# The Readout Controller Unit

The TPC Readout Controller Unit (RCU) (see figure 1) [1] is responsible for controlling the readout of the TPC, and initialising and monitoring the Front-End Cards (FECs). In total 216 RCUs will connect 4500 FECs, with a maximum of 25 cards connected to each RCU. The amplifying, shaping, digitizing, processing and buffering of the TPC signals is done on the FECs. A custom integrated circuit, the ALTRO (ALICE TPC Read Out), is dedicated to the digitalisation and the further processing of the digitized data. This chip is initialised and controlled directly by the RCU. The communication between the RCU and the ALTROs on the Front-End Cards (FECs) is implemented via a custom bus based on a shielded ribbon cable and a custom protocol.

The RCU collects the data from the FECs, assembles a subevent, compresses the data if necessary and sends the compressed, packed subevent to the PCI Readout Receiver Card (RORC). In addition, the RCU monitors and initialises the FECs. The supervision includes readout of events for monitoring purposes, statistics (readout of number of datastrobes and number of triggers received), temperature variation monitoring, current measurement and power consumption monitoring for hardware fault detection which will be achieved via a separate slow-control bus. The initialisation of the Altros (including pedestal values for the digital filter responsible for tail-cancellation) is done via the main front-end bus. Calculation of all parameters for the ALTRO can be performed on the RCU.

The RCU design involves schematics design, PCB design, model design and firmware design. Several test benches have to be designed in order to test the controller unit. There are a number of design tasks: A simulation model of the complete readout chain starting from simulated signals from the TPC, via FEC, through RCU and RORC. On the RCU there will be a memory bank able to store the ALRTRO data, the size of the local memory is currently under evaluation. The shipping of data to the RORC (through optical fibre) is accomplished via a custom interface, the Detector Digital Link (DDL), which is available both as a simulation model and as a hardware interface prototype. A readout sequence of the TPC is initiated by a common ALICE trigger-signal. The trigger is distributed to the RCU by the central trigger processor. The Slow Control bus from the RCU will be connected to the central Detector Control System (DCS) of the ALICE experiment. The data received from the FECs could (optionally) be compressed before being placed on the optical fibre (DDL). For this purpose, Huffman compression algorithms have been developed [2].

The use of SRAM-based FPGA necessitates special attention to single event upset. To monitor the functionality of the chip, checksums are being calculated and compared to checksums stored both on board and externally (via slow control). Should an error occur the FPGA can be reprogrammed from on board EPROMs. An option to reprogram from an external source will also be included.

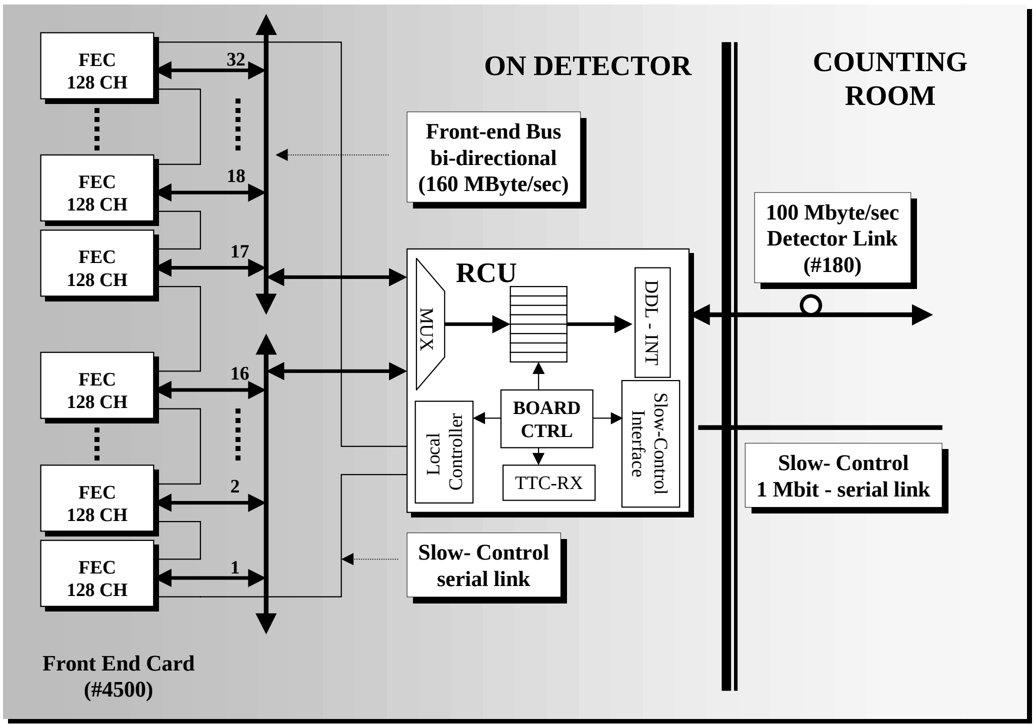

Figure 1 Front-end electronics system architecture and readout controller unit.

