Real-time data processing pipeline for Trigger Readout Board based Data Acquisition systems

A. Malige, G. Korcyl, M. Firlej, T. Fiutowski, M. Idzik, B. Korzeniak, R. Lalik, A. Misiak, A. Molenda, J. Moron, N. Rathod, P. Salabura, J. Smyrski, K. Swientek, P. Wintz

***Abstract*—Large scale physics experiments running at high interaction rates place high demand on the data acquisition system responsible for transporting the data from the detector to the storage. PANDA at FAIR is one such experiments of the future which will not use fixed hardware triggers; instead, the event selection is based on real-time feature extraction, filtering, and high-level correlations. A firmware framework for such real-time data processing has been developed and tested with hardware setup for a PANDA Forward Tracker prototype. The solution is applicable for other detector subsystems based on the so-called Trigger Readout Board data read-out system.**

***Index Terms*—Data acquisition systems, tracking detectors, real-time data processing.**

I. Introduction

The PANDA experiment [1] at the Facility for Anti-proton and Ion Research (FAIR), in Darmstadt (Germany) [2], will be an example of a large system with several detector subsystems. PANDA will operate at high beam-target interaction rates reaching 20 MHz [3], with a broad variety of event topologies and a dedicated list of features to distinguish and identify interesting physics processes. As a consequence, data collection generates a continuous data stream of about 200 GB/s [4]. The continuous read-out allows for more flexible measurements but also gathers excess data, even when no hits are detected in some subsystems. For instance, the Forward Tracker (FT), a detector system in PANDA can produce from 1 GB/s to 10 GB/s depending on the PANDA mode of operation [5, 6]. This large amount of raw data could be significantly reduced in the initial stages of the Data Acquisition System (DAQ).

The DAQ for PANDA is designed with three major components: detector read-out Front End Electronics (FEE), Data Concentrators (DC), network infrastructure and Online Processing Nodes (OPN) for synchronous event processing, as shown in Fig. 1.

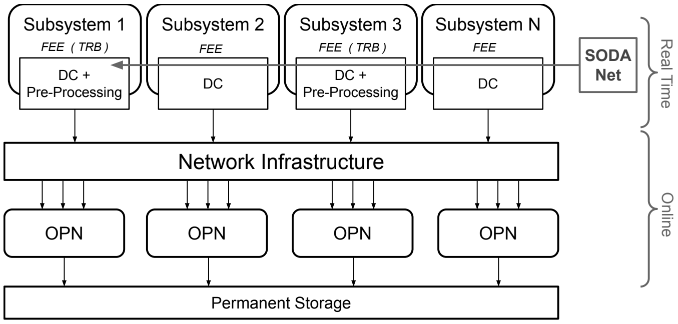

Fig. 1. The overview of the PANDA read-out system along with the pre-processing pipeline in the Data concentrators (DC) for TRB based detector systems and Online Processing Nodes (OPN) for synchronous event processing.

The FEE detects particle hits, prepares raw analog signals and performs digitization in dedicated units. One of the most commonly used digitization components of the FEE is the Trigger Read-out Board (TRB) [7, 8, 9]. It is designed for a wide range of applications in various experiments [10, 11, 12]. An FPGA-centric hardware architecture, well-defined network protocols, firmware modules for intra and extra FPGA communication along with software tools to interface the system functionalities make the TRB a complete system for read-out and data acquisition. The Straw Tube Tracker (STT), Forward Tracker (FT) and Barrel Detection of Internally Reflected Cherenkov light detector (DIRC) are some of the detector subsystems in PANDA that will use the TRB for time measurement. The digitized data from the TRB are sent to the Data Concentrator. The DC is designed to collect the digitized data from all the PANDA FEEs as well as to distribute detector synchronization signals. Both data collection and synchronization signal distribution is done using a protocol called Synchronization Of Data Acquisition Network (SODANet) [3, 4] by providing a common clock signal and timestamps for all PANDA subsystems.

This section of the DAQ uses custom hardware which is common to all PANDA subsystems and should operate in a real-time regime. The network infrastructure transmits digitized data from all the PANDA subsystems. The data packets belonging to the same time frame are sent to a single OPN for further online analysis. The OPN will perform processing of complete time frames from all the subsystems using various algorithms implemented in CPU and GPUs. It is

Manuscript received 29 April 2022; revised 20 June 2022. This work has received financial support from the Jagiellonian University DSC grant No. 2020-N17/MNS/0000028 and the Foundation for Polish Science grant no. TEAM/2017-4/39. This research was funded by the Priority Research Area Digiworld under the program Excellence Initiative – Research University at the Jagiellonian University in Kraków.

A. Malige, G. Korcyl, B. Korzeniak, R. Lalik, A. Misiak, N. Rathod, P. Salabura and J. Smyrski are with the Faculty of Physics, Astronomy and Applied Computer Science, Marian Smoluchowski Institute of Nuclear Physics, Jagiellonian University, 30-348 Kraków, Poland (e-mail: mailme.akshaym@gmail.com).

M. Firlej, T. Fiutowski, M. Idzik, A. Molenda, J. Moron and K. Swientek are with the Wydział Fizyki i Informatyki Stosowanej, Akademii Górniczo-Hutniczej, 30-059 Kraków, Poland.

P. Wintz is with the Nuclear Physics Institute, Forschungszentrum, 52425, Jülich, Germany.

therefore important to reduce the size of the raw data stream at the Data Concentrator level by filtration and extraction of useful information only.

