

Design and Commissioning of the ATLAS Muon Spectrometer RPC Read Out Driver

A. Aloisio^{a,b}, L. Capasso^{a,b}, F. Cevenini^{a,b}, M. Della Pietra^{b,c}, D. Della Volpe^{a,b} and V. Izzo^b
on behalf of the ATLAS MUON collaboration

^aUniversità di Napoli “Federico II”, Dipartimento di Scienze Fisiche, Via Cintia - 80126 Napoli, Italy

^bI.N.F.N. Sezione di Napoli, Dipartimento di Scienze Fisiche, Via Cintia - 80126 Napoli, Italy

^cUniversità di Napoli “Parthenope”, Dipartimento per le Tecnologie, Centro Direzionale - 80143 Napoli, Italy

Abstract

The RPC subsystem of the ATLAS muon spectrometer provides the Level-1 trigger in the barrel and it is read out by a specific DAQ system. On-detector electronics pack the RPC data in frames, tagged with an event number assigned by the trigger logic, and transmit them to the counting room on optical fibre. Data from each sector are then routed together to a Read-Out Driver (ROD) board. This is a custom processor that parses the frames, checks their coherence and builds a data structure for all the RPCs of one of the 32 sectors of the spectrometer. Each ROD sends the event fragments to a Read-Out subsystem for further event building and analysis. The ROD is a VME64x board, designed around two Xilinx Virtex-II FPGAs and an ARM7 microcontroller. In this paper we describe the board architecture and the event binding algorithm. The boards have been installed in the ATLAS USA15 control room and have been successfully used in the ATLAS commissioning runs.

I. INTRODUCTION

THE Large Hadron Collider (LHC) is a proton-proton collider, with a centre of mass energy of 14 TeV, which started operation in Sept. 2008 at CERN. The ATLAS experiment has been installed at one of the four LHC's beam interaction points. ATLAS is an “all-purpose” detector designed to have almost a 4π geometry around the interaction vertex and it has a cylindrical symmetry along the beam axis; it is made of different sub-detectors, each with its own dedicated read-out electronics. Its inner tracking detector is located inside a 2 T axial magnetic field; outside there is a liquid argon electromagnetic calorimeter and a hadronic calorimeter using scintillating tiles in the barrel and liquid argon in the endcaps. In the outer region, the muon spectrometer is instrumented with precision measurement chambers and trigger chambers. The bending magnetic field in the muon spectrometer is generated by an air-core toroid made by 8 coils.

Resistive Plate Chambers (RPC) are used both for trigger and readout purposes in the barrel region (figure 1). The RPC chambers are arranged in projective towers to form three cylinders concentric with the beam axis and have a 16-fold segmentation in the azimuthal plane, following the eightfold azimuthal symmetry of the magnetic structure. The whole Muon Spectrometer barrel is divided into 32 physical sectors (16 within each half barrel).

In order to reduce the interaction rate from 1 GHz to 100 Hz, the ATLAS trigger has been designed with a three

level architecture. The raw rate of 1 GHz proton-proton interactions is reduced to 75 kHz by the Level-1 trigger system, that also flags interesting events with a *Level-1 Accept* signal (L1A) with a fixed latency of 2.5 μ s.

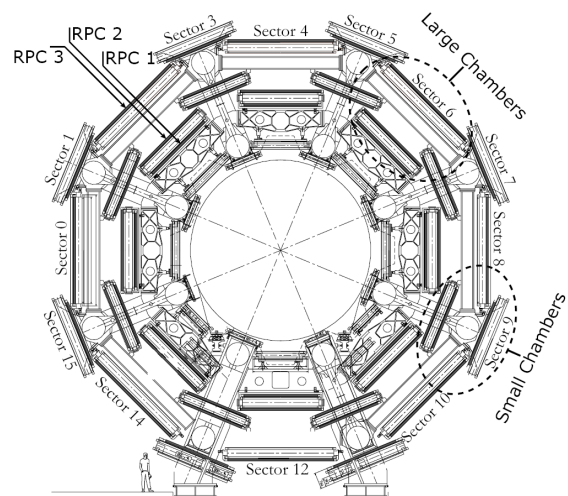


Figure 1. ATLAS barrel muon spectrometer view in the azimuthal plane, showing RPC Trigger stations.

