

Development of new data acquisition system for COMPASS experiment

Abstract

This paper presents development and recent status of the new data acquisition system of the COMPASS experiment at CERN with up to 50 kHz trigger rate and 36 kB average event size during 10 second period with beam followed by approximately 40 second period without beam. In the original DAQ, the event building is performed by software deployed on switched computer network, moreover the data readout is based on deprecated PCI technology; the new system replaces the event building network with a custom FPGA-based hardware. The custom cards are introduced and advantages of the FPGA technology for DAQ related tasks are discussed. In this paper, we focus on the software part that is mainly responsible for control and monitoring. The most of the system can run as slow control; only readout process has realtime requirements. The design of the software is built on state machines that are implemented using the Qt framework; communication between remote nodes that form the software architecture is based on the DIM library and IPBus technology. Furthermore, PHP and JS languages are used to maintain system configuration; the MySQL database was selected as storage for both configuration of the system and system messages. The system has been design with maximum throughput of 1500 MB/s and large buffering ability used to spread load on readout computers over longer period of time. Great emphasis is put on data latency, data consistency, and even timing checks which are done at each stage of event assembly. System collects results of these checks which together with special data format allows the software to localize origin of problems in data transmission process. A prototype version of the system has already been developed and tested the new system fulfills all given requirements. It is expected that the full-scale version of the system will be finalized in June 2014 and deployed on September provided that tests with cosmic run succeed.

Keywords:

FPGA, hardware eventbuilding, Qt

1. Introduction

This paper presents development and recent status of the hardware and software part of a new data acquisition system (DAQ), based on the Field Programmable Gate Array (FPGA) technology, for the COMPASS (Common Muon Proton Apparatus for Structure and Spectroscopy) experiment at CERN. COMPASS [10] is a fixed target experiment situated at CERN's SPS particle accelerator that studies hadron structure and hadron spectroscopy with high intensity muon and hadron beams. During the previous years, it had a usual data rate of approximately 1500 MB/s during approximately 10 seconds on-spill with the off-spill time between 30 and 50 seconds, depending on SPS super cycle. The

original DAQ of the experiment was built during years 1999-2001. The Data Acquisition and Test Environment (DATE) software [1], originally developed for the ALICE at CERN, was used to control DAQ and event building in old system. Both software package and usage of FPGA-based cards have been widely studied [8], [3], [5], [4] and as the result a design of the new DAQ was prepared.

Development of the new DAQ software and hardware was started to improve reliability and speed of system. Main idea of the hardware upgrade is to use FPGA technology for event building purposes and consequently reducing number of used computers to only eight. Hardware event building was previously investigated by the CDF experiment [2] at Fermilab and the NA48 exper-

iment [12, 13] at CERN. Both these experiment returned to software event-building; still reliable, flexible and cost-effective hardware event-building can be prepared today thanks to improvements in FPGA technology. The new software has to cope with challenges linked to control of such new hardware event-building network and has to allow users to operate whole system efficiently.

2. COMPASS DAQ architecture

2.1. Old DAQ architecture

The original COMPASS DAQ system connected to experimental setup consisted of several layers. The frontend electronics that form the lowest layer continuously preprocess and digitize analogue data from the detectors. There are approximately 300 000 detector channels. Data from multiple channels are readout and assembled by the concentrator modules called CATCH, GeSiCA, and GANDALF. These modules receive the signals from the time and trigger system; when the trigger signal arrives, the readout is performed. By adding the timestamp and the event identification to the data the subevent is created. Parts of the old DAQ, up to this point, are used in unchanged form in the new DAQ.

Next layer of the original DAQ was event building network composed of readout buffers and event builders. Readout buffers were standard servers equipped with a custom PCI cards with optical receiver called spillbuffers. Spillbuffers were used for buffering of subevents which allowed to distribute the load through the full cycle of the SPS accelerator. Finally, subevents were sent over the Gigabit Ethernet to the event builders that assembled full events. Assembled events were stored temporarily on event builder's local disks before being transferred to the CERN Advanced STORAGE manager (CASTOR). Described structure of the original DAQ architecture is shown in Figure 1 [8, 5].

