RUN CONTROL COMMUNICATION FOR THE UPGRADE OF THE ATLAS MUON-TO-CENTRAL-TRIGGER-PROCESSOR INTERFACE (MUCTPI)

R. Spiwoks[†], A. Armbruster, G. Carrillo-Montoya, M. Chelstowska, P. Czodrowski, P.-O. Deviveiros, T. Eifert, N. Ellis, P. Farthouat, G. Galster[1], S. Haas, L. Helary, O. Lagkas Nikolos, A. Marzin, T. Pauly, V. Ryjov, K. Schmieden, M. Silva Oliveira, J. Stelzer, P. Vichoudis, T. Wengler, CERN, Geneva, Switzerland
[1]also at Niels Bohr Institute, Copenhagen, Denmark

Abstract

The Muon-to-Central-Trigger-Processor Interface (MUCTPI) of the ATLAS experiment at the Large Hadron Collider (LHC) at CERN will be upgraded to an ATCA blade system for Run 3, starting in 2021. The new design requires development of new communication models for control, configuration and monitoring. A System-on-Chip (SoC) with a programmable logic part and a processor part will be used for communication to the run control system and to the MUCTPI processing FPGAs. Different approaches have been compared. First, we tried an available UDP-based implementation in firmware for the programmable logic. Although this approach works as expected, it does not provide any flexibility to extend the functionality to more complex operations, e.g. for serial protocols. Second, we used a SoC processor with an embedded Linux operating system and an application-specific software written in C++ using a TCP remote-procedure-call approach. The software is built and maintained using the framework of the Yocto Project. This approach was successfully used to test and validate the MUCTPI prototype. A third approach investigated is the option of porting the ATLAS run control software directly to an embedded Linux instance.

THE ATLAS EXPERIMENT AT THE LHC

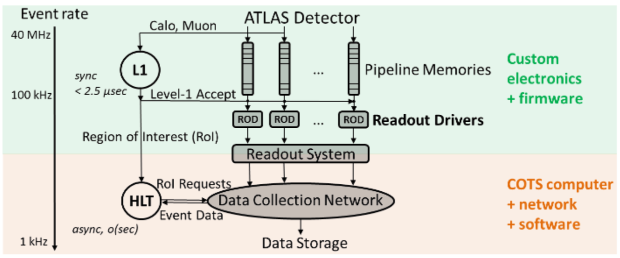

Figure 1: The Trigger and Data Acquisition System of ATLAS.

The ATLAS experiment [1] is a general-purpose experiment at the Large Hadron Collider (LHC) at CERN. It observes proton-proton collisions at a centre-of-mass energy of 13 TeV. With about 25 interactions in every bunch crossing (BC) every 25 ns, there are 10^9 interactions per second potentially producing interesting physics. The trigger system selects those events which are interesting to physics and which can be recorded to permanent storage at a reasonable rate. The ATLAS trigger system (see Figure 1) consists of a Level-1 trigger, based on custom electronics and firmware, which reduces the event rate to a maximum of 100 kHz, and a high-level trigger system based on commercial-off-the-shelf computers, network components, and software which reduces the event rate to around 1 kHz.

THE LEVEL-1 TRIGGER SYSTEM

The Level-1 trigger system (see Figure 2) uses reduced-granularity information from the calorimeters and dedicated muon trigger detectors. The trigger information is based on multiplicities and topologies of trigger candidate objects. The muon trigger is based on Resistive Plate Chambers (RPC) in the barrel region and Thin-Gap Chambers (TGC) in the end-cap region. The Muon-to-Central-Trigger-Processor Interface (MUCTPI) [2] combines the muon candidate counts from the RPC and TGC taking into account double counting of single muons that are detected by more than one chamber due to geometrical overlap of the chambers and the trajectory of the muon in the magnetic field. It sends the results to the Central Trigger Processor (CTP) which combines all trigger object multiplicities from the calorimeter trigger and from the MUCTPI, as well as the topology flags from the Topological Processor to make the final Level-1 decision based on rules described in a trigger menu. The CTP then sends the Level-1 decision back to the detector front-end electronics.