II. THE ATLAS SYNCHRONIZATION SYSTEM

In the LHC collider [1], protons are grouped in “bunches” that interact every 25 ns. From the point of view of the trigger system, ATLAS is a synchronous system working at 40 MHz, the *bunch crossing* frequency of the LHC. All the trigger electronics are driven by a common clock signal, synchronized with the *bunch crossing* frequency of the collider. Different from the trigger, the DAQ dataflow does not have a fixed latency and can be considered asynchronous with respect to the bunch crossing. However, the RPC DAQ system is also synchronized with the same clock, in order to share resources with the L1 trigger and optimize the routing of the distribution system.

Front End electronics associate the data in each event to a number identifying the bunch crossing that generated the collision (*Bunch Crossing Identifier*, or BCID) and to a unique progressive number (*Event Identifier*, or EVID) identifying that event. The BCID is incremented every LHC's clock cycle and the EVID is incremented every time a L1A pulse is generated by the trigger processor and the; both these numbers are managed by counters on the front end boards synchronously to the LHC's clock. In

order to initialize and handle these two identifiers (i.e. EVID and BCID), two signals are transmitted: the *Event Counter Reset (ECR)* is issued on request by one of the subsystems of the ATLAS apparatus, if a malfunction has occurred; the *Bunch Counter Reset (BCR)* is transmitted periodically after every orbit, in order to rewind the BCID counters.

Because of the large dimensions of the detector, the reference clock, the LIA, the ECR and the BCR signals must be transmitted over distances up to hundreds meters. In order to allow the synchronization of the DAQ systems with the machine clock, a physical layer has been chosen, able to distribute the clock and control signals to all elements of the ATLAS apparatus, with programmable *skew* and low *jitter*. This is made by the *Trigger Control System (TCS)* and the *Timing Trigger and Control (TTC)* system.

The *TCS* is in charge of the generation of the trigger and timing signals. The *TTC* system is responsible for their distribution to the entire detector. The *TTC* receives from the *TCS* the 40 MHz clock signal, the *LIA*, *BCID*, *EVID*, *BCR*, *ECR* and other service signals. All these signals are coded and optically transmitted, over a tree structure. At the destination, the receiver board *TTCrq* [2] reconstructs the signals and adapts them to the protocols of every sub-detector.

III. THE RPC READ-OUT SYSTEM

The trigger and RPC readout system of the ATLAS' muon spectrometer is split between on-detector and off-detector sections. The level-1 muon trigger in the barrel region is based on a fast geometric coincidence [3] between different planes of the RPC detectors. On-detector electronics execute, every 25 ns, the trigger algorithm and send, via optical links, the relevant trigger information to the off-detector Trigger Processor, that can validate or reject the event with a fixed latency of 2.5 μ s.

Data produced by the RPC detectors are written in *FIFO* memories on the *on-detector* electronics and are kept in the buffers during the decision time of the Trigger processor. If an event is accepted by the first level trigger, a *LIA* pulse is generated and transmitted across the *TTC* system together with the pertinent *EVID* and *BCID*. After the arrival of the *LIA* pulse, data stored in the *FIFO* buffers are transferred to the off-detector electronics over the same optical link used for the transmission of trigger information. Trigger and read-out data of each of the 32 sectors of the spectrometer are managed by a *Read Out Driver (ROD)* crate (see figure 2).

The main task of the *RX-SL* boards is to receive and elaborate trigger and *read-out* data from the on-detector electronics. The *RX/S�* boards pre-process the trigger information and sent them to the Trigger Processor through the Muon Central Trigger Processor Interface board (μ CTPI) [4], across a custom backplane. The *RX/S�* boards also arrange readout data in an event frame (*RX Frame*) and transmit them to the adjacent *Read Out Driver* across a custom backplane *RODbus* via a high speed serial link.

