Acceleration of Cherenkov angle reconstruction with the new Intel Xeon/FPGA compute platform for the particle identification in the LHCb Upgrade

Christian Faerber
CERN Openlab fellow, CERN EP-LBC, CH-1211 Geneva 23, Switzerland
E-mail: Christian.Faerber@cern.ch

Abstract. The LHCb experiment at the LHC will upgrade its detector by 2018/2019 to a 'triggerless' readout scheme, where all the readout electronics and several sub-detector parts will be replaced. The new readout electronics will be able to readout the detector at 40 MHz. This increases the data bandwidth from the detector down to the Event Filter farm to 40 TBit/s, which also has to be processed to select the interesting proton-proton collision for later storage. The architecture of such a computing farm, which can process this amount of data as efficiently as possible, is a challenging task and several compute accelerator technologies are being considered for use inside the new Event Filter farm.

In the high performance computing sector more and more FPGA compute accelerators are used to improve the compute performance and reduce the power consumption (e.g. in the Microsoft Catapult project and Bing search engine). Also for the LHCb upgrade the usage of an experimental FPGA accelerated computing platform in the Event Building or in the Event Filter farm is being considered and therefore tested. This platform from Intel hosts a general CPU and a high performance FPGA linked via a high speed link which is for this platform a QPI link. On the FPGA an accelerator is implemented. The used system is a two socket platform from Intel with a Xeon CPU and an FPGA. The FPGA has cache-coherent memory access to the main memory of the server and can collaborate with the CPU.

As a first step, a computing intensive algorithm to reconstruct Cherenkov angles for the LHCb RICH particle identification was successfully ported in Verilog to the Intel Xeon/FPGA platform and accelerated by a factor of 35. The same algorithm was ported to the Intel Xeon/FPGA platform with OpenCL. The implementation work and the performance will be compared. Also another FPGA accelerator the Nallatech 385 PCIe accelerator with the same Stratix V FPGA were tested for performance. The results show that the Intel Xeon/FPGA platforms, which are built in general for high performance computing, are also very interesting for the High Energy Physics community.

1. Introduction
The LHCb experiment will be upgraded during the Long Shut-down 2 (2018-2019) to take data with an instantaneous luminosity of $2 \times 10^{33} \text{cm}^{-2}\text{s}^{-1}$ and to collect a dataset of at least 50 fb$^{-1}$. This goal can only be achieved when the whole detector readout chain will be modified to make a much more flexible 40 MHz detector readout possible [2]. Furthermore, the new readout scheme will use not a hardware trigger any-more, which would reduce the trigger efficiency in hadronic decays too much. The triggering will happen only in a software-based trigger running on a large Event Filter farm, which has to process and filter an input bandwidth of 40 TBits/s.
In addition, the trigger efficiency for hadronic decays will be increased. This is a challenging task and several compute accelerator technologies are being considered for use inside the new Event Filter farm. Here we present the tests with an FPGA-based compute accelerator, the Intel(R) Xeon/FPGA.

This accelerate compute platform uses an Intel(R) Xeon(R) CPU E5-2680 v2 at 2.80 GHz linked via Quick Path Interconnect (QPI) to a high performance Stratix V FPGA, on which an accelerator is implemented. QPI is the standard Intel point-to-point processor interconnect. First, the Event Filter farm in the LHCb Upgrade is shortly introduced, followed by a description of the Intel Xeon/FPGA platform. Afterwards studies are shown done with the RICH algorithm for the Cherenkov angle reconstruction[3].

1.1. The Event Filter Farm in the LHCb Upgrade
After the LHCb Upgrade the raw data from the detector will be sent by roughly 12,000 optical links with a bandwidth of 4.8 GBit/s each to 500 Event Building nodes (see the schematic in figure 1). The optical data transmission uses a custom made radiation hard optical called link the GBT[4]. Each of the Event Building nodes will host a custom PCIe receiver board, the PCIe40, which converts the GBT protocol to the PCIe protocol and gets the data into the server. In each Event Building node only a part of the whole detector information arrives, so the data fragments have to be combined for each event from every Event Building node, which are connected via a 40 TBit/s full duplex network. This network will be realized with a high-speed interconnect technology (100 GBit/s) e.g. Intel Omni-Path. Afterwards, the completed events are sent to the Event Filter farm, where a software Trigger is used for selecting the events. The processing of the data in the Event Filter farm has to be very fast, as the decision is needed within O(10 μs). In order to sustain the designed throughput, different technologies are under study, including FPGA compute accelerators, GPUs and other compute accelerators like the Intel KNL.

