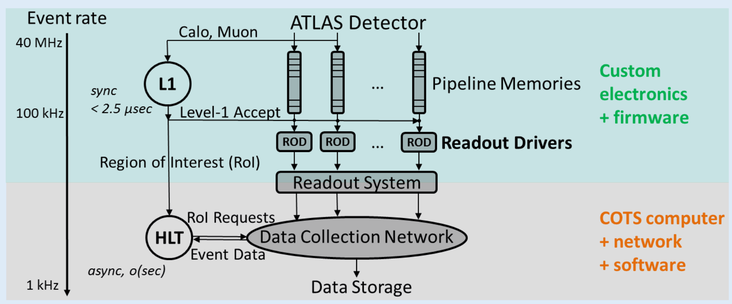
Run Control Communication for the Upgrade of the ATLAS Muon-to-Central-Trigger-Processor Interface (MUCTPI)

A. Armbruster[a], G. Carrillo-Montoya[a], M. Chelstowska[a], P. Czodrowski[a], P.-O. Deviveiros[a], T. Eifert[a], N. Ellis[a], P. Farthouat[a], G. Galster[a,b], S. Haas[a], L. Helary[a], O. Lagkas Nikolos[a], A. Marzin[a], T. Pauly[a], V. Ryjov[a], K. Schmieden[a], M. Silva Oliveira[a], R. Spiwoks[a], J. Stelzer[a], P. Vichoudis[a], T. Wengler[a]
a) CERN, Switzerland, b) NBI Copenhagen, Denmark

ATLAS Experiment @ LHC

The ATLAS experiment is a general-purpose experiment at the Large Hadron Collider (LHC) at CERN. It observes proton-proton collisions at an energy of 13 TeV. With about 25 interactions in every bunch crossing (BC) every 25 ns, there are 10^9 interactions per second. 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 consists of a Level-1 trigger based on custom electronics which reduces the event rate to a maximum of 100 kHz, and a high-level trigger system based on commercial computers which reduces the event rate to around 1 kHz.

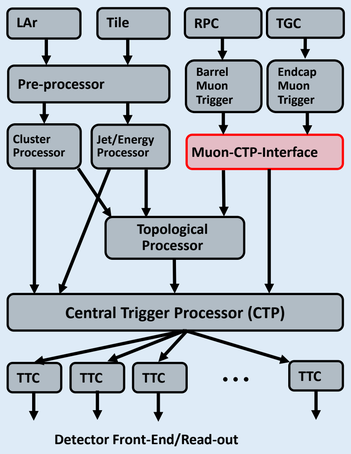
The Trigger and Data Acquisition System



Level-1 Trigger System

The first-level trigger 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) 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 the trigger information from the calorimeter trigger, the MUCTPI, and the Topological Processor to form the final Level-1 decision.

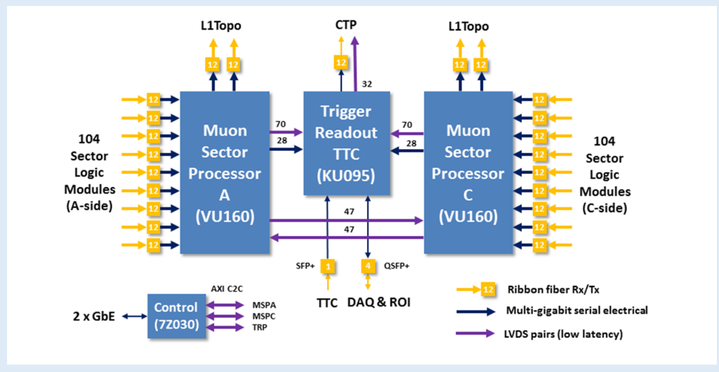
The Level-1 (L1) Trigger System



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 of the muon trigger system, to be installed during the shutdown of 2019 and 2020. The new MUCTPI will use optical links to replace bulky electrical cables. Those links will allow the muon trigger detectors to send more muon candidates with more precise information. The new MUCTPI will provide improved overlap handling and full-precision information to the Topological Processor. The new MUCTPI will be built as a single ATCA blade. It will receive 208 optical links and will use two FPGAs for the overlap handling, counting of muon candidates, and sending candidates to the Topological Processor. A third FPGA will provide the total count of muon candidates to the CTP and read out data to the data acquisition system. A System-on-Chip will integrate the new MUCTPI into the ATLAS run control system.

