High-speed, Low-latency Readout System with Real-time Trigger Based on GPUs

M. Caselle, L.E. Ardila Perez, S. Chilingaryan, T. Dritschler, A. Kopmann, H. Mohr, L. Rota, M. Vogelgesang, M. Balzer, M. Weber

***Abstract*–At the HL-LHC, proton bunches collide every 25 ns and produce an average of 140 pp interactions per bunch crossing. To operate in such an environment, the CMS experiment will need a L1 hardware trigger, able to identify interesting events within a latency of 12.5 µs. Therefore, the novel L1 trigger will make use of data coming from the silicon tracker to constrain the trigger rate. The architecture that will be implemented in future to process tracker data is still under discussion. In this paper we present a heterogeneous L1 track system formed of FPGA-based boards combined to high-end graphics processor units (GPUs). The proposed track finding algorithm is based on the Hough transform method executed at a GPU farm. To keep the real-time constrains and to reduce data transfer latency, new readout electronics has been developed. It is based on Direct Memory Access to transfer the stub data from the front-end directly into the GPU memories without any intermediate buffering, therefore offloading the CPU, avoiding OS jitter effects.**

I. Introduction

SIGNIFICANT challenges continuously arise from the High Energy Physics (HEP) experiments at the Large Hadron Collider (LHC) at CERN. The quest for rare new physics phenomena leads to the evaluation new computing concepts. A potential and very powerful architecture are provided by Graphics Processing Unit (GPU). Their use in high-level trigger (HLT) systems not only provides faster and more efficient event selection, but also includes the possibility of developing new complex triggers that were not feasible previously. At HLT efficient many-core parallelization of event reconstruction algorithms is possible. The benefit of significantly reducing the number of the farm computing nodes in further processing steps is evident. At lower levels, where typically severe real-time constraints are present, we envisioned the possibility to meet the real-time constrains and to reduce data transfer latency and its fluctuations, by injecting readout data directly from the FPGA into the GPU memories without any intermediate buffering, therefore offloading the CPU, avoiding OS jitter effects. In order to satisfy such constraints at lower levels, we have developed a custom FPGA-based readout card and implemented a new concept of Direct Memory Access (DMA) capable to move the data from FPGA to system memory and/or GPU memory. For the GPU algorithm, a tracking algorithm for transverse momentum p_T trigger is evaluated on a NVIDIA Tesla K40 GPU using Hough-transform methods. Benchmarks for latency and bandwidth for the proposed readout system are discussed, followed by a performance analysis on case studies of the GPU-based low level trigger for the CMS experiment.

II. Readout System and “GPU-Direct”

The Institute for Data Processing and Electronics (IPE) is specialized in development and commissioning of experimental setups that require on-line data processing in the range of GB/s [1]. The integration of GPUs in trigger and data acquisition systems is currently being investigated for several applications in HEP and photon science. Because of their parallel architecture, GPUs can efficiently process compute-intensive work-loads and have the potential to achieve real-time data analysis. However, communication between GPUs and other devices limit the performance of these systems. To reduce this constrain, we have developed a readout architecture that enables fast communication between FPGA, CPUs and GPUs via Direct Memory Access (DMA).

In this paper we present a high throughput platform based on direct FPGA-GPU communication. The architecture consists of a Direct Memory Access (DMA) engine compatible with the Xilinx PCI-Express core, a Linux driver and high-level software to manage direct memory transfers using the “GPU-Direct” technology of both major GPU vendors AMD’s and NVIDIA.. The readout card is shown in Fig. 1. It is equipped with a Xilinx Vitex-7 FPGA and it is connected to the GPU farm by a PCIe generation 3 link with 16 lanes. The readout card has been designed as multipurpose platform for the photon science and for a first demonstrator of the CMS track trigger. Two FMC VITA-57 compliant high pin count connectors, provide a wide number of high speed interconnection for application specific cards. A DDR3 - 4GB memory device has been integrated and the firmware is optimized to operate with up to 120 Gb/s. A high data throughput FPGA infrastructure has been developed. It is capable to operate in continue data streaming with a throughput of more than 6.5 GB/s. The readout is based on a high performance scatter-gather DMA engine capable to transfer data from FPGA to both GPU and system memories.

