

New data acquisition system for the COMPASS experiment

This content has been downloaded from IOPscience. Please scroll down to see the full text.

2013 JINST 8 C02009

(<http://iopscience.iop.org/1748-0221/8/02/C02009>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 137.138.125.163

This content was downloaded on 08/07/2014 at 09:05

Please note that [terms and conditions apply](#).

RECEIVED: October 29, 2012

REVISED: December 6, 2012

ACCEPTED: December 20, 2012

PUBLISHED: February 5, 2013

TOPICAL WORKSHOP ON ELECTRONICS FOR PARTICLE PHYSICS 2012,
17–21 SEPTEMBER 2012,
OXFORD, U.K.

New data acquisition system for the COMPASS experiment

M. Bodlak,^a V. Frolov,^{b,c} V. Jary,^{a,1} S. Huber,^d I. Konorov,^d D. Levit,^d A. Mann,^d
J. Novy,^a S. Paul^d and M. Virius^a

^aCzech Technical University in Prague,
Žitkova 4, 166 36, Praha 6, Czech Republic

^bJoint Institute for Nuclear Research,
Joliot-Curie 6, 141980 Dubna, Moscow region, Russia

^cEuropean Organization for Nuclear Research - CERN,
CH-1211, Genève 23, Switzerland

^dPhysik-Department E18, Technische Universität München,
James-Frank-Straße, 85748 Garching, Germany

E-mail: Vladimir.Jary@cern.ch

ABSTRACT: The modern market offers low cost high performance FPGA integrated circuits equipped with dozens of multi gigabit serial links making them ideal devices for data transmission and data sorting applications. Therefore we have designed the new DAQ system that would perform the detector readout and event building in a custom made FPGA based hardware. The software part will provide the control and monitoring function. Currently, the prototypes of the new FPGA card are being tested and the control and monitoring software is being prepared for the tests with the real hardware.

KEYWORDS: Software architectures (event data models, frameworks and databases); Data acquisition concepts; Digital electronic circuits

¹Corresponding author.

Contents

1	Introduction	1
2	COMPASS experiment at CERN	2
3	Existing data acquisition system of the experiment	2
3.1	Hardware of the system	2
3.2	Software of the system	2
4	New DAQ system	3
4.1	AMC module design	4
4.2	Firmware architecture	5
4.3	Software architecture	6
4.3.1	Core processes	7
4.3.2	Reporting facility	7
5	Summary and outlook	7

1 Introduction

Today, modern particle physics experiments produce data in quantities never seen before which poses strong requirements on the data acquisition (DAQ) systems. This paper focuses on the development of the new DAQ system for the COMPASS experiment at CERN [1].

After twelve years of successful data taking the COMPASS experiment requires a change of the obsolete PCI technology, which served as an interface between the front-ends and the online computers. An exchange of the basic DAQ modules also required a revision of the COMPASS event building architecture based on the Ethernet Gigabit network. The event building has a specific data transmission topology and time pattern that even the modern advanced switches handle inefficiently and require an external traffic shaping [2, 3]. At the same time, there is a long term trend in the FPGA technology of increasing serial link IO bandwidth and support for high performance SDRAM memories. Even the low cost FPGA families are equipped with several 3 Gbps links [4, 5]. Therefore it was decided to develop the new COMPASS DAQ using modern FPGAs.

Firstly, the experiment and the existing DAQ system is introduced. Then, an integrated solution of the FPGA based event builder and the control and monitoring software is presented. Finally, the current project status is summarized and the future steps in the development are outlined.

2 COMPASS experiment at CERN

COMPASS is the particle physics experiment with the fixed target situated at the Super Proton Synchrotron (SPS) particle accelerator at CERN in Geneva, Switzerland. The scientific program was approved by CERN in 1997; it covers the studies of the gluon and quark structure and the spectroscopy of hadrons using high intensity muon and hadron beams. After several years of preparations and commissioning, the process of taking physical data started in 2002. Recently, the experiment has entered its second phase known as the COMPASS-II [6] that focuses on the studies of the Primakoff scattering, the Drell-Yan effect, or the Deeply Virtual Compton Scattering (DVCS).

