Particle identification and tracking in real time using Machine Learning on FPGA

F. Barbosa,[a], L. Belfore,[b], C. Dickover,[a], C. Fanelli,[a,c], S. Furletov, [a], Y. Furletova,[a], L. Jokhovets,[d], D. Lawrence,[a], and D. Romanov[a]

[a]Jefferson Lab, U.S.A.
[b]Old Dominion University, U.S.A.
[c]William & Mary, U.S.A.
[d]Juelich Research Centre, Germany

July 25, 2022

Project ID:

Project Name Particle identification and tracking in real time using Machine Learning on FPGA

Contact Person: Sergey Furletov (furletov@jlab.org), D. Romanov (romanov@jlab.org)

Abstract

This project is a multi-disciplinary endeavour between Physics, Electrical Engineering, and Computer Engineering. The purpose is to develop and implement an FPGA(*) based Machine Learning algorithm for real-time particle identification, filtering, and data reduction. This is important research that can be applied to streaming readout systems being developed now at JLab and other facilities. Real-time data processing is a frontier field in experimental physics, especially in HEP. The application of FPGAs at the trigger level is used by many current and planned experiments (CMS, LHCb, Belle2, PANDA). Usually they use conventional processing algorithms. LHCb has implemented ML elements for real-time data processing with a triggered readout system that runs most of the ML algorithms on a computer farm and is using the Allen system which does much of the work on GPUs. The project described in this proposal aims to test the ML-FPGA algorithms for streaming data acquisition. There are many experiments working in this area and they have a lot in common, but there are many specific solutions for detector and accelerator parameters that are worth exploring further. We propose evaluating the ML-FPGA application for a full streaming readout and the first target is EIC experiment. The first goal is particle identification (e, π, μ) using multiple detectors (CAL, GEMTRD, GEM Trackers) in real time using neural networks on FPGA. The results of this project would be useful for other experiments worldwide, especially in nuclear physics, such as EIC, SoLID, PANDA (FAIR), etc.

(*) Field Programmable Gate Array

1 Introduction

With the increased luminosity of accelerator colliders, alongside an increased granularity of detectors for particle physics, more challenges fall on the readout system to transfer data from front-end detectors to the computer farm and long term storage. Modern data acquisition systems (LHC, KEK, Fair) employ several stages for data reduction. The CMS experiment at LHC has a Level 1 trigger that makes a decision in $\sim 4\mu s$ and rejects 99.75% of events. Their High Level Trigger (software), makes a decision in 100 ms, and rejects 99% of the data from Level 1. Modern concepts of trigger-less readout and data streaming will produce a very large data volume to be read from the detectors. Most of this will be uninteresting and ultimately discarded. Handling this large volume using traditional means would require either a huge farm for real time processing, or a very large volume of data stored on tapes. From a resource standpoint, it makes much more sense to perform both the pre-processing of data and data reduction at earlier stages of acquisition.

The growing computational power of modern FPGA boards allows us to add more sophisticated algorithms for real-time data processing. Some tasks, such as clustering and particle identification, could be solved using modern Machine Learning (ML) algorithms which are naturally suited for FPGA architectures.

While the large numerical processing capability of GPUs is attractive, these technologies are optimized for high throughput, not low latency. FPGA-based filters and data acquisition systems have extremely low, sub-microsecond, latency requirements that are unique to particle physics. Machine learning methods are widely used and have proven to be very powerful in particle physics. However, the exploration of such techniques in low-latency FPGA hardware has only just recently begun.

ML particle identification (PID) methods can be applied individually for various subdetectors such as: RICH, DIRC, calorimeters, dE/dx in tracking detectors, transition radiation detectors (TRD), etc. By combining data from all subdetectors it is possible to provide global particle identification. This takes into account the responses of all subdetectors and provides better particle information for physics analysis in real time. It also allows for the filtering of data based on the topology of physics events and to control data traffic based on physics.

Real-time reconstruction and identification of particle tracks can help suppress background and single photon noise in RICH detectors, especially in the case of SiPM readout. EIC data traffic estimation shows RICH detector dominance by 2-3 orders of magnitude (1.8 Tb/s) compared to other detectors due to single photon noise.

2 Expected Results

This is an interdisciplinary R&D project that requires efforts from physicists, computer engineers, and electrical engineers who have expertise with FPGAs. The goal is to develop and build a functional demonstrator for FPGA Machine Learning applications, described here as the ML-FPGA. The ML-FPGA project will be used to identify and optimize artificial neural network algorithms and topologies suitable for real time FPGA applications.