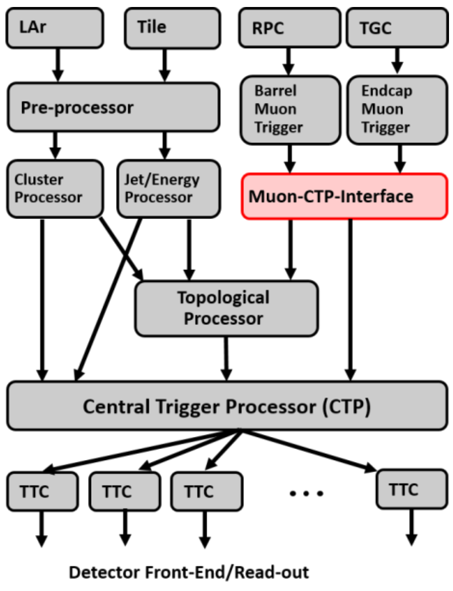

Figure 2: The Level-1 Trigger System of ATLAS.

† Ralf.Spiwoks@cern.ch

THE UPGRADE OF THE MUCTPI

The MUCTPI upgrade is part of the overall upgrade of ATLAS on the road to High-Luminosity LHC and is in line with the development of the New Small Wheel [3] of the muon trigger system to be installed during the shutdown of 2019 and 2020. The new MUCTPI will use optical links replacing bulky electrical cables. Those links will allow the muon trigger detectors to send more muon candidates with more precise information. The MUCTPI will provide improved overlap handling and full-precision information of the muon candidates to the Topological Processor.

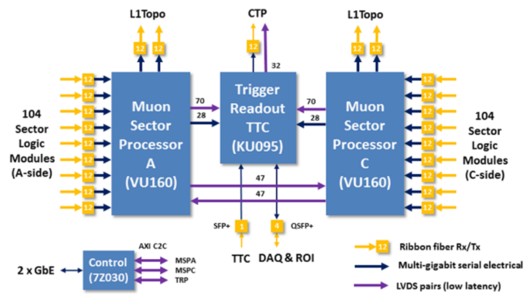
L1Topo
CTP
L1Topo
104 Sector Logic Modules (A-side)
Muon Sector Processor A (VU160)
Trigger Readout TTC (KU095)
Muon Sector Processor C (VU160)
104 Sector Logic Modules (C-side)
70
28
32
47
SFP+
QSFP+
TTC
DAQ & ROI
2 x GbE
Control (7Z030)
AXI C2C
MSPA
MSPC
TRP
Ribbon fiber Rx/Tx
Multi-gigabit serial electrical
LVDS pairs (low latency)

Figure 3: The new MUCTPI of ATLAS.

The new MUCTPI (see Figure 3) will be built as a single ATCA blade, compared to 18 VME modules in the current system. It will receive 208 optical links and will use two state-of-the-art FPGAs (Xilinx Virtex Ultrascale+) for the overlap handling, counting of muon candidates, and sending candidates to the Topological Processor. A third FPGA (Xilinx Kintex Ultrascale) will provide the total count of muon candidates to the CTP and readout data to the data acquisition system. A System-on-Chip (SoC, Xilinx Zynq 7000) [4] will act as a control processor and integrate the MUCTPI into the ATLAS run control (RC) system.

THE RUN CONTROL OF THE MUCTPI

The RC system of the ATLAS experiment provides control of the MUCTPI, e.g. start and stop commands; loading of configuration data, e.g. overlap lookup table (LUT) data and collection of monitoring data, e.g. counter values. Due to the new technology (ATCA), new forms of communication between the MUCTPI and the RC system had to be investigated. A SoC with a processor part and a programmable logic part will be used for the communication with the RC system and the processing FPGAs of the MUCTPI (see Figure 4). The processor system runs embedded Linux and the programmable logic provides the communication with the processing FPGAs using Xilinx chip-2-chip links, the configuration of the FPGAs using the Xilinx slave serial protocol, and the configuration and monitoring of the MUCTPI hardware with its power, clock and optical modules using serial buses like I2C and SPI.

Three models of communication with the RC system have been investigated and will be discussed below. The software for the operating system, a kernel module for accessing memory, including the use of DMA (for Model #2 and #3), the port of the RC software (for Model #3), and all the applications are built and maintained using the framework of the Yocto Project [5] from the Linux Foundation. Recipes for all packages have been developed, in order to fetch, configure, and compile the software, and create all images necessary to boot the processor system.

Model #1: IPBus

IPBus is provided by the CMS experiment [6] and is based on firmware and software. The firmware is implemented in the programmable logic part of the SoC, receives UDP packets, and performs read/write accesses on the processing FPGAs of the MUCTPI (see Figure 5). Run controllers on a PC directly connected to the SoC use the software library μHAL. Note that for this model, the processor system of the SoC is not needed, and that it can actually be implemented using an FPGA.

