Readout Unit Prototypes for the CMS DAQ System

Antchev G. [1], Cano E. [1], Cittolin S. [1], Erhan S. [2], Faure B. [1], Gigi D. [1], Gutleber J. [1], Jacobs C. [1], Meijers F. [1], Meschi E. [1], Ninane A. [3], Orsini L. [1], Pollet L. [1], Racz A. [1], Samyn D. [1], Sinanis N. [4], Schleifer W. [1], Sphicas P. [1, 5]

[1] CERN, European Organization for Nuclear Research, Geneva, Switzerland
[2] University of California Los Angeles, USA
[3] Universite catholique de Louvain, Louvain-la-Neuve, Belgium
[4] ETH Zurich, Switzerland
[5] Massachusetts Institute of Technology, USA
E-mail: Gueorgui.Antchev@cern.ch

Abstract

In the context of developing a DAQ Prototype System for the CMS experiment CERN we have designed a Readout Unit (RU). The RU is a part of the DAQ Readout Column and has up to 512 MBytes memory. It is capable of handling data events at 400MB/s bandwidth. This unit is based on PCI (PMC) modularity and exhibits a reconfigurable structure. This paper describes the hardware implementation of the RU and its components.

I. INTRODUCTION

Future experiments in High Energy Physics, such as the Compact Muon Solenoid (CMS) [1] for the LHC at CERN, raise the demand for high performance Data Acquisition Systems (DAQ). Multilevel filtering DAQ architectures require fast buffering for intermediate storage of raw data while the event is processed in various levels. Usually a fast dual-port memory is used as data buffer. It also offers the possibility to collect large amounts of data that span the sizes of multiple events. Standard bus interfaces such as the Peripheral Component Interconnect (PCI) [2] are used whenever possible. PCI Mezzanine Cards (PMC) [3] are intended to be used where slim, parallel board mounting is required for host modules with the logical and electrical layers based on PCI.

II. CMS DAQ ARCHITECTURE

The architecture of the CMS DAQ system is shown in Figure 1. The RU is a major part of the readout system. It resides between the Front-End Drivers (FED) and the Builder Data Network (BDN). The RU catalogues events according to their Event-Ids, which are provided by the Event Manager (EVM) and serves as an intermediate buffer. The RU is capable of receiving event data from the FED at 400MB/s and can send requested event fragments at the same time to the BDN at 400MB/s.

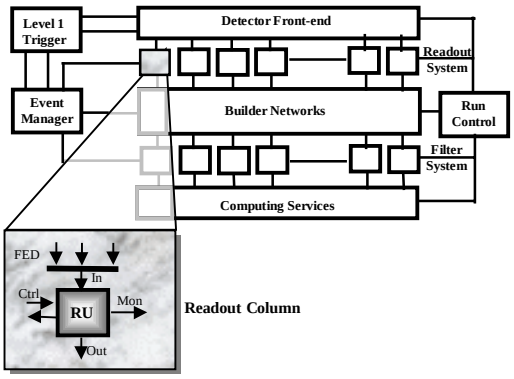

Figure1: CMS DAQ System

III. READOUT UNIT

A. RU Functions and Requirements

The RU functional blocks are shown in Figure 2. It has four communication ports and comprises the following basic functional and structural components: Readout Unit Input (RUI) - input for event data (up to 4KB) at a rate of 100KHz, Readout Unit Output (RUO) - output for sending data to the BDN, Readout Unit Memory (RUM) - dual-port memory up to 512 MBytes size for storing the event fragment data and a Readout Unit Supervisor (RUS) for additional control and monitoring. A fast interconnect is used between all components of the RU.

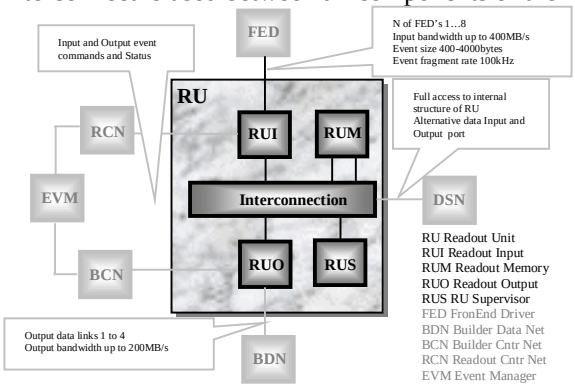

Figure 2: RU Functional Blocks

B. RU Block Diagram

The block diagram of the RU is shown in Figure 3. It is implemented by two physical boards, called Readout Unit Memory (RUM) and Readout Unit Input Output (RUIO) respectively. Both boards are designed as long-size 64bit PCI units, which operate at 33/66MHz and can be plugged into any standard 64bit PCI slot.

