Software environment for controlling and re-configuration of Xilinx Virtex FPGAs – TWEPP-07

D. Fehlker[ab], J. Alme[a], T. Alt[c], S. Bablok[a], Th. Beierlein[b], R. Campagnolo[d], C. González Gutiérrez[d], H. Helstrup[e], R. Keidel[f], T. Krawutschke[g], D. T. Larsen[a], V. Lindenstruth[c], C. Lippmann[d], L. Musa[d], M. Richter[a], D. Röhrich[a], K. Røed[e], B. Schockert[f], K. Ullaland[a]

[a] Department of Physics and Technology, University of Bergen, Norway
[b] University of Applied Sciences Mittweida, Germany
[c] Kirchhoff Institute of Physics, University of Heidelberg, Germany
[d] CERN, European Organization for Nuclear Research, Geneva, Switzerland
[e] Faculty of Engineering, Bergen University College, Norway
[f] Center for Technology Transfer and Telecommunications, University of Applied Sciences Worms, Germany
[g] Institute of Communication Engineering, University of Applied Sciences Cologne, Germany
Dominik.Fehlker@ift.uib.no

Abstract

The Time Projection Chamber is one of the detectors of the ALICE experiment, that is currently being commissioned at the Large Hadron Collider at CERN. The Detector Control System is used for control and monitoring of the system. For the TPC Front-End Electronics (FEE) the control node is a Readout Control Unit that communicates to higher layers via Ethernet, using the standard framework DIM. The Readout Control Unit is equipped with commercial SRAM based FPGAs that will experience errors due to the radiation environment they are operating in. This article will present the implemented hardware solution for error correction and will focus on the software environment for configuration and controlling of the system – TWEPP-07.

I. INTRODUCTION

One of the tasks of the ALICE Detector Control System (DCS) is the controlling, configuration and monitoring of the front-end electronics. Detailed information about ALICE, the DCS components and its architecture can be found in [1, 2].

The control system is detached from the data-flow. From the Front-End Electronics (FEE) the data is fed into an optical link and transported to the Data Acquisition system (DAQ). Throughout the FEE, FPGAs and other programmable logic devices are used, which require configuration and monitoring. FPGAs may experience errors due to radiation during the experiment which could lead to malfunctioning of the system or corrupted data. Hence it is of high importance to detect and correct these errors as soon as possible.

Although this article will describe the hardware of the TPC DCS for the Front-end electronics and focus on the implemented software environment, other detectors e.g. the electromagnetic calorimeter - PHOS, share similar hardware and use the same software environment as presented here. Furthermore parts of this software environment are kept generic in order to support other parts of the detector, e.g. the PHOS trigger system.

II. SYSTEM OVERVIEW

The Time Projection Chamber (TPC) is the main tracking detector (Fig. 1). In the axial center, a high voltage electrode provides an electric field. The barrel is filled with a mixture from Ne, CO_2 and N_2. Charged particles from the collision ionize the gas volume on their way through the detector, and the electrons drift through the electric field towards the end caps where the charge is amplified and collected.
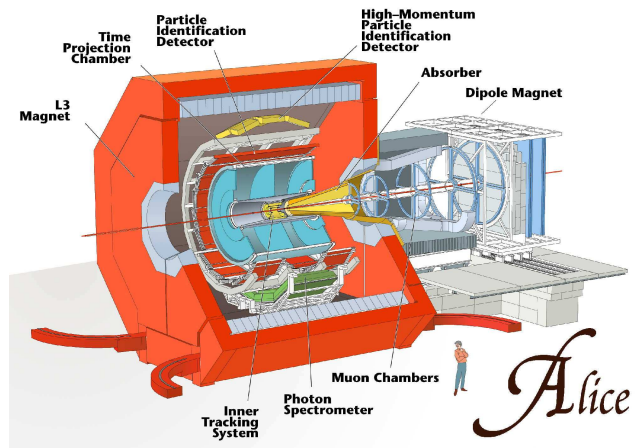

Figure 1: The ALICE detector

The end caps of the TPC barrel are divided into 18 sectors on each side, and the readout chambers are located here. Each sector has one inner and one outer readout chamber where the charge is amplified and ascertained by a 2-dimensional readout system. Together with the drift time this provides 3-dimensional spatial information. Sensor pads in this 2-dimensional readout system connect to the FEE which is located behind the readout chambers at the end caps. From there the data is passed on to the DAQ.

