DUAL-PORT MEMORY WITH RECONFIGURABLE STRUCTURE

Gueorgui ANTCHEV, Dominique GIGI
CERN EP/CMD Division
CH-1211 Geneva 23 Switzerland
Gueorgui.Antchev@cern.ch, Dominique.Gigi@cern.ch

Abstract

In the acquisition system for CMS, the RDPM is a dual-port memory (up to 256Mbytes) used to buffer and filter events. The third prototype is currently under study and development. It will be a PCI board with three PCI busses: an input bus to receive data form the DDU, an output bus to send data to the computer through a switch and a control bus (each one with 64-bit @ 33-66MHz to reach the 400MB/s data bandwidth). The board will be built with FPGA components. This is an advantage to reprogram the board to be flexible and to test different events organizations. The third prototype will be integrated in the DAQ system demonstrator

1. INTRODUCTION

The future experiments in the High Energy Physics, as Compact Muon Solenoid (CMS) at LHC in CERN needs complex Data Acquisition System (DAQ) [1]. Multilevel DAQ systems structures required fast buffer for intermediate storage of data before transferring between the levels. Usually as a data buffer is used fast dual-port memory, with possibility of collect a big amount of data, corresponding to the event size. Standard bus interfaces are used for designing of modules for such DAQ system. Most useful bus standard interface becomes Peripheral Component Interconnect (PCI)[2]. 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 the PCI.

2. CMS DAQ STRUCTURE

2.1 CMS DAQ Readout Column Description

Block diagram of the CMS DAQ Readout Column is shown on Fig.1. The RU is a major part of the Readout Column and it is placed between the Front-end Devices (FED) and Builder Data Network (BDN). This unit is used as an intermediate data buffer capable of receiving event data from FED at 400MB/s and sending at the same time requesting event data to the BDN at 200MB/s. Event organisation according to the Event-ID# is required to be implemented inside the RU. The same functionality and structure has the Builder Unit (BU) major part of the Filter Column.

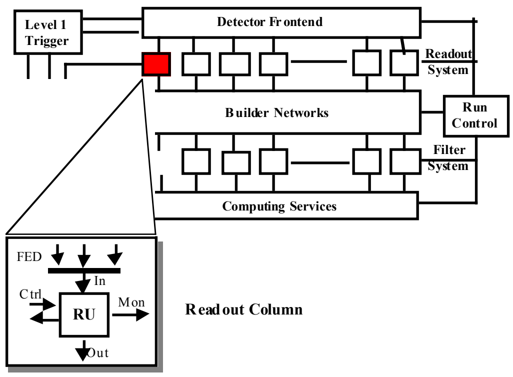

Fig.1. CMS DAQ Readout Column Block Diagram

3. READOUT UNIT

3.1 RU Functions and Requirements

RU functional diagram is shown on Fig.2. The RU has four ports and contains the following basic functional and structural components: Readout Unit Input (RUI) - input for event (up to 4KB) data size at 100KHz rate; Readout Unit Output (RUO) - output for sending data to the BDN; Readout Unit Memory (RUM) - dual-port memory up to 512MB size for storing the event data and Readout Unit Supervisor (RUS). Fast Interconnect is using between all components of the RU. Additional ports for Control and Monitoring are also available on the unit.

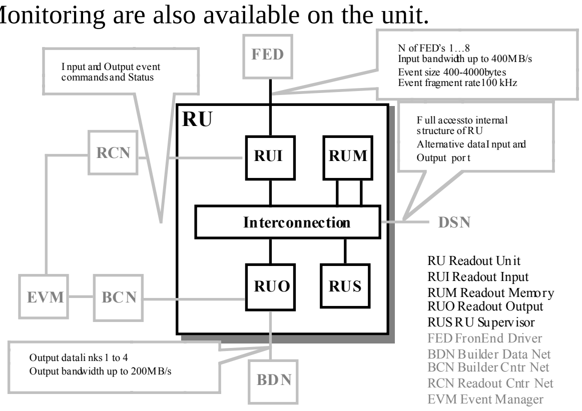

Fig.2. RU Functions and Requirements

3.2 RU Block Diagram

All functions of the RU were possible by dividing the hardware implementation on two boards called RUM and

Readout Unit Input Output (RUIO). Those are long size 64bit at 33/66MHz PCI boards connected together. Block diagram of hardware implementation of RU is shown on Fig.3.