2.2. New DAQ architecture

The event building network has been replaced with two layers of special FPGA Data Handling Cards (DHC) as shown in Figure 3 in the new DAQ. This newly designed event building part allows usage of more compact control system. The hardware event builder performs online data consistency check and includes programmable error recovery algorithm with configurable error tolerance level. These features of the design of the new DAQ will make the system more reliable. From this part the full events are transferred to

8 readout engine computers, where they are received by PCI-e Spillbuffer cards, copied via DMA transfer to RAM, converted to the DATE format, and stored temporarily on the local disks before being transferred to the CASTOR. All DHC cards are controlled over separate network by processes using the IPBus package originally developed for the CMS trigger level 1 upgrade. This package consists of firmware part and software part. The firmware part mediates access to registers and memory of an FPGA card through Ethernet. The software part is implemented in C++ language and contains all classes needed for a connection to the interface of the firmware part.

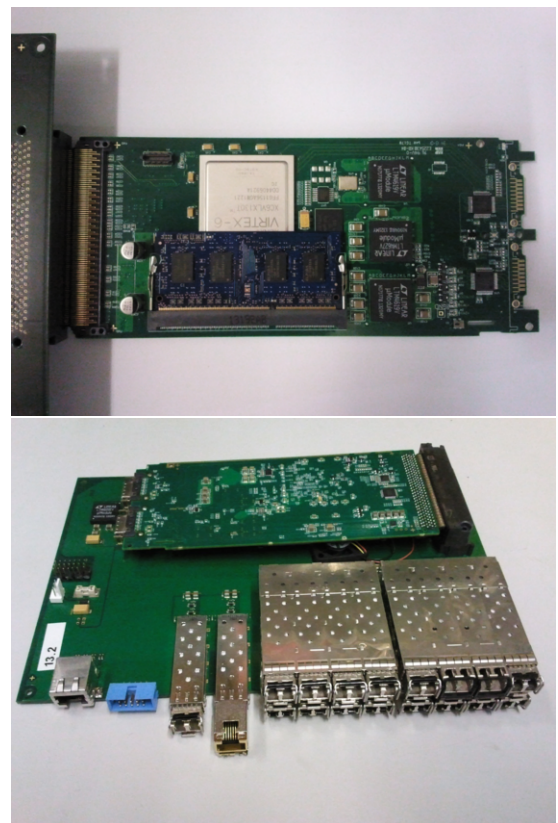


Figure 3: New DHC-Switch/Multiplexer FPGA card

2.3. Technologies of the new DAQ design

Many different technologies are used for the new DAQ with the FPGA being the key one. FPGA chips with their ability to change structure by loading different firmware and with many high speed serial links, high speed DDR3 memory, optical fiber transceivers, and fast Ethernet for control purposes, has made possible to create DAQ module for event-building.