The FEE in the TPC consists of Front-End Cards (FECs) , Readout Control Units (RCUs) and DCS boards. The FECs are

mounted to the readout chambers and connect to the sensor pads of the readout system via short kapton cables. The inner readout chamber has 43 FECs, the outer chamber 78 and up to 25 FECs are connected to one RCU. [3]
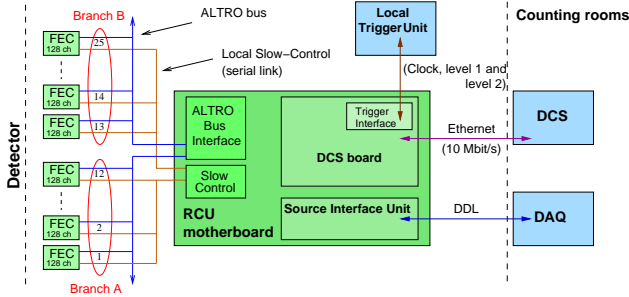

Figure 2: Components of the TPC FEE and dataflow

For each RCU the backplane bus is divided into two branches. The RCU motherboard holds a DCS board mezzanine card which is used to link the FEE into the DCS. In overall there are 216 RCUs with attached DCS boards controlling 4356 FECs serving roughly 560000 data channels. (Fig. 2).

The main FPGA on the RCU is a Xilinx Virtex-II Pro SRAM based FPGA which performs the task of controlling, monitoring, readout and configuration of the FECs as well as providing them with trigger information. A Source Interface Unit (SIU) is plugged onto the RCU and enables the RCU to send the data to the DAQ Readout and Receiver Card of the DAQ system. The SIU is the standard interface to the DAQ in ALICE and provides a 2 GBit/s optical link. The other mezzanine card plugged to the RCU is the DCS board which provides the interface to the DCS (Fig. **??**). [4]

The DCS board [5] is used in several detectors of ALICE. It hosts an ALTERA EPXA1 FPGA with an integrated ARM CPU that runs an embedded Linux. In addition the DCS board features 32 MB RAM, 8 MB Flash ROM, a 10 Mbit/s Ethernet connection and a dedicated interface to the RCU. The Flash ROM is used to store the firmware of the board and the Linux embedded operating system. From the DCS board Linux registers and memory of the RCU can be accessed via dedicated Linux device drivers. Among others this is the MessageBufferInterface driver which provides access to the RCU and the Xilinx Virtex device driver which is described in more detail in Section V..

III. RADIATION CONCERNS

Since the FEE is in the radiation area, Single Event Upsets (SEUs) can occur in the SRAM based devices. SEUs can lead to functional interrupts on the firmware design level. SEUs are not permanent, they can be corrected by reloading the firmware of the device from the Flash memory. Especially it has to be avoided that several SEUs combined do permanent damage to the device, i.e. by changing the I/O directions. In the case of the DCS board, SEUs can be corrected by rebooting the device with the data from the Flash memory. This leads to occasional short downtime while rebooting, but since the DCS board is not part of the data path, this can be tolerated.

