



Contribution ID: 33

Type: Oral presentation to parallel session

Implementation of a PC-based Level 0 Trigger Processor for the NA62 Experiment

Tuesday, 15 October 2013 17:25 (20 minutes)

The performance of “level 0” (L0) triggers is crucial to reduce and appropriately select the large amount of data produced by detectors in high energy physics experiments. This selection must be accomplished as fast as possible, since data staging within detectors is a critical resource. For example, in the NA62 experiment at CERN, the event rate is estimated at around 10 MHz, and the L0-trigger should reduce it by a factor of 10 within a time budget limit of 1ms.

So far, the most common approach to the development of an L0 trigger system has been based on custom hardware processors, so event filtering has been performed by algorithms implemented in hardware. More recently, the implementation of custom processors has been based on FPGA devices, whose hardware functionalities can be configured using specific programming languages.

The use of FPGAs offers greater flexibility in maintaining, modifying, improving filter algorithms, however even small changes require a hardware re-configuration of the systems, and changes to the algorithm logic can be limited by hardware constraints that have not been foreseen at the development time.

So, even if this approach guarantees fast processing, strong limitations still remains in the available flexibility when changing filtering algorithms on the fly or testing more filtering conditions at the same time could be an “added-value”, as required during the data-taking phase of the experiment.

In this contribution we present an innovative approach in the implementation of an L0-trigger system based on the use commodity PC, describing the architecture that we are developing for the NA62 experiment at CERN.

Data streams coming from the detectors are collected by an FPGA installed on a PCI-Express board plugged on a commodity PC. The FPGA receives data from detectors via giga-bit data channels, and stores them into the main memory of the PC. The PC then performs the filter algorithms on data available on its own memory, and writes back results to the FPGA for routing to the appropriate destination.

In our case we have used a commodity board with an Altera Stratix IV FPGA, 4 gigabit channels and a X8 Gen2 PCI-Express link delivering a peak bandwidth of 4 GB/s per direction.

In this presentation we focus on the description of the logic inside the FPGA to interface with the PCI-Express bus, and on the software organization including the Linux driver that allows the software filtering algorithm to read and write data to and from the FPGA.

We also analyze performances, and investigate ways to move quickly data to and from the FPGA.

Since the filtering program runs on a commodity PC, algorithm changes are much simpler, as they do not impact on the rest of the hardware system, and do not require to re-configure the FPGA.

Primary authors: PIVANTI, Marcello (University of Ferrara and INFN Ferrara); SOZZI, Marco (Sezione di Pisa (IT)); DALPIAZ, Pietro (Universita di Ferrara (IT)); SCHIFANO, Sebastiano Fabio (University of Ferrara and INFN Ferrara)

Co-author: GIANOLI, Alberto (Universita di Ferrara (IT))

Presenter: GIANOLI, Alberto (Universita di Ferrara (IT))

Session Classification: Data Acquisition, Trigger and Controls

Track Classification: Data acquisition, trigger and controls