# Intelligent FPGA Event Builder and Data Acqiusition System for the COMPASS Experiment

Yunpeng Bai*, Martin Bodlak†, Vladimir Frolov‡§, Vladimir Jary†, Stefan Huber*, Igor Konorov*, Dmytro Levit*, Josef Novy†, Richard Salac†, Dominik Steffen*§, Miroslav Virius†, Stephan Paul*

*Physikdepartment E18, Technische Universität München

†Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University

‡Joint Institute for Nuclear Research

§European Organization for Nuclear Research – CERN

***Abstract*—High reliability and compactness are key features of the Data Acquisition system for the COMPASS experiment at CERN's SPS. The system was commissioned in 2014 for the Drell-Yan physics run and succesfully operated during 2015 run. For 2016 run the system will be employed in full scale with on spill data rate of 1.5GB/s and a sustained rate of 500MB/s. The DAQ has a hybrid FPGA-software architecture, where the event building task is performed entirely by FPGAs, while the software is responsible for overall system control, configuration, monitoring and moving data to the central data reording facility. The FPGA firmwares perform data consistency checks and guaranties synchronous data merging and uninteraptable data taking. The event builder hardware consists of nine Virtex-6 VLX130T FPGA cards. The architecture and performance of iFDAQ will be discussed.**

## I. Introduction

The Common Muon and Proton Apparatus for Structure and Spectroscopy (COMPASS) [1] [2] is a multi-purpose fixed target experiment at the M2 beamline of the Super Proton Synchrotron at CERN. The experiment runs with the muon and the hadron beams with the momentum of around 170 GeV/c. The polarized muon beam is used to study the spin structure of the nucleons and deeply virtual Compton scattering. The hadron beams are used to study the Primakoff reactions, exotic mesons, glueballs and Drell-Yan processes.

From 2012 to 2014 the COMPASS experiment has undergone an upgrade to prepare the spectrometer for the deeply virtual Compton scattering and Drell-Yan measurements. During the break a new COMPASS DAQ system was developed. It has a single trigger level with a typical rate of 30kHz. With an average event size of 50 kB, the data rate is 1.5 GB/s during spill. A hybrid FPGA-software approach was selected for the upgrade of the DAQ to reduce the system's complexity, improve performance and reliability. Full events are built entirely in the FPGAs of the custom designed Data Handling Cards (DHC) in two stages as shown in Fig. 1FIDAQ architecturefigure.caption.1. The first stage of the system consists of 8 DHC multiplexers, which perform channel multiplexing and sub-event building. Then sub-events are sent to DHC switch, which performs final event building and sends complete events to the read-out engines computers. The 8 read-out engine PCs are equipped with the commercial

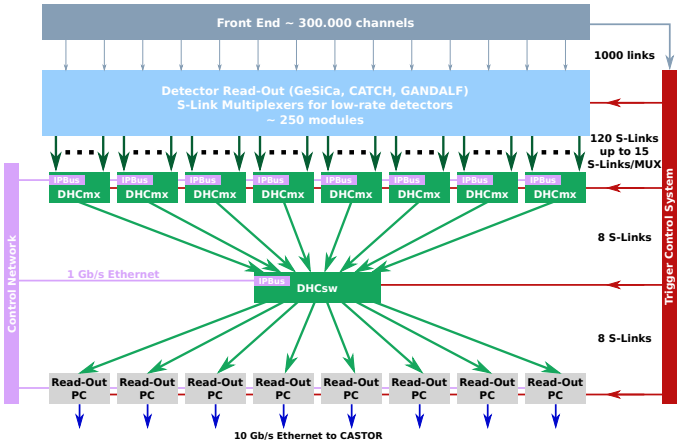


Figure 1: FIDAQ architecture

FPGA PCIe cards. The data are then buffered on the hard drive and later transfered to the central storage system. All event building stages are synchronized by the TCS [3]. The data consistency is monitored on all stages of the event building process and reported to the control software.

## II. DAQ Hardware

The DHC is designed as a compact AMC card which complies with the ATCA standard, see Figure 2DHC AMC module.figure.caption.2. Here are main characteristics of the module:

- Xilinx XC6VLX130T-2 FPGA type;
- Multi boot flash;
- 4GByte of DDR3 memory, 6GB/s bandwidth;
- 16 high speed serial links, each link may run up to 6.5Gbps;
- 1000Base Ethernet interface for slow control;
- Trigger Control System interface.