Many physics experiments use early feature extraction techniques for trigger generation [13]. As PANDA operates under continuous read-out mode, feature extraction will be used for data reduction. A data processing pipeline for such feature extraction has been developed and has been adapted successfully in the read-out for the PANDA FT detector prototype. As the hardware for the final PANDA DC is still under development, the processing pipeline has been implemented and tested on a commercially available board, Xilinx ZCU102, which is equipped with a Zynq UltraScale+ MPSoC. The system and the processing algorithm and some of the obtained results are presented in this article.

In section II we describe the FT detector and its read-out system for which the pre-processing pipeline has been developed. In section III we describe the general layout of the processing pipeline along with its components, functionalities, usage, and its scheme for development and testing. Section IV describes the evaluation of the developed system and the results obtained from various tests performed with proton beams and radioactive sources. The optimization prospects and future developments are described in Section V. The paper concludes with the scope and application in the PANDA experiment in section VI.

II. Description of the System

A. Forward Tracker

The Forward Tracker (FT) is designed for the momentum measurement of charged particles deflected in the field of the dipole magnet of the PANDA Forward Spectrometer (FS) [3]. It is based on straw tube detectors, with a tungsten-rhenium wire located in the axis of the tube as an anode [14]. The high voltage applied between the cathode and anode causes the ionization electrons to drift to the anode wire. The longest drift time (DT) corresponding to the maximum distance between the particle track and the anode wire (0.5 cm) is about 180 ns. The ionization electrons reaching the anode wire are multiplied in the strong electric field near the anode. The resulting ionization electrons and positive ions drift in the electric field, to the anode and cathode, respectively, inducing an electrical signal on the electrodes.

The electric signals are processed by dedicated Front End Electronic Boards (FEB), a part of the FEE that prepares raw analog signals and is connected directly to the detector. It is equipped with two 8 channel charge-sensitive front-end ASIC called PASTTREC (PANDA Straw Tube Tracker REad-out Chip) [15]. Here the signals are shaped, amplified, and discriminated with a common threshold set by an integrated Digital to Analog Converter (DAC) . The drift time of the ionization electrons in the straw is calculated from the time difference between the rising edge of the signal and the time of flight of the particle through the straw, which is registered by a T0 detector. In turn, the difference between the rising edge and falling edge of the signal, referred to as the Time-Over-Threshold (TOT) of the straw pulse, is used to determine the particle energy loss in the straw. The digital part of the FEE for FT uses TRB for time measurement with a precision TDC (20 ps RMS time resolution), data movement, read-out process control and inter-FPGA communication. The PANDA FT consists of 12,224 straws arranged in three pairs of tracking stations: one pair (FT1, FT2) is placed before the dipole magnet, the second pair (FT3, FT4) inside the dipole magnet, and the third pair (FT5, FT6) is placed after the magnet, as shown in Fig. 2.

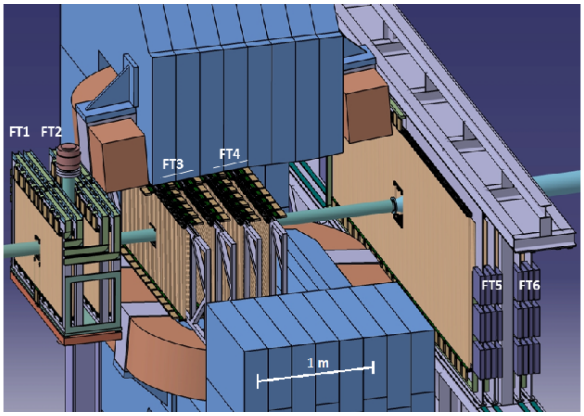

Fig. 2. Forward Tracker stations (FT1-FT6) in the PANDA Forward Spectrometer with the dipole magnet [5].

B. System Requirements

The FT in PANDA will use the TRB5sc, latest hardware of the TRB family [16]. Each TRB5sc provides 64 TDC channels, sufficient to read-out 4 FEBs and 9 TRB5sc boards will be mounted on a custom crate as shown in Fig. 3. Each crate will house a master board which acts as an interface between the slave TRBs in the crate and the DC as shown in Fig. 3. The DC will function as a bridge between the output of the master board and the network infrastructure. The data in the DC are stamped with time information, synchronized and are read-out to the network infrastructure at a rate defined by the SODANet (see further). The proposed intermediate pre-processing pipeline in the DC has to unify the data streams from master boards arriving at the DC and filter the data before they are sent over the network.

TABLE I
PANDA-FT read-out requirements.

	FT 1,2	FT 3,4	FT 5,6
No. of channels	2304	3328	6592
Average hit rate per straw	35 kHits/s	31 kHits/s	9 kHits/s
TRB5sc's	36	52	103
Master boards	4	6	12
Total bandwidth	324 MB/s	410 MB/s	237 MB/s

The number of TRB5sc, master boards and expected data rates for each of the PANDA FT station pairs are presented in Table. I. The pre-processing pipeline must be compact enough to be implemented on the FPGA that will be used in the DC hardware (Xilinx Ultrascale+ KU15P) and guarantee filtering the data from all the links without compromising any valuable information.