It will also be used to perform beam tests in Hall-D with the GEM-TRD and calorimeter prototypes. They will be used as PID detectors to estimate the performance of ML on an FPGA in a real-time environment. Test results will be used to calculate resource scaling for planned large scale experiments (EIC, SOLID, etc). The performance results and price will also serve as a feasibility study for building a larger scale ML-FPGA selector/filter for current experiments such as CLAS12 and/or GlueX.

Project Goals:

1. design and build a functional demonstrator ML-FPGA in which the particle identification and tracking ML algorithm runs on FPGA, for testing various ML algorithms.

2. evaluate the performance, efficiency, and resources used by ML-FPGA compared to other solutions.

3. evaluate scalability of the ML-FPGA to the future experiments (EIC,SOLID,PANDA, e.t.c.)

4. use the results in decision of building the ML-FPGA with higher performance for some running experiment (GlueX,CLAS12)

5. EIC SRO/DAQ has options for multiple stages of data aggregation and reduction based heterogeneous hardware solutions that also includes FPGAs, so the system can be used to optimize and evaluate the performance of various hardware solutions.

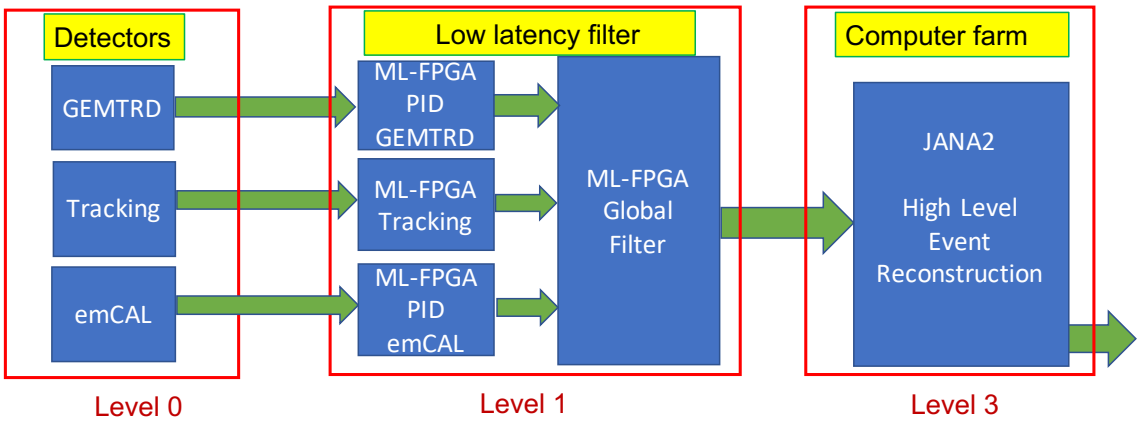

Figure 1: Data processing chain.

3 Proposal Narrative

To demonstrate the operating principle of the ML FPGA, and estimate the performance of the ML-FPGA, we propose using the input data from existing detectors. The detectors used for ongoing EIC R&D projects are the "GEM based Transition radiation detector (TRD) and tracker" ([2]), a prototype calorimeter ([3]) and GEM tracker. Currently, a small 10x10 cm GEM-TRD prototype is being readout with several fADC125s [4] and can generate up to 18 GB/s of raw data traffic. This detector, in addition to a track coordinate (μTPC mode), is capable of electron identification or electron/hadron separation. This is highly important for EIC physics. The size of the calorimeter prototype is 3x3 cells and is read out by an FADC250. For the GEM-TRD project we already use offline Machine Learning tools (JETNET, ROOT-based TMVA) [5] ,[6]. The results of which can be used for validating the proposed implementation of FPGA-based neural networks and to discover potential FPGA-based Machine Learning algorithms for real-time systems. A FPGA-based Neural Network application would offer real-time, low latency, particle identification. It would also allow for data reduction based on physical quantities during the early stages of data processing. This will allow us to control data traffic and offers the possibility of including detectors with PID information for online high-level trigger decisions, or online physics event reconstruction.

To start this project we plan on using a standard Xilinx evaluation board to test the ML algorithms, rather than develop a custom FPGA board. These boards have the functionality and interfaces sufficient to provide proof of principle for the ML-FPGA. This will significantly speed up the work and gives us the freedom to choose the type of FPGA that we find best suited for ML applications while we work on optimization. FPGA platforms are a good solution for achieving online real-time processing for several key reasons. First, current FPGA technology offers massive raw computational performance. The proposed Xilinx evaluation board includes the Xilinx XCVU9P which has 6,840 DSP slices. Each slice includes a hardwired optimized multiplication unit and collectively offers a peak theoretical performance in excess of 1 Tera multiplications per second. Second, the internal layout can be optimized for a specific computational problem and can remove any irrelevant elements in the chain during compilation. The internal data processing architecture can support deep computational pipelines offering high throughputs. Furthermore, many ML algorithms can be mapped to make very effective use of FPGA resources. Third, the FPGA supports high speed I/O interfaces including Ethernet and 180 high speed transceivers that can operate in excess of 30 Gbps.