3 Existing data acquisition system of the experiment

The COMPASS experiment collects 1.5 GB of data per second during a 10 s spill which is followed by a break of 30–40 s depending on the SPS super cycle mode. The front-ends cannot store significant amount of data locally, thus they transmit all the collected information to the DAQ via approximately 100 S-Links at 160 MBps per link. This creates a long term traffic pattern; the random nature of particle interaction processes creates a short term pattern in the data acquisition. Therefore, the data transmission is highly coherent and has a very spiky burst structure saturating the serial links for a short period of time during the spill. COMPASS adopted a scheme of the data buffering for the entire spill followed by the event building during the full accelerator cycle, which is done over the Gigabit Ethernet network.

3.1 Hardware of the system

The DAQ system [7] consists of several layers. The first layer is the front-end electronics, it serves approximately 250 000 detector channels. The first part of the subevent building takes place in VME modules called CATCH and GeSiCA that act as the data concentrators. COMPASS uses 130 CATCH and GeSiCA modules. The data taking is synchronized by the Trigger Control System (TCS) [1] that distributes the reference time and the trigger and event identification information. The modules create subevents by combining the data blocks received from the attached subdetectors with a header provided by the TCS. Then, the subevents are transferred to the readout buffer (ROB) servers via the S-Link optical interfaces. Each of the 32 ROBs contains four Spillbuffer PCI cards equipped with 512 MB of memory. Finally, the subevents are moved through the switched Gigabit Ethernet network to 16 event building servers (known as event builders) which form the complete events. After being stored on hard disk, the data are copied to the CERN permanent storage CASTOR.

3.2 Software of the system

The DAQ system is powered by the DATE software package [8] that was originally developed for the ALICE experiment. The package performs the readout and event building tasks and also provides the run control, reporting, monitoring, and event sampling facilities to the distributed network environment. The DATE package was adapted for the use in the COMPASS experiment environment.