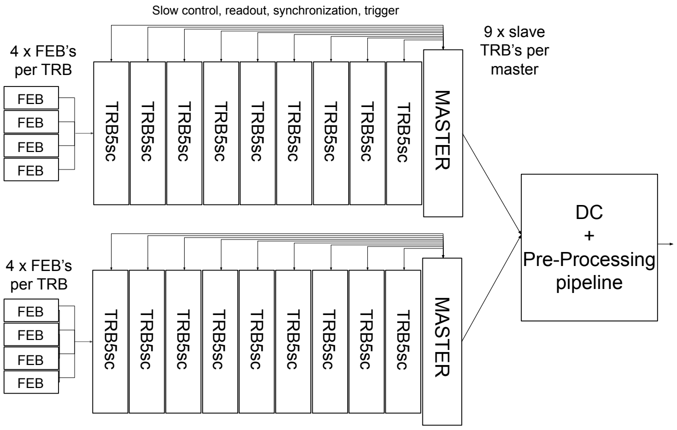
Slow control, readout, synchronization, trigger
4 x FEB's per TRB
9 x slave TRB's per master
FEB
FEB
FEB
FEB
TRB5sc
MASTER
DC + Pre-Processing pipeline
4 x FEB's per TRB

Fig. 3. Schematic view of the TRB5sc system for the Forward Tracker read-out. Four FEBs (analog FEE) are connected to a TRB5sc (digital FEE) with a ≈10 meter long 40-pin ribbon cable. A master board interfaces 9 TRBs placed in a crate to the DC hardware.

In the PANDA experiment, interaction of the antiproton beam with the internal proton target will last for 2 μs and will be followed by an interval of 400 ns. This interval is used to gain time for the data processing. Such a 2.4 μs cycle is called burst and 16 such bursts make up one 'super-burst'. Detector data are read out from the DC on each super-burst update i.e at ∼26 kHz (SODANet frequency) [4]. The pre-processing pipeline has to be capable of filtering the data within this time limit (∼38.4 μs). Additionally, as DC is a common system for all detectors in PANDA, the pre-processing pipeline is expected to be easily adapted by other detector systems using TRB based read-out or other read-out hardware that use similar data formats.

III. Pre-processing pipeline

A. General Layout

The developed processing pipeline is designed to be placed within the DC firmware, in between the data receivers from the detector subsystems and the common network. The data are received, prepared for processing, analyzed for feature extraction by the processing component, prepared for transmission, and transmitted in a push-forward manner as shown in Fig. 4. The extracted feature can be either used as a parameter to filter out meaningless data from the stream (packet validity) or can be transmitted as an additional data packet along with the original data. The user can decide how the data have to be handled in the pipeline. The three modes of operation of the designed module that can be selected are listed below.

- Filtering mode: Only the filtered data packets with meaningful features are transmitted and the raw data from the TRB and the data packets with no interesting features are dropped, thus reducing the data volumes.
- Marking mode: All the raw data from the TRB is transmitted and the data packets with meaningful features are marked with a header word, thus preserving the raw data for further analysis. This mode allows estimating the effect of filtering via offline analysis.
- Bypass: Raw data from the TRB links is forwarded out to the network without any pre-processing.

The pre-processing unit has a modular structure and the components share a common, simple interface that is described below.

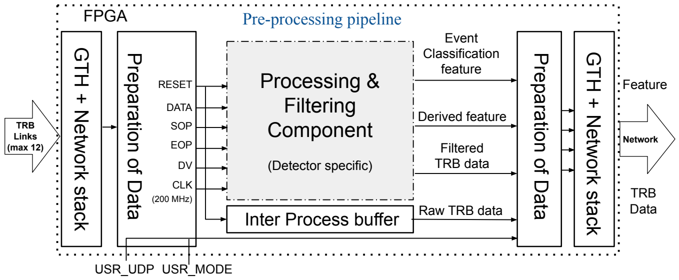

Fig. 4. Scheme of the processing components and pre-processing pipeline. The layout represents the components of the system implemented in the FPGA. It consists of a transceiver network stack, data preparation modules, inter process buffers and the data processing and filtering component (see text for details).

The communication modules in the processing pipeline implement a basic set of network protocols (ARP, DHCP, ICMP, IP, UDP) that allow to directly connect the TRB Ethernet output links to the pre-processing pipeline over the Gigabit transceivers (Xilinx GTH). The module accepts the data from all the TRB boards and forwards them along with the Start-Of-Packet (SOP), End-Of-Packet (EOP), and Data-Valid (DV) signals for the preparation of the data in the pipeline.

Data from different links can arrive at different times and can have different packet sizes. The data packets are stored in de-randomization buffers until the data from all the links (EOP) arrive and a single, parallel data stream is constructed. These data are then mirrored into two separate data paths: one preserves the original, raw data and the other one is used for applying the processing and filtering mechanisms. They are both synchronized in time to react to the processing results, e.g. drop, forward or mark current data packet upon feature extraction decision.