# Firmware design

The Resource and Priority Manager (RPM) is the master state machine, controlling all subsystems (see figure 2):

- FEC Bus Control. Two bus branches are serviced.
- FEC Slow Control. Interface between the FEC slow control system and DCS.
- SIU Control. Interface to the DDL.
- DCS Interface. Low level for monitoring heart beat and critical temperature/voltages; high level for access of RCU/ALTRO registers.
- Health Agent. External circuit with watchdog functionality for monitoring single event upsets.
- Trigger interface. Receiving trigger information via TTCrx.
- Debugging interface. PCI core for development, optional in the final version.
- Huffman decoder for data compression – optional.

The firmware design is based on a finite state machine design implemented in VHDL.

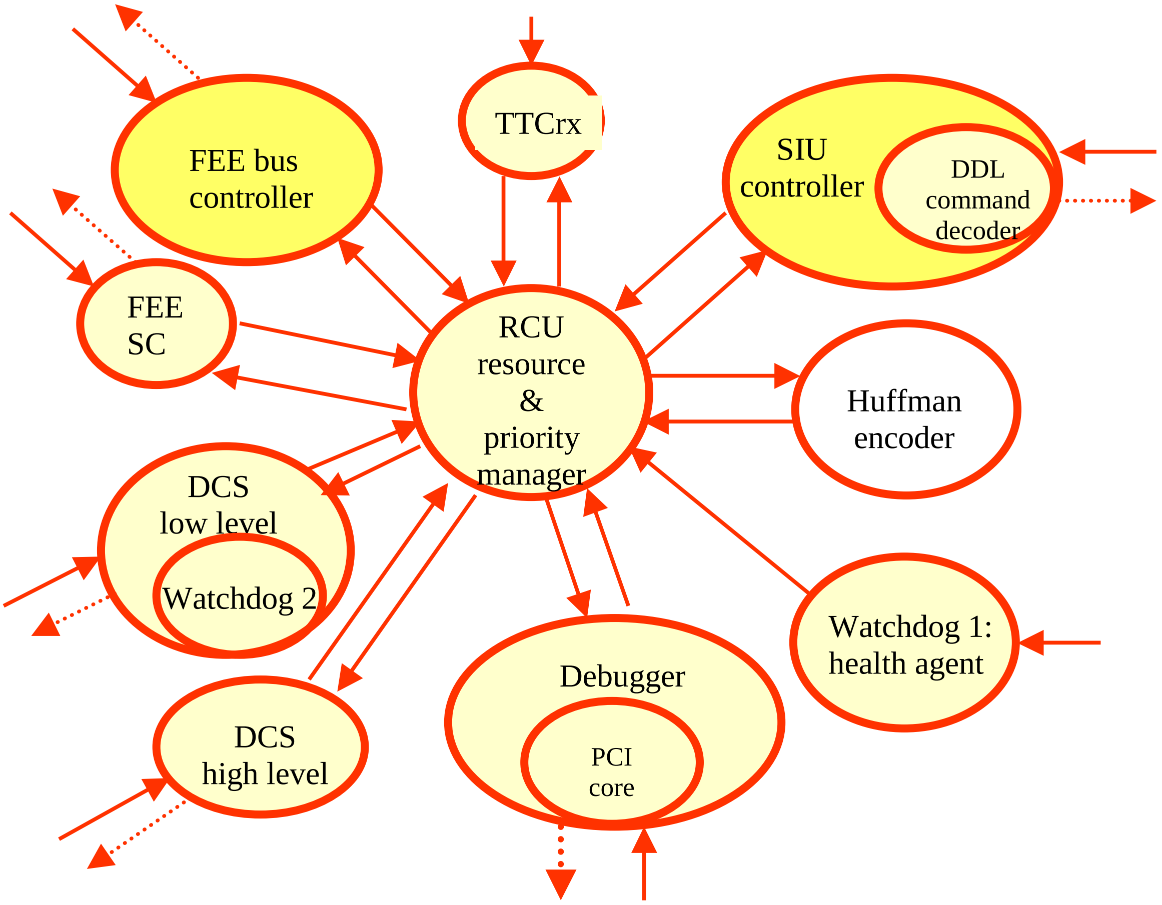

Figure 2 Schematic firmware design.