Another important part of the project is evaluating the advantages of a "global PID" compared to the standalone PID from each detector. To test the global PID performance, we propose using a setup with two detectors: the EIC calorimeter prototype (3x3 modules) and a prototype of the

GEMTRD. Preprocessed data from both detectors, including a decision on the particle type, will be transferred to another ML-FPGA board with a neural network for global PID decision.

Initial beam testing is planned in Hall D, where there is already a test beam site that can be used for testing the prototype GEM-TRD, ECAL, and Modular RICH detectors. This part of the work depends on the availability of the beam, but can be done parasitically while GlueX is running. In order to test the performance of the system as a PID, we plan to test it on an external facility that provides an electron and hadron beam (FermiLab/CERN). Depending on the performance of the ML-FPGA demonstrator, one might consider building a full scale filter/selector for current and planned experiments.

4 Methods

The anticipated hardware platform will use high performance Xilinx devices. The candidate products include the new Xilinx VersalTM series adaptive compute acceleration platform (ACAP) platform along with the XCVU9P. In addition to providing FPGA programmability, the Versal platform includes "intelligent engines" that are very long instruction word (VLIW) single instruction multiple data (SIMD) processors that can be programmed to accelerate ML/AI computations.

The system will be able to receive data from any front-end board with a fiber interface. But, for the GEM/TRD use case we will be using the prototype SRO125 (currently being manufactured). The VME version of this board, the fADC125, currently provides processed data for the offline ML system described previously. The streaming version (SRO125) will allow for an apt comparison of online and offline results. The SRO125 runs a 16 bit bus at 125 MHz with a 2.5 GB/s transceiver. The interface between the SRO125 and the development board will utilize a custom serial protocol with a fixed latency (described in attachment 1). The fixed latency protocol will allow for a synchronized clock to be recovered and used on all front end boards and for embedded control signals to arrive deterministically. This interface model has been used for both the Hall B RICH detector and the Hall D DIRC. For this project the event building portion will be modified to provide data to the ML block efficiently and will be organized in a manner useful for the algorithm (noted in figure 1 as high speed interface logic).

For the initial hardware implementation we will use a triggered system. A trigger can be received on the SRO125 from either an input connector or from the fiber interface itself. A self-triggering mode has also been developed, but will require additional logic for trigger supervision that is currently handled by a separate trigger module. In order to support validation of the data processing hardware and other types of analysis, a passthrough mode is implemented where the readout from downstream input are combined with the inferences. A separate FPGA application will be implemented for each detector. Aggregate detection decisions will be made with a global ANN which will receive data and their respective inferences. In addition, results from the global ANN can be used to control the data volume of passthrough data.

Considering the FPGA architecture, after receiving the data and unpacking it, the event trigger identifies events of interest and passes the information to the data clustering module. After clustering, the data is passed to the neural network which generates the inference. To the right, the embedded processor sets the configuration for processing the data and monitors progress the data processing. The embedded processor is otherwise not directly involved with the data processing. The embedded microcontroller coordinates a separate diagnostic mode where results can be sampled and validated separately.

Because of the required high data rates, the modules will be implemented at the register transfer language (RTL) level so that state machine operation controlling the data processing for each module can be optimized. In addition, pipelining will be used extensively so that throughput can be maintained because of inference latencies. Furthermore, FIFOs will be deployed where elasticity is necessary due to the occurrence of burst data or as required to cross clock domains. Figure 2 shows data flow in the experiment. Green arrows represent data streams from detectors.

The data from the detectors after pre-processing and pre-selection at the ML-FPGA are sent to the farm running the online physics event reconstruction software - JANA2 [8]. It is a modern C++ multi-threaded framework for offline and online applications, which is being used in a number of projects (GlueX, EIC (eJANA), BDX, Indra Astra and other streaming readout test stands) at Jefferson Lab and is backed by LDRD FY18-20. The data prepared by the ML-FPGA is accessed through its high-performance IO and sent to nodes via TPC, utilizing a messaging middleware. JANA will be used to disentangle the input stream. Event boundaries are determined and put

into parallel processing, where raw data is reconstructed, filtered, and recorded; incorporating a software L3 trigger functionality. One of the important possibilities made available by using JANA is subevent parallelism, which allows us to effectively run batch calculations on a GPU or TPU. In the future this will allow us to bring emerged low latency FPGAs and traditional GPU or TPU based ML algorithms together, providing an ultimate ML solution for data processing.