The processing component is the module in the pipeline which is dedicated to the extraction of the features from the data. The Features extracted by the processing unit are of two kinds: the event classification feature and a derived feature. The event classification feature is used for zero-suppression by resolving the validity of the data packet after checking it across user-defined conditions. The derived feature is the meaningful information generated additionally, after analyzing the zero-suppressed data. All detector-specific operations and algorithms are performed here and this component can be easily replaced to serve other applications. With modern technologies like High-Level Synthesis (HLS) [17], complex detector-specific analysis can be made simple and adapted in this module, as long as the pipeline's interface and timing are maintained. HLS allows to accelerate development and debug cycles of the algorithmic and computational parts of the pipeline through their implementation in C++ and then conversion to regular HDL components. This also enables for the direct reuse of source code used in offline analysis,

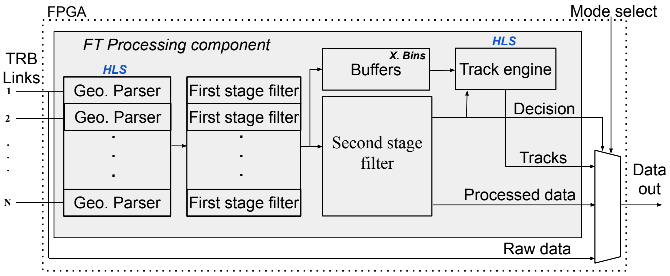

Fig. 5. Schematic view of the component of the processing pipeline for the FT with the geometry parser, first stage filter, second stage filter and the tracking engine (see text for details).

ensuring consistency. The raw TRB data, filtered data from the processing component, and the features extracted, all arrive at the preparation stage for transmission. Depending upon the operation 'mode' (USR_MODE), the data (FD_DATA) to be transmitted is packed and transmitted to the designated destination (USR_UDP).

B. Pre-processing for FT

The task of the FT in PANDA is the reconstruction of particle trajectories. Therefore, the key information to be extracted from the data stream is the identification of data packets with tracks and their parameters. Being able to identify data packets with valid tracks originating from the target will eliminate background induced from 'out-of-the-target' interactions.

In the context of this work, the time duration between two read-outs is called a time frame. It can contain many interaction events with multiple tracks. Additionally, the straw tubes are slow detectors with a maximum drift time of 180 ns. In order to identify and separate these tracks, the time and TDC channel information transferred from the TRB must be translated into hits in the detector geometry, sorted in time order of signal arrival, and later checked across the track recovery conditions, so that accidental coincidences are discarded. The analysis module of FT data is made of four components displayed in Fig. 5: Geometry parser, first-stage filter, second-stage filter, and tracking-engine. These are described in detail below.

C. Geometry Parser

The Geometry Parser is the first module in the 'FT processing component'. The data received by the pre-processing unit are prepared for processing and forwarded to the geometry parser. A lookup table describing the relation between the TDC channel and the detector geometry is provided to the parser. In this way, the data are stamped with the geometry parameters like detector layer, module, straw and its coordinates (x, y, and z) in the global reference system which will further be necessary for reconstructing tracks from data. The processing is parallelized by a dedicated instance of geometry parser for each of the input links (TRB crate) in the processing component. This module is implemented using HLS, as it entirely relies on the detector geometry and could be modified easily for any change in the detector setup.

D. First Stage Filter

The first-stage filter assures the quality of the raw data. The raw data may consist of packets with no hits (empty packets), hits with minor glitches (noise), or corrupted, i.e. TDC artifacts. This includes hits with repeated edges (rising or trailing), missing time-stamps, or unrealistic TOT. Such data words can produce fake multiplicities, thus affecting the track reconstruction conditions in the further stages. The role of the first-stage filter is to reject such data. Firstly, a 64-bit word is constructed by utilizing the calibration and geometry attributes from the geometry parser, as described in Table II.

TABLE II
DATA FORMAT FOR FT PROCESSING COMPONENT

Bits	Attribute
0	Pair valid flag
1	LT pair check flag
2	Epoch time corrupt flag
3	Hit valid flag
4	Rising-Falling edge
5 - 10	Straw
11 - 14	Straw layer
15 - 18	Detector Module
19 - 50	TDC Time
51 - 63	Unused

Every data word arriving at the first-stage filter module with a rising edge waits for the next word in a shift register. The data word is pushed forward only if the next word is a trailing edge. This ensures to have a rising and falling edge pair and mitigates the risk of having fake hits. In order to eliminate corrupted hits, the time stamp in the data word is compared with the reference time of the associated TDC and the data are accepted only if the difference is within a reasonable range. An instance of the first-stage filter is created for every input link in the system.

E. Second Stage Filter

In a continuous read-out mode, the data units represent time-frames of fixed length. Particle tracks can be extracted from a pool of hits only after sorting them into well-separated groups inside a given time period corresponding to the physical event. For this purpose, the entire time-frame is divided into smaller time-bins and all the hits in the time-frame are sorted into these time-bins based on the rising edge of the respective hit. These time-bins are treated individually from here on and the presence of a potential track can be recognized if the time-bin fulfills the following configurable coincidence conditions:

- Number of hits in a time-bin must be above the minimum number of hits required to reconstruct a track.
- Hits must be present in defined layers of the detector.
- Hits in a time-bin must be in pairs for a defined number of detector layers (FT straw layers consist of two staggered straw planes).

With all these conditions fulfilled, the group of hits in this time-bin is defined to originate from the same event and constitutes a track.

