Contribution ID: **151** Type: **Oral**

A framework for porting the NeuroBayes machine learning algorithm to FPGAs

Tuesday 29 September 2015 11:10 (25 minutes)

The NeuroBayes machine learning algorithm is deployed for online data reduction at the pixel detector of Belle-II. In order to test, characterize and easily adapt its implementation on FPGAs, a framework was developed. Within the framework a HDL model, written in python using MyHDL, is used for fast exploration of possible configurations. Under usage of input data from physics simulations figures of merit like throughput, accuracy and resource demand of the implementation are evaluated in a fast and flexible way. Functional validation is supported by usage of unit tests and HDL simulation for chosen configurations.

Summary

The envisaged luminosities for SuperKEKB are expected to generate extremely high hit rates for the detectors closest to the beam pipe. This is especially true for the pixel detector of the Belle-II experiment. Here data rates are estimated to reach up to 26 GBit/s. Combined with data from other subdetectors, this rate is not manageable for the employed DAQ system. To solve this problem data reduction mechanisms are executed online. One of them is the NeuroBayes machine learning algorithm, that is going to be used on FPGAs close to the pixel detector. It has the capability to reliably separate hits in pixel detector, originating from different particles. However this algorithm was primarily developed to run on CPUs, thus a port for FPGAs has to be developed. It has to identify particles with the same reliability as the reference implementation in software. At the same time a required throughput 200 million analysed hit clusters per second have to be achieved. Since the FPGA chosen for deployment, already hosts data organization mechanisms, the allowed resource demand is bounded and cannot be exceeded. Additionally the characteristics of the sensors used in the pixel detector can change over time, for example the pedestal charges, as a result easy and flexible adaptation are needed.

To match these requirements, we developed a framework that allows for easy and flexible porting of the NeuroBayes algorithm to FPGAs. It is used as a tool for the evaluation of the port's capability to correctly classify hits in the pixel detector caused by different particles. Facilitating fast exploration of the design space, the framework uses a model of the algorithm's hardware architecture in software. The model is written in python using MyHDL, which is a library that allows hardware modelling by introducing bit-level data types and operations. This way different configurations of the hardware architecture, like the bit width for fix point arithmetic or pipeline depth of vector multiplications, are quickly explored. Functional validation is conducted under usage of the basf2 simulation framework. It is capable of generating pixel data comparable to realistic scenarios in Belle-II. The software model is validated against a reference implementation under usage of unit tests. For chosen configurations a testbench is generated for subsequent simulation of post synthesis models. In case characteristics of the pixel detector change, reevaluation is performed using new pixel data from basf2 incorporating the new circumstances as input.

After evaluation and selection of a suitable configuration that is fulfilling throughput, resource and efficiency requirements, the framework generates HDL packages that are used in the hardware architecture of the algorithm for subsequent implementation.

In this paper the architecture of the framework will be presented. The flow from acquiring pixel data until generation of the bistream to be used is depicted. Results for the design space exploration inside the framework

and verification using the most recent batch of simulated pixel detector data will be shown.

Primary author: BAEHR, Steffen (Karlsruhe Institute of Technology)

Co-authors: Prof. BECKER, Juergen (Karlsruhe Institute of Technology); Dr HECK, Martin (Karlsruhe Institute of Technology); Prof. FEINDT, Michael (Karlsruhe Institute of Technology); Dr SANDER, Oliver (Karlsruhe Institute of Technology)

Presenter: BAEHR, Steffen (Karlsruhe Institute of Technology)

Session Classification: Programmable Logic, design tools and methods

Track Classification: Logic