Design and Evaluation of an FPGA Online Feature Extraction Data Pre-Processing Stage for the CBM-TRD Experiment

Cruz de Jesús García Chávez and Udo Kebschull

***Abstract*—Feature extraction is a data pre-processing stage of the Transition Radiation Detector (TRD) data-acquisition chain (DAQ) as part of the Compressed Baryonic Matter (CBM) experiment. The feature extraction stage delivers event-filtered and bandwidth-reduced data to the First Level Event Selector (FLES). The feature extraction stage implements multiple processing algorithms in order to find and extract regions of interest within time series signals. Algorithms such as peak-finding, signal integration, center of gravity and time-over threshold were implemented for online analysis. On the other hand, a local clustering algorithm allows to find cluster members and to implement even further data reduction algorithms. A feature extraction framework for automatic firmware generation has been tested for the CBM-TRD data acquisition chain. The framework allows the generation of Field Programmable Gate Array (FPGA) designs that implement feature extraction algorithms. Such designs are FPGA-platform independent and are described by a file written in a Domain Specific Language (DSL). The result of using the mentioned feature extraction framework for the TRD feature extraction stage is presented and discussed.**

I. Introduction

Feature extraction is a common problem in the data acquisition of high-energy particle experiments. The design and evaluation of Hardware Description Languages (HDL) based designs can be a time consuming task. Because of the complexity of the required processing algorithms for feature extraction, firmware development can take up to a couple of years until a fully functional production design is up and running. Moreover, the HDL designs tend to be made in a way to solve very specific problems and cannot be implemented in different hardware platforms nor can they use different front-end electronics without adaption of the design to the new hardware platform. Constraints like data-format, transport links and overall resource management make reusing and maintaining HDL designs a difficult task for a firmware designer. According to the problem at hand, a previously presented feature extraction framework for FPGA firmware generation has been evaluated [1], where by a high-abstraction description file an FPGA design can be generated without the necessity of writing a single line of HDL code. The automatically generated designs have been used for the TRD feature extraction processing stage.

Manuscript received June 3, 2016.

This work was supported by the German Federal Ministry of Education and Research No. 05P12RFFCM.

C. de J. García Chávez and Udo Kebschull is with the Infrastructure and Computer Systems for Data Processing (IRI) at the Goethe University Frankfurt am Main, Germany (e-mail: garcia@iri.uni-frankfurt.de, kebschull@rz.uni-frankfurt.de).

978-1-5090-2014-0/16/$31.00 2016 European Union

II. The Compressed Baryonic Matter (CBM) Experiment

The Compressed Baryonic Matter (CBM) experiment is a high-energy physics experiment that explores the QCD phase diagram in the region of high baryon densities using high-energy nucleus-nucleus collisions. The experiment is part of the International Facility for Antiproton and Ion Research (FAIR). FAIR is located in Darmstadt, Germany, and is currently under construction. The CBM experiment is conformed by several detectors arranged in two main configurations. A muon detector setup and an electron detector setup. Depending on what is required to be measured, certain detectors can be arranged in different configurations as illustrated in figure 1. The CBM data acquisition chain expects high even rates of about 10 MHz and a total data rate of 1 TB/s.

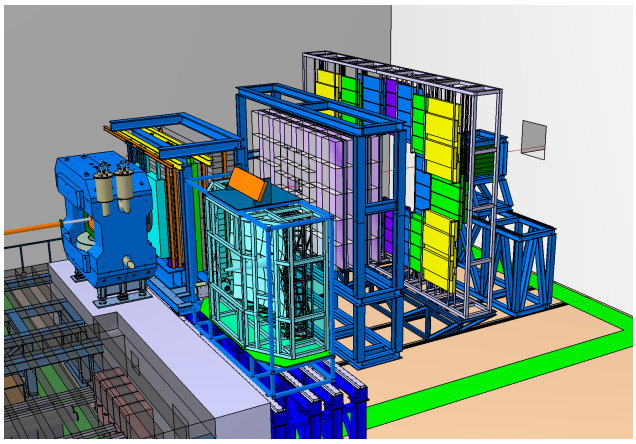

Fig. 1. The CBM detector setup for the SIS100

A. The Transition Radiation Detector (TRD)

The Transition Radiation Detector is one of the 8 sub-experiment detectors of the CBM experiment. It consist of one station with four layers of detectors (SIS100 configuration). The TRD aims to track particles with a position resolution of σ = 300-500 μm along the pad as well as for electron identification for a momenta p $\geq$ 1 GeV/c and a pion suppression of 100 at 90% electron efficiency.

