

External scrubber implementation for the ALICE ITS Readout Unit

Magnus Rentsch Ersdal^{*a}, Johan Alme^a, Matthias Bonora^c, Piero Giubilato^d, Matteo Lupi^c, Simon Voigt Nesbø^{a,b}, Attiq Ur Rehman^a, Dieter Röhrich^a, Gianluca Aglieri Rinella^c, Joachim Schambach^e, Arild Velure^c, Shiming Yuan^a

^aUniversity of Bergen (UiB)

^bWestern Norway University of Applied Sciences (HVL)

^cEuropean Organization for Nuclear Research (CERN), Geneva, Switzerland

^dUniversita e INFN, Padova, Italy

^eThe University of Texas at Austin, Physics Department, Austin, Texas, United States

E-mail: magnus.ersdal@uib.no

The ALICE Inner Tracking System is currently being upgraded for the LHC run 3. The whole detector and its associated systems, including the readout electronics are being designed and built. The Readout Unit is responsible for the first level of data aggregation and detector control in the ITS detector. 192 of the Readout Units will be operational in the ALICE cavern, where their proximity to the beam makes radiation tolerant design for all components mandatory. The Readout Unit is designed with a Xilinx Kintex Ultrascale at its core, providing high-density reconfigurable logic and high-speed IOs to handle the data-load from the sensors. Inclusion of a Microsemi flash-based auxiliary FPGA on the Readout Unit enables fault-tolerant operation, by implementing periodic blind scrubbing to correct single event upsets in the configuration memory of the SRAM-based Xilinx FPGA. This contribution discusses the external scrubber implementation for the Readout Unit for the ALICE ITS upgrade.

Topical Workshop on Electronics for Particle Physics TWEPP2019
2-6 September 2019
Santiago de Compostela - Spain

*Speaker.

1. Introduction

One of the major upgrade-projects for the ALICE detector in Long Shutdown 2 is the replacement of the inner tracking detector. The upgraded Inner Tracking System (ITS) is constructed of seven cylindrical barrels with all together 24 120 pixel sensor chips arranged around the beam pipe in seven cylindrical layers. The Readout Unit (RU) is designed to control and read data from the detector. The 192 Readout Units are located in the radiation area, ~4 meters from the interaction point. The RUs send the sensor data on optical GigaBit Transceiver (GBT) links to the counting room, where 24 Common Readout Units (CRUs) aggregate the data from 8 RUs each. The GBT links are also used for monitoring and control of the detector and the RUs. 2 CRUs are installed in dedicated computers called First Level Processors (FLPs). These communicate both to the Offline-Online system for data offloading, as well as the Detector Control System (DCS).

The expected radiation environment in the location of the RU is ≥ 20 MeV high-energy hadron flux at 1 kHz/cm^2 [1] and with a Total Ionizing Dose (TID) of < 100 Gy for run 3. The fairly low TID is not considered to be a problem for the RU as numerous irradiation campaigns of the RU have confirmed. The high-energy hadron flux is on the other hand a concern. The RU has been designed with a modern Xilinx Kintex Ultrascale XCKU060 FPGA (see section 2 and fig. 1) at its core, where the Configuration RAM (CRAM) is SRAM based. It is therefore vulnerable to Single Event Upsets (SEUs), which is defined as a radiation induced change of value in a memory element. The expected SEU rate in the CRAM is 0.0004 Hz per FPGA and mitigation strategies are needed to avoid functional interrupts: (1) the design of the Xilinx FPGA must be protected, for instance by Triple Modular Redundancy (TMR), and (2) the SEUs in the CRAM must be continuously corrected to avoid accumulation of errors, commonly known as scrubbing. This paper describes how the scrubbing mechanism is implemented on the RUs of the ITS detector.

2. Readout unit overview

The Xilinx FPGA, from now on labeled main FPGA, interfaces the sensor elements, a dedicated power board, and it sends data and receives triggers from the CRU and the trigger system using the three radiation hard GBTx ASICs [2]. In addition, one radiation hard GBT-Slow Control ASIC (SCA) is used for slow control communication. The GBTx data frame has a data-field of 80 bits, which is used for communication with the main FPGA, and a 4 bit slow control field which is use for communication with the GBT-SCA chip. The SCA interfaces to the second FPGA on the RU; the flash-based Microsemi ProASIC3 A3PE600L (PA3). An external flash memory (Samsung K9WBG08U1M [3]) provides the storage for the main FPGA configuration files. The PA3, the flash memory and the SCA are the components comprising the external scrubber design.