The module is mounted on a double width VME carrier card. All 16 high speed links are connected to SFP+ cages and accessible via the front panel. The Ethernet, TCS and JTAG interfaces are also placed at the front panel.

The VME crate provides power and cooling. A photo of the event builder from 2015 run is shown on Figure 3VME


Figure 2: DHC AMC module.

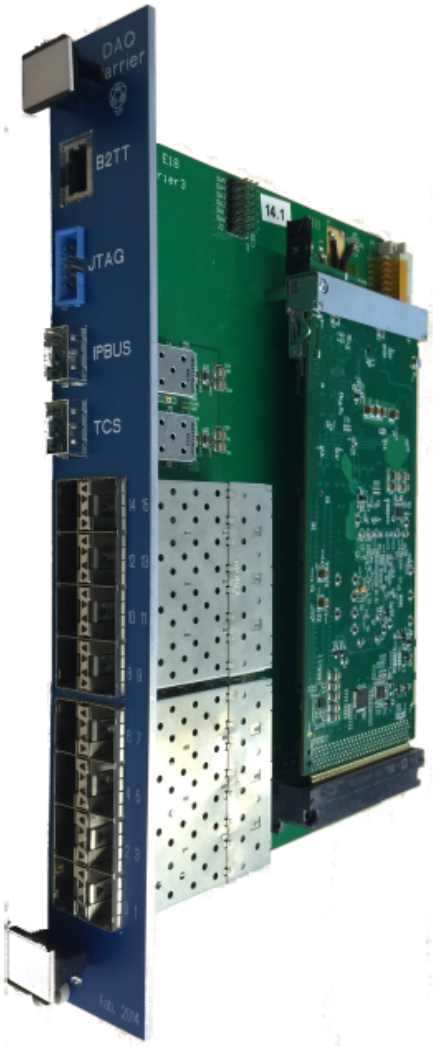


Figure 3: VME carrier card for DHC module.

carrier card for DHC module.figure.caption.3.

## III. DHC Firmware Architectures

Although all DHC mdules are identical eight of them are programmed as DHCmx(multiplexr) and one as DHCsw(switch).

A DHCmx firmware functional diagram is shown in Figure 5Firmware functional diagram of DHCmx's FPGA.figure.caption.5. DHCmx receives data from front-end

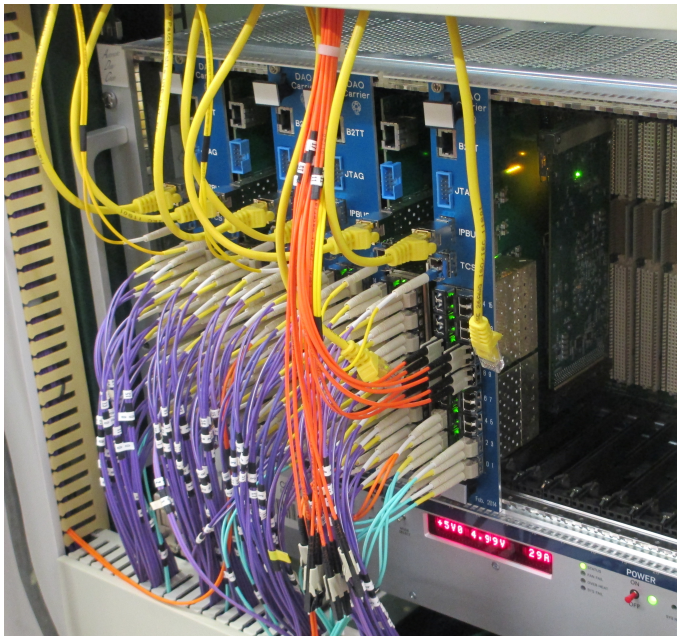

Figure 4: Hardware event builder

modules via 15 optical interfaces running S-Link protocol [4]. The data checker verifies data consistancy while copying them to DDR3 memory. It guaranties data integrity for further stages of the DAQ in all possible failures and does not block data flow. Depending on configuration, the corrupted data are either abandoned or corrected. In case of failure the diagnostic information is propagated to the DAQ together with the data and via IPbus. If the error is unrecoverable, the problematic port is disabled till the next spill.