In traditional DMA architectures, data is first written to the main system memory and then sent to the GPUs for final processing, Fig. 2 (yellow data flow). The main memory is involved with a certain number of read/write operations,

Manuscript received May 30, 2016. All authors are with the Institute for Data Processing and Electronics (IPE) of Karlsruhe Institute of Technology (KIT), Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany. Author contact: michele.caselle@kit.edu.

depending on the specific HW or SW implementation. The total throughput and latency of the system is therefore limited by the memory bandwidth/operations.

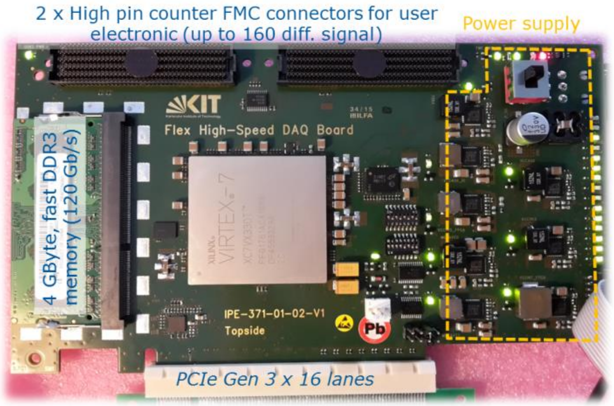
2 x High pin counter FMC connectors for user electronic (up to 160 diff. signal)
Power supply
4 GByte, fast DDR3 memory (120 Gb/s)
KIT
Flex High-Speed DAQ Board
VIRTEX-7
IPE-371-01-02-V1
Topside
PCIe Gen 3 x 16 lanes

Fig. 1. A multi-purpose and high-performance readout card used as demonstrator for CMS L1 track finding system.

Using direct GPU communication, Fig. 2 (blue data flow), the DMA engine has direct access to the GPU memory, therefore latency and bandwidth of the system are drastically improved.

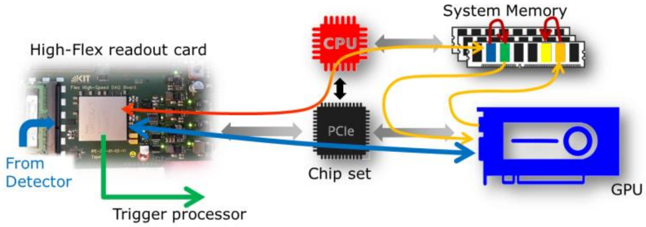

Fig. 2. Data flow from FPGA to GPU with “GPU-Direct” technology.

The latency has been measured as the total time of a data transfer from FPGA to GPU. The total time splits in an initial latency and the time to transfer a certain amount of data. The total time versus the data size for a NVIDIA Tesla K40 graphic card is shown in Fig. 3.

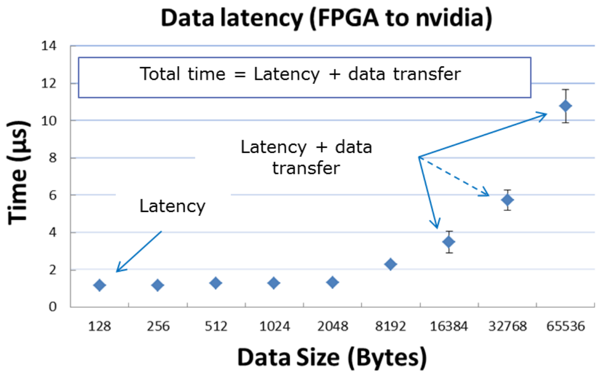

Fig. 3. Total transfer time from FPGA to GPU by “GPU-Direct” technology versus the data size.