B. Front-End Electronics

The TRD detectors are read-out by the Self-Triggered Pulse Amplification and Digitization ASIC (SPADIC) chip. The latest version of the ASIC, the SPADIC 1.0 [2] which has been implemented in the UMC 180 nm process, provides features such as 32-channel charge-sensitive amplifiers, free-streaming mode, forced neighbor readout and self-timestamped messages.

III. Feature Extraction

A. Concept

Feature extraction algorithms select and extract relevant set of features for a given input vector. In this work, an input vector is defined as a discrete time-series. As a definition, discrete time-series is a sequence of data points made over a discrete time interval. For an input vector, multiple features can be extracted by applying different algorithms over the same input vector. Known algorithms in the field of high-energy physic experiments are Time-Over-Threshold (ToT), Center of Gravity (CoG) and Peak Finder, just to mention a few of them. Feature extraction algorithms provide a dimension reduction of the input vector and the resulting feature vector contains a representation (usually in a higher abstraction) of the input vector with informative and non-redundant features.

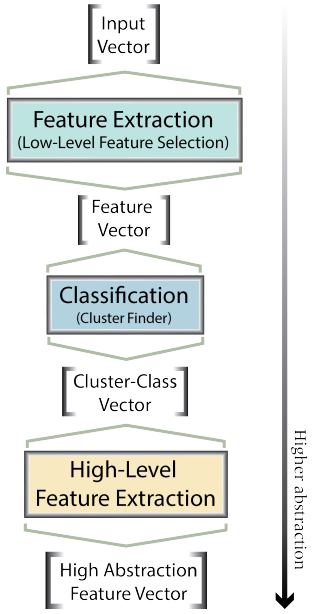

Fig. 2. Classification of feature extraction algorithms

As shown in figure 2, feature extraction algorithms can be classified as follows:

1 Feature Extraction:
These algorithms extract the best set of features for dimension reduction while maintaining their discriminatory information as much as possible.
2 Classification:
Classification algorithms assign a class to a feature vector. Common algorithms for classification are cluster finder algorithms.
3 High-Level Feature Extraction:
Aims to perform a further dimension reduction of a given class of feature vectors.

B. The Framework - Overview

A feature extraction framework has been developed and presented for automatic FPGA firmware generation [1]. The details about the framework implementation, performance and results are beyond the scope of this work. Nevertheless, the presented framework has been used to generate FPGA designs to be used under the context of the TRD feature extraction. A simplified workflow overview of the framework is shown in figure 3.

The main features provided by the feature extraction framework are:

1 Feature extraction designs are described in a Domain Specific Language (DLS) file, specifically developed for the framework. The DSL allows the designer to:
    1 Instantiate feature extraction algorithms called “Cores”.
    2 Define a network called “Stream Network”. This defines the interconnection among the selected “Cores”.
    3 Declare a “Stream Model” for the input stream-messages. This defines the structure in which the Front-End Electronics wraps its messages.
    4 Finally, select a target FPGA platform. If the platform is not defined in the framework, the designer can opt to define a new one.
2 New “Cores” can be declared by the user, written in VHDL or Verilog.
3 Automatic data stream handling. Messages from Front-End Electronics usually contain both metadata words (which provides information like timestamp or fired channel) and payload words. Therefore, the required logic for stream processing and buffering is automatically generated by the framework.
4 Internal stream transport standarized to AXI4-Streamer interface.
5 System Manager: clock manager, FIFOs as well as input/output pins of the FPGA fabric are automatically managed by the framework.
6 Generation of synthesis projects for Xilinx tools (ISE and Vivado). A feature extraction project is defined in a DSL file and therefore a synthesis project is generated for every file.

IV. CBM/TRD Readout Architecture

The CBM-TRD readout architecture is illustrated in figure 4. Particles generated after the collision pass through the TRD detector. While the particle is crossing the detector, a charge around the particle trajectory is induced on the neighbor pads. This induced charge is then read-out by the Front-End Electronics. This analog signal is digitized and sent to a concentrator, where data from multiple front-end electronics are merged together.