## Implementation

The RCU board design is based on a single FPGA, containing all logic and a fraction of (or all) memory (see figure 3). The FPGA is interfaced to all subsystems either via commercial chips (DCS: Profibus slave ASIC, Ethernet chip; FEC: GTL bus drivers) or custom chips (Trigger: TTCrx). The SIU is interfaced via a CMC mezzanine card. For debugging purposes the board has an additional PCI interface. A memory controller (FIFO structure) is implemented in the FPGA in order to access external SRAM if necessary.

## Status

A first prototype of the RCU has been developed using the Altera EP20K400 FPGA on a commercial PCI-board (PLDA) [3]. Only the custom front-end bus protocol is implemented in this FPGA. A memory controller (FIFO structure) for accessing internal and external banks is also implemented in the FPGA. The SIU interface has been developed and successfully tested.

A second prototype of the RCU has been developed and is being tested. The prototype is based on a PCI mother board carrying two mezzanine cards, an SIU board and a board interfacing the two FEC-busses, the TTCrx and the DCS system. The aim of the board is to develop the trigger and DCS interfaces, to evaluate the performance of the interface to the FECs and the SIU and to optimise memory usage, i.e. to minimise external SRAM. Various scenarios to cope with single event upsets in the SRAM-based FPGA will also be studied .

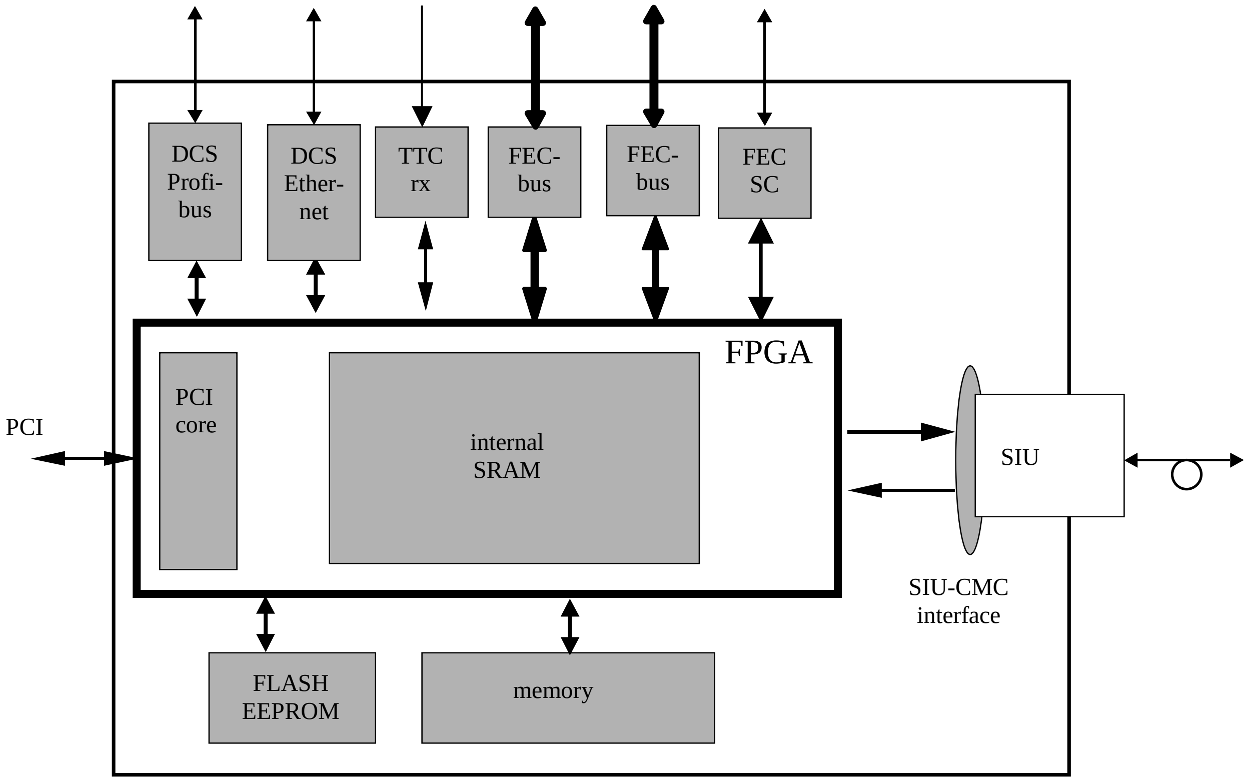

Figure 3 Schematic RCU board design.

# References

[1] ALICE collaboration, Technical Design Report of the Time Projection Chamber, CERN/LHCC 200-001 (2000).

[2] T. Jahnke, S. Schoessel and K. Sulimma, EDA group, Department of Computer Science, University of Frankfurt (2000).

[3] http://www.fi.uib.no/~dieter/ALICE/alice_fee_cern_02_final.ppt