These virtual time-bins must be large enough to capture all the hits belonging to a track and small enough to avoid mixing

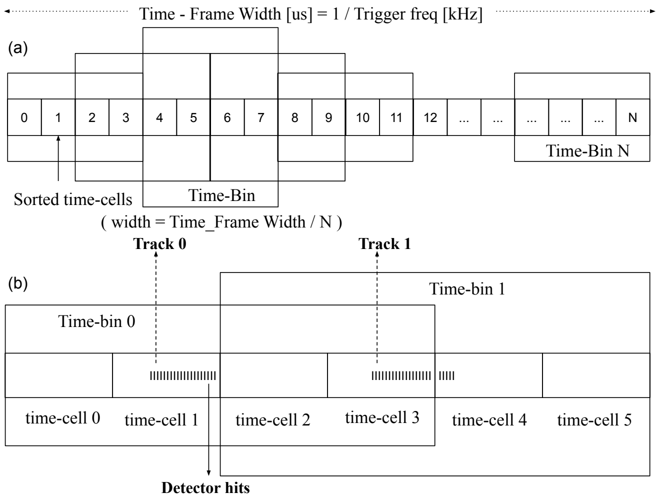

Fig. 6. Schematic representation of a time-frame divided into 'N' number of time-bins consisting of four time-cells each (a). Schematic illustration of a case with more than one track in a time-bin. 'Track 1' is separated from 'time-bin 0' to 'time-bin 1' due to the overlap of time-cells (b).

of tracks. To achieve this task, each time-bin is further divided into four 'time-cells' of suitable width, into which the hits will be assigned based on its TDC rising time as shown in Fig. 6. The number of time-bins per time-frame depends strictly on the time-frame length, drift time of straws ($\approx$ 180 ns), and on the resources available in the pre-processing hardware. To avoid the hits belonging to a track being spread across adjacent time-bins, they are overlapped by two time-cell units. With this, we can expect all the hits from a track to be in a common time-bin and can be checked for the track recovery conditions. When all the filtering conditions are satisfied, the time-frame is recognized to be valid and a positive decision is issued from the second-stage filter along with the index of the time-bin containing the potential track. The sorting of hits into the time-bins and the coincidence checks are done in a push-forward manner as the data passes through the component. The decision on the presence of a potential track is issued five clock cycles after the last data word (EOP) in the time-frame has arrived (for a total of 25 ns in the hardware used). The width of the time-cells, number of time-bins per time-frame, the overlap offset, and the coincidence conditions in this stage are configurable and can be tuned based on the operational conditions of the detector.

F. Tracking Engine

A copy of the data supplied to the 'second-stage filter' is buffered when the data are processed for feature extraction (see Fig. 5). The index of these buffers is equivalent to the index of the time-bins in the second-stage filter. On the arrival of a positive decision and the index of the time-bin with potential tracks from the second-stage filter, the tracking engine reads data from the respective buffers. The tracking engine is an HLS component written in C++ capable of performing a simple reconstruction of tracks in 2D space based on the position of anodes in the vertical straw layers (no drift time information is used). In the first stage, hits in every detector layer are grouped into 'clusters' and 'hit-pairs'. A 'cluster' is a group of hits in an individual straw layer while a 'pair' is a set of two neighboring straws in a cluster. The ambiguity in track position is later solved by selecting the combination of these hit-pairs with the least χ^2 obtained from the linear fit. The parameters of this track are the derived feature from the data and are described by a 64-bit word. Multiple tracks in a time-frame can be simultaneously reconstructed by using an instance of the tracking engine for every time-bin. This tracking component for the FT data consists of loops and inline functions required to solve the combinations. The number of track combinations to solve in a time-bin can vary. Due to this, the component has a variable latency of $\approx$6 μs. As the tracks appearing in a time-frame are sorted into time-bins, a tracking engine interfacing these time-bins has abundant time (length of the time-frame) to complete the task until another track appears in the same time-bin in the upcoming time-frames.

G. Resource Usage

The processing pipeline has been developed keeping in mind the scalability of the system. The resources required to implement the pipeline entirely depend on the number of data links required in the detector read-out. An instance of all the components in the processing pipeline is dedicated for each input link except for the common processing and filtering component. Projected resource consumption of a processing pipeline with 12 TRB links (FT 5,6) for both the evaluation ZCU102 board and the FPGA in the targeted DC hardware (Kintex UltraScale+ KU15P) are presented in Table III.

TABLE III
RESOURCE CONSUMPTION

Component	Block RAM Blocks	CLB LUTs	DSP Slices
Available in DC (KU15P)	984	522,720	1,968
Available in (MPSoC ZCU102)	912	274,080	2,520
Receiver, Clocks, Geo-parser, first-stage filter, buffers, transmitter (for 12 links)	846	106,500	1512
Processing component (second-stage filter)	0.0	8565	0.00
Track buffers + tracking engine	0.5	9624	6

H. Test Scheme

A generic testing and development scheme has been realized for the evaluation of the pre-processing system. The raw data to be processed can be streamed to the processing module directly from the TRB via GbE links but requires an active detector and the DAQ system ready for operation. An alternative approach is to convert the pre-collected raw data from the TRB to the adequate binary format and stream it to the pre-processing module to mimic the data from the detector. In both cases, the processed data output and the features extracted by the module are sent back to the PC. The same raw data can be analyzed by the offline analysis software to benchmark the performance of the processing module by comparing the results. It can also be sent to the FPGA simulator for further development of the pre-processing components under realistic conditions, hence completing the full development and testing cycle.