2. Intel Xeon/FPGA system
The Intel Xeon/FPGA prototype is a two socket server machine, where the first socket hosts an Intel(R) Xeon(R) E5-2680 v2 CPU and the second socket hosts an Altera Stratix V GX A7 FPGA (see Figure 2). In the FPGA, 234,720 Adaptive Logic Modules (ALM) are available, where each contains inputs, several LUT-based resources and four registers to realize any boolean function with up to six inputs. Furthermore, the FPGA hosts 940,000 registers and 256 DSP blocks. These digital signal processor (DSP) blocks are crucial for any algorithm using floating point calculations. Between the two sockets with the CPU and the FPGA a QPI bus is used to connect the two chips with a high bandwidth and low latency interconnect. In addition, the FPGA and the CPU have cache-coherent access to the main memory.

In the standard work flow, the CPU allocates a block inside the main memory large enough for configuration, data to process and results. Afterwards the CPU writes the data to process into the main memory and gives the FPGA a pointer to this location. The FPGA accesses and processes the data with this address, and writes the result back to a destination address directly in the main memory. This is a real advantage versus the standard PCIe FPGA accelerator cards, because for the PCIe card two additional data copies are required: back and forth from the main memory into the local memory of the card. This reduces the potential FPGA performance dramatically, because the pipelines are not used throughout the whole running time of the algorithms[5].

The programming of the algorithms running on the FPGA can be written in Verilog or OpenCL. For the FPGA bitstream synthesis the Intel/Altera standard software Quartus II is used. The user has to write the accelerator blocks himself, but the QPI interface is already available via an encrypted Verilog block. So the user block has to fit to the Intel Verilog block
Figure 1. Schematic of the upgraded LHCb Readout electronics, Event Building network and the Event Filter farm.

Debugging the bitstream can be done with the FPGA usual tools like ModelSim and SignalTap. OpenCL is an open standard, written in C/C++ maintained by the Khronos Group [6]. It offers a alternative programming model for non FPGA programmers, and due to its higher level of abstraction, it reduces the development time dramatically. OpenCL is a Standard to run code on heterogeneous platforms.
3. FPGA-based accelerator for the RICH reconstruction

One of the most time consuming algorithms in the LHCb High Level Trigger is the RICH photon reconstruction, which is used for the particle identification and crucial for the LHCb physics program. The current software trigger for the RICH cannot handle the full data rate presented to it, and is therefore pre-scaled.

The LHCb software trigger gets all events triggered by the hardware trigger and processes them in two stages. First the data are processed by the High Level Trigger 1, which reduces the number of events and stores the data on hard drives for buffering. Afterwards, during a break of the LHC, when no proton-proton collisions are generated, the data on the hard drives are processed with the High Level Trigger 2, which has a better selection efficiency, due to the fact that in addition detector calibration constants from the last collision period are used.

The RICH photon reconstruction is a good candidate to be accelerated. For each proton-proton collision hundreds of particle tracks are measured, each of them travelling through the RICH detectors creates $O(10)$ Cherenkov photons. For every combination of particle track and photon hit inside the RICH detectors the same calculation has to be done to find the corresponding Cherenkov cone to every particle track. This takes roughly 30% of the second High Level Trigger (HLT2).

3.1. Algorithm

The algorithm implemented on the Xeon/FPGA machine is a time consuming sub process of the Cherenkov angle reconstruction, which is used for the particle identification. Cherenkov radiation is emitted by every charged particle travelling faster than the speed of light in a medium and the photons are arranged on a cone around the particle track. The opening angle of the cone depends on the speed of the particle. The general formula for the Cherenkov angle is: $\cos(\Theta_c) = \frac{1}{\beta n}$, $\beta = \frac{v}{c}$ and $n$ is the refractive index. The analytic solution depends on solving a quartic equation, calculating a cubic root, a rotation matrix and several cross and scalar products [3]. In Figure 3 a schematic related to the reconstruction is shown.

4. Implementation of Cherenkov Angle reconstruction

The idea of the design is to stream the data with full bandwidth of the QPI interface through the FPGA, so there should be no bottleneck of FPGA design. Two versions were developed, one in Verilog and one in OpenCL.