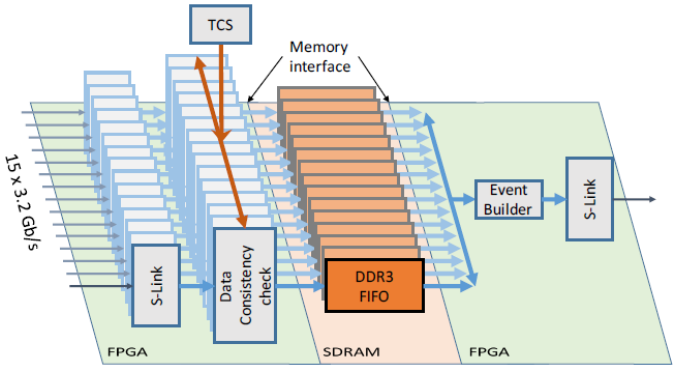


Figure 5: Firmware functional diagram of DHCmx's FPGA.

The DHCsw firmware functional diagram is shown in Figure 6DHCsw firmware diagram.figure.caption.6. The switch module receives sub-events from the DHCmx modules. The data are checked again for consistency and actions identical to DHCmx firmware are taken. The DDR memory is divided into blocks of a maximum possible event size. In contrary to the DHCmx the event is assembled inside the DDR memory. Event fragments are skewed at the input ports in a way that all 8 checker components process data in parallel. The DHCsw has a throughput of 3GB/s, which is very close to the theoretical maximum of the DDR memory.

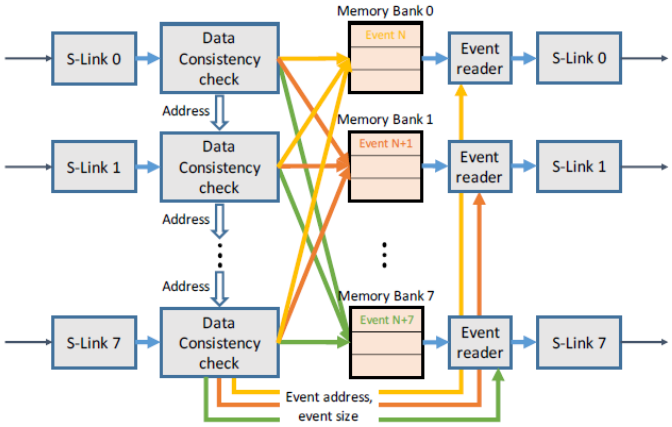


Figure 6: DHCsw firmware diagram.

## IV. DAQ Software

The DAQ software includes the slow control master, control slave processes for FPGA cards, real-time read-out slaves, configuration and monitoring GUI, message logger, and configuration database. The master process is a central process in the system which controls and synchronizes the other processes in the system by communicating with them using the DIALOG library specially developed for the iFDAQ. DIALOG has subsituted the DIM library [5] which was used before and showed some instability. The read-out slaves processes run at the read-out engine PCs and are the only real-time software processes in the system. They read out the data over PCIe and check the events for errors. Then the events are decoded by the read out processes into the DATE data format which makes them compatible with the existing analysis software and the decoded events are written to the hard disk.

## V. Conclusions and Outlooks

The DAQ system was commissioned in 2014 and used succesfully for physics run in 2015. The system proved a viability of the FPGA based event building concept. It will be extended to the nominal setup with 8x8 switch configuration for 2016 run.

The system development will be continued. The next step will be migration of the DHC modules to ATCA shelf. A new full size ATCA carrier card is being developed now. It will house 144x144 cross-point switch to extend flexibility and reliability performance of the system. The system will consists of three ATCA carrier cards and provide 9 slots for the DHC modules and three remaining slots for hot swappable DHC modules.

## References


[1] P. Abbon et al. (the COMPASS collaboration), *The COMPASS experiment at CERN*, In: Nucl. Instrum. Methods Phys. Res., A 577, 3 (2007) pp. 455–518

[2] Ch. Adolph et al. (The COMPASS Collaboration), *COMPASS-II Proposal*, CERN-SPSC-2010-014; SPSC-P-340, May 2010.

[3] I. Konorov, H. Angerer, B. Grube, W. Liebl, S. Paul, and L. Schmitt, *The trigger control system for the compass*, In: Nuclear Science Symposium Conference Record, 2001 IEEE, vol. 1, Nov 2001, pp. 98–99 vol.1.

[4] H. van der Bij, R. McLaren, O. Boyle, and G. Rubin, *S-link, a data link interface specification for the lhc era*, In: Nuclear Science, IEEE Transactions on, vol. 44, no. 3, pp. 398–402, Jun 1997.

[5] C. Gaspar, M. Dönszelmann, and P. Charpentier, *Dim, a portable, light weight package for information publishing, data transfer and inter-process communication*, In: Computer Physics Communications, vol. 140, no. 1–2, pp. 102 – 109, 2001, CHEP2000.