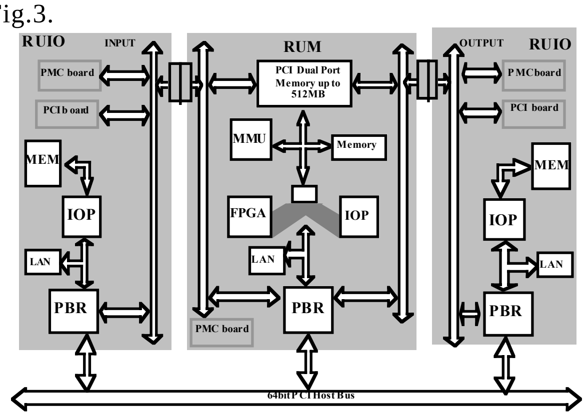

Fig.3. RU Block Diagram

In this hardware configuration from each side of the RUM is connected one RUIO board. All of them are configured via common (host) PCI bus. For internal connection between the boards we chose also 64bit PCI bus protocol running at 33/66MHz. Possible configurations for RU are using only RUM and mixed input/output port with control port, or using RUM and one only RUIO board.

3.2.1 RUM

RUM board is a general part of the RU. Readout Unit Memory is a PCI dual-port memory with third PCI bus for control. RUM receive the event header (Event-ID#, word-count, first memory block and status) and event data from the input. Event header is transferred to the Memory Management Unit (MMU) on board and is stored into the sequencer memory. Event data is stored into the data memory according to the memory block organisation. Four PCI Bridges (PBR) is using to connect the input, output, control and local bus. And fast local bus is using between MMU, Memory Controller (MC), PCI Interface Controllers and PBR. Block diagram of the RUM is shown on Fig.4. and contains the following general blocks: PBR; MMU; MC; PCI Interfaces; Memory; Local Bus controller (IOP or FPGA like unit) and PCI/PMC connectors. For all mention above general blocks we are using fast reprogrammable devices (as FLEX10K and APEX series from Altera) and for the local bus controller IOP480 from PLX Corp. Data memory is based on Synchronous DIMM modules with possibility of increasing the size up to 512MB.

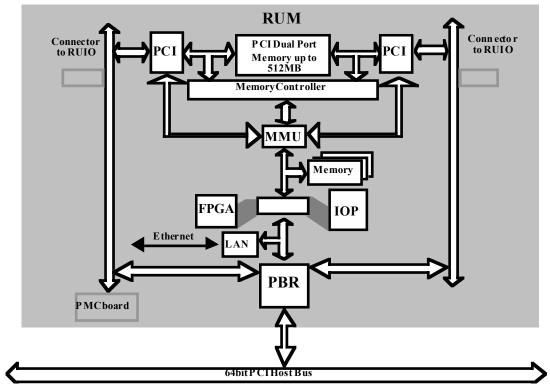

Fig.4. RUM Block Diagram

3.2.2 RUIO

RUIO board shown on Fig.5. contains the following general blocks: Three PCI Bridges (PBR); PCI to Local bus IOP480 controller; Memory – Flash, SRAM and DIMM; Ethernet Controller and PCI/PMC connectors. Basic function of the RUIO is to extend the input and output bus of the RUM and provide more flexible control of the RU. Three PBR is implemented in FLEX10K200 component. Commercial available interface links board (as Myrinet, ATM, FC etc.) can be plugged into PCI/PMC connectors to connect RU to the BDN. The first five RUIO boards are produced and tested successfully.