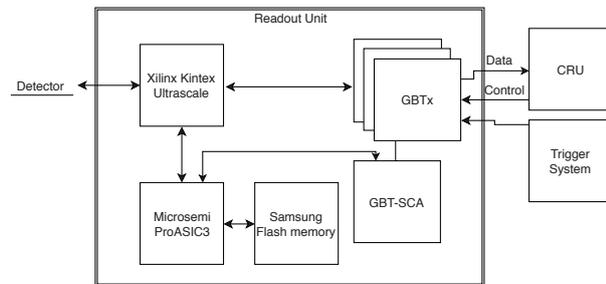


Figure 1: Simplified Readout Unit block diagram, highlighting the external scrubbing solution

Radiation mitigation techniques on the RU TMR is a common mitigation strategy which is implemented by triplicating SEU sensitive components (registers/memory/logic) followed by a

majority voter on the output. TMR is used on both FPGAs on the RU to protect the design from being affected by SEUs. There are multiple implementation schemes for TMR, and although similar, the effectiveness is decided by the FPGA technology [4, pp 73-80]. For ProASIC flash-based FPGAs, local TMR is normally adequate, as the configuration flash-cells in these FPGAs are immune to SEUs [5]. The flash memory is using a larger feature size, thus it requires more charge injected to change state of the flash cell, consequently lowering the probability of SEUs. Still, the flash memory bits has a data-dependent SEU cross-section of 10^{-21} or 10^{-16} cm²/bit depending on whether the data in the flash cell is high ('1') or low ('0') [6]. For radiation tolerance, and proper operation, the files stored on the flash memory are encoded using an error correcting code (ECC), that enables two bit detection and one bit correction. The ECC codes are applied to blocks of 128 bytes, and interleaved in the data stream. Additionally, the Xilinx configuration files typically have a 20:1 ratio of '0' to '1', and the cross-section for the flash memory is data-dependent. Consequently, the data is stored inverted on the flash. The radiation tolerance of the RU is further improved by duplicating the content of the flash, and by the implementation of an external scrubber in the PA3, which will be discussed in the following section.

3. External Scrubber implementation

The main objectives of the external scrubber are: (1) To configure the main FPGA from the flash memory on power-on or later by a command, and (2) to perform blind scrubbing of the main FPGA CRAM to avoid accumulation of SEUs. Blind scrubbing implies that no read-back of the CRAM is done prior to writing. Since the introduction of the Soft Error Mitigation (SEM) IP, external scrubbing is not officially supported by Xilinx. The SEM IP is an on-chip solution for repairing errors in the Xilinx configuration memory [7, ch. 1], and was considered early in the design process. However, based on irradiation campaigns of the SEM IP of a Xilinx Kintex 7, it was concluded that external scrubbing was more reliable. Consequently, an external scrubber solution was selected, similar to what was done for other ALICE sub-detectors in the previous runs [8].

The scrubbing can be done in two ways, either in discrete steps by a command controlled by the DCS software, or continuously. In continuous mode, the scrubbing rate is 0.58 Hz, which is significantly faster than the SEU rate of 0.0004 Hz, hence the chances of experiencing accumulated errors in the CRAM are small (the expected number of accumulated errors in a 10h run for the entire detector becomes significantly smaller than 1 with scrubbing).

Generation of scrubbing files When generating the files for blind scrubbing, two things are important: (1) the BRAM (dedicated RAM circuits) part of the configuration stream is not possible to scrub, and it must not be overwritten, and (2) if LUTRAMs (logic cells configured as RAM) are used in the design, these configuration bits must be masked. The latter is done by setting the appropriate value in the configuration registers in the Xilinx FPGA, and the first by not addressing the configuration frames containing the BRAM elements. RAM elements cannot be scrubbed and must have other means of mitigation. The valid frame addresses for the scrubbing file is extracted from the tool-chain JTAG Configuration Manager (JCM) [9]. In our system the scrubbing and configuration files are automatically generated by custom scripts as part of the development procedure. These scripts also calculates the ECC codes by the algorithm described in [10], and interleave these in the datastream so that the files are fully prepared for being stored on the flash memory.

PA3 FPGA design overview A simplified block diagram highlighting the main data-paths in the PA3 FPGA design is seen in fig. 2. During an initial configuration or a scrubbing (path II in fig. 2), the Configuration Controller takes charge of both the flash interface and the SelectMAP interface in the PA3. There are only minor differences in the scrubbing and initial configuration procedure.

A data width of 8-bits was chosen for the Wishbone (WB) bus and the SelectMAP, because the external flash memory has an 8-bit data width. Furthermore, in order to optimize data transfer from the GBT-SCA, a 7-bit address width was chosen for the WB bus, which allows the 7-bit I²C address to be mapped directly to the WB bus.

An ECC decoder for decoding the data on the flash memory is implemented in the PA3. The decoding is done in parallel with a FIFO with a depth of 256 words (see fig. 2). When a 128 byte block is buffered then the syndrome can be calculated. Any single bit error is then corrected on the fly. This solution only adds 393 clock cycles of latency, as the depth of the FIFO ensures a pipe-lining behavior of the file reading. In case of a double bit error, the ECC module will halt the current operation to avoid writing faulty data. Restarting the scrubbing and fixing the errors on the flash will require intervention from the DCS.