In case of the RCU downtime is not acceptable because it is part of the data path and vital information would be lost. For this reason the RCU has the possibility to reconfigure the Xilinx Virtex-II Pro FPGA while it is under full operation, which is called Active Partial Reconfiguration (APR). For the APR feature the RCU is equipped with an Actel ProASICplus APA075 Flash based FPGA and Macronix Flash memory which are both radiation tolerant for the dose and flux that are estimated for ALICE [6]. The Actel communicates with the Flash memory device and the SelectMAP interface of the Xilinx Virtex-II Pro FPGA. The SelectMAP interface is the fastest available connection to the configuration memory of the Xilinx Virtex-II Pro.
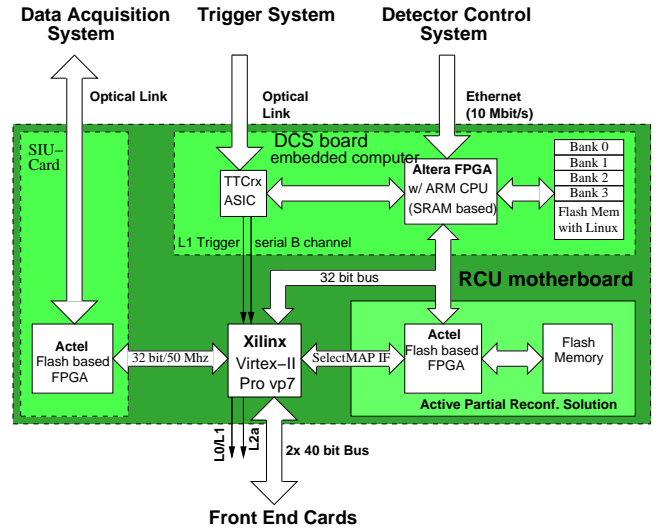

Figure 3: Conceptual sketch of the RCU, the devices used for APR are framed bottom right

The firmware of the Actel FPGA has the task to control and monitor the operation of the Xilinx Virtex-II Pro. It provides three different modes of operation: Initial configuration, scrubbing - which is overwriting the configuration memory of the Xilinx Virtex-II Pro whether there was an error or not - and frame by frame read-back and verification. The latter has the ability to read back the frames one by one, compare them with the ones stored in the Flash memory device and overwriting the ones where an error has been found. During scrubbing and readback mode counters keep track of how many times APR was performed for the whole memory. During read-back and verification the number of occurred errors can be determined.

Tests have shown that with the flux and dose estimated in ALICE there are about 3-4 functional errors for all 216 RCUs during 4 hours of beamtime to be expected. For the Xilinx Virtex-II Pro vp7 the rate of SEUs was measured to be in the order of 10-20 times higher then the SEFIs ([6]).

IV. Software Architecture and Components

A. Functional Layers

A three layered structure in hierarchical architecture is common in other experiments and also used in the ALICE DCS. It consists of Field layer, Control layer and Supervisory layer.
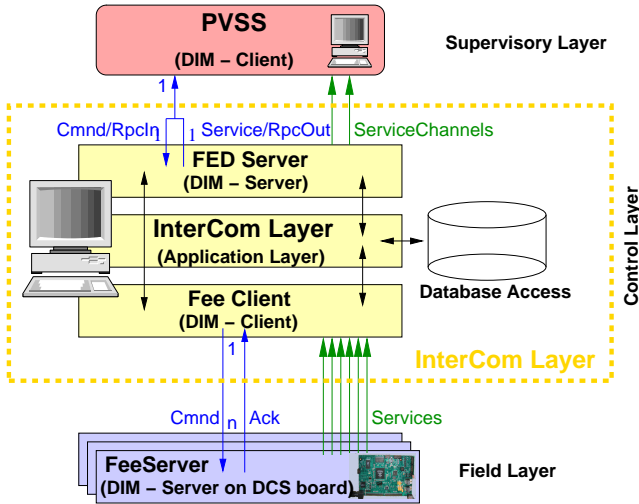

Figure 4: Software components and data flow

In the Field layer are mostly sensors, actuators, power supplies as well as the RCU with attached DCS board and the readout electronics.

The Control layer runs on several standard PCs, PLCs and PLC like devices. It connects to the Supervisory layer via ethernet. The Control layer sends commands and configuration data from the Supervisory layer to the Field layer. Vice versa it collects information from the Field layer and sends it to the Supervisory layer. A configuration database supports the tasks of the Control layer.

In the Supervisory layer are mostly servers and operator nodes. It provides the user interfaces to the operator. The Supervisory layer interfaces external systems and services like the LHC.

Dedicated applications carry out the tasks of the three layers. They work in parallel and feed the operator with useful information about the status of the system and respond to commands given by the operator. The three layered structure is pictured in Fig. 4.