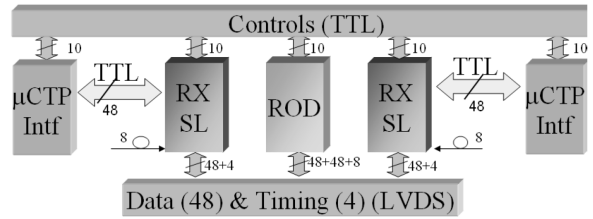


Figure 2: Scheme of the crate that hosts the ROD board.

The main task of the ROD is to perform a further framing of the readout data, before transmitting them across the optical link S-Link to the next acquisition levels, i.e. to the *Read Out Systems (ROS)*. The ROD also manages the timing signals of the trigger and DAQ system. For this purpose, the ROD hosts a *TTCrq* receiver module from which it receives the ATLAS' timing and control signals to be forwarded to the *RX/S�* boards on the *RODbus*.

On the *RODbus*, data and timing signals are transmitted in LVDS standard to achieve high rate, low *skew* and *jitter*; the serial links between each *RX/S�* and ROD have an aggregate bandwidth of ~ 2 Gbit/s (48bit@40MHz) [5]. Control signals run at lower rate and are transmitted using the TTL standard. The *RODbus* also hosts a 48-bit TTL bus that allows the *RX/S�* boards to transmit trigger data to the μ CTPI boards.

The ROD boards developed for the other ATLAS sub-detectors [6, 7] implement different logic and functionalities, in order to fulfil the specific detector requirements. However, they all share the same logical output format and optical physical layer. In this way, the ROS design is unique for the entire apparatus, making it possible to use the same architecture for the higher level DAQ systems [8].

A ROS unit is implemented with a 3.4 GHz PC housing 4 custom PCI-x cards (ROBIN), each hosting 3 S-Link optical receivers (Read Out Buffers). Therefore, a ROS unit receives and stores up to 12 event fragments from different channels of the ATLAS detector. Such fragments can be forwarded to the Event Builder, if the event is accepted by the Level-2 trigger, otherwise it will be dropped. The Event Builder is in charge of merging the event fragments coming from the ROS into a full event.

IV. THE READ OUT DRIVER BOARD

The ROD (figure 3) has the form factor of the VME 64X 6U board and is equipped with two VIRTEX II XILINX FPGAs, labelled as *VME FPGA* and *ROD FPGA*. The board also hosts an ARM7 microcontroller, the *TTCrq* receiver, the S-Link transmitter and the two deserializers (*RX SerDes*) that receive data via *RODbus* backplane from the *RX/S�* boards.

The main task of the *VME FPGA* is to interface the whole board with the *VMEbus*; the *VME FPGA* allows a user to access the *ROD FPGA* memory locations and configuration registers and to read the microcontroller's

data. The *VME FPGA*'s clock is obtained from an on-board 40 MHz oscillator multiplied by 2 by the internal Digital Clock Manager.

The *ROD FPGA* performs the *event building* algorithm on data transmitted by the *RX/SL* boards. The *ROD FPGA* also hosts registers for the configuration and control of the event builder engine. In the same fashion as the *VME FPGA*, the *ROD FPGA* clock is obtained multiplying by 2 the 40 MHz board clock. The *ROD FPGA* receives 48-bit words and the recovered 40 MHz clock from each *RX SerDes*. It is also interfaced with the *TTCrq* module - from which it receives the *TTC* timing signals and the 40 MHz *LHC*'s clock - and to the *S-Link* transmitter, that is fed by a 40 MHz clock derived by the 80 MHz internal clock.