IV. Evaluation of the system

The quality of the features extracted by the module could be evaluated by comparing the data processed by the hardware against the results obtained using offline software data analysis tools as described in Sec.III-H. For this purpose, the pre-processing pipeline has been extensively tested with the data streams from in-beam experiments and radioactive sources obtained with detector prototypes. For the test of the system under beam conditions, the detector prototype (described below) was exposed to proton beams with momenta of 3 GeV/c and with a peak intensity of ≈20 kHz at the COSY, Forschungszentrum Jülich (Germany) accelerator facility in early 2019. The gathered data were digitized and streamed to the hardware. For the test of the system with a radioactive source, the pre-processing system was connected directly to the detector read-out (TRB).

A. Test setup: Detector and readout

The FT prototype used in tests was built with eight double layers of straws of length 125 cm, each having one module i.e 32 straws. The layers were arranged in 0°, 5°, -5°, 0°, 0°, 5°, -5°, 0° inclination to the vertical axis respectively. A 7.5 $\times$ 7.5 cm^2 wide plastic scintillator detector was placed before the straw modules and was used for the reference time (T0)[18].

The TRBv3 based master-slave architecture was used to read-out the detector FEBs as the TRB5sc hardware was in the stage of development during this period. A master TRBv3 with the Central Trigger System (CTS) [7, 19] firmware was used to emulate SODANet time-frames for the two slave TRBv3. One of the slave TRBv3 contains four 48 channel TDCs. The second TRBv3 contains three 48 channel TDCs and an FPGA configured to read-out scintillator signals. The collected data were sent to the Zynq UltraScale+ MPSoC ZCU102 hardware configured with the pre-processing pipeline using a Gigabit Ethernet (GbE). The read-out system is shown in Fig. 7. The output links of the pre-processing hardware were connected to the event-builder PC to store the data and control the system. The detector was read-out in a continuous mode with time-frames of 50μs and the processing component was configured with 127 time-bins of width 1.2μs each (320 ns per time-cell). The track parameters extracted by the pre-processing pipeline were stored in the same PC and were synchronized with the raw TRB data using the trigger number.

B. Results

The processing system identifies the time-frame that contains tracks and filters out the non-track time-frame. The pre-processing pipeline stage was operated in the ”marking mode” which simply marks the time-frame that contains tracks while retaining all the other time-frames in the data output. This allowed us to compare these marked and unmarked time-frames with the track reconstruction performed by the off-line analysis. In an ideal case, all the time-frames from which the tracks are extracted by the off-line analysis must be marked by the pre-processing pipeline and the excess marking of time-frames due to wrong identification must be zero.

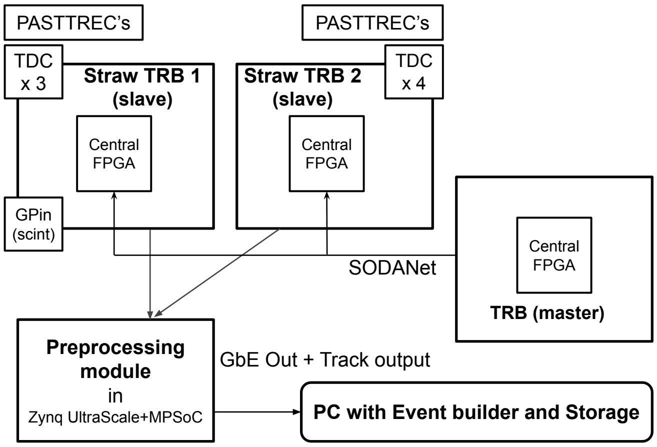

Fig. 7. A schematic representation of the DAQ for the test setup showing the connection between PASTTREC front-end-boards, TRB and pre-processing board.

The collected data contained 3.9 million time-frames. They were analyzed by the offline software and tracks were reconstructed from 23.07% of these time-frames. When the same data was streamed through the pre-processing pipeline, 25.54% out of the 3.9 million time-frames were marked by the hardware as potential track frames. Over 99% of time-frames found in the off-line analysis are also marked by the pre-processing hardware. Less than 0.01% of the time-frames were not marked by the hardware due to the errors in the TDC that corrupted a single data word in the stream, resulting in the rejection of this hit belonging to the track in the first-stage filter, but could be recovered in the software analysis. A surplus of 2.47% of time-frames not found by the offline analysis was marked by the hardware, as it does a more lenient analysis of the data compared to the offline (example: stricter time-coincidence with the reference detector). This marking efficiency of over 99% proves the track recognition capability of the pre-processing pipeline. Being able to efficiently extract this information from the data stream in real-time will prevent all the meaningless data from entering the network.

The system was also tested with detector data from radioactive sources and cosmic rays, where the pre-processing hardware was streamed with data directly from the detector read-out and the marking efficiency remained the same, proving the stability of the system irrespective of the source from which the data are streamed.

The data reduction capability of the pre-processing system depends entirely on the environment (mainly radiation intensity) to which the detector is subject to. For the conditions in which the system was tested, around 74.54% of time-frames were rejected when the detector was exposed to the proton beam and over 99% rejected when exposed to radioactive source/cosmic rays due to empty frames. The rejected data for track-less time-frames are mostly empty data packets with the essential header and data from a few co-incidental detector hits. These data packets still constitute 35% of the data volume in the case of the beam data and over 99% in the case of radioactive source/cosmic rays.