The Distributed Information Management (DIM) protocol is used as a basis for communication between all layers. DIM is an open source development of CERN. Like most communication systems it is based on the client/server paradigm. It is specially suited for network-transparent inter-process communication in distributed and heterogeneous environments. [7] provides more thorough information about the DIM framework.

A specialized application, the Intercom Layer, carries out the task in the Control layer. It runs on dedicated machines outside

of the radiation area independently from other systems and acts as an abstraction layer. It separates Supervisory and Field layer. Details about the Intercom layer can be found in [8].

B. The FeeServer

The Front-end electronics Server (FeeServer) is the software component to receive commands and monitor data points specific to the underlying hardware in the ALICE DCS. The FeeServer runs on the DCS board and its purpose is to abstract from the underlying Front End Electronics and also to cover the tasks of interfacing the data sources in the hardware and to publish the data. Further tasks are receiving commands and configuration data for controlling the Front-end electronics as well as providing self tests and watch dogs.

The common functionality, mainly concerning the communication to the DIM framework, is built into its device independent core. Because the FeeServer is used in the DCS of different detectors the device/detector specific functions are encapsulated in the so called ControlEngine (CE). Detailed information about the FeeServer can be found in [9].

C. The FeeServer Control Engine

The main purpose of the CE is to contact the field devices. Depending on the detector the source code for the CE will be compiled together with the FeeServer. This solution was chosen to avoid slow and fault prone inter-process communication. Communication to the underlying hardware devices is established via device drivers. The initialization of these drivers has to be done during startup of the CE. Mainly the communication is writing instruction codes to a certain address and reading the result(s) from other addresses. The values of the monitored items like temperatures, voltages and currents are retrieved in a similar procedure.
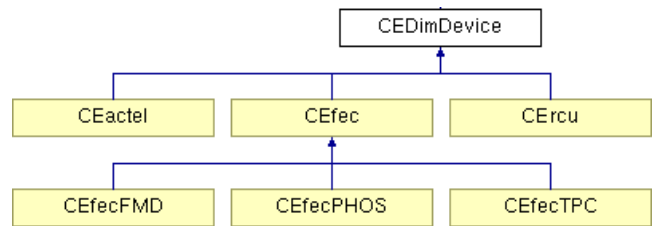

Figure 5: Software devices in the FeeServer Control Engine

Different software devices which describe the actual hardware are introduced in the FeeServer CE. The main device is the RCU. To fit the different FECs in the different detectors that the FeeServer is going to be used in, three different types of software devices for the FECs are introduced here: for the FMD, TPC and PHOS detector. Fig. 5 shows the existing software device representations which all inherit parts of their functionality from the CEDimDevice. In this way they are equipped with a basic state machine and command handling. In addition the DIM state channel is automatically published.

V. XILINX VIRTEX DEVICE DRIVER

A Linux device driver was developed to access the SelectMAP interface of the Xilinx Virtex-II Pro FPGA directly from the DCS board embedded Linux. It handles the control lines of the SelectMAP interface and provides read and write functionality to the registers and memory of the FPGA [10]. For the Xilinx Virtex-II Pro the device driver is also able to read back frames of the configuration memory.

The Xilinx Virtex device driver is not only able to provide access to the memory but also to issue an abort command to the Xilinx Virtex-II Pro FPGA. In case it gets stuck in its operation the abort command can bring the Xilinx Virtex-II Pro back to a defined state where it is able to receive commands.

The device driver also creates an entry in the Linux proc file system. There it provides the up to date values of the 15 control registers of the SelectMAP interface. This allows for easy debugging.

A. Support for Xilinx Virtex-4 and Virtex-5

In other parts of the ALICE experiment (e.g. the PHOS trigger system) the Xilinx Virtex-4 is used. These devices also feature the SelectMAP interface to access the configuration memory. The SelectMAP interface is the fastest available interface to configuration memory of the FPGA and common among FP-GAs from the Xilinx Virtex family. Several parts of the PHOS trigger system that use Xilinx Virtex-4 FPGAs are described in [11]. For future developments it is probable that the Xilinx Virtex-5 would be used. The design and implementation of the Xilinx Virtex device driver fits these devices as well.

