

TOPICAL WORKSHOP ON ELECTRONICS FOR PARTICLE PHYSICS
BERGEN, NORWAY
19–23 SEPTEMBER 2022

Data-driven design of the Belle II track segment finder

K.L. Unger,^{a,*} M. Neu,^a J. Becker,^a E. Schmidt,^b C. Kiesling,^b
F. Meggendorfer^{b,c} and S. Skambraks^a

^a*Institut für Technik der Informationsverarbeitung (ITIV),
Karlsruhe Institute of Technology (KIT), Kaiserstraße 12, Karlsruhe, Germany*

^b*Max Planck Institute for Physics (MPI), Föhringer Ring 6, Munich, Germany*

^c*Technical University of Munich (TUM), Arcisstraße 21, Munich, Germany*

E-mail: kai.unger@kit.edu

ABSTRACT: The Belle II experiment relies on a level-1 trigger system to reduce noise background and preselect events of interest for particle physics. The Central Drift Chamber is the main track detector which makes its trigger system important for online track reconstruction. To improve its hit efficiency, an extension of the track segment finder for low angle tracks is proposed. By combining hardware and software development flows, an automated data-driven pipeline is created and three different-sized hardware concepts are implemented. The operation point is adjustable to balance hit efficiency against hit purity in the trigger system.

KEYWORDS: Trigger algorithms; Trigger concepts and systems (hardware and software)

*Corresponding author.

Contents

1	Introduction	1
2	Concept	2
3	Evaluation	4
4	FPGA implementation	5
5	Conclusion and outlook	6

1 Introduction

The Belle II experiment at KEK in Tsukuba (Japan) investigates CP violations in B mesons. For this purpose, collisions with the record luminosity $L = 4.21 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ (30.6.2022) are achieved in the asynchronous electron positron collider SuperKEKB. To keep the quantity of data manageable, a two-stage trigger is implemented. The first stage consists of an FPGA based pipelined dead-time free first level trigger (L1) [1] that operates on the data from the Central Drift Chamber (CDC) [2], Electromagnetic Calorimeter (ECL) and the Muon detector (KLM). Subsequently, a final decision is made with the complete detector data in the High Level Trigger (HLT) [3]. The L1 decision time is defined as the time between the moment of data readout to the moment of the final trigger decision. It provides an upper boundary for the maximum latency of 5 μs . Figure 1 depicts the L1 CDC trigger chain for online track reconstruction. The track trigger receives data from the CDC readout system [4] which consists of 56 layers arranged in nine super layers (SL). The innermost SL features eight layers and all other layers feature six layers. For the first filter stage, also described as track segment finder [5], one FPGA is utilized for each SL to process 14336 drift wires inside the CDC in parallel. Five layers of each SL are connected to the trigger system. This filtering is based on a geometric hourglass shape in SL one to eight, as shown in figure 2 on the left. In addition a triangle shape is featured in SL zero. To generate a valid track segment (TS) at least four of the five layers in these geometric shapes must be hit at the same time. This limits the radius of curvature of a particle to about 30° around the detector vertex. The reduced data is then sent to the 2D Finder and the Event Time Finder [6] which process the data and send it to the 3D Finder [7] and the z-Vertex Track Trigger (Neuro Trigger) [8]. The current L1 CDC trigger is not optimized for the detection of long-lived particle tracks with a low angle of incidence. Such tracks are classified as background noise, since only tracks from the detector vertex are taken into account. In order to enable the detection of flat tracks, a novel track trigger system is required. Therefore the current track segment finder (TSF) has to be extended with the displaced vertex trigger.

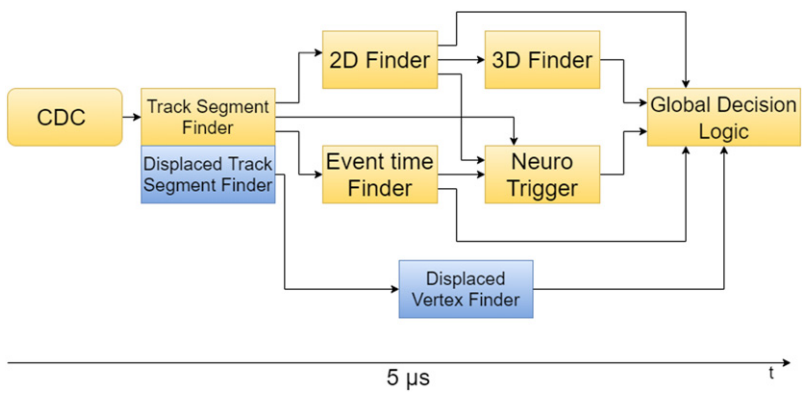


Figure 1. The Belle II CDC L1 trigger chain. Planned extensions are blue whereas the existing system is yellow. This contribution comprises the *Displaced Track Segment Finder*.