Design choices for radiation tolerance Local TMR is employed in all parts of the design in the PA3, and the FIFOs are triplicated. Additionally, the use of PLLs are avoided for the system clock, because of their higher cross-section. A simple Johnson ring counter is used instead, a solution that is less sensitive to single event effects. The index to the files on the flash is also refreshed on every configuration/scrubbing cycle, which is implementing a kind of scrubbing mechanism for these registers in the PA3.

Flash Structure The flash memory can be delivered with up to 320 memory regions with defects. These are called invalid blocks and they are labeled by a bit in the beginning of the memory region. A special tool has been developed to read and database these blocks. Because the invalid blocks occur randomly, each RU needs to have the capability to store the configuration data in different locations. This is solved by using the first page, which is guaranteed to be valid, as an index for the configuration and scrubbing files. During a configuration or scrubbing process, the index is always read first. This is also beneficial for long term operation of the RU, as aging will cause more invalid blocks to form over the lifetime of the experiment.

An additional feature of the flash memory is that it contains two separate ICs embedded in the same package, i.e. it is possible to store a double copy of the content on the flash without any time-penalty.

Uploading files to the flash memory For programming the flash memory a FIFO interface is implemented in the main FPGA. This uses the main payload of the GBT down-link (path III in fig.2) instead of the slow-control. This data-path provides a maximum of 1.66 MB/s for download of the

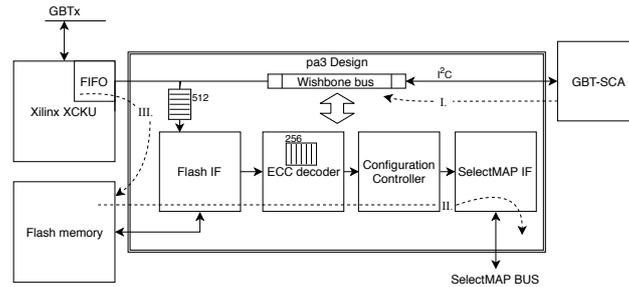


Figure 2: Simplified block diagram of the PA3 highlighting data flow

configuration files. As a backup, it is possible to program the flash via the I²C interface provided by the SCA (path I in fig. 2). A prerequisite for using the upload path via the main FPGA is that this device is configured. This is ensured by an additional pointer in the flash-index for a main FPGA golden image, which is a guaranteed stable version of the firmware that is used as fall-back.

4. Conclusion

The external scrubber design has been through several months of commissioning at CERN, with successful operation with the rest of the system. The initial configuration takes ~2 seconds, and one scrub-cycle ~1.7 seconds, which gives a 1400 times faster correction rate compared to the expected SEU rate. The programming time for initial configuration and scrubbing file to the flash is less than one minute. This operation can be done in parallel on each FLP, hence it can be expected that updating the flash will take ~15 minutes in the final system.

References

- [1] J. Schambach, M. J. Rossewijn, K. M. Sielewicz, G. A. Rinella, M. Bonora, J. Ferencei, P. Giubilato, and T. Vanat, *ALICE inner tracking system readout electronics prototype testing with the CERN “giga bit transceiver”*, *Journal of Instrumentation*, vol. 11, no. 12, pp. C12074–C12074, Dec. 2016. DOI: 10.1088/1748-0221/11/12/c12074.
- [2] P. Moreira, J. Christiansen, and K. Wyllie, *GBTX manual*, Draft 0.16, CERN.
- [3] *2G x 8 Bit / 4G x 8 Bit / 8G x 8 Bit NAND Flash memory user manual*, 1.0, Samsung, 2007.
- [4] M. Berg, *New developments in FPGA SEUs and fail-safe strategies from the nasa goddard space flight center perspective*, NASA, Tech. Rep. TN18767, 2014.
- [5] Actel, *ProASIC3E production FPGA features and advantages*, 2007.
- [6] G. Mikkelsen, *Integration and design for the ALICE its readout chain*, Master’s thesis, University of Bergen, 2018.
- [7] *PG036 SEM IP user guide*, Xilinx.
- [8] J. Alme, *Firmware Development and Integration for ALICE TPC and PHOS Front-end Electronics: A Trigger Based Readout and Control System operating in a Radiation Environment*, 2008.
- [9] A. Gruwell, P. Zabriskie, and M. Wirthlin, *High-speed programmable FPGA configuration through JTAG*, in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, Aug. 2016. DOI: 10.1109/fpl.2016.7577336.
- [10] I. Micron Technology, *Tn-29-08: Hamming codes for NAND flash memory devices overview*, Tech. Rep., 2005.