B. Performance

Tests have shown that reading back the frames needed in the Flash memory from the configuration memory of the Xilinx Virtex-II Pro with the help of the Xilinx Virtex device driver take about 30 sec. Performing the same task via the Message-BufferInterface driver, which was the only possible way before the development of the the Xilinx Virtex device driver, needs about 2 minutes 30 seconds. This improvement is due to using the SelectMAP interface of the FPGA. In this operation the amount of data handled is about 1.3 MBytes.

VI. ACTEL DEVICE IN THE FEESERVER CE

To control and monitor the functionality of the Xilinx Virtex-II Pro FPGAs on the RCUs, the FeeServer CE was equipped with an Actel software device which represents the state of the hardware solution chosen for APR. A finite state machine (FSM) which controls the mode of operation of the APR is used. Services are provided to monitor the status of the hardware, e.g. the number of occurred errors. A command channel is created which enables sending of commands and binary command blocks to the Actel device. The Actel device then processes the binary commands accordingly.

This software part can also be used in the PHOS detector since it is equipped with the same type of RCUs.

A. Services

Values reflecting the status of the hardware are published to the higher layers via services. A client subscribing to the FeeServer can monitor these values. Among others these services publish the number of entire scrubbing or read-back verification cycles over all frames and the number of occurred errors found in the Xilinx configuration memory during read-back and verification.

Especially the number of occurred SEUs in the Xilinx configuration memory is important since it is planned to use this information to observe the luminosity of the beam. Using the number of occurred errors from all RCUs the overall beam luminosity can be monitored. To illustrate the distribution over the end caps, all SEUs in RCUs from all sectors with the same distance to the interaction point (e.g. TPC-FEE_0_x_0, $0 \leq x \leq 17$, which would match the A side, sectors 0 to 17, all RCUs with number 0) can be added up.

B. State machine

The state machine provides states for: *OFF*, *ON*, *scrubbing*, *read-back and verification*, *failure* and *error*. In the inital state - *OFF* - no checks have been performed on the hardware to ensure a proper operation of the devices. During the transition to the state *ON* a series of internal checks (concerning pointers in the Flash memory, status and error registers, . . .) are performed to ensure the proper function of the device. From the *ON* state the *scrubbing* and *read-back verification* states can be reached during which the hardware performs the according tasks.
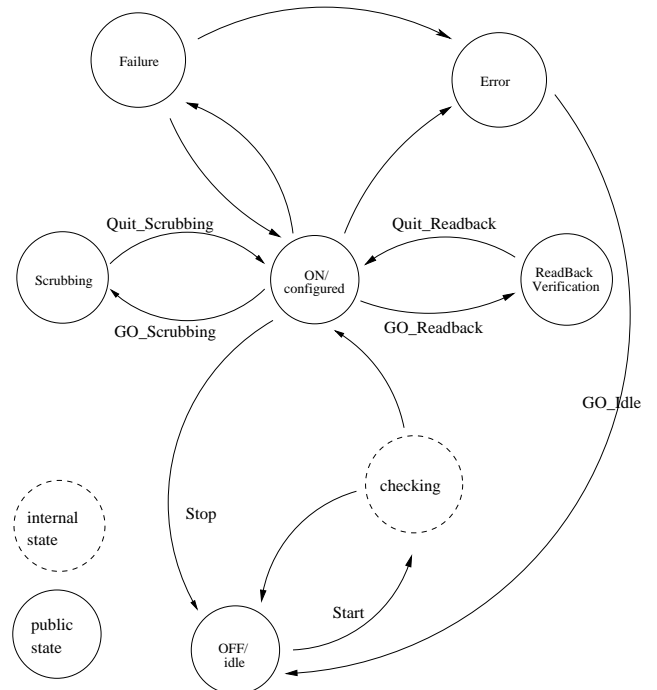

Figure 6: The state machine of the Actel device

Failure and error states are provided by the FSM and can be

reached from any state. In case of minor errors the Actel device enters the failure state from where it tries to recover by itself. Only in case of a severe error (e.g. wrong pointers in the Flash memory device) where intervention of an operator is needed the error state is entered. Fig. 6 shows a sketch of the Actel state machine.