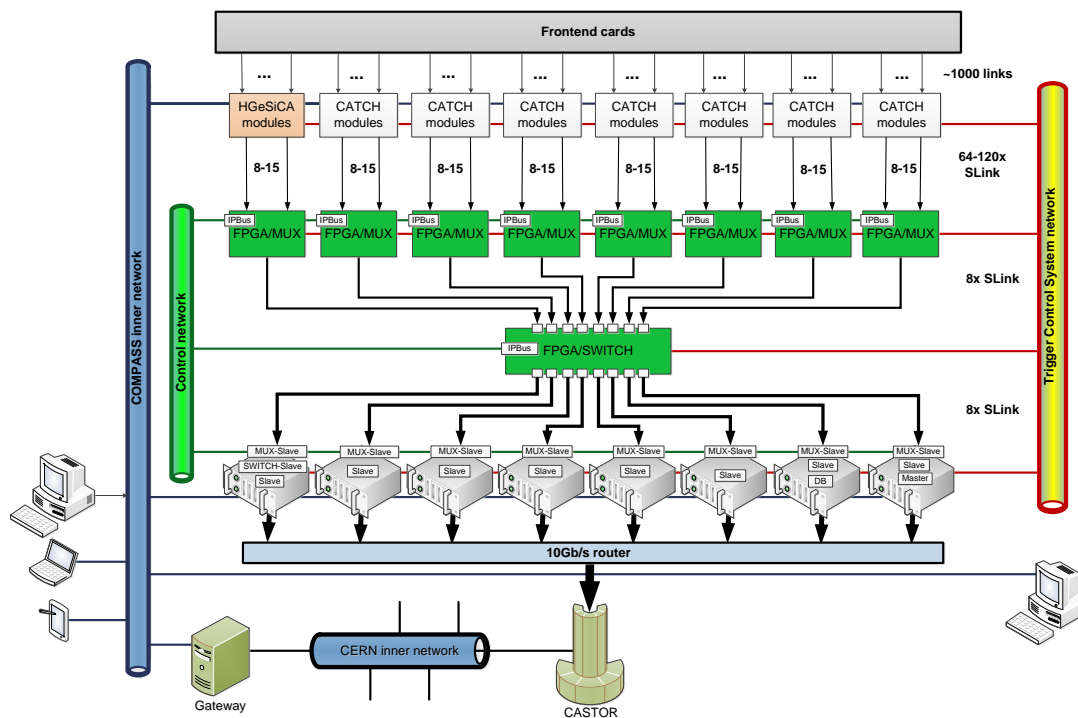


Figure 1. The new hardware and software architecture. The FPGA modules perform smoothing of the data rate and complete the event building. The control network serves for configuration and run control message exchange. The TCS network broadcasts the event IDs with fixed latency for the data synchronization.

4 New DAQ system

The event building functionality is implemented in nine newly developed FPGA modules. Each FPGA module has 16 programmable serial links with a maximum speed of 3.25 Gbps each and 2 GB of DDR3 memory. The architecture of the DAQ is shown in figure 1. The first eight modules are configured as multiplexers and reduce the number of serial links from 100 to 8. The combined memory size of the FPGA modules is 16 GB and is about the volume needed to store the data of the entire spill. The sustained data rate after the multiplexer is about 60 MBps per each link taking into account the SPS duty cycle. The final event building is performed with the same type FPGA card programmed as a 8×8 switch. The interfaces between the FPGA modules are implemented using a custom protocol running at maximum speed of 3.25 Gbps. The assembled events are sent sequentially to one of eight Inveium PCIe cards [9] mounted in the online computers. The system throughput is defined by the switch module bandwidth and is expected to be 3.2 GBps.

Each DAQ FPGA module, including the PCIe cards, is attached to the TCS and receives event ID information prior to receiving the corresponding data from the front-end electronics. This information is used for synchronization and recovery in the case of problems.

The configuration, monitoring, and data flow control implemented over a dedicated control network via the Ethernet based IPbus protocol [12] are other important aspects of the system. The control network provides access to internal registers and memory blocks of the FPGA modules.

4.1 AMC module design

The FPGA card has to fulfill the following requirements of the data multiplexer/switcher module:

- more than 10 high speed serial links with more than 2.0 Gbps bandwidth to be compatible with the HOLA S-Link standard,
- 200 MB of SDRAM per serial link,
- interface to the Trigger Control System,
- Ethernet interface for the slow control and the run control.

The FPGA module was designed in a compact AdvancedMC form factor with respect to other possible applications such as the PANDA TPC [10] and ATCA based event builder [11]. One ATCA carrier card may host up to eight of the FPGA modules. The optical transceivers are placed either on an Rear Transition Module (RTM) or on a dedicated full size AMC module. The FPGA modules in the COMPASS DAQ are expected to be placed close to the front-end electronics in different locations inside the experimental area, thereby an ATCA shelf would be an oversized standard. Therefore a special standalone carrier card was designed.

The functional diagram and an artistic view of the AMC card with the carrier card are shown in figure 2. After comparison of the FPGA chips from three main vendors (i.e. Xilinx, Altera, and Lattice), we have concluded that Lattice ECP3 FPGA LFE3-150 provides the optimum price/performance ratio for our application. This FPGA has 16 high speed serial links running upto 3.2 Gbps, a DDR3 memory controller with the clock frequency of 400 MHz (6.4 GBps), the 150 K logical units, and 6.8 Mb of block RAM.

The module is equipped with the main and the monitor FPGA of the same type in 1156 and 672 pins packages respectively. The main FPGA has 16 serial interfaces wired to the AMC connector and a DDR3 SODIM socket for a 2 GB memory module. A Si5338 chip synthesizes 4 low jitter reference clocks for the serial links. The main FPGA generates a 400 MHz clock, derived from a 100 MHz oscillator clock, for the SDRAM. The main FPGA is exclusively dedicated to the buffering. The monitor FPGA has two Ethernet and one TCS interfaces which are also wired to the AMC connector. The TCS information is broadcasted via the serial interfaces; it is biphase mark encoded at 155.52 MHz. The monitor FPGA firmware is loaded from a parallel flash S29GL064N at power up, while the main FPGA is loaded by the monitor FPGA afterwards from the second bank of the same flash memory. When both FPGAs are loaded, the monitor FPGA controls the main FPGA via a parallel bus which extends the IPbus internal interface.