4.1 ML development tools

The Xilinx Vivado HLS (High-Level Synthesis) tool provides a higher level of abstraction for the user by synthesizing functions written in C,C++ into IP blocks, by generating the appropriate ,low-level, VHDL and Verilog code. Then those blocks can be integrated into a real hardware system. High-level synthesis bridges hardware and software domains and significantly decreases development time. A neural network trained in Root/TMVA can be exported to C/C++ code. The C/C++ code of the trained network, including weights, is used as input for Vivado HLS.

An offline trained neural network is usually far from optimal for an FPGA application, with its limited resources in terms of network size and computational accuracy. Neural network weights often have many zeros and double precision is often unnecessary. Thus, it is possible to reduce the size of the network by removing weights close to zero. Also, lowering the precision of floating point calculations will save resources in the FPGA.

Now there is a software package HLS4ML that helps in the design and optimization of a neural network for FPGA ([11]). HLS4ML supports common layer architectures and model software, highly customizable output for different latency and size needs, simple workflow to allow quick translation to HLS.

A typical workflow for developing a neural network in HLS4ML is shown in Figure 2.

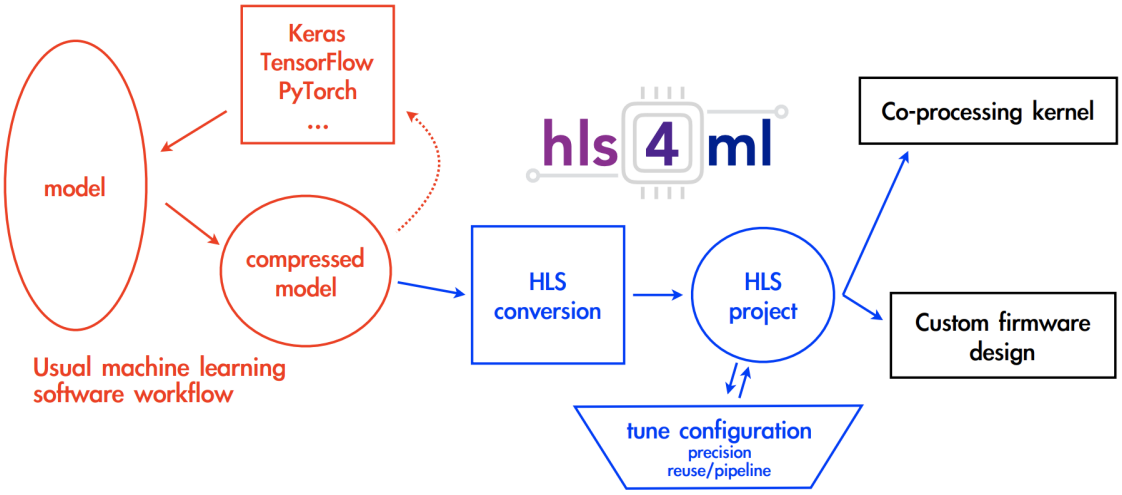

Figure 2: A typical workflow to translate a model into a FPGA implementation using hls4ml [11].

5 Resources

Project work involves conceptual development, computer simulations (Geant4, code development, testing, analysis, documentation) and will take place at JLab's CEBAF Center. The detector setup building will take a place at Hall D. The work will be carried out by JLab staff at a fractional effort (F. Barbosa, (5% FTE, electronics) C. Dickover (10% FTE, FPGA expert), S. Furletov (20% FTE, physics, ML,FPGA), Y. Furletova (5% FTE, physics), D. Lawrence (0% FTE, consulting), D. Romanov (10% FTE, software). Office space and administrative support will be provided by JLab's Physics and Fast Electronic divisions. FPGA algorithms implementation will be supported by a graduate student (ODU) at 50% FTE, supervised by the ODU Prof. L.Belfore. The graduate student will be enrolled in the graduate program at his/her university and participate in project work as part of the thesis research. Certain identified tasks will be carried out by consultants

D. Lawrence (JLAB) and C. Fanelli (W&M,JLAB, DIRC ML algorithm). The FPGA tracking algorithms consultant (L.Jokhovets) will perform her work remotely.