C. Binary commands

The Actel software device is able to receive commands and command blocks from the upper layers through the established communication channels. These command blocks are in binary format and can contain data needed for configuring the APR solution. At present, 4 binary command blocks have been designed: erase Flash memory, write initial configuration data, write scrubbing data and write the data for read-back verification.

These commands allow to set up the Active Partial Reconfiguration solution for the three available modes in a convenient way from the upper layers. This is of special importance during commissioning of the TPC since configuring all 216 RCUs would be a time-consuming process if done manually. The required data for the binary command blocks and the blocks itself can be generated in controlled lab conditions using low level tools.

The binary command block for the read-back verification is the largest since it has to hold the data for all the frames of the Xilinx Virtex-II Pro that can be verified. The size of this command block is about 1.3 MBytes, whereas the scrubbing and initial configuration command block are of not more then 600 KBytes in size.

The binary command blocks can be stored in the configuration database. Control panels are under development that will allow to send configuration data to all present RCUs in the TPC and PHOS from the Supervisory layer.

VII. Summary

In this article the implementation of the DCS for FPGA configuration and error correction in the TPC FEE has been introduced. In the field layer the DCS consists of an RCU motherboard with DCS board and several FECs attached. On the RCU an SRAM based Xilinx Virtex-II Pro FPGA is essential for the read-out chain. Since the Front-End-Electronics operates in a radiation environment Single Event Upsets can occur.

The presented Xilinx Virtex device driver is already in use to configure the RCUs in the TPC, PHOS and parts of the PHOS trigger system and has proven its effectiveness. As a future prospect it is planned to enable reading back frames via the device driver for Xilinx Virtex-4 and Xilinx-Virtex-5 FPGAs.

The Actel device in the CE of the FeeServer will be used for the TPC and PHOS. It has been successfully tested on several test setups at the University of Bergen, Norway. The updated version of the FeeServer has been introduced in a lab at CERN and is about to be tested there at a larger scale. With the information provided by the Actel device it is possible to monitor the status of the hardware and the radiation environment.

References

[1] ALICE Collaboration, "ALICE Technical Proposal for A Large Ion Collider Experiment at the CERN LHC", CERN/LHCC 1995-71 (1995)

[2] ALICE Collaboration, "Technical Design Report: Trigger, DAQ, HLT, DCS", CERN/LHCC/2003-062 (2004)

[3] L. Musa *et al.*, "The ALICE TPC Front End Electronics", in Proc. IEEE Nuclear Science Symposium, 2003

[4] C. González Gutiérrez *et al.*, "The ALICE TPC Readout Control Unit", in Proc. 10th Workshop on Electronics for LHC Experiments, 2004

[5] H. Tilsner *et al.*, "Hardware for the Detector Control System of the ALICE TRD", in Proc. 9th Workshop on Electronics for LHC Experiments, 2003

[6] K. Røed *et al.*, "Irradiation tests of the complete ALICE TPC Front-End Electronics chain", in Proc. of the 11th Workshop on Electronics for LHC and future Experiments, Heidelberg, Germany, 2005

[7] C. Gaspar *et al.*, "DIM, a Portable, Light Weight Package for Information Publishing, Data Transfer and Interprocess Communication", Presented at the Int. Conference on Computing in High Energy and Nuclear Physics, Padova, Italy, 2000

[8] M. Richter *et al.*, "The control system for the front-end electronics of the ALICE time projection chamber", in Trans. Nuclear Science 53 (2006) 980.

[9] S. Bablok *et al.*, "Front-End-Electronics Communication software for multiple detectors in the ALICE experiment", In Proc. Nuclear Instruments and Methods in Physics Research, 2006

[10] Xilinx Corp., "Virtex-II Pro and Virtex-II Pro X FPGA User Guide", (ug012, V4.1), 2007

[11] J. Alme *et al.*, "Radiation-tolerant, SRAM-FPGA Based Trigger and Readout Electronics for the ALICE Experiment", In Proc. 15th Real Time Conference (IEEE-NPSS Technical Committee on Computer Applications in Nuclear and Plasma Sciences), 2007