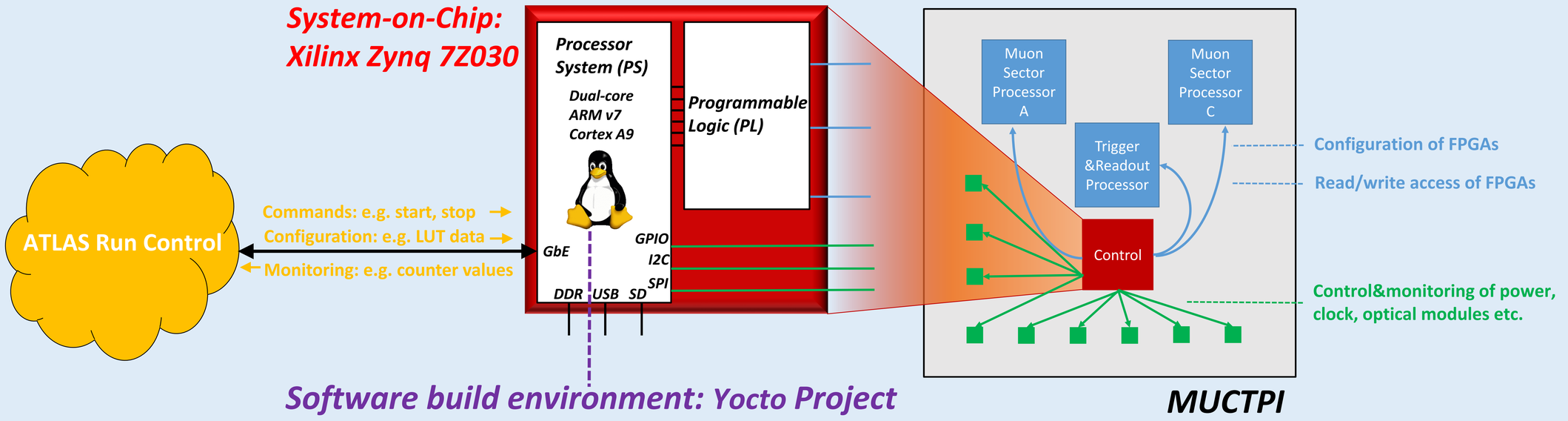
The new MUCTPI



Run Control of the MUCTPI using a System-on-Chip and Embedded Linux

The run control 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 ways of communication between the MUCTPI and the run control system had to be investigated. A System-on-Chip (SoC) with a programmable logic part and a processor part will be used for the communication with the run control system and the processing FPGAs of the MUCTPI. The processor system runs embedded Linux and the programmable logic provides the communication with the processing FPGAs using Chip-2-Chip links, the configuration of the FPGAs using the 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.

Run Control of the MUCTPI using a SoC

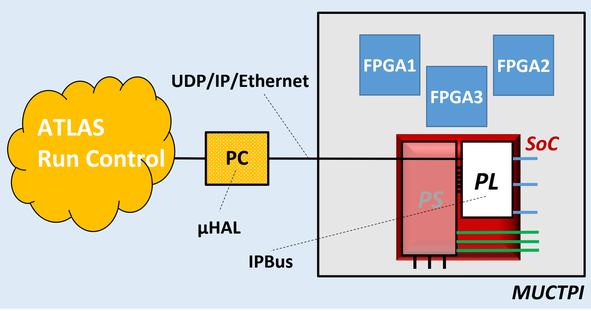


The software for the operating system, a kernel module for accessing memory, including use of DMA (Model #2, and #3), the port of the run control software (Model #3), and all the applications are built and maintained using the framework of the Yocto Project from the Linux Foundation. Recipes for all 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 and is based on firmware and software. The firmware is implemented in the programmable part of the SoC, receives UDP packets, and performs read/write accesses on the processing FPGAs of the MUCTPI. Run controllers, i.e. run control processes, 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 could therefore be implemented in an FPGA.

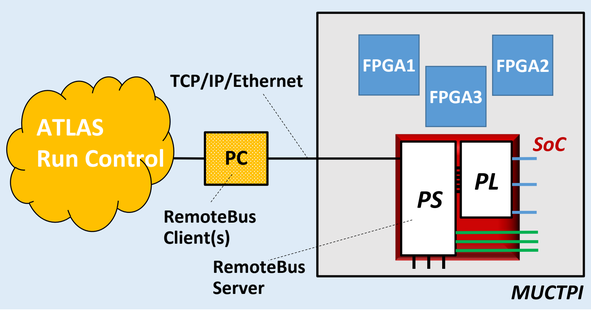
Use of IPBus on the Programmable Logic of the SoC



Model #2: RemoteBus

RemoteBus was developed by the ATLAS L1CT team and is based on software using an approach like the 'Remote Procedure Call'. Run controllers on a PC directly connected to the SoC implement RemoteBus clients which send requests using TCP/IP to the RemoteBus server on the processor system 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. Queuing of requests allows them to be executed in one go and mitigate latency overhead. C++ provides flexibility to extend the functionality. A minimal latency of 75 µs for a request-response transaction and sustained data rates of 50 MB/s have been measured.

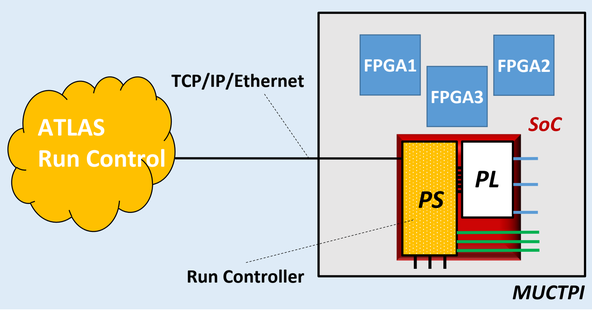
Use of RemoteBus on the SoC



Model #3: Run Controller

The ATLAS software for a run controller was ported to the embedded Linux by the L1CT team. The run controller runs on the processor system of the SoC and communicates directly with the ATLAS run control system using TCP/IP. Low-level software in the run controller performs read/write accesses of the processing FPGAs of the MUCTPI.

Use of Run Controller on the SoC



Summary

A System-on-Chip (Xilinx Zynq) has been used successfully to provide communication between the ATLAS run control 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 software implementation using an approach like the 'Remote Procedure Call' and 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 to embedded Linux was achieved and provides the highest flexibility, 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.