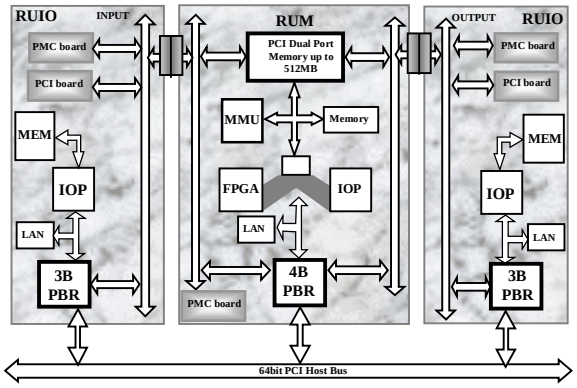

Figure 3: RU Block Diagram

The boards are configured via the common 64bit PCI host bus in which they are plugged. For internal connection between the boards, we also use the 64bit PCI bus protocol at 33/66MHz.

1) RUM

The Readout Unit Memory board is a central part of the RU. The RUM is a PCI dual-port memory with a third PCI bus for control purposes. Four PCI Bridges (4B PBR) are used to connect the input, output, control and a local bus. The local bus is used to connect the Memory Management Unit (MMU), the Memory Controller (MC), the PCI interface controllers and the PBR. The RUM board can be seen in Figure 4. It contains also memory and PCI/PMC connectors. For the mentioned building blocks we use fast re-programmable devices (e.g. the APEX series from Altera [4]). An IOP480 from PLX Corp. [5] acts as a local bus controller. The data memory is based on Synchronous DIMM modules with a possible capacity of up to 512 MBytes. The Synchronous DIMM modules can operate at up to 133MHz-clock frequency and in full-page burst mode. For each input and output ports two bi-directional 2Kx36 bits FIFO's are available as deep data buffer between the memory and the PCI interface controllers. The boards are currently undergoing functional tests.

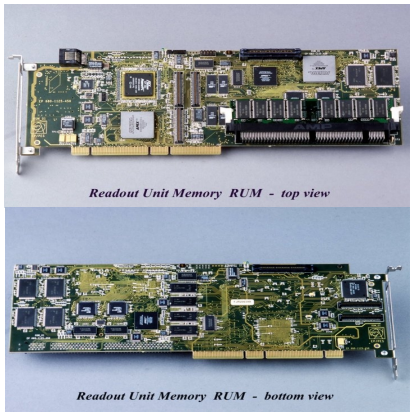

Figure 4: Readout Unit Memory

2) RUIO

The RUIO board is shown in Figure 5. It contains the following generic building blocks: three PCI bridges (3B PBR), a PCI to Local Bus IOP480 controller, Memory – (Flash, SRAM and DIMM), an Ethernet controller and PCI/PMC connectors. The core tasks of the RUIO are to extend the input and output bus of the RUM and to provide flexible control of the RU. Three PBRs are implemented using one FLEX10K200 component [4]. Commercial-off-the-shelf network interface cards (e.g. Myrinet [6], Ethernet, ATM and FC) can be plugged into PCI/PMC connectors to link a RU to the FED and BDN.

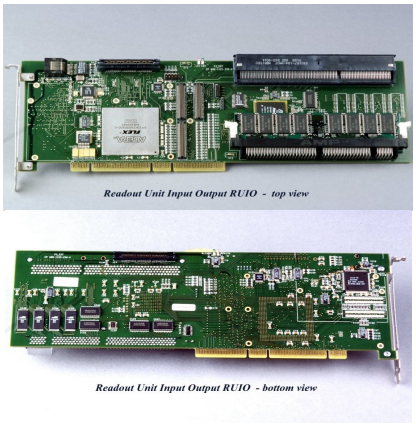

Figure 5: Readout Unit Input Output

The IOP480 controller has a 32bit/33MHz PCI bus interface and a 32bit local bus running at 60MHz, integrated memory SRAM/FLASH/SDRAM controller for up to 256MBytes size of memory, DMA controllers, a serial interface RS-232 and an I_2O (Intelligent I/O) [7] compliant messaging unit. As an Ethernet controller we use a 21143 PCI 10/100Base-T LAN controller from Intel. The chip is optimised for low power systems. The first five RUIO boards have been produced and have been functionally tested successfully.