Microcontroller VME FPGA RX SerDes ROD FPGA

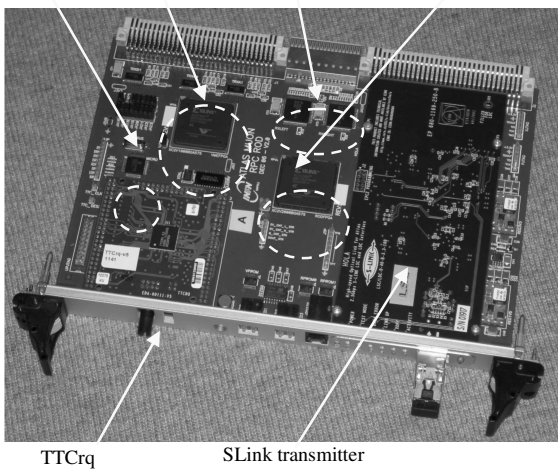


Figure 3: A photo of the *ROD* board.

Figure 4 shows a simplified block diagram of the *ROD* board. The *ROD FPGA* communicates with the *VME FPGA* via a serial synchronous custom protocol, carried out by two point-to-point unidirectional lines with a data rate of 80 Mbit/s. The main advantages of a serial link are a simpler PCB layout, the use of a small number of *FPGA* pins and limitations of *ground bounce* effects.

The *VME FPGA* is the *Master* of the serial link, managing both the write (for data and for address) and read operations. As a consequence, the *ROD FPGA* can transmit data only if the *VME FPGA* has previously requested them. The serial protocol allows the user to set once an 8-bit target address and then to perform an arbitrary number of 32-bit read or write accesses to the selected location in the *ROD FPGA*.

The main task of the *ARM7* microcontroller is to program the *TTCrq* receiver, via *I²C* protocol. This makes it possible to access all the *TTCrq* registers, both for configuration and monitoring purposes. The microcontroller also allows reading, via the internal *ADC*, the three power supplies on the *ROD* board (5V, 3.3V, 1.5V). The power supply and temperature on the *RODbus* are acquired from a remote *ADC* installed on the backplane via an *I²C* bus. The microcontroller's output data can be read via a *RS232* port or can be redirected on

the *VMEbus*, through a 16-bit parallel bus handled by the *VME FPGA*.

The *ROD* board is the meeting point of trigger signals and different readout data streams from the *Muon Barrel Spectrometer*. Besides the internal 80 MHz *FPGA* clocks, the 40 MHz *LHC* clock, the two 40 MHz *SerDes* clocks and the 40 MHz *S-Link* clock run all over the board. Even if these clock signals have the same frequency, they have an unpredictable phase relationship and should be handled as domains asynchronous to each other. All these clocks are present in the *ROD FPGA*, which is the most complex and critical section of the board. In order to decouple the clock domains and to guarantee their coexistence on the *ROD FPGA*, *FIFO* memories have been extensively used.

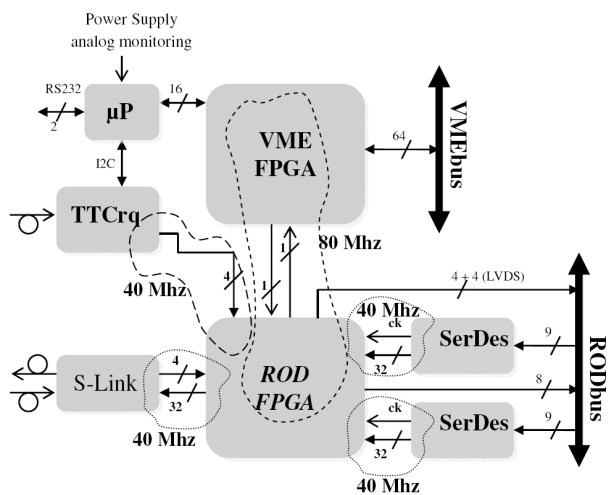


Figure 4: Block diagram of the *ROD* board.