The AMC module is housed on the COMPASS carrier card. The carrier card provides the following external interfaces:

- 2×8 SFP+ optical transceivers of data transmission network,
- SFP+ for optical or RJ45 Ethernet control network using IPbus protocol,
- SFP+ optical receiver for the TCS.

The module is placed in a 19 inch box for rack mounting. Two COMPASS carrier cards can be fixed within one box.

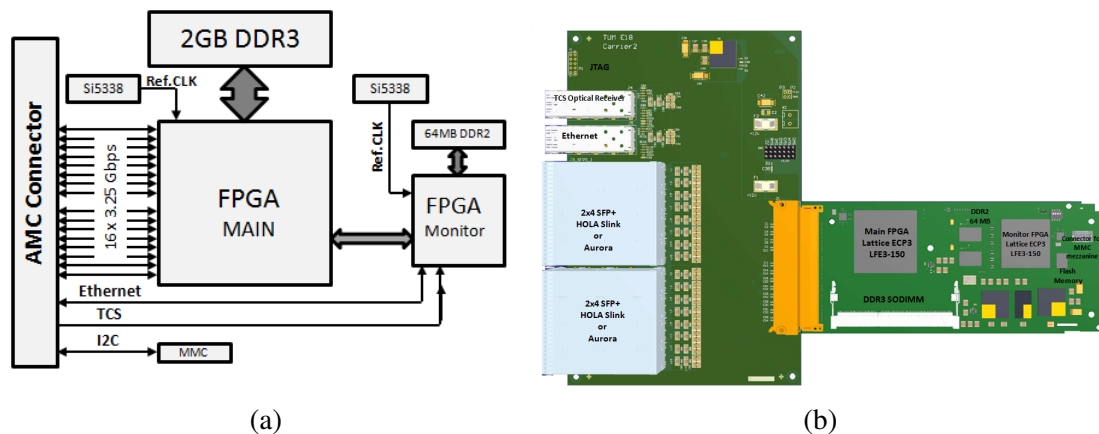


Figure 2. The AMC module functional diagram (a) and artistic view of the module with the standalone carrier card (b) as it will be used in COMPASS.

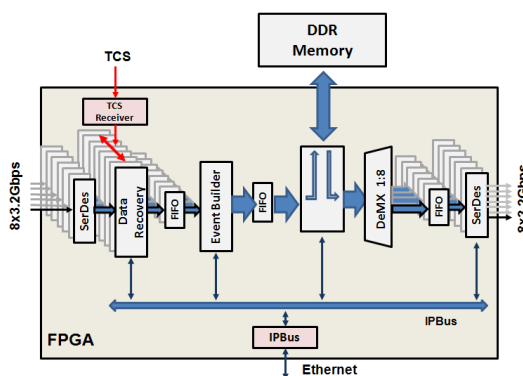


Figure 3. The FPGA 8×8 switch architecture. The data are checked and then stored to the DDR3 memory organized as multiple FIFOs.

4.2 Firmware architecture

A simplified diagram of the firmware architecture of the switch is shown in figure 3. The maximum switch size that can be implemented in the current FPGA is 8×8 . The multiplexer is another case of the switch, with a single output, which can be as large as 15×1 . The data processing chain is similar for both types of modules and differs only by number of inputs and outputs.

The TCS distributes 155.52 MHz clock together with the biphasic mark encoded trigger, the accelerator spill structure, and the event identification information. The TCS receiver utilizes the LatticeEPC3 deserializer, which supports the SONET STS-3 physical layer. The receiver decodes the event ID information, the start of the burst (SOB) and the trigger signals. The trigger time is measured locally within the FPGA relatively to the SOB. This information is provided to the data recovery logic blocks. The feature of the TCS to distribute any information synchronously can be used for changing the event building network topology on the fly by switching on and off certain system components or changing the data paths around faulty port or online computer. This feature will be used at COMPASS in case redundant hardware modules are added to the system.