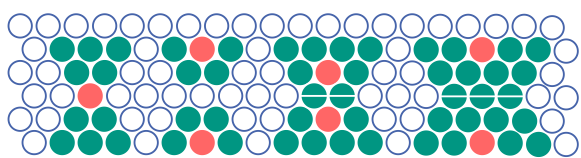


Figure 2. Exemplary visualization of the TSF hitmaps. The leftmost hitmap is implemented in the current TSF. The other three versions, from left to right, are called LUT-5, LUT-9 and LUT-12 TSF. The green dots represent the TS hitmap while the red dot shows the TS address.

2 Concept

The novel displaced vertex TSF must meet the following requirements. First, it must provide similar noise suppression compared to the current system while removing the 30° restriction. Second, the system must be implemented on the current Universal Trigger Board 4 (UT4) in parallel to the existing TSF, adhering to the tight latency constraint of 200 ns. The concept refrains from using the hourglass structure but divides the original hitmap into two parts, defined as upper and lower. The changed architecture implies two differences from the current system. First, the number of TS and therefore the number of addresses is doubled. Second, the width of all hitmaps is increased. As a result, the following changes are required to compensate the previously utilized hourglass structure.

As the track segments are only two to three layers in height in comparison to five layers for the current TSF, it is no longer possible to assume a minimum number of hits per SL. To enable noise filtering, a look-up table approach has been chosen. Each signal wire is labeled with a unique id which resembles a memory address. In figure 2, all hitmap implementations are depicted. On the left, the state-of-the-art approach using fixed layers is shown. To the right the LUT-5 version, which allows 32 combinations, is illustrated. For this hitmap, six wire configurations resemble valid track patterns. As the number of patterns increases exponentially with the number of wires per hitmap, it is not possible anymore to calculate valid patterns of the LUT-9 and LUT-12 TSF versions by hand. To enable the evaluation of all possible patterns per hitmap, a data-driven software training

framework is developed that allows training of LUT patterns with simulated and recorded data from the experiment. Figure 3 shows the pattern training flow. The data sources are shown in green and are either taken from the simulation tool basf2 or from the detector. The training process is split into three sub procedures, of which the first is called pattern evaluation. Based on the supplied event data, a metric score is calculated for every pattern determining how likely it corresponds to a valid track segment. Currently three metrics are implemented in the framework, namely the precision score (purity), the recall score (efficiency) and the f1-score. The second procedure determines the selection of suitable patterns depending on the previously calculated scores. In a naive approach, a threshold between zero and one is determined by the developer. This threshold describes the fraction of valid patterns from all possible patterns that will be hypothesized as valid track segments. This parameter is used to select a configuration depending on the desired noise suppression. In the third and last procedure of the training framework a synthesis configuration file is generated. It contains an array of single bits, each indicating a positively or negatively hypothesized pattern. This array is then stored in either distributed memory or in BRAM cells. Based on the system architecture, the threshold can be chosen freely and does not impact timing closure on the target FPGA. As of now, the system is designed in such a way that the DAQ transmission limit is always complied with. In the case that more track segments are classified positively per event than optical link bandwidth is available, remaining track segments are discarded. To avoid discarding relevant experiment data the noise suppression must be increased, i.e. the threshold parameter must be lowered. It is therefore recommended to analyze the distribution of the underlying dataset to find a suitable operation point. In addition it is noted that each super layer is handled by a separate UT4 board. It is therefore possible to program individual pattern configurations depending on the detector condition and board position inside the detector.

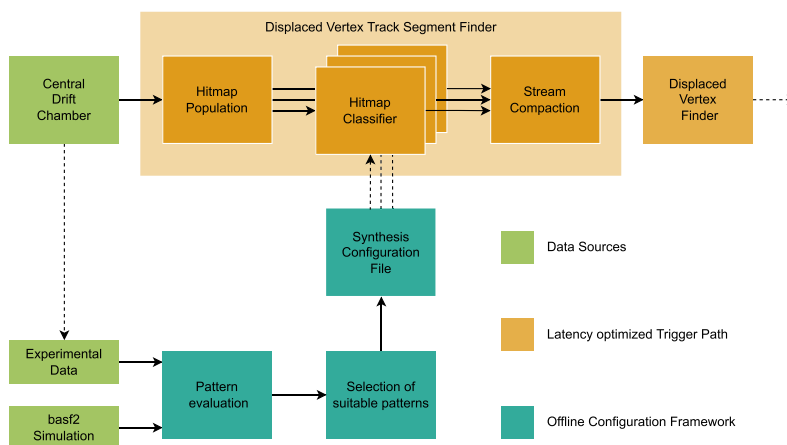


Figure 3. Architecture of the data-driven track segment finder. The pattern training framework is depicted in mint on the bottom half of the figure. Latency optimized firmware implementations are orange.

