Nuclear Instruments and Methods in Physics Research A 629 (2011) 260–268

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

Fault injection as a test method for an FPGA in charge of data readout for a large tracking detector

K. Røed [a,b,*], J. Alme [c], D. Fehlker [c], H. Helstrup [b], M. Richter [c], D. Röhrich [c], K. Ullaland [c]

[a] CERN CH-1211, Genève 23, Switzerland
[b] Bergen University College, P.O. Box 7030, Nygårdsgaten, 112 5020 Bergen, Norway
[c] University of Bergen, P.O. Box 7803, 5020 Bergen, Norway

ARTICLE INFO

Article history:
Received 2 February 2010
Received in revised form
5 December 2010
Accepted 6 December 2010
Available online 17 December 2010

Keywords:
Fault injection
Radiation effects
Single event upsets
Partial reconfiguration
FPGA

ABSTRACT

This paper describes how fault injection has been implemented as a test method for an FPGA in an existing hardware configuration setup. As this FPGA is in charge of data readout for a large tracking detector, the reliability of this FPGA is of high importance. Due to the complexity of the readout electronics, irradiation testing is technically difficult at this stage of the system commissioning. The work presented in this paper is therefore motivated by introducing fault injection as an alternative method to characterize failures caused by SEUs. It is a method to study the effect that a configuration upset may have on the operation of the FPGA.

The target platform consists of two independent modules for data acquisition and detector control functionality. Fault injection to test the response of the data acquisition module is made possible by implementing the solution as part of the detector control functionality.

Correct implementation is validated by a simple shift register design. Our results demonstrate that fault injection can assist in measuring the effect of an implemented mitigation technique in the final design of the FPGA.

1. Introduction

The ALICE experiment [1] is an experiment at the Large Hadron Collider (LHC) that is currently under commissioning at CERN. High energy beams of particles (Lead–Lead and proton–proton) will be collided. A shower of new particles will be produced in the collisions. These new particles will be detected by different sub-detectors of ALICE in order to determine the particle type, energy and momentum. An important node of the readout electronics in the main tracking detector of ALICE is the Readout Control Unit (RCU) [2]. It uses a Xilinx Virtex-II Pro 7 Field Programmable Gate Array (FPGA) for data readout, hereafter called the RCU main FPGA. Simulations of the TPC detector radiation environment [3] have shown that the RCU main FPGA will experience a flux of energetic ($E > 10$ MeV) protons, neutrons and pions of approximately 100–400 particles/cm^2/s. In contrast to applications for space environments it will not experience a significant heavy ion flux. The main cause of single event upsets is therefore charge deposited by inelastic nuclear reaction fragments produced in the device material. Due to the high number of RCU nodes and several hours of continuous data taking it is expected that the system will experience functional failures due to Single Event Upsets (SEU) [4]. An SEU is a change of state (logical 1–0 or vice versa) of a configuration memory cell or any other type of memory cell. It is a soft error as it can be corrected by rewriting the originally stored value to the memory cell. During its temporary existence it can however lead to loss of readout functionality of detector data which should be avoided if possible.

1.1. Fault injection as a test method

Partial reconfiguration is a feature offered by several Xilinx FPGAs that allows the reconfiguration of a subset of the configuration memory without interrupting the operation of the device [5]. This functionality has already been implemented in the RCU system in order to detect and correct SEUs [6]. By continuously reading back the configuration memory, an SEU is detected by comparing the read back data to the original data stored in a separate location. If different, the corrupt part of the configuration memory is reconfigured with the correct data. While the main purpose during normal operation is to keep the original data of the configuration memory intact, this very same solution can be used to write incorrect data to the configuration memory. The latter is referred to as fault injection. It allows the injection of errors in the configuration memory with the purpose of studying any effects this may have on the operation of the RCU main FPGA.

* Corresponding author at: Bergen University College, P.O. Box 7030, Nygårdsgaten, 112 5020 Bergen, Norway.
E-mail address: ketil.roeed@cern.ch (K. Røed).

0168-9002/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2010.12.033

Due to the complexity of the readout electronics, irradiation testing is technically difficult at this stage of the system commissioning. The work presented in this paper is therefore motivated by introducing an alternative test method for the RCU main FPGA. By implementing fault injection on the existing hardware there was no need for a separate test setup. Consequently fault injection can be run in the normal configuration of the readout system. This allows us to study the effect a configuration memory upset may have on the readout of detector data.

1.2. Related work

The work presented in this paper is constrained by the system implementation. The main purpose was not to develop a general fault injection scheme optimized to test the efficiency of various SEU mitigation techniques. Other tools like for instance the Flipper injection platform described in Ref. [7] are already available for this purpose. However, it is difficult to see how the Flipper platform could have replaced the RCU in order to run fault injection while connected in the data readout path of the detector. Therefore, the only possible solution was to integrate fault injection in the existing detector readout electronics. This allows us to test and improve the effect of specific mitigation techniques implemented to prevent loss or corruption of data during readout of the detector.