IV. FPGA FLEXIBILITY

In order to implement all functions of the RU we decided to use reprogrammable logic devices (FLEX, APEX etc.). This decision is based on our experience from previous versions of the Readout Unit Memory (RDPM see [8]). An approach based on FPGA's provides us with enough resources to implement numerous different functions with no modification to the physical board.

A. PCI Bridge

Commercially available fast interface protocols such as the 64bit 33/66MHz PCI standards delivered the required bandwidth that is necessary to transfer data from the FED to the BDN. Data and control from one bus to the other are transferred using multiple PCI bridges that are part of the RUM and the RUIO. They are implemented in FPGA's. The basic structure of a version with four PCI busses is shown in Figure 6.

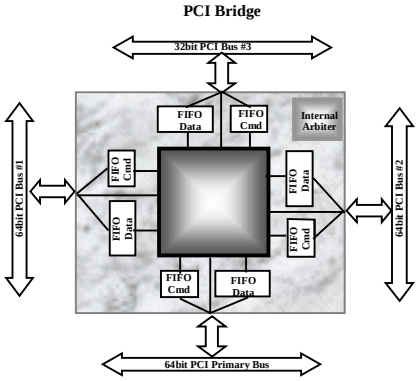

Figure 6: Four PCI Bridges

The core elements of the structure are unidirectional FIFO's for sending or receiving commands and data over the PCI busses. In addition, a PCI arbiter for each bus is implemented.

B. Memory Management Unit

The MMU is a device that organises event data in blocks using tables and pointers. It is receiving event header information from the RUM input PCI interface. Each header contains an Event-ID number, a word-count, the first memory block address and status information. The MMU also contains logic for releasing memory areas in the data memory. These areas are associated with events that are stored in the RUM. For these functions the MMU is using SRAM and has a direct connection also to the Memory Controller on board. The block diagram of the MMU is shown in Figure 7.

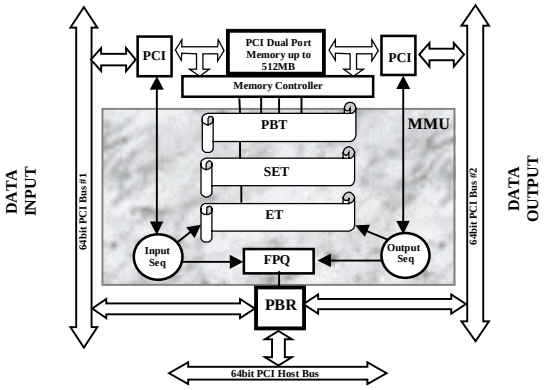

Figure 7: Memory Management Unit

C. Memory Controller

The Memory Controller (MC) is a device, which generates physical memory addresses from logical event data information that originates from MMU or PCI interface units. The MC also contains an internal arbiter for read/write access from/to the PCI ports, accessing address counters and control logic of the FIFO's. The Block diagram of the MC is shown in Figure 8.

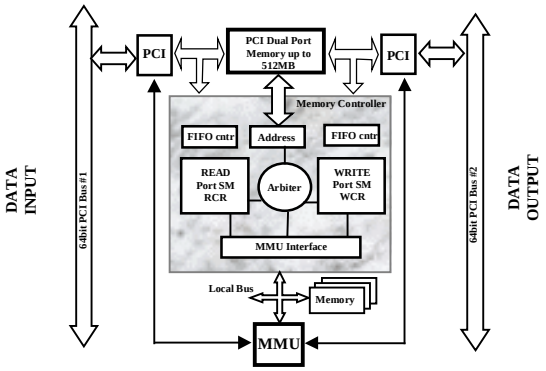

Figure 8: Memory Controller

D. Readout Unit Controls

The Readout Unit devices are configured via the Host PCI bus, see Figure 3. The data flow control in the RUM and RUIO designs is done by internal 32bit PCI bus. We envisage two different schemes of implementation of the data flow control. The first is to use a commercially available I/O processor (IOP480 from PLX Corp.). The second is to deploy programmable logic devices and to replace the control of the RUM with simple protocol. The former approach has been successfully implemented in the RUIO prototype. Experience with the IOP and the I_2O protocol will be accumulated during the prototype development. The block diagram of the data flow control implementation in the RUM is shown in Figure 9.