For smaller data sizes, lower than 2 kByte, the total time is dominated by the latency of the system. For a larger datasets, the total time increases linear with the size of transmitted data. Therefore, the latency measurements could be estimated by the plot in Fig. 3 to ~ 1.15 µs (average) with a time jitter < 100 ns. Using “GPU-Direct”, FPGA devices can directly read and write CUDA/OpenCL host and device memory, eliminate unnecessary memory copies, dramatically lower the CPU overhead and reduce latency. All results in significant performance improvements for data transfer in HEP trigger applications.

III. Direct Memory Access Architecture

We have developed a DMA engine that minimizes resource utilization while maintaining the flexibility of a Scatter-Gather memory policy [2]. The engine is compatible with the Xilinx PCIe 2.0/3.0 IP-Core [3] for Xilinx FPGA families 6 and 7. DMA data transfers between main system memory and GPU memory are supported. The complete architecture of the DMA engine is shown in Fig. 4. The DMA can operate in bus master and bus slave mode. Four different engines to transmit and receive data (TX and RX) using the PCIe core are developed to cover different applications. The TX-master engine is used to move data to system memory and GPU memory. The RX-master is used to read specific memory location from GPU. The DMA architecture has been developed to allow a “Direct DMA through InfiniBand”. The feature is capable to move data from FPGA readout electronics to GPU clusters by InfiniBand network link. For this purpose, the TX and RX slave engines have been developed also to satisfy the requirement for a direct InfiniBand communication. The architecture also includes a Base Address Registers (BAR) space, which is used to configure the DMA engine. Two FIFOs, operating at 250 MHz and a data width of 256 bits, act as user-friendly interfaces with the user custom logic. The user logic and the DMA engine are configured by the host system through PIO registers. The physical addresses of the host's memory buffers are stored into an internal memory and are dynamically updated by the driver or user, allowing highly efficient zero-copy data transfers. The maximum size associated with each address is 2 GB.

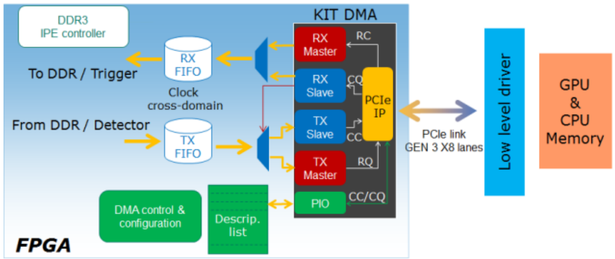

Fig. 4. DMA firmware architecture.

Measurements with a Gen3 x8 links show a throughput of over 6.5 GB/s for transfers to GPU memory and to system memory [2]. We also assessed the architecture for low latency requirements as shown in Fig. 3.

IV. “GPU-Direct” - Software Implementation

The software implementation of “GPU-Direct” has been developed for both GPU vendors NVIDIA and AMD. The procedure is similar for both platforms. In the following descriptions it is given for AMD implementation. On the host side, AMD's "DirectGMA" technology, an implementation of the bus-addressable memory extension for OpenCL 1.1 and later, is used to write from the FPGA to GPU memory and from the GPU to the FPGA. Fig. 5 shows the main mode of operation: to write into the GPU, the physical bus addresses of the GPU buffers are determined with a call to *clEnqueueMakeBuffersResidentAMD* and set by the host CPU in an internal registers of the FPGA (1). The FPGA then writes data blocks autonomously in DMA fashion (2). To signal events to the FPGA (4), the control registers can be mapped into the GPU's address space passing a special AMD-specific flag and passing the physical BAR address of the FPGA configuration memory to the *clCreateBuffer* function. From the GPU, this memory is seen transparently as regular GPU memory and can be written accordingly (3). In our setup, trigger registers are used to notify the FPGA on successful or failed evaluation of the data. Using the *clEnqueueCopyBuffer* function call it is possible to write entire GPU memory regions in DMA fashion to the FPGA. In this case, the GPU acts as bus master and push data to the FPGA.

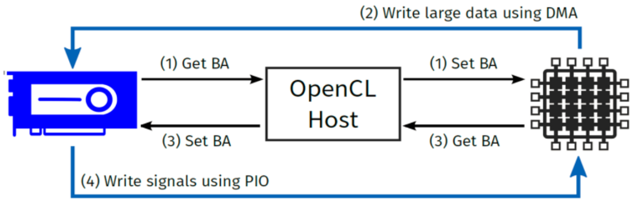

Fig. 5. "GPUDirect" – software implementation.