4.1. Verilog implementation

For the Verilog version the algorithm was realized in a 753 clock cycle long pipeline. The design uses almost the complete FPGA. In Table 1, the FPGA resource usage after the synthesis
Figure 3. Schematic related to the Cherenkov angle reconstruction. The emission point (E) the curvature centre of the mirror (C) and the detection point (D) are known, and the point (M) has to be calculated. The Cherenkov angle is calculated between the thick black particle track vector (t) and the green photon momentum vector (P). The picture is from [3].

Optimization is shown. It has to take into account that the QPI interface uses already 30% of the FPGA ALMs to implement the protocol, and after the optimization 88% of all the ALMs are used for the whole design. The DSP blocks are used to implement all soft-core floating point calculation blocks and these don’t limit the design, like the registers. Only 50% of the Stratix V registers are used for the design. The pipeline was optimized to run at least with a frequency of 200 MHz, which makes a calculation for a single photon within 5 ns possible when the pipeline is completely filled.

Table 1. FPGA resource usage for Cherenkov Angle reconstruction

| FPGA Resource Type | FPGA Resource used [%] | Dedicated for QPI [%] |
|--------------------|------------------------|-----------------------|
| ALMs               | 88                     | 30                    |
| DSPs               | 67                     | 0                     |
| Registers          | 48                     | 5                     |

4.2. OpenCL implementation

The OpenCL kernel was realized in only 250 lines of codes, and was much easier to implement. If the resource usage for both implementations is compared (see table 2), it is seen that the OpenCL version uses more DSPs, less ALMs as also less registers. The OpenCL compiler realized the pipeline in a different way, broader and not so deep.

Table 2. Compare FPGA resource usage for RICH kernel with Verilog and OpenCL

| FPGA Resource Type | FPGA Resource used [%] for Verilog | FPGA Resource used [%] for OpenCL |
|--------------------|-------------------------------------|-----------------------------------|
| ALMs               | 88                                  | 63                                |
| DSPs               | 67                                  | 82                                |
| Registers          | 48                                  | 24                                |
5. Compare Verilog - OpenCL results

In the following the results of both Verilog and OpenCL implementations will be compared to the standard Xeon(R) E5-2680 v2 CPU single thread version. The compute acceleration was measured for different number of photons from 1 up to 2,000,000 photons. In figure 4 the processing time vs. the number of photons for the Verilog version and the CPU version are shown. For a number of photons below 200, the CPU version is faster, due to the latency of the data transfer and the pipeline processing, but for higher number of photons the FPGA version gets faster than the CPU version. The speed-up obtained by the Stratix V reaches 35x compared to the CPU single-threaded version. The acceleration of the Verilog version is limited by the bandwidth to the FPGA, and the result is that the photon pipeline is used only 50% of all clock cycles. This was verified on the running FPGA by counting in how many clock cycles the pipeline got new data to process. Therefore, the developed photon pipeline on the Stratix V could run a factor 64 faster than the Xeon CPU alone. An idea to overcome this bottleneck is to cache hits and tracks in the FPGA for a physics event. This would reduce the read bandwidth from the main memory and the data could be cached in the FPGA local RAM blocks. So the population of the pipeline could be increased, due to the fact that the internal bandwidth from the local FPGA RAM blocks through the pipeline is much higher.

![Figure 4. Time for reconstructing the Cherenkov angle on a CPU compared to the runtime on an FPGA.](image)

The speed-up between the OpenCL program and the CPU version increases with an increasing number of photons. Table 3 compares the results for both implementations. A major advantage of the OpenCL work flow is the much shorter development time and the readability, maintainability and the smaller code base. From the performance point of view the Verilog version is faster. This is important if you want to optimize a kernel to use as much performance of the expensive FPGA as possible. Two kernels, a cube root calculation, which is a sub-function of the RICH kernel, and the RICH kernel itself are tested. In the OpenCL version for the cube root there is already an optimized function inside the OpenCL mathematics library, and therefore very easy to implement, for the Verilog version the floating-point cube-root block had to be developed, which took 4 weeks with all simulations.
Table 3. Verilog kernel and OpenCL kernel comparison

|                      | Verilog | OpenCL |
|----------------------|---------|--------|
| Development time     | 2.5 months | 2 weeks |
| Lines of code        | 3,400 | 250    |
| Performance RICH kernel | x35 | x26    |

6. PCIe - QPI interconnect comparison

The OpenCL version is also interesting, because it is easy to compare with a PCIe FPGA accelerator card. The comparison with Verilog would take too long, because it is not the standard programming flow for the Nallatech card. For this test a Nallatech 385 with the same Stratix V FPGA was used. The work flow for the Nallatech card is very similar to the Intel Xeon/FPGA OpenCL work flow. The performance of the pipeline inside the FPGA depends only on the FPGA and is the same for both systems, but if we compare the complete performance including copying the data back to the main memory for the PCIe card the Intel Xeon/FPGA is a factor 3 faster then the Nallatech card (see figure 5). The reason for this difference is the cache-coherent, high-bandwidth and low latency interface between the CPU and the FPGA of the Intel Xeon/FPGA.