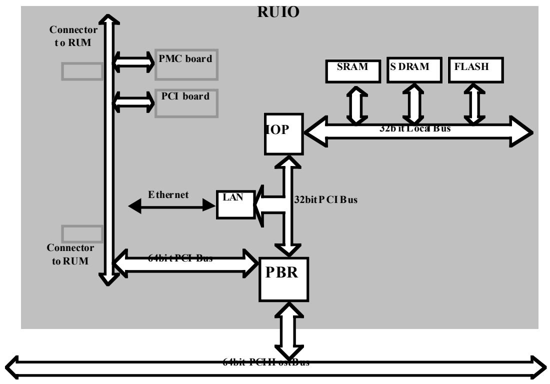

Fig.5. RUIO Block Diagram

The IOP480 processor has 32bit 33MHz PCI bus interface and 32bit Local bus running at 60MHz, integrated memory SRAM/FLASH/SDRAM controller for up to 256MB of memory, DMA controllers, serial interface RS-232 and I2O Ready Messaging Unit. For Ethernet controller was chosen 21143 PCI 10/100Base-T LAN controller from Intel. The chip supports both 100-Mb/s and 10-Mb/s data rates and is optimized for low power based systems.

4. FPGA FLEXIBILITY

In order to implement all functions of the RU and its corresponding parts we decided to use reprogrammable logic devices as FLEX, APEX etc. This is based also on our experience from previous versions of the Readout Unit (RDPM see [4]). Flexible architecture of these devices is useful to reprogram and implement different functions and structures without changing the hardware.

4.1 PCI Bridge

High bandwidth of data transfer from FED to the BDN and complex control of the RU required using commercial available fast interface protocols as 64bit 33/66MHz PCI. Transferring the data and control from one to the other bus was possible by developing the multiple PCI Bridges as general part of the control port for the RUM and RUIO. There are two versions of the PCI Bridge – three and four bridges in one component, implemented respectively in RUIO and RUM. Basic structure of the four PCI Bridges is shown on Fig.6.

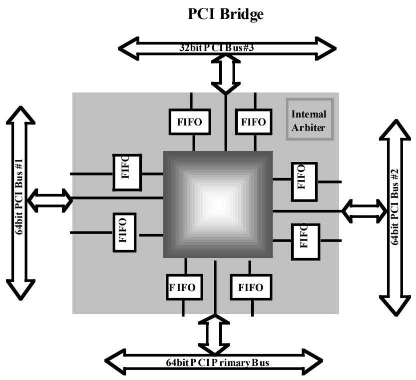

Fig.6. Four PCI Bridges

The general parts of the structure are unidirectional FIFO's for sending or receiving commands and data over the PCI busses. Also PCI arbiter for each bus is implemented inside.

4.2 Memory Management Unit

Memory Management Unit is receiving the event header from input and output PCI interfaces. Each header contains Event-ID number, word-count, first memory block address and status. MMU is the device that keeps internal event data memory structure organised in blocks using event table and pointers. MMU also contains algorithm for freeing locations inside the data memory according to the transferred event out from the RUM. For these functions MMU is using SRAM and has a direct connection also to the Memory Controller on board. Block diagram of the MMU is shown on Fig.7.

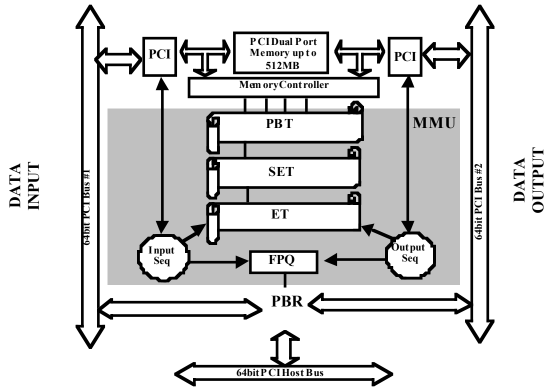

Fig.7. Memory Management Unit

4.3 Memory Controller

Memory Controller is device that control directly event data memory by generating the physical address to the memory according to the information receiving from MMU or PCI interface units. MC also contains internal arbiter for read/write access from the PCI ports, read and write address counters and control logic for the FIFO's. Block diagram of MC is shown on Fig.8.