This model works as expected and we consider it a fallback solution. The use of a protocol which can lose packets, and the lack of flexibility for extending the protocol to more complex operation using serial protocols on the MUCTPI, like I2C and SPI, made us look for alternatives.

Model #2: RemoteBus

RemoteBus [7] was developed by the ATLAS Level-1 Central Trigger team and is based on software using a Remote-Procedure-Call-like [8] approach. Run controllers on a PC directly connected to the SoC implement RemoteBus clients which send requests using TCP/IP to

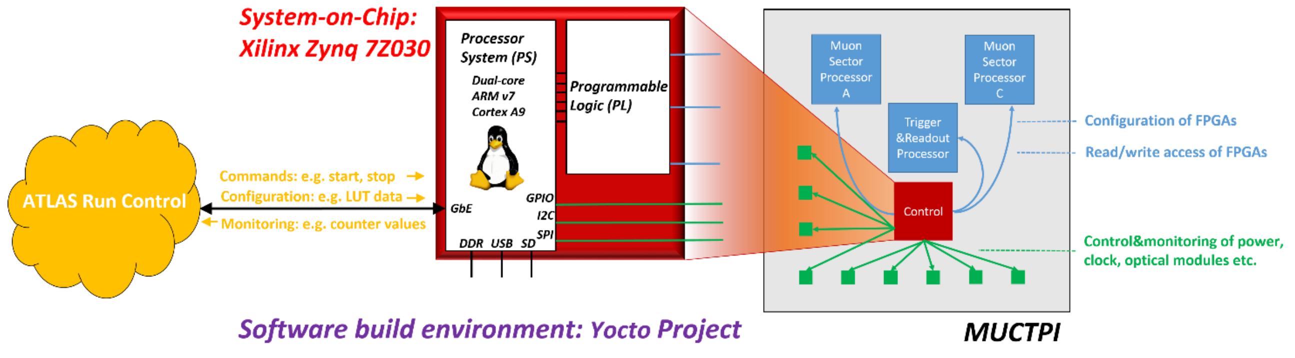
System-on-Chip: Xilinx Zynq 7Z030
Processor System (PS)
Dual-core ARM v7 Cortex A9
Programmable Logic (PL)
ATLAS Run Control
Commands: e.g. start, stop
Configuration: e.g. LUT data
Monitoring: e.g. counter values
GbE
GPIO
I2C
SPI
DDR USB SD
Muon Sector Processor A
Muon Sector Processor C
Trigger &Readout Processor
Control
Configuration of FPGAs
Read/write access of FPGAs
Control&monitoring of power, clock, optical modules etc.
Software build environment: Yocto Project
MUCTPI

Figure 4: Run Control of the MUCTPI using an SoC.

the RemoteBus server on the processor system of the SoC (see Figure 6).

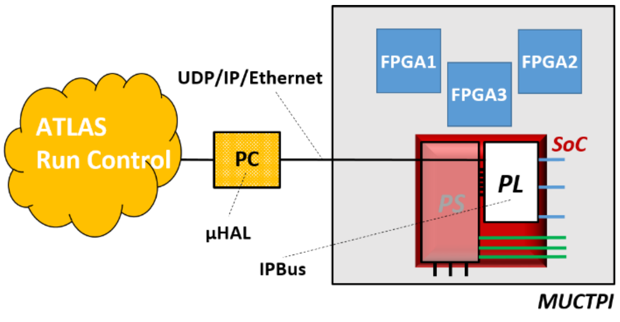

Figure 5: Model #1 - Use of IPBus using the Programmable Logic of the SoC.

A dedicated thread per client executes read/write accesses of the processing FPGAs of the MUCTPI or more complex functions for serial protocols like I2C, SPI, etc. C++ inheritance provides the flexibility to extend the functionality. All parameters are added to the message or retrieved from it in a stack-like way. Queuing of requests allows them to be executed in one go and mitigate latency overhead. A minimal latency of 75 µs for a request-response transaction and sustained data rates in excess of 50 MB/s were measured.