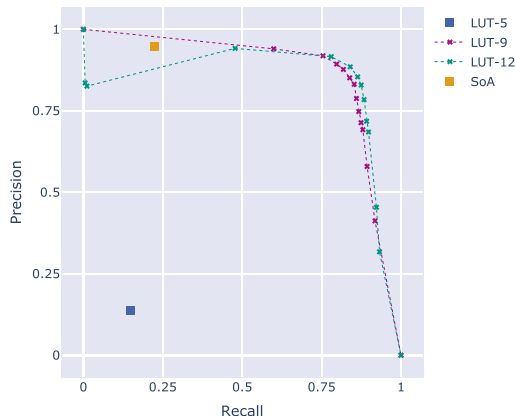


Figure 4. Precision vs. recall with all types of TSF.

3 Evaluation

The performance estimation of the displaced vertex TSF is based on simulated events which have been generated with basf2. A train test ratio of 90% training's data to 10% test data has been used. Each collision event contains labels for all sense wire inside the CDC. Possible labels are no hit, true hit or noise hit. Therefore an exact knowledge of which wires belong to particle tracks and which to belong to noise events is given. The noise power level is adjusted accordingly to measurements of the previous runs of the Belle2 experiment. Labels of valid sense wire hits are generated from previously not found displaced vertex decays. Since the displaced vertex TSF acts as a binary classifier on hypothesized track segments, precision (hit purity) and recall (hit efficiency) are utilized as metrics for the evaluation. Plotting the precision over the recall score yields a receiver operating characteristic curve (ROC curve) for each version of the TSF. In the following, the performance of the three novel versions are compared to the performance of the current TSF on the same dataset. Figure 4 depicts the performance of the three novel TSF implementations and the current TSF. The current TSF and the LUT-5 TSF version are not parameterized and therefore only support one operation point. The LUT-5 TSF shows the worst precision and the worst recall score compared to all other solutions. The current TSF features a precision score of 94.7% and a recall score of 22.3% which serves as a baseline. In practice a track segment detected by the current TSF will be correctly classified, but many track segments remain undetected. This behavior reflects drastically in the recall score, as the evaluation set contains many events with displaced vertices which often feature tracks with a shallower angle than 30° . The LUT-9 and LUT-12 TSF versions are parameterized by the threshold value. Therefore it is possible to evaluate a range of operation points. Each evaluated operation point is indicated by a cross and connected with a dotted lines. This linear interpolation between operation points serves as rough guideline for the expected performance. In figures 4 and 5 it can be seen that the LUT-9 and LUT-12 TSFs provide a similar precision to the current TSF. By adjusting the threshold value it is possible to increase the recall score, indicating the fraction of detected wire hits, by a large number without reducing the precision score greatly compared to the current TSF. Lastly it is noted, that the LUT-12 TSF performs slightly better than the LUT-9 TSF. For trigger applications, high recall is of particular interest. Here the

LUT-12 performs slightly better than the LUT-9 version and yields a vast improvement compared to the current solution. It is important to note that the ROC curves of the LUT-9 and the LUT-12 may change depending on the training and evaluation data used. However, the concept remains the same.

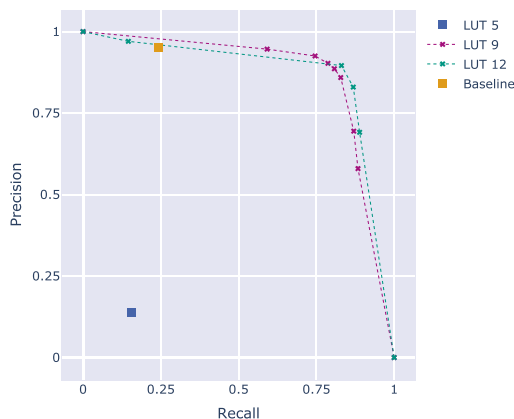


Figure 5. Precision vs. recall with LUT-9 and LUT-12 combined with the current TSF.