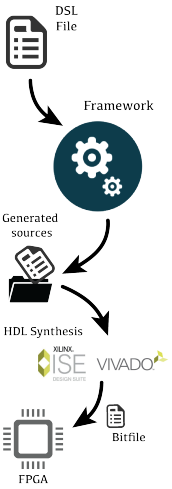

Fig. 3. Workflow overview of the feature extraction framework

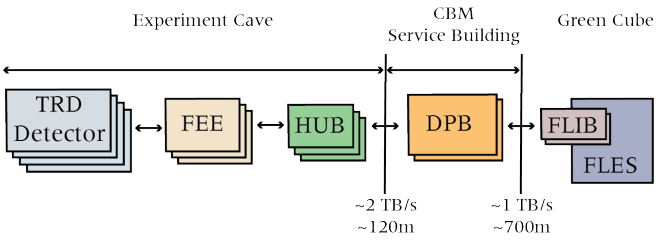

Fig. 4. CBM-TRD readout architecture

The Data Processing Board (DPB) implements different algorithms. First of all, it is in charge of readout the SPADIC 1.0. It also sorts messages in time and finally implements feature extraction algorithms for data reduction. Finally this set of vectors are sent to the First-Level Event Selector where offline analysis is performed.

The CBM data acquisition chain expects to produce around 2 TB/s of data before the Data Processing Board and deliver around 1 TB/s of data to the FLES.

V. Experimental Design

A data acquisition chain has been installed at the Infrastructure and Computer Systems for Data Processing (IRI) research group of the Frankfurt University. This laboratory setup emulates the final detector setup expected for the CBM-TRD acquisition chain.

The feature extraction firmware has been developed on the feature extraction framework and tested on a SysCore3 FPGA board. The SysCore3 is a Spartan-6 based FPGA universal read-out controller and data processing board developed at IRI [3]. The design contains two data transfer links called CBMnet 2.0. These links provide deterministic latency messages as well as retransmission logic. The SysCore3 allows to interface up to three SPADIC 1.0 front-end electronic boards and an optical link based on CBMnet 2.0 to a First Level Interface Board (FLIB) installed in a PC.

The FPGA firmware architecture is illustrated in figure 5. Data transmission links CBMnet 2.0 send and receive both control and data transport messages from back end to front end. The feature extraction block was instantiated after data from the SPADIC 1.0 FEEs had been gathered together.

The processing "cores" inside the feature extraction block are explained below:

1 Pre-processing:
Signal conditioning algorithms are implemented here, such as overflow detection, pedestal reduction and baseline correction.

2 Feature Extraction:
Feature extraction algorithms like Time-Over-Threshold, Center-of-Gravity and Peak-Finder are instantiated here.

3 Cluster Finder:
Messages generated by neighbor pads in the TRD detector are included to a cluster class according to a given discriminator like timestamp, cluster size or similar center of gravity.

4 Post-processing:
Finally, cluster-class members can be merged in a higher abstraction class, called "Hit Class".

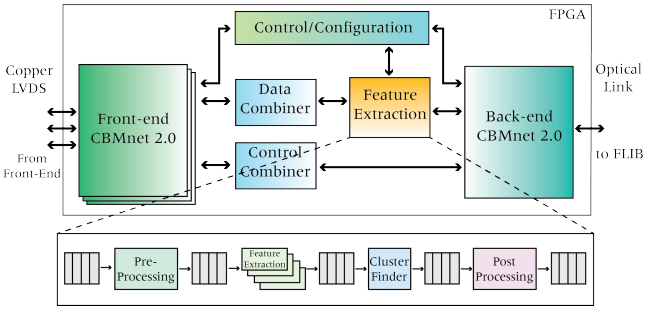

Fig. 5. Architecture of the feature extraction firmware

VI. Outline and Future Work

A feature extraction design has been developed and tested for the CBM-TRD data acquisition chain based on the feature extraction framework. This framework allows the automatic HDL code generation for feature extraction algorithms without the necessity of developing HDL code.

The feature extraction firmware design has been evaluated during a CERN-SPS (2015) and CERN-PS (2014) beam test runs. For the upcoming years (2016 and 2017) it is necessary to integrate the new read-out components to the DAQ, like the new Data Processing Board, namely AFCK, based on a Xilinx Kintex-7 FPGA. Together with a release by the end of the year 2016 of the SPADIC 2.0, which now includes a GBTx data transport link, it would increase the number of FEE boards that can be processed by a single DPB. These upgrades would allow the evaluation of the current feature extraction design for larger setups.

References

[1] C. de J. García Chávez et al., "Status update of the feature extraction framework for automatic FPGA firmware generation", CBM Progress Report 2015

[2] T. Armbruster., "SPADIC - a Self-Triggered Detector Readout ASIC with Multi- Channel Amplification and Digitization". PhD thesis, ZITI, Heidelberg University, 2013.

[3] J. Gebelein et al., "SysCore3 A universal Read-Out Controller and Data Processing Board",CBM Progress Report 2012