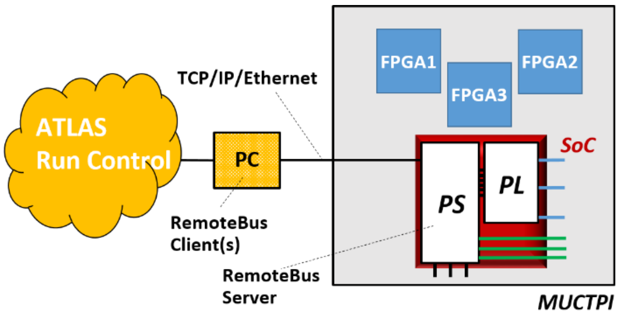

Figure 6: Model #2 - Use of RemoteBus on the SoC.

This model provides a reliable protocol, the flexibility for extending the protocol to more complex operation and performance well in excess of the previous VME based model. It is our baseline for controlling the MUCTPI.

Model #3: Run Controller

The ATLAS software for the RC system [9] was ported to the embedded Linux by the Level-1 Central Trigger team. Recipes in the Yocto framework were used to build a minimal RC application running on the processor system of the SoC and communicating directly with the ATLAS RC system using TCP/IP. This RC application will be instrumented with low-level software which exists in the RemoteBus model (see above) in order to perform read/write accesses of the processing FPGAs of the MUCTPI (see Figure 7).

This model provides reliable communication and the highest degree of flexibility, as the run control application is running directly on the processing system of the SoC. Further development of this model and an investigation of the maintenance effort required are currently being carried out.

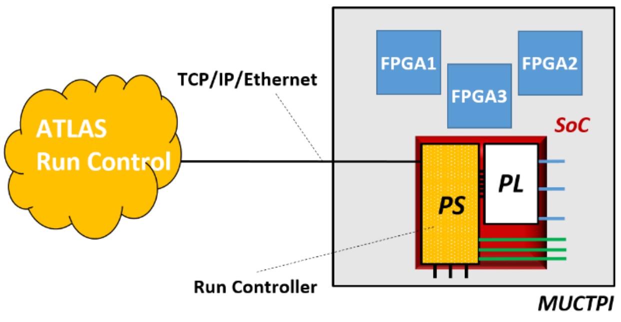

Figure 7: Model #3 - Use of a Run Controller on the SoC.

SUMMARY

A SoC (Xilinx Zynq) has been used successfully to provide communication between the ATLAS RC system and the new MUCTPI. Several models of communication have been compared. An available UDP-based implementation uses firmware for the programmable logic, and although this works as expected it uses an unreliable protocol and does not provide sufficient flexibility to extend its functionality to more complex operations on the MUCTPI. A Remote-Procedure-Call-like software implementation running on the processor system with embedded Linux provides reliable communication and its implementation in C++ provides the possibility to extend its functionality to more complex operations, like serial protocols to be executed on the processor system. Porting of the ATLAS run control software to embedded Linux was achieved and provides the highest flexibility by removing the need for intermediate layers of software. All software developments for the operating system and the application software were successfully maintained using the framework of the Yocto Project. Using a full-blown operating system on the SoC has proven to be very useful in the debugging of the MUCTPI prototype.

REFERENCES

[1] The ATLAS Collaboration: The ATLAS Experiment at the CERN Large Hadron Collider, Journal of Instrumentation, JINST 3 S08003, 2008.

[2] *P. Vichoudis et al., The ATLAS Muon-to-Central-Trigger-Processor Interface, Proc. Topical Workshop on Electronics for Particle Physics, 2017, in preparation.*

[3] *B. Stelzer for the ATLAS Muon Collaboration, The New Small Wheel Upgrade Project of the ATLAS Experiment, Nuclear and Particle Physics Proceedings 273-275 (2016) 1160-1165.*

[4] *Xilinx Inc., Zynq-7000 All Programmable SoC, https://www.xilinx.com/products/silicon-devices/soc/zynq-7000.html.*

[5] *The Yocto Project, https://www.yoctoproject.org.*

[6] *T. Williams et al., IPBus: a flexible Ethernet-based control system for xTCA hardware, JINST 10 (2015) C02019.*

[7] *R. Spiwoks et al., The ATLAS Muon-to-Central-Trigger-Processor Interface (MUCTPI) Upgrade, Proc. International Conference on Technology and Instrumentation in Particle Physics, Beijing, People's Republik of China, 2017, in preparation.*

[8] *J. Bloomer, Power Programming with RPC, O'Reilly & Associates, Inc., 1992*

[9] *The ATLAS Collaboration, ATLAS High-Level Trigger, Data Acquisition and Controls Technical Design Report, CERN/LHCC/2003-22, 2003, see https://cdsweb.cern.ch/record/616089.*