The second feature extracted (derived feature) from the data using the tracking engine in the pre-processing pipeline (hardware), i.e the parameters of the track are compared with the one obtained from the offline analysis (software). The tracks were straight lines (y = mx + C) inside the detector geometry and the calculated track parameters (offset and the slope) of these tracks for both offline and hardware-based track reconstruction are presented in Fig. 8. The track parameters presented for tests of the system under cosmic rays cover a larger angular range as expected. The parameters of the track show a very high agreement between the software and hardware-based tracking for both perpendicular and inclined tracks as seen in the difference plot (Fig. 8c, Fig. 8d). The slight difference observed in the parameters of the track is due to accidental coincidences that the tracking engine implemented in hardware could not resolve. This can be mitigated by narrowing the width of the time-bin and introducing additional filtering conditions. As the efficiency of the system is adequate, streaming just the extracted features instead of both data and the feature would significantly reduce the data volume in the future.

These track parameters can be also used to determine the origin of a particle track in an experimental setup. By using this selection, only data for the tracks originating from the interaction vertex can be accepted and the rest (i.e secondaries) of the data can be rejected from entering the network. This feature of the FT pre-processing pipeline can also serve as a platform for the visualization of the data in real-time and online monitoring during the experiments. This is an excellent example of a system that requires complex computing like solving combinations being easily and effectively implemented on FPGA using technologies like HLS.

Additionally, T0 and drift time (DT) are the two features that are important for the analysis of the data in the further stages of the DAQ. The mean rising time of all the hits in a track can be treated as T0 after some offset subtraction and can be used to calculate the DT. Tests of the detector under beam condition show that the DT calculated using this method is similar to the value obtained using a reference detector like a scintillation detector as shown in Fig. 9. This information can be extracted instantaneously in the second-stage filter stage of pre-processing. Collectively, all of these results can make up a self-reliant read-out that can provide the OPN with refined FT data.

V. Optimization prospects

The time measurement component in the TRB produces a 32-bit word for every hit in the detector containing channel number, rising time (relative to the super-burst update time) and TOT [20]. With the super-burst update frequency of 26 kHz (time-frame width = 38.46 μs), it requires at least 17 bits (0.5 ns/bit) to represent the rising time of a hit. Rather than transmitting the rising time for every hit in an event, it is sufficient to transmit a single reference time (T0) for an event and the rising time of the hit with respect to this T0, which is also the DT. DT is a small number (≈180 ns) that can be represented in 9 bits (1 bit = 0.5 ns), allowing a hit to be represented in just 24 bits. This encoding would reduce the input data rate of the FT by 25%. Due to the efficient isolation of events in the time-bins of the second-stage filter, T0 and DT can be easily obtained. Furthermore, Straw Tube Tracker (STT) will be another detector in PANDA, constructed using similar straws and read-out components, that can benefit from this pre-processing pipeline.

From simulation studies for PANDA, it is known that tracks from secondary particles produced by the interaction

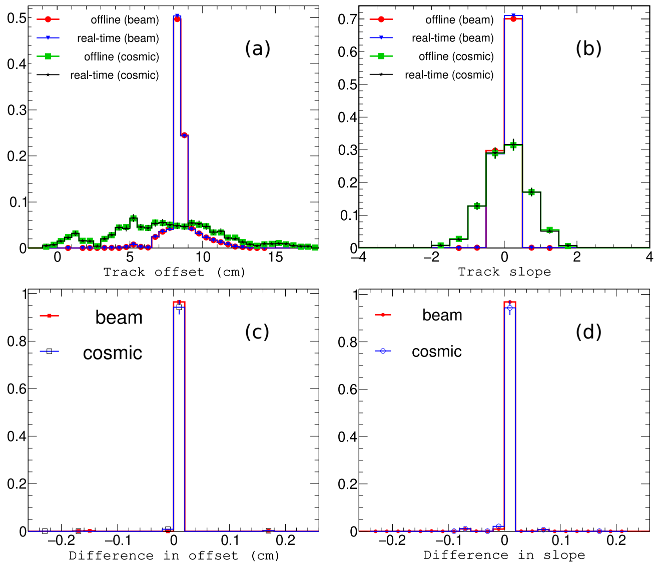

Fig. 8. The parameters of the tracks obtained from in-beam and radioactive/cosmic ray measurements. Distribution of the track offset (C) (a) and distribution of slope (m) (b) for the tracks reconstructed in the pre-processing hardware using offline software analysis for the in-beam data (black), using real-time pre-processing for the in-beam data (red), using offline software analysis for the radioactive source/cosmic rays (green) and using real-time pre-processing for the radioactive source/cosmic rays (blue). Difference between offline and real-time values of offset (c) and track slope (d) normalized to 1.

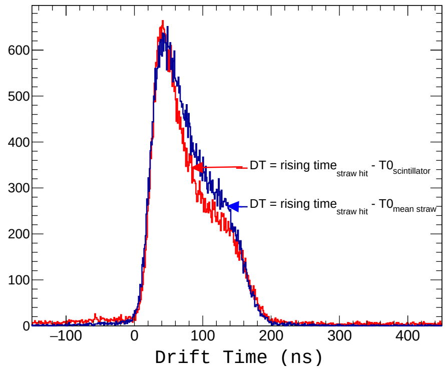

Fig. 9. Drift time (DT) calculated using the reference time T0 from a scintillation detector placed behind the FT (blue) compared to the DT calculated using the mean rising time of the hits in a track (red) in the offline analysis, presented after offset correction.

of particles with the elements of the PANDA detection system amount to 1.5 times the primary tracks [3] and the goal is to reject these tracks. The average time distance between hits in FT from two events is ≈200 ns (High-Resolution Mode). As the hits are sorted into the time-bins, by instantiating multiple tracking engines, multiple tracks in a time-frame can be reconstructed simultaneously. The reduction in data rates due to the rejection of tracks from secondary interactions is expected to be significant. Furthermore, the features extracted from the pre-processing pipeline (T0, DT, and track parameters) can be stored in some buffer to be later utilized by the OPN and will reduce time required for synchronous processing. The future development of the project includes the development of a tracking engine capable of reconstructing tracks in 3D space and study the data reduction by the rejection of secondary tracks.

