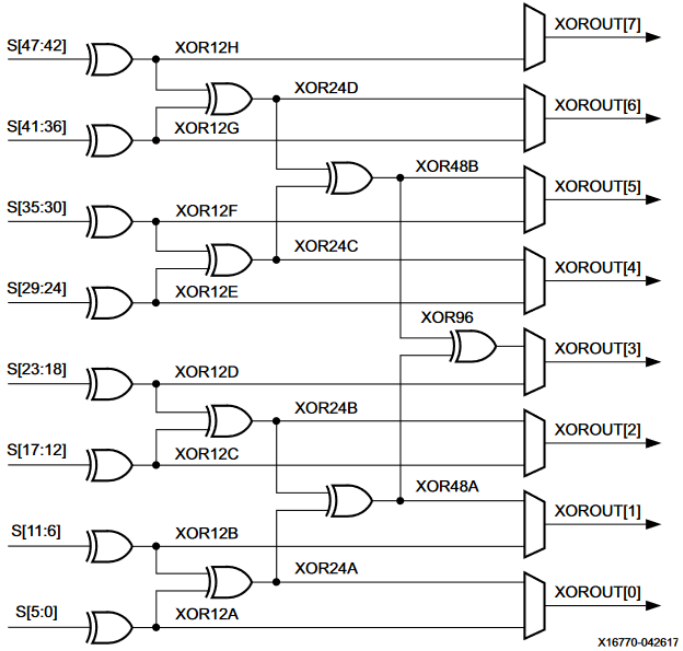
Net (i/h/o)		Used	Util% (UT4 80)	Used (all)	Util% (UT4 80)
25 / 100 / 2	CLB	9790	14.57 %	210960	313.93 %
	LUT	47793	10.72 %	1147032	257.35 %
25 / 75 / 2	CLB	8017	11.93 %	192408	286.32 %
	LUT	39020	8.75 %	936480	210.11 %
25 / 50 / 2	CLB	5866	8.73 %	140784	209.5 %
	LUT	28589	6.41 %	686136	153.94 %

[COR] http://www.cs.cornell.edu/courses/cs5740/2019sp/lectures/03-nn.pdf

DSP

- Xilinx DSP:
- DSP48E1
  - 48-bit logic
- DSP48E2
  - 96-bit wide XOR function
    - Eight 12-bit wide XOR
    - Four 24-bit wide XOR
    - Two 48-bit wide XOR

- Less resources need due to DSP use for XOR calculation
- But the VHDL implementation is time consuming

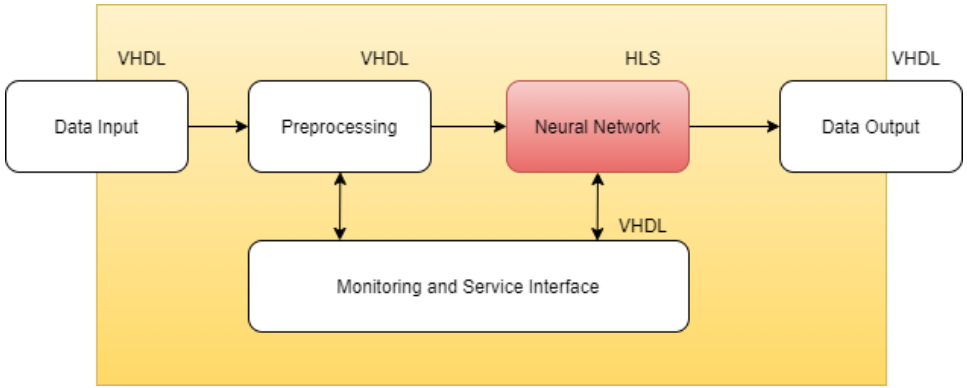


Tooling

- Development of neural networks in VHDL is inflexible and takes a very long time
- Automated frameworks are getting better
- Divide the hardware architecture into a VHDL part and a high-level language part
  - Time-critical parts are kept in VHDL (In/Output, Preprocessing ..)
  - Algorithmic part is written in high level language and automatically translated into VHDL

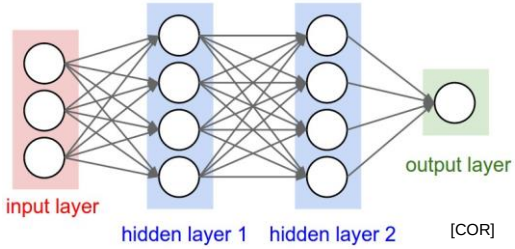
QKeras

hls4ml



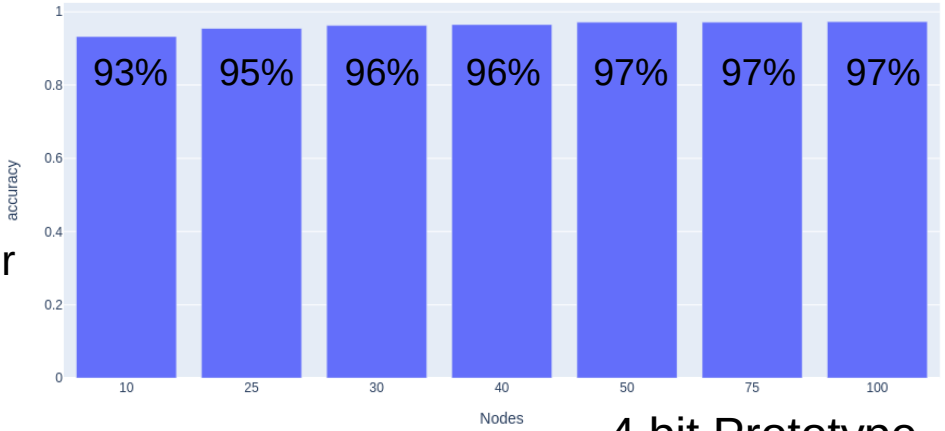
First results

- Different models tested
  - Multilayer perceptron because it need to be small and fast
- From the experiment given:
  - 4 clock cycles calculation time
    - Input layer, 2 hidden layer, output layer
    - Fix point NN (4 bit)
    - Accuracy >90
- Next steps:
  - Smaller Networks
    - Input layer, 1 hidden layer, output layer
  - Smaller bit width
    - (2 bit) (1 bit)
  - Build tool flow with hls4ml



[COR]

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$



4 bit Prototype

[COR] http://www.cs.cornell.edu/courses/cs5740/2019sp/lectures/03-nn.pdf

Summary

- Two new Track Segment Finder approaches are possible
  - It is not yet decided which one will be used
- Lookup Table Track Segment Finder
  - Use pattern stored in Lookup Tables to determ valid tracks
  - Bigger Lut better SNR
- Neural Network Track Segment Finder
  - Use neural network to determ valid tracks
  - Hard to fit so may neural networks on FPGA