6 Anticipated Outcomes/Results

Outcomes/Results of this project

1. Software and hardware system to test various ML algorithms on FPGA.
2. Implemented ML FPGA PID core for GEMTRD prototype
3. Implemented ML FPGA TRACKING for GEM prototype
4. Implemented ML FPGA PID core for EmCAL prototype
5. Latency and real time performance test results of ML FPGA PID/Tracking
6. Implemented ML FPGA "global" PID using GEMTRD, EmCAL and GEM.
7. Estimation of scalability the system to the full size experiment (EIC)

7 Budget

Table 1 below summarizes the Jefferson Lab budget request for FY23.

Table 1: **JLAB:** FY23 request.

	Request	**-20%**	**-40%**
2 FPGA boards	$20,000	$20,000	$20,000
Xilinx Software License	$3,000	$3,000	$3,000
Optical cables, transceivers	$1,000	$1,000	$1,000
Development computer/workstation	$3,000	$3,000	$0
Beam Test Travel	$10,000	$0	$0
conferences/workshops	$5,000	$5,000	$0
Sub Total	$42,000	$32,000	$24,000
Overhead	$6,822	$3,822	$2,064
Total	**$48,822**	**$35,822**	**$26,064**

Table 2 summarizes the ODU budget request for FY23.

Table 2: **ODU:** FY23 request.

	Request	**-20%**	**-40%**
PhD student	$23,250	$18,800	$14,100
Travel	$5,000	$0	$0
Xilinx Software	$4,295	$4,295	$4,295
Overhead (60%)	$19,677	$13,857	$11,037
Total	**$52,222**	$36,952	$29,432

Table 3: **A total** FY23 request.

	Request	**-20%**	**-40%**
JLAB	$48,822	$35,822	$26,064
ODU	$52,222	$36,952	$29,432
Total	**$101,044**	**$72,774**	**$55,496**

8 Manpower/Personnel

Jefferson Lab (JLAB):

F. Barbosa	Electronics Engineer	5%	electronics
C. Dickover	Electronics Engineer	10%	FPGA expert
S. Furletov	Research Scientist	20%	physics, ML,FPGA
Y. Furletova	Research Scientist	5%	physics
D. Romanov	Research Scientist	10%	software, ML
D. Lawrence	Research Scientist	0%	consulting

William & Mary,JLab:
C. Fanelli , Research Scientist, 0% FTE, consulting

Old Dominion University (ODU):
L.Belfore Professor 0% FTE
PhD student % 50% FTE

Forschungszentrum Jülich, Germany:
L. Jokhovets , 0% FTE, consulting

9 Publications

References

[1] S. Furletov et al, *Machine learning on FPGA for event selection*, 2022 JINST **17** C06009

[2] F. Barbosa et al., *A new Transition Radiation detector based on GEM technology*, NIM A, **942** (2019), doi:10.1016/j.nima.2019.162356.

[3] T. Horn et al., *Scintillating crystals for the Neutral Particle Spectrometer in Hall C at JLab*, Nucl. Instrum. Meth. A, **956** , 2020

[4] G. Visser et al., *A 72 channel 125 MSPS analog-to-digital converter module for drift chamber readout for the GlueX detector*, IEEE Nuclear Science Symposuim & Medical Imaging Conference, 2010, pp. 777-781, doi: 10.1109/NSSMIC.2010.5873864.

[5] The Toolkit for Multivariate Data Analysis with ROOT (TMVA), https://root.cern.ch/tmva.

[6] C. Peterson, T. Rögnvaldsson, L. Lönnblad, *JETNET 3.0-A versatile artificial neural network package.*, Computer Physics Communications. 81, 185–220 (1994).

[7] R. Aaij, et al. *Design and performance of the LHCb trigger and full real-time reconstruction in Run 2 of the LHC* , Journal of Instrumentation **14** 2019 P04013

[8] D. Lawrence, A. Boehnlein, N. Brei, *JANA2 Framework for Event Based and Triggerless Data Processing*, 10.1051/epjconf/202024501022, EPJ Web Conf. **14** 2020, P01022

[9] L. Jokhovets et al., *Improved Rise Approximation Method for Pulse Arrival Timing*, IEEE Transactions on Nuclear Science, vol. 66, no. 8, pp. 1942-1951, Aug. 2019.

[10] L. Jokhovets et al., *ADC-Based Real-Time Signal Processing for the PANDA Straw Tube Tracker*, IEEE Transactions on Nuclear Science, vol. 61, no. 6, pp. 3627-3634, Dec. 2014.

[11] J. Duarte et al., *Fast inference of deep neural networks in FPGAs for particle physics*, JINST, 2018, doi:10.1088/17480221/13/07/p07027