VI. Summary and Outlook

The described pre-processing pipeline provides a promising concept of a fully integrated solution for FT and STT detector data processing in real-time. The PANDA experiment will use TRB-based read-out in several detector systems and the DAQ architecture has been designed keeping in mind the online and real-time data processing requisites. As the described pipeline is modular and expandable, it presents a high potential of use in TRB based readout systems, in PANDA or other experiments.

The tests conducted on the developed pre-processing pipeline have shown excellent event recognition and track reconstruction efficiency from the data stream in real-time. This proves the effectiveness of the technique that is used. The development of complex computing and data processing on SoC devices will assist the real-time particle tracking, event filtering, data reduction, and online monitoring for the FT in the future. The pre-processing pipeline will be an essential and easily adaptable extension of the TRB ecosystem performing data processing required for both the DAQ and the end-user of the data.

Acknowledgment

The TRB platform and the accompanying firmware and software are developed by the TRB Collaboration (www.trb.gsi.de). The project could be realized thanks to the accelerator team at COSY Forschungszentrum Jülich, Germany who provided a high quality and stable proton beam. The project could be realized thanks to the support of the Xilinx University Program and its donations.

References

[1] Guenther Rosner. “Future facility: FAIR at GSI”. In: *Nuclear physics B-Proceedings supplements* 167 (2007), pp. 77–81.

[2] *PANDA Experiment Website*. [Online]. Available: https://panda.gsi.de/.

[3] Panda Collaboration et al. “Technical Design Report for the: PANDA Straw Tube Tracker”. In: *arXiv preprint arXiv:1205.5441* (2012).

[4] Paweł Strzempek. “Development and evaluation of a signal analysis and a readout system of straw tube detectors for the $\bar{P}$ANDA spectrometer”. PhD thesis. PhD thesis (Jagiellonian University, Krakow, Poland, 2017), 2017.

[5] W Erni et al. “Physics performance report for PANDA: strong interaction studies with antiprotons”. In: *arXiv preprint arXiv:0903.3905* (2009).

[6] Cahit Ugur et al. “Implementation of a high resolution time-to-digital converter in a field programmable gate array”. In: *Proc. Sci,(Bormio2012)* 15 (2012).

[7] G Korcyl et al. *A users guide to the trb3 and fpga-tdc based platforms*. [Online]. Available: http://jspc29.x-matter.uni-frankfurt.de/docu/trb3docu.pdf. 2015.

[8] Grzegorz Korcyl. “A novel data acquisition system based on fast optical links and universal readout boards”. PhD thesis. AGH-UST, Cracow, 2015.

[9] M Traxler et al. “A compact system for high precision time measurements ($<$ 14 ps RMS) and integrated data acquisition for a large number of channels”. In: *Journal of Instrumentation* 6.12 (2011), p. C12004.

[10] J. Michel et al. “Electronics for the RICH detectors of the HADES and CBM experiments”. In: *Journal of Instrumentation* 12.01 (Jan. 2017), p. C01072. DOI: 10.1088/1748-0221/12/01/c01072.

[11] Carsten Schwarz et al. “The Barrel DIRC detector of PANDA”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 936 (2019), pp. 586–587.

[12] Andreas Neiser. “Current Status and Performance of the Crystal Ball and TAPS Calorimeter”. In: *J. Phys. Conf. Ser.* 587.1 (2015), p. 012041. DOI: 10.1088/1742-6596/587/1/012041.

[13] R Achenbach et al. “The ATLAS level-1 calorimeter trigger”. In: *Journal of Instrumentation* 3.03 (2008), P03001.

[14] J. Smyrski et al. “Pressure stabilized straw tube modules for the PANDA Forward Tracker”. In: *JINST* 13.06 (2018), P06009. DOI: 10.1088/1748-0221/13/06/P06009.

[15] D. Przyborowski et al. “Development of a dedicated front-end electronics for straw tube trackers in the $\bar{P}$ANDA experiment”. In: *JINST* 11.08 (2016), P08009. DOI: 10.1088/1748-0221/11/08/P08009.

[16] *The TRB family of FPGA based readout boards*. [Online]. Available: http://trb.gsi.de/.

[17] Xilinx. *Vivado Design Suite User Guide: High-Level Synthesis (UG902)*. 2019.

[18] A Malige et al. “Development of Forward Tracker”. In: *Journal of Physics: Conference Series*. Vol. 1667. 1. IOP Publishing. 2020, p. 012028.

[19] Jan Michel et al. “The HADES DAQ system: trigger and readout board network”. In: *IEEE Transactions on Nuclear Science* 58.4 (2011), pp. 1745–1750.

[20] M. Kavatsyuk et al. “Technical Design Report for the: $\bar{P}$ANDA Data Acquisition and Event Filtering”. In: (Jan. 2021).