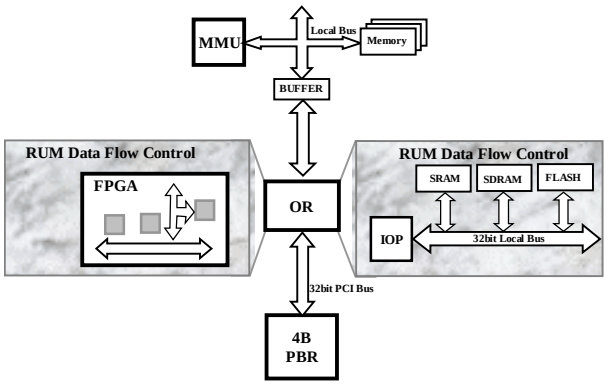

Figure 9: Data Flow Control Implementation in the RUM

V. RU CONFIGURATIONS

A Readout Unit is a set of physically connected PCI devices, namely a RUM and RUIO. The flexibility of each of these devices provides the possibility of building a RU by choosing one RUM and two RUIO's or one RUM and one RUIO. The first configuration is exhibited in Figure 10. There, two RUIO modules are connected to a RUM module. The RUIO's implement the functions of RUI and RUO. The input uses PMC and the output uses PMC and PCI boards.

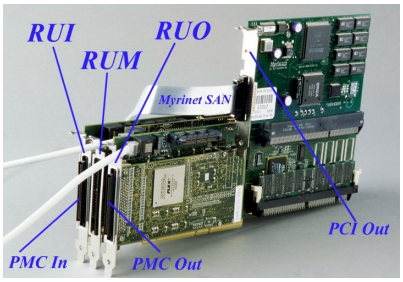

Figure 10: RUM and two RUIO

The second configuration with one RUM and only one RUIO is shown in Figure 11. The functions of RUI and RUO are realised by the RUIO module. For control and data we are using two independent PCI busses.

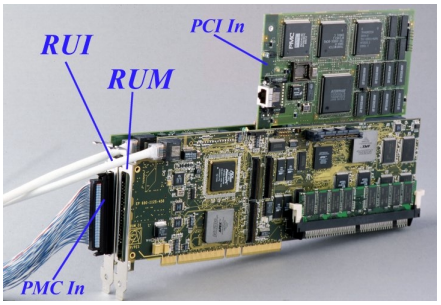

Figure 11: RUM and one RUIO

VI. SOFTWARE DEVELOPMENTS

For control and monitoring of the input, output and local bus of the RUM and RUIO, a Real-Time Operating System (VxWorks) has been ported to the IOP480 [9]. To accomplish this task a Board Support Package (BSP) has been designed which interfaces the hardware to the OS software layers. The next step, an adaptation of Linux for the mentioned processor is currently under development.

VII. CONCLUSIONS

The RU prototype follows closely the requirements of the CMS data acquisition system and is therefore a fundamental element for testing switched event builder systems under realistic conditions. It can also be considered as a stand-alone firmware DAQ system for test beam environments or mini experimental set-ups. The use of latest generation FPGA components provides a flexible way to implement and further improve the functions of a Readout Unit.

VIII. REFERENCES

1. CMS TriDAS Computing Controls, CMS Document 1997-090, CERN.
2. PCI Local Bus Specification Revision 2.0, April 30,1993.
3. Draft Standard for a Common Mezzanine Card Family: CMC IEEE P1386/Draft 2.0, April 4, 1995.
4. Altera Corporation, 101 Innovation Drive, San Jose, California 95134, USA http://www.altera.com/
5. PLX Technology, Inc. 390 Potrero Avenue Sunnyvale, California 94085 USA http://www.plxtech.com/
6. Myricom Inc., Arcadia, CA, USA http:://www.myri.com
7. I_2O Special Interest Group (SIG) http://www.i2osig.org
8. CMS FPGA dual port memory prototypes. D.Gigi - Third Workshop on Electronics for LHC Experiments. London, September 22-26,1997.
9. Software developments for the Readout Unit Prototypes for CMS DAQ System. Bellato, M. 6th Workshop on Electronics for LHC Experiments. Krakow, September 11-15, 2000.