Other fault injection implementations are also reported in the literature. An extensive study is presented in Ref. [8] where fault injection is used to study the effect of a Triple Modular Redundancy (TMR) design. Fault injection studies to investigate the mitigation of Xilinx Virtex-II input/output blocks are presented in both Refs. [9,10]. In Ref. [11] fault injection is demonstrated as a test methodology to ensure that a design has been hardened. Compared to, for example, accelerated beam testing it proved more effective in terms of cost and test coverage.

Fault injection is not a method to characterize the SEU sensitivity of the FPGA configuration memory to radiation. The SEU cross-section can only be determined by using accelerated beam testing. However, fault injection can be used to investigate any abnormal behaviour that may be caused by a configuration memory upset. A long beam time is needed to achieve reasonable statistics for this type of testing. Fault injection offers a systematic approach and can be used to support beam tests and therefore reduce the amount of beam time needed. For example, for the studies reported in Refs. [12,13] it is reported that fault injection can be an effective alternative to extensive beam testing. Their fault injection results show very good agreement with irradiation testing using accelerated protons.

1.3. Structure of the paper

The scope of this paper is to describe how fault injection has been implemented in the RCU system. Section 2 introduces the existing system focusing on the reconfiguration network that has been used to implement fault injection. Section 3 further explains the implementation of fault injection in the system and introduces the test scenarios that were used to validate that it was working. Results of the test runs are then presented followed up by a discussion on limitations and prospective use.

2. System overview

The Time Projection Chamber (TPC) is the main tracking detector of ALICE. A main node in the readout electronics of this detector is the RCU. It consists of the RCU motherboard, the Detector Control System board (DCS) and the Source Interface Unit (SIU). In total 216 nodes are present in the system serving 4356 front end cards and roughly 560,000 individual data readout channels from the detector. A simplified sketch of the RCU is shown in Fig. 1 and more detailed information can be found in Refs. [2,14,15].

During normal operation of the ALICE experiment its radiation environment will prevent physical access to the system. Remote access is therefore provided by an Ethernet connection to the DCS board which is an embedded computer running Linux. Its main task is to monitor and configure the front end cards and RCU motherboard. As it is not a direct part of the data path occasional downtime can be accepted. In contrast, the RCU main FPGA on the RCU motherboard is in charge of the data readout and if interrupted by an SEU may lead to loss of experimental data. The system therefore implements a solution to correct the configuration memory of the

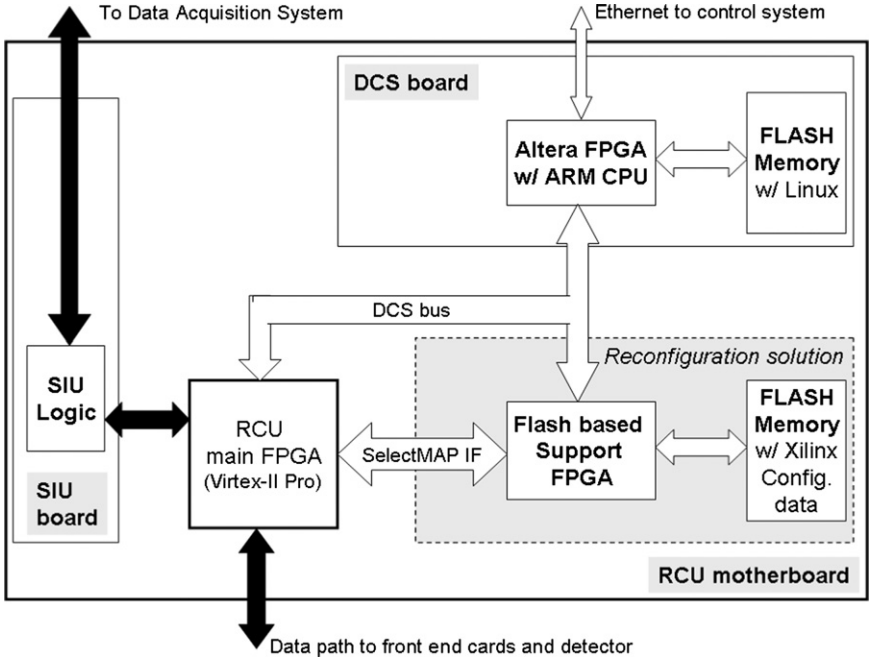

Fig. 1. Conceptual schematic of the RCU motherboard. The data path of the system is marked with black arrows. As can be seen it passes through the Xilinx Virtex-II Pro FPGA.