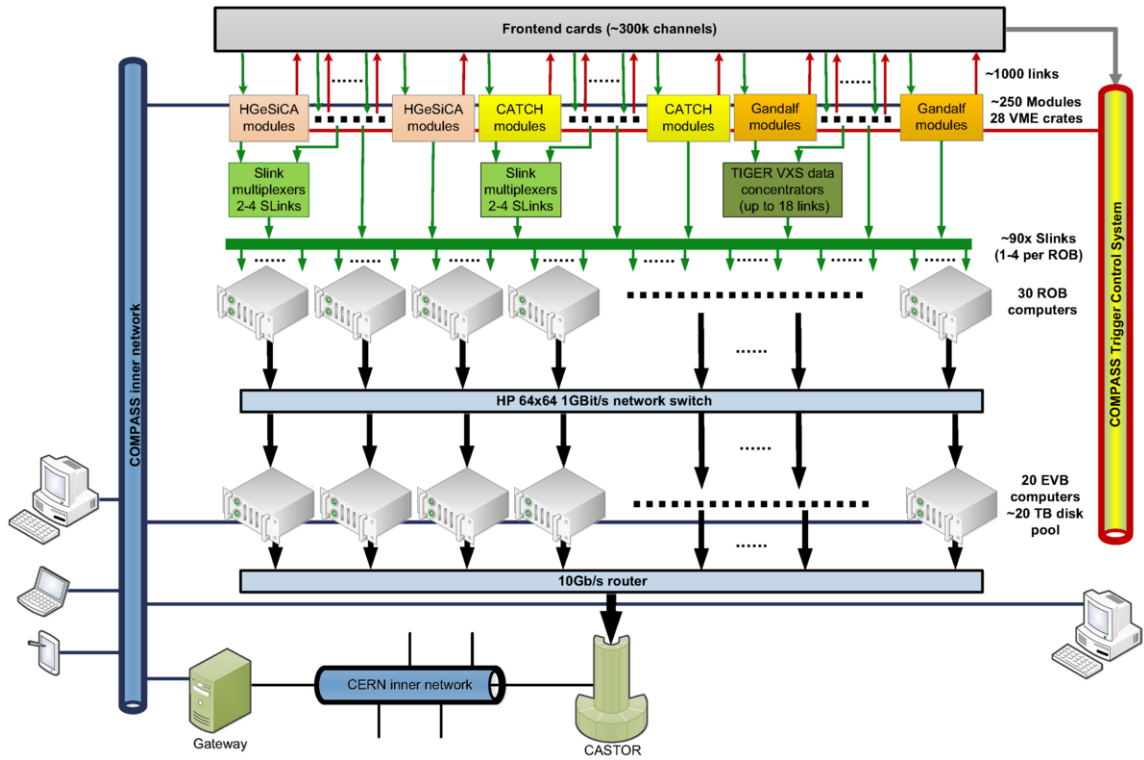


Figure 1: The original DAQ architecture

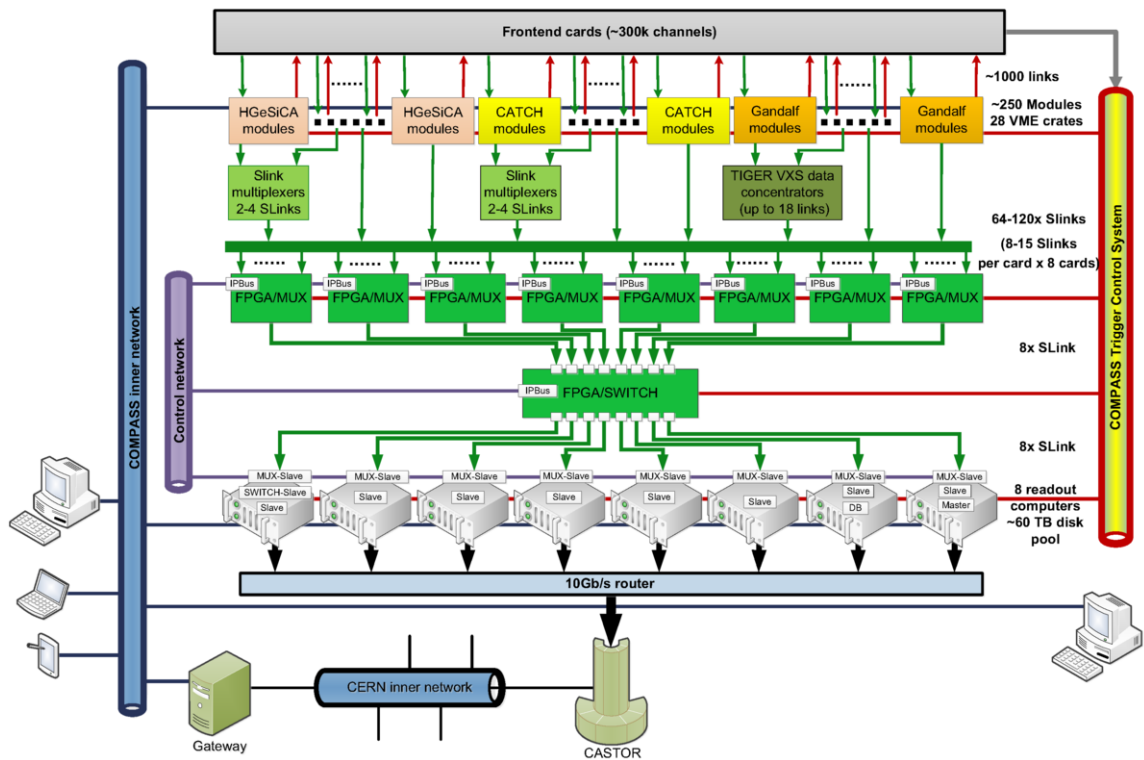


Figure 2: The new DAQ architecture

Major part of the new DAQ software has been implemented in C++ language. It is supported by MySQL for database access and Python with bash scripts for minor tasks. PHP, HMTL, javascript, and AJAX technologies have been used for development of web-based configuration interface. The Qt framework, a cross-platform application framework, has been used for all main graphical user interfaces (GUIs) and to speed up development of core applications. Some support GUIs, written in Tool Command Language (TCL), were taken from the original DAQ and reused in the new one.