V. THE EVENT BINDING ALGORITHM

The dataflow in the *ROD FPGA* is shown in figure 5. Input data (coming from a specific *RX/SL* board) are stored in the corresponding *FIFO* (*RX SerDes FIFO*). *EVID* data from *TTC* are stored in the *EVID FIFO*. Event builder output data are stored in the *S-Link FIFO* and then read out by the *S-Link* transmitter and sent across the optical link to the *ROS*. The transfer protocol with the transmitter *S-Link* module is implemented in the *ROD FPGA*.

The basic algorithm of the *Event Builder Engine* is shown in figure 6. The *Event Builder Engine* has been designed as a cluster of *Finite State Machines*, in order to guarantee real-time performance.

The *ROD MUON FRAME* produced by the engine is made of 32-bit words [9]. The frame starts with a *ROD Header* (pertinent to a specific *EVID* value), includes as a payload the frames coming from the *RX/SL* boards and ends with a *Footer* containing status and error flags.

The received *RX/SL* data are formatted in frames that are made of a *RX Header*, a certain number of data words (*payload*) and a *Footer*. The *Header* contains the *EVID* and *BCID* of the event. The *Footer* contains a control code.

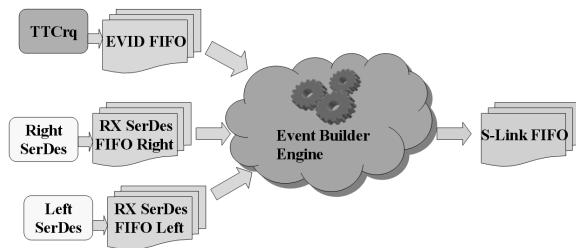


Figure 5: The dataflow in the ROD FPGA

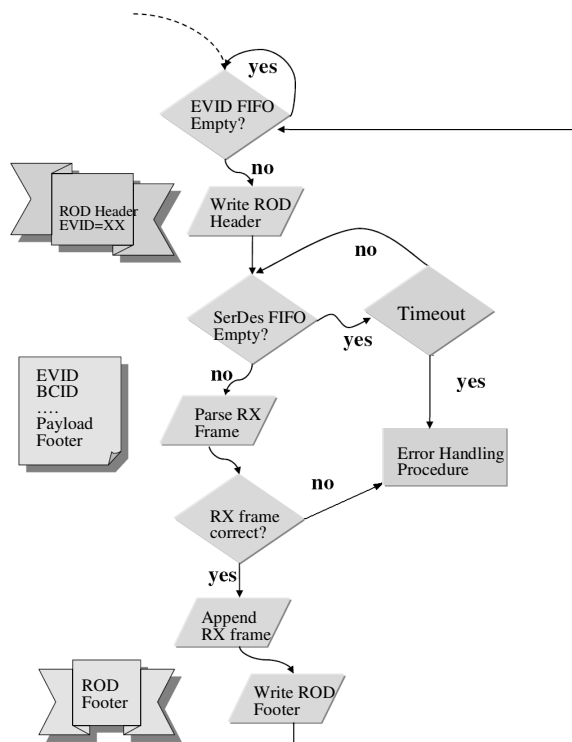


Figure 6: Basic algorithm of the Event Builder Engine.

Every *ROD Muon Frame* is relative to a specific EVID value received by the TTC. The *Event Builder Engine* checks the empty flag of the EVID FIFO and waits for a EVID to be processed.

When a EVID is available, the *Event Builder engine* starts writing a valid header into the output S-Link FIFO. The *Header* contains nine control words, such as the *Start of Frame*, the board identifier code and information about the current EVID and BCID value.

Then the *Event Builder Engine* waits for data from the *RX/SL* boards. As in the previous step, the *Event Builder Engine* checks the empty flag of each *SerDes* FIFO. The received RX frames are parsed to find a *Header*. If the frame is correctly formatted and the embedded EVID and BCID match the current one, it is appended to the ROD frame. The ROD event builder does not inspect the payload of the RX frames: it only counts its total length.