DMA can be operated in both configurations: "standard" where the data is transferred from FPGA to system memory "CPU" and "Direct GMA" where the data is transferred from FPGA to GPU. The very low latency achieved enable the proposed readout architecture as prominent system for a very flexible low-level trigger based on GPUs.

V. Implementation of Hough transform by GPU

The Hough transform is a technique often used in image analysis to identify lines in the image. The Hough transform is particularly adapted to L1 tracking requirements. This method is fast, robust against fakes, and provides encouraging results in harsh environment. In addition, the Hough transform is naturally tolerant against missing hits or hits that do not exactly fit the candidate features. Sample input data was generated using Monte Carlo simulation of a simple detector model where only the transverse plane is considered. Fig. 6 shows the result of 50 simulated stubs in a beam condition with a pile-up of 140 protons per bunch-crossing. Each stub in the detector results in a straight-line in the Hough parameter space. A track is by a local peak in the 2D histogram of the Hough parameter space. In Fig. 6 a dataset with 2 tracks with a transverse momenta over 3 GeV/c are detected.

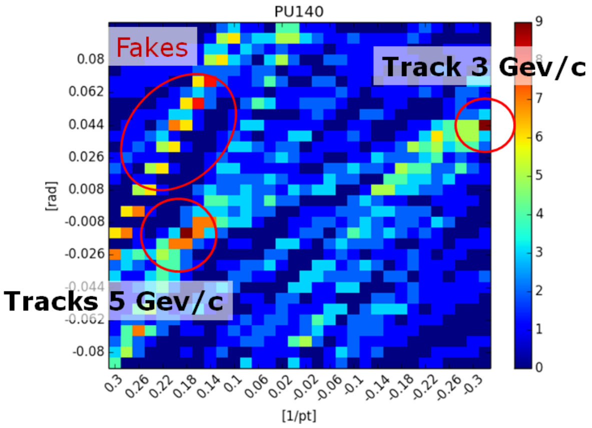

Fig. 6. Hough transform parameter space applied to a simulated data with beam condition of pileup 140.

The performance of the tracking algorithm based on Hough-transform has been evaluated on a NVIDIA Tesla K40 GPU. The results show that 500 stubs can be processed in only 7 µs with a single GPU.

VI. CONCLUSIONs

As part of the HL-LHC upgrade, the CMS experiment plans to build a new tracker with enhanced trigger capabilities. Several trigger concepts with simple track reconstruction in the L1 trigger are being explored by the CMS collaboration. They rely on a combination of ASICs and commercial FPGAs for very low latency pattern matching and track fitting. In this proceeding we presented an alternative architecture where the track finding by Hough transform is implemented on a NVIDIA Tesla K40 GPU. Preliminary results show that 500 stubs can be processed in only 7 µs. A high performance DMA architecture has been developed and is used in a first demonstrator. It is capable to transfer data directly from the front-end to GPUs with microsecond latency only. These results show that low GPU execution time combined with low latency and high throughput electronics open a new prospective for a flexible GPU-based architecture of low-level trigger system in HEP.

References

[1] M. Caselle, M Balzer, S. Chilingaryan, M. Hofherr, V. Judin, A. Kopmann, N. J. Smale, P. Thoma, S. Wuensch, A. –S. Müller, "An ultra-fast data acquisition system for coherent synchrotron radiation with terahertz detectors", Journal of Instrumentation (JINST), JINST 9 C01024, Jan. 2014.

[2] L. Rota and M. Caselle, S. Chilingaryan, A. Kopmann, M. Weber, "A PCIe DMA Architecture for Multi-Gigabyte Per Second Data Transmission", Nuclear Science, IEEE Transactions, DOI: 10.1109/TNS.2015.2426877, June 2015.

[3] Xilinx, "Virtex-7 FPGA Gen3 Integrated Block for PCI Express", July 2015.