Figure 5. Performance comparison for PCIe and QPI interface for the OpenCL version of the Cherenkov angle reconstruction.

7. Future tests

Several additional LHCb High Level Trigger Algorithms like Vertex Tracking or decompression and re-formatting of packed binary data from the detector in the Event Building will be tested. Also very important the performance will be compared to the new Intel Xeon/FPGA with Arria 10 and faster interconnect in a multi chip package (MCP). This chip hosts hardened floating point multiply and accumulate blocks, which use less logic and run at a higher frequency. Both will make the Arria 10 FPGA outperform the Stratix V FPGA for algorithms using floating point calculations. Also other FPGA compute accelerators like the 385A Nallatech PCIe board
with the same Arria 10 FPGA will be compared. The tests to compare the performance of algorithms written in Verilog and in OpenCL as also the development time which is also very important, will be continued. Furthermore measurement of the power consumption will be done and it will be compared with GPUs and CPUs, because the performance per Joule is expected to be much better for FPGAs [5]. This would be an interesting and important point for the planning of the future Event Filter farm.

8. Summary
The LHCb experiment will be upgraded 2018-2019 to make a much more flexible 40 MHz detector readout possible. Afterwards no hardware trigger will be used any-more, only a complete software based trigger will be implemented to select the interesting proton-proton collision. The triggering will happen on a large Event Filter farm, which has to process and filter a input bandwidth of 40 TBits/s. This is a challenging task and several compute accelerator technologies are being considered to be used inside the new Event Filter farm.

Therefore, a study was performed to investigate the possible usage of the new Intel Xeon/FPGA compute accelerator, which is about the acceleration of the Cherenkov angle reconstruction for the particle identification. Different work flows were tested, the Verilog and the OpenCL work flow. Both show an encouraging acceleration of 35x for the Verilog version and 26x for the OpenCL version. The OpenCL version is much faster to develop and easier to maintain, which is also an important consideration if a lot of functions have to be accelerated. Furthermore, the Intel Xeon/FPGA is compared to an Nallatech 385 PCIe Stratix V FPGA accelerator card. The cache-coherent, high-bandwidth and low latency interface between the CPU and the FPGA of the Intel Xeon/FPGA has a strong influence on the performance, because the copy of the data local to the card memory and back is avoided. The Intel Xeon/FPGA shows a factor 3 higher performance than the Nallatech accelerator using the same FPGA.

These results are very encouraging and the High Energy Physics community may benefit tremendously from these new devices, especially with the new coming Arria 10 Xeon/FPGAs in a single package and faster interconnect between CPU and FPGA. Also compared to GPUs the performance per Joule will be interesting, due to the lower power consumption of FPGAs, and other algorithms like decompression and re-formatting of packed binary data from the detector for the Event Building are very promising and will be tested in the near future.

References
[1] LHCb Collaboration, Framework TDR for the LHCb Upgrade : Technical Design Report.CERN, Geneva, 2012.
[2] LHCb Collaboration, LHCb Trigger and Online Upgrade Technical Design Report.CERN, Geneva, 2014.
[3] R. Forty and O. Schneider, RICH pattern recognition.CERN, Geneva, 1998.
[4] P. Moreira et al., The GBT Project.in proceedings of the Topical workshop on electronics for particle physics, Prague, Czech Republic, 3-7 Sept. 2007, pp. 332-336
[5] S. Sridharan et al., Accelerating particle identification for high-speed data-filtering using OpenCL on FPGAs and other architectures.IEEE FPL16, Lausanne, Switzerland, 29 Aug - 2 Sept. 2016, DOI: 10.1109/FPL.2016.7577351
[6] H. Lee, The OpenCL Specification Version: 2.1 Document Revision: 23. Khronos OpenCL Working Group. 11 Nov. 2015
[7] J. Blinn, Floating Point Tricks.Computer Graphics & Applications, IEEE. 17 (4): 80. DOI:10.1109/38.595279