The data recovery logic block provides a data repair procedure in case of errors. Information about detected errors is collected in error registers until it is read out by the run control. The errors that can be detected by the data recovery logic are the following:

- the transmission errors detected either by the S-Link or by the custom protocol,
- the truncation errors, which can occur in the front-end electronics when buffers overflow; in this case there is a mismatch between the declared and the real data block size,
- the inconsistency of the event ID, a mismatch between the actual and the expected event ID and/or the event time tag,
- the missing data, i.e. data are not received within a programmable time interval.

All the data blocks with errors are dropped during the normal data taking and the empty blocks with correct event ID information are generated instead with the exception of a block with truncation error which is recovered by adding a header with real data block size. The recovery mechanism guarantees the data consistency and the decodable data structure. In the DAQ test mode the recovery logic can be configured to accept data blocks with any type of errors.

The SDRAM memory is organized as sizeable, multiple, FIFO-like buffers, where the number of buffers is equal to the number of the incoming data links. The buffer size is programmable in steps of 2 MB and allows arranging the memory sharing proportionally to the amount of the transmitted data. The memory controller supports a 2 GB single rank memory module. The memory runs at 400 MHz clock speed and provides a theoretical bandwidth limit of 6.4 GBps. The memory bandwidth exceeds the total maximum data rate of the incoming links of 2.4 GBps and guarantees lossless data transmission. The memory access is optimized for the speed with a simple algorithm implemented in the arbiter logic. If all ports have the same priority, the access is granted sequentially to each port. If a port does not have the highest currently existing priority or if there are no data to transmit, it is skipped. The priority logic is different for the multiplexer and the switch. In case of the multiplexer the priority is given to the incoming data to avoid data losses. In case of the switch it is given to the outgoing links in order to keep the online computers permanently loaded at maximum, because they have a lower bandwidth than the switch.

The location of the (sub)event building logic is one important difference between the multiplexer and switch firmware. In the multiplexer the event building is done after the memory, while in the switch the event building is done before.

Configuration of the FPGA, monitoring, and data flow control is implemented by several processes running on general purpose PCs inside the control network. The first version of the control network support in the FPGAs was implemented using Mico32 and uBlaze softcore CPU for Lattice and Xilinx FPGA respectively. Recently it was switched to a simpler and unified solution of the IPbus protocol.

4.3 Software architecture

The initial idea was to deploy most of the software nodes directly on the FPGA softcore CPUs, however after adopting the IPbus protocol all the processes were transferred to standard server machines.

The control and monitoring software for the new DAQ architecture is a multilayer system centered around the master process which acts as a run control process. The multi-platform DIM library [13], that was originally developed for the transportation of messages in the DAQ of the DELPHI experiment at CERN, is used for the communication between the master process and other processes. DIM distinguishes two types of processes — servers and clients. Servers “publish” their services by registering them with the name server (usually once, at start-up). Clients “subscribe” to these services by asking the name server which server provides the actual service and then contact the server directly, providing the type of service as a parameter. The name server keeps an up-to-date directory of all the servers and services available in the system. Each process in the DAQ acts as a DIM server and a DIM client at the same time.

4.3.1 Core processes

The main DAQ chain incorporates four types of processes. The slaves for the control and monitoring of FPGA modules are the first type of processes. These slaves can access the memory and registers of the connected FPGA module through the IPbus. The readout slave process is the second type; this process must be deployed on every readout computer. It serves for monitoring and control of these computers and for readout and storing the full events from the spill-buffer cards.

The master is the most important process; it is the only process that has a full access to the configuration database. It receives all the status information from the slaves and propagates it to a Graphical user interface (GUI). The master process also contains almost all application logic. The system is controlled by state machines implemented in the master and all the slave processes. The GUI is the last type of core process; it is being developed using the Qt framework [14]. Modularity of the GUI is a matter of great importance, thus users should not be overwhelmed by the amount of information. On the other hand, experts should still be able to find details they need for the error identification.