The Distributed Management System (DIM) library is used for communication between processes of the DAQ. The DIM is a multi-platform library that serves for an asynchronous one to many communication through the Ethernet. It was originally developed for the DEPHI experiment at CERN. The IPBus package is used for communication between FPGA modules and control processes. It was developed for the level one trigger update of the CMS experiment.

2.4. The new hardware parts

The DHC card is the main new piece in the hardware design. This card has been designed as a compact AMC card [8, 7] and features 2 GByte of DDR3 memory, 16 high speed serial links with configurable bandwidth from 1 Gbps up to 6.25 Gbps, Gigabit Ethernet, and COMPASS Trigger Control System receiver.

These cards will be used with two different functionalities - multiplexer and switch - which correspond to two stages of the event building process. Their basic function is shown in Figure 4. Changes between functionalities are purely matter of firmware. The first stage of FPGA cards will multiplex up to 120 incoming links to 8 outgoing ones, providing one more data concentration level. The second stage consists of only one DHC card, which performs the event building and the event distribution functions. Distribution will be based on the lookup table.

A bandwidth of the DHC DDR3 memory is 6 GB/s which exceeds combined maximum data rate of incoming links for both DHC-Multiplexer and DHC-Switch architectures and allows to collect data without throttling data transmission even at maximum rate. The multiplexer will combine data from up to 15 front-end modules providing about 240 MB/s at the outgoing link on average. The links between the multiplexers and the switch have a bandwidth of 300 MB/s each allowing transmission of data without storing them in local memory.

The Spillbuffer card is another new piece of equipment in the new DAQ. This card is a commercially

available FPGA based PCI-Express card developed by Inrevium company with optical input interface that buffers a data stream from data producers before copying them to a computer memory.

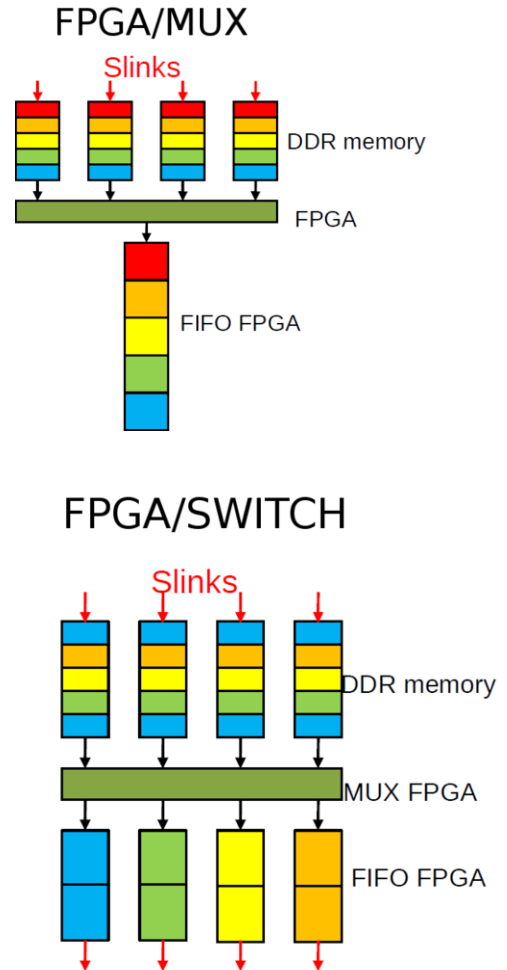


Figure 4: Two different functionalities of DHC card

2.5. The new DAQ software architecture

The new data acquisition system provides six main functions [4]. The configuration of the hardware is the first function; it includes interface for changing of configuration. Monitoring of data taking process is the second function, it incorporates the control of status of all hardware devices, the status of software, and the quality of collected data. Information about hardware contain the status of network communication, the load of event

builder's CPU and memory, level of FPGAs FIFO, error levels of hardware, and status of HDD. The remote control of hardware is the third function. The subsystem starts and stops the data taking, gives commands to the FPGA card, and performs other tasks. Full list of tasks is not yet finalized. The data flow control is the fourth function. With the help of information provided by the second function the system must control data flow, in a such way that data are sent to the least busy hardware on every level. This function is important for the uniform load of a hardware, thus no node should be overfilled with the work and no node should be unemployed. The logging of information and errors is the fifth function. The log browsing is the last function. It must work only with the database and must be independent on other parts of DAQ. It must be easy to use and well arranged, [5].