The ROD frame is closed by a *Footer*, containing status words, error flags, the total word count and the

elapsed time to build the frame. Then the *Event Builder Engine* restarts and waits for a new EVID value.

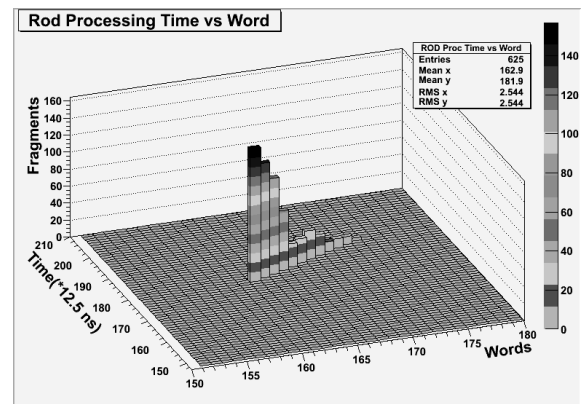


Figure 7: Histogram of the ROD processing time, as a function of the words in the *RX/SL* frame.

The timing information written in the ROD Frame Footer allow us to perform a real-time analysis on the elapsed time needed to build a frame. Figure 7 shows the histograms of the ROD processing time, as a function of the size of the *RX/SL* frames. This analysis has proved to be a very useful tool for the fine tuning of the ROD Event Builder Engine, giving us information about the average time to build an event or a timeout occurrence. In the plot, the linearity between the ROD processing time and the size of the *RX/SL* frame is clearly visible.

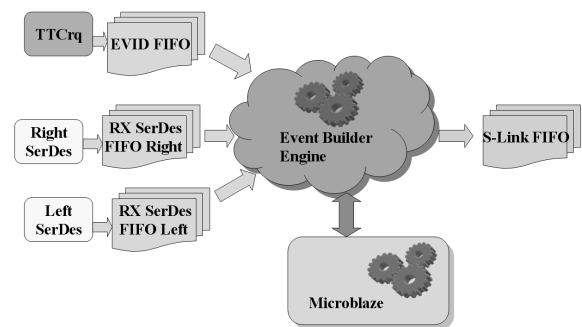


Figure 8: The MicroBlaze connection to the Event Builder.

A data analysis architecture, based on the Xilinx MicroBlaze embedded microprocessor, is under development in our laboratories in Napoli. It can be downloaded in the ROD FPGA and it has been specifically designed in order to control the Event Builder performance. With this data analysis tool, the status and the timing of the Event Builder FSM can be monitored. Presently, even using such MicroBlaze-based data analysis architecture, about 35% resources in the ROD FPGA are still available for further improvement of the Event Builder logic. The timing performance of the FPGA are unchanged. The Event Builder engine with MicroBlaze is shown in figure 8.

VI. THE COMMISSIONING OF THE READ OUT DRIVERS

The ROD boards have been fully tested in the stand alone mode in I.N.F.N. laboratories in Napoli and then installed at CERN in the USA15 counting room. The RPC ROD crates in the USA15 counting room are shown in Figure 9.

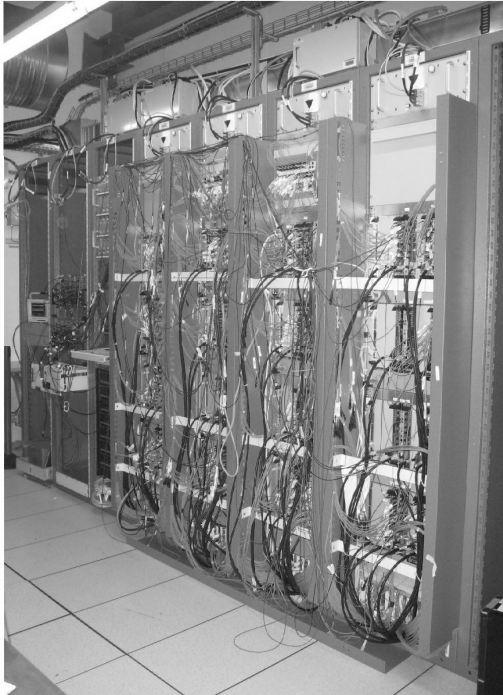


Figure 9: The RPC ROD crates in the USA15 counting room.