4.3.2 Reporting facility

Several software nodes are used for monitoring of the system. The master and all the slave processes are able to generate informative and error messages. These messages are sent to the message logger process that evaluates them and eventually stores them into the online database. The message logger uses the DIM library for communication with the other processes of the system.

Any message stored by the message logger can be viewed using the message browser application which also provides an intuitive interface for the message filtering.

5 Summary and outlook

The new COMPASS DAQ will significantly reduce number of system components and online computers which will consequently improve the reliability of the system. In addition, the new DAQ architecture is based on low cost FPGA modules and can be scaled up to provide extra resources for redundancy.

The hardware of the system has been tested and the modules are fully functional. The firmware of the system is still being developed, although the functionality of several building blocks like SDRAM memory controller, serial link protocols, TCS receiver, and IPbus has already been tested.

The new DAQ will be extensively tested and commissioned during the shutdown of the CERN accelerators in 2013. It will be employed in the COMPASS data taking since 2014.

Acknowledgments

The work on this project has been supported in part by the following grants: MŠMT LA08015 and SGS 11/167. It is being also supported by the Maier-Leibnitz-Labor, Garching and the DFG cluster of excellence “Origin and Structure of the Universe”.

References

- [1] COMPASS collaboration, P. Abbon et al., *The COMPASS experiment at CERN*, *Nucl. Instrum. Meth. A* **577** (2007) 455 [[hep-ex/0703049](#)] [[INSPIRE](#)].
- [2] T. Uchida et al., *New Communication Network Protocol for a Data Acquisition System*, *IEEE Trans. Nucl. Sci.* **53** (2006) 286.
- [3] J. Adamczewski-Musch, H. Essel, N. Kurz and S. Linev, *Dataflow engine in DAQ backbone DABC*, *IEEE Trans. Nucl. Sci.* **57** (2010) 614 [[INSPIRE](#)].
- [4] *Spartan-6 Family Overview* (2012), http://www.xilinx.com/support/documentation/data_sheets/ds160.pdf.
- [5] *LatticeECP3 EA Family Data Sheet* (2012), http://www.latticesemi.com/dynamic/view_document.cfm?document_id=31998.
- [6] COMPASS collaboration, *COMPASS-II Proposal*, [CERN-SPSC-2010-014](#) (2010).
- [7] L. Schmitt et al., *The DAQ of the COMPASS experiment*, in the proceedings of *13th IEEE-NPSS Real Time Conference 2003*, Montreal Canada, 18–23 May 2003, pg. 439.
- [8] ALICE DAQ PROJECT, *ALICE DAQ and ECS User’s Guide*, CERN EDMS 616039, [CERN-ALICE-INT-2005-015](#) (2006).
- [9] *TB-5V-LX50T/110T/SX50T-PCIEXP User’s Manual*, Tokyo Electron Device Limited, x2bdg@teldevice.co.jp.
- [10] M. Ball et al., *Technical Design Study for the PANDA Time Projection Chamber*, [arXiv:1207.0013](#) [[INSPIRE](#)].
- [11] A. Mann et al. *An AdvancedTCA Based Data Concentrator and Event Building Architecture*, in the proceedings of *IEEE-NPSS Real Time Conference*, Lisbon Portugal (2010), pg. 1.
- [12] *The CACTUS project*, February 2013, <https://cactus.hepforge.org/>.
- [13] C. Gaspar and M. Dönszelmann, *DIM — A Distributed Information Management System for the DELPHI Experiment at CERN*, *Proceedings of the 8th Conference on Real-Time Computer applications in Nuclear, Particle and Plasma Physics*, Vancouver Canada (1993).
- [14] J. Blanchette and M. Summerfield, *C++ GUI Programming with Qt 4 (2nd ed.)*, Prentice Hall, Qt preview for Symbian S60 (2008).