There are six types of processes fulfilling these six functions in the new DAQ [7]: Master, Slave-control, Slave-readout, Runcontrol GUI, MessageLogger, and MessageBrowser. The Master process is responsible for control of the system by retranslation of messages from user to slaves according to configuration loaded from database. It has access to all slaves through DIM services and direct access to MySQL database. It also has integrated error recovery functions to cope with problems caused by misbehaving slave processes. The Slave-control process supervises connected FPGA card by accessing registers via IPBus. The full scale system will contain 17 Slave-control processes which will be distributed over the readout computers. Communication by IPBus is shown as dash-dot line in Figure 5. The Slave-readout process is the most complex and demands most of CPU resources in the new DAQ. It is a multi-threaded process that monitors readout activities and checks consistency of accepted data. A Spillbuffer card is used as the data source in both the old and in the new DAQ. The data are transferred between threads via signal-slot connections mechanism of Qt by blocks of about 1000 events. Events are distributed to 8 processing threads before final checks and preparation of the final data format. Portion of data is, simultaneously with storing on the HDD, distributed to monitoring outputs according to the set filter parameters. The main graphical user interface is implemented in Qt framework. It has been designed and developed with emphasis on ergonomics and flexibility. It provides DAQ status information for expert and nonexpert users. It runs in one of two modes: runcontrol and monitoring. There is only one runcontrol GUI allowed in the system; it controls and monitors state of system. The number of running monitoring GUIs is not limited, as they are used only

for monitoring. MessageLogger and MessageBrowser are the last two programs to be discussed [5]. The MessageLogger receives messages from all parts of the system and stores them in the database. The MessageBrowser is a visualization tool for browsing through these messages. The master process and slave processes are based on state machines.

3. Present status

The DAQ is presently running with 75 links connected to 5 DHC/MUX card and through 1 DHC card programmed as switch to 1 readout engine computer. It is used for commissioning of COMPASS spectrometer setup for run in November 2014. Maximum speed of readout, tested so far, is approximately 150 MB/s. Limitation in rate with single computer is caused by usage of Slink protocol. Change to Aurora protocol is foreseen in future. Tests with more readout computers are planned for end of October 2014.

4. Conclusion

Design of the DAQ has been prepared with respect to demands and restrictions which were extracted from the initial studies of the present DAQ of the COMPASS experiment at CERN [4, 5, 3] and discussion among collaboration of the experiment. All processes were implemented in the C++ language using the QT library. PHP, MySQL, javascript, bashscript, and python were chosen as languages for support function and web interface. The COMPASS typical data rate is 1500 MB/s during spill which is collected from more than 100 front-end modules. A maximum aggregated throughput of the designed system is 1.5 GB/s, but taking into account accelerator duty cycle and significant local memory resources, it has a safety margin of 200-300% and possibility of future improvement by exchange of Slink protocol for Aurora protocol. Prototype version is currently taking data from the COMPASS experiment.

Acknowledgments

The work on this project has been supported in part by the following grants: MSMT LA08015 and SGS 11/167. It is also being supported by the Maier-Leibnitz-Labor, Garching and the DFG cluster of excellence Origin and Structure of the Universe.