4 FPGA implementation

The implementation is realized on the Universal Trigger Board 4 (UT4) which is also utilized in the experiment. The platform features either a XCVU80 or a XCVU160 Xilinx Virtex Ultrascale FPGA. For a comprehensive evaluation of the module, the parameters are configured to match the size of the outermost super layer (SL8). It features the largest number of detector inputs, 1920 sense wires in total, and therefore its resource utilization is deemed critical for the deployment of the system. In the final system the implementation is realized in parallel to the current TSF, i.e. both modules execute concurrently on the same FPGA. Therefore it is required to keep the utilization of the novel TSF as low as possible. Table 1 shows the resource utilization of the standalone implementation of the TSF in the UT4 FPGA (XCVU80 and XCVU160). Comparing the utilization of the examined modules some distinct differences can be made. First, the LUT-5 and LUT-9 versions of the TSF do not utilize any BRAM cells, i.e. all patterns are mapped into distributed memory. This method is preferred by the synthesis algorithms, because only one bit (hit or miss) is saved per address (pattern). For the larger LUT-12 version, storing all patterns in distributed memory would require vast amounts of the device resources. Therefore a implementation in BRAM cells is preferred. Second, the utilization of the remaining resource types is nearly constant and only depends on the number of detector inputs. Since the LUT utilization amounts to 20.42% for the LUT-12 version on the small FPGA, all versions can be implemented in parallel together with the state-of-the-art system. In addition it is noted, that the processing latency amounts to eight clock cycles at a system frequency of 127 MHz. As a result the latency adds up to 62 ns and therefore meets the requirements of 200 ns for the whole system including receiving and transmitting the data.

Table 1. Resources utilization of the three module implementations on SL8 for the UT4 board.

Features	LUT-5		LUT-9		LUT-12	
	XCVU80	XCVU160	XCVU80	XCVU160	XCVU80	XCVU160
CLB logic	21.01%	10.11%	20.42%	9.82%	20.42%	9.82%
CLB memory	1.02%	0.35%	9.05%	3.15%	1.02%	0.35%
CLB registers	4.56%	2.20%	4.74%	2.28%	4.60%	2.24%
CARRY8	7.94%	4.24%	7.94%	4.24%	7.94%	4.24%
BRAM	0%	0%	0%	0%	26.69%	11.70%

5 Conclusion and outlook

This paper presents a novel FPGA-based track segment finder for displaced vertex tracks. It is based on a pattern recognition algorithm which can be reconfigured between experiment runs. Patterns are stored in distributed memory or BRAM cells inside an FPGA and are evaluated offline using a data-driven training framework. The different TSF configurations were trained and tested with different threshold values. These values were then plotted in a ROC curve to evaluate an optimal operation point based on the system requirements. With the help of the training framework, the new displaced TSF can be quickly adapted to the changing background conditions. Three new FPGA hardware implementations have been designed, implemented and tested for deployment. The resource utilization for all modules has been evaluated on two configurations of the UT4 platform. All implementations meet the resource constraints of the trigger system in its current state. The next step is the integration of the modules into the Belle II trigger system during long shutdown one, so that a displaced vertex track trigger can be tested in operation after its completion.

Acknowledgments

Funded by the German Federal Ministry of Education and Research under “Verbundprojekt 05H2021 (ErUM-FSP T09) — Belle II: Pixeldetektor, Software und erste Datenanalysen”.

References

- [1] Y.T. Lai et al., *Development of the Level-1 track trigger with central drift chamber detector in Belle II experiment and its performance in SuperKEKB 2019 Phase 3 operation*, 2020 *JINST* **15** C06063.
- [2] Belle II collaboration, Z. Doležal and S. Uno, *Belle II Technical Design Report*, KEK Report 2010 (2010). Available: <https://arxiv.org/abs/1011.0352>.
- [3] R. Itoh, T. Higuchi, M. Nakao, S.Y. Suzuki and S. Lee, *Data flow and high level trigger of Belle II DAQ system*, *IEEE Trans. Nucl. Sci.* **60** (2013) 3720.
- [4] N. Taniguchi et al., *All-in-one readout electronics for the Belle-II central drift chamber*, *Nucl. Instrum. Meth. A* **732** (2013) 540.

- [5] K.L. Unger, S. Bähr, J. Becker, Y. Iwasaki, K. Kim and Y.T. Lai, *Realization of a state machine based detection for track segments in the trigger system of the Belle II experiment*, Proceedings of Topical Workshop on Electronics for Particle Physics, Proceedings of Science, Vol. 370 (2020).
- [6] E. Won and H. Moon, *Development of an event time finding algorithm for multi-wire drift chamber-based Level-1 trigger system in the Belle II experiment*, *J. Korean Phys. Soc.* **80** (2022) 1.
- [7] E. Won, J.B. Kim and B.R. Ko, *Three-dimensional fast tracker for the central drift chamber based level-1 trigger system in the Belle II experiment*, *J. Korean Phys. Soc.* **72** (2018) 33.
- [8] S. Baehr et al., *Low latency neural networks using heterogenous resources on FPGA for the Belle II trigger*, Connecting the Dots and Workshop on Intelligent Trackers (2019).