The RODs have been successfully tested also in the ATLAS data acquisition environment. The testbench comprised the entire chain, with a ROD board, *RX/SL* boards and μ CTPI boards, connected via a *ROD*bus backplane. *RX* frames have been correctly acquired and processed by the ROD which also was in charge to distribute TTC timing signals on the *ROD*bus.

In our tests in the ATLAS DAQ, we did not encounter any errors. The impact on the builder logic of corrupted frames have been evaluated in *stand alone* mode by writing frames with missing or wrong fields in the *SerDes* FIFOs.

The boards have been successfully used in the ATLAS commissioning runs. In this phase, several millions cosmic muons have been acquired through the Read-Out Driver without any failure of the event building logic. During several tests of the DAQ system with random triggers, a L1A rate of 100 KHz has been reached, thus fulfilling the ATLAS specifications for the L1A rate.

VII. CONCLUSIONS

We summarized the design and the development of the ROD board for the RPC *read-out* system of the ATLAS muon spectrometer. The ROD is a VME board that receives data generated by one of the 32 sectors of the

spectrometer. The ROD verifies their logical and syntactical coherence and builds a Frame which is then transmitted on optical fibre to the next level of the DAQ system.

The framing engine fulfils all the ATLAS requirements. Additional features have been designed in order to allow the user to obtain information about events, errors occurred and to check the internal status of the hardware. Some specific fields have also been included in the ROD frame, in order to perform timing analysis on the ROD Event Builder performance. A data analysis architecture, based on the Xilinx MicroBlaze embedded microprocessor, presently being developed.

The RODs have been tested and installed in the ATLAS USA15 counting room in the early 2008. They have been successfully used in the ATLAS commissioning runs and several millions of cosmic muons have been acquired without any failure.

REFERENCES

- [1] B.G. Taylor, "Timing Distribution at the LHC", presented at the 8th Workshop on Electronics for LHC Experiments, Colmar, September 2002. [Online] Available: <http://ttc.web.cern.ch/TTC/intro.html>
- [2] P. Moreira, "TTCrq Manual". [Online] Available: <http://proj-qpll.web.cern.ch/proj-qpll/images/manualTTCrq.pdf>
- [3] V. Bocci et al., "The Coincidence Matrix ASIC of the Level-1 Muon Barrel Trigger of the ATLAS Experiment", *IEEE Trans. Nuclear Science*, Vol.50, no.4, 2003
- [4] G. Schuler, *The MICTP Module of the Muon CTP Interface Demonstrator User's Guide*. [Online] Available: <https://edms.cern.ch/file/340643/1/mictp.pdf>
- [5] *DS90CR483 / DS90CR484 48-bit LVDS Channel Link SER/DES datasheet*.
- [6] I. Efthymiopoulos, "The readout driver (ROD) for the ATLAS liquid argon calorimeters", *Nucl. Instrum. Meth. Phys. Res.*, A 461, (2001), pp. 481-482
- [7] H. Boterenbrood et al., "The read-out driver for the ATLAS MDT muon precision chambers", *IEEE Trans. Nuclear Science*, Vol. 53, 2006, Issue 3, Part 1, pp. 741-748
- [8] M. Abolins et al., "Integration of the Trigger and Data Acquisition Systems in ATLAS", *IEEE Trans. Nuclear Science*, Vol. 55, 2008, Issue 1, Part 1, pp. 106 - 112
- [9] C. Bee et al., "The raw event format in the ATLAS Trigger & DAQ", ATLDAQ-98-129. [Online] Available: <http://atlas.web.cern.ch/Atlas/GROUPS/DATABASE/project/event/TDAQ-eventformat-0019v24.pdf>