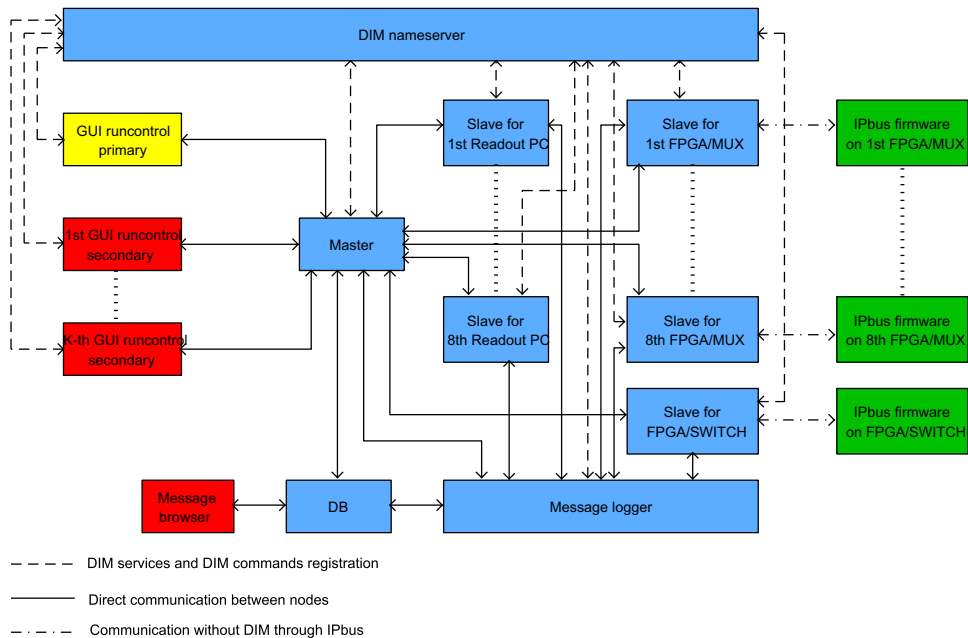


Figure 5: Communication diagram

References

- [1] T. Anticic, et al. (ALICE DAQ Project): *ALICE DAQ and ECS User's Guide* CERN, EDMS 616039, January 2006.
- [2] T. M. Shaw, et al.: *Architecture and development of the CDF hardware event builder* IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 36, NO. 1, AUGUST 1989
- [3] V. Jary: *Analysis and proposal of the new architecture of the selected parts of the software support of the COMPASS experiment* Prague, 2012, Doctoral thesis, Czech Technical University in Prague
- [4] J. Novy: *COMPASS DAQ - Basic Control System* Prague, 2012, Master thesis, Czech Technical University in Prague
- [5] M. Bodlak: *COMPASS DAQ Database Architecture and Support Utilities* Prague, 2012, Master thesis, Czech Technical University in Prague
- [6] M. Bodlak, et al. *Developing Control and Monitoring Software for the Data Acquisition System of the COMPASS Experiment at CERN*. Acta polytechnica: Scientific Journal of the Czech Technical University in Prague. Prague, CTU, 2013, issue 4. Available at: <http://ctn.cvut.cz/ap/>
- [7] M. Bodlak, et al. *FPGA based data acquisition system for COMPASS experiment*. Journal of Physics: Conference Series. 2014-06-11, vol. 513, issue 1, s. 012029-. DOI: 10.1088/1742-6596/513/1/012029. Available at: <http://stacks.iop.org/1742-6596/513/i=1/a=012029?key=crossref.78788d23de2b4a6a34d-127c361123b8c>
- [8] M. Bodlak, et al. *New data acquisition system for the COMPASS experiment*. Journal of Instrumentation. 2013-02-01, vol. 8, issue 02, C02009-C02009. DOI: 10.1088/1748-0221/8/02/C02009. Available at: <http://stacks.iop.org/1748-0221/8/i=02/a=C02009?key=crossref.a76044facdf29d0fb21f9e-efe3305aa5>
- [9] M. Bodlak, V. Jary, J. Novy: *Software for the new COMPASS data acquisition system*. In: COMPASS collaboration meeting, Geneva, Switzerland, 18 November 2011
- [10] P. Abbon, et al.(the COMPASS collaboration): *The COMPASS experiment at CERN*. In: Nucl. Instrum. Methods Phys. Res., A 577, 3 (2007) pp. 455518
- [11] L. Schmitt, et al.: *The DAQ of the COMPASS experiment*. In: 13th IEEE-NPSS Real Time Conference 2003, Montreal, Canada, 1823 May 2003, pp. 439444
- [12] E. Bal, et al.: *The NA48 Data Acquisition System* IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 45, NO. 4, AUGUST 1998
- [13] M. Wittgen, et al.: *The NA48 Event-Building PC Farm* IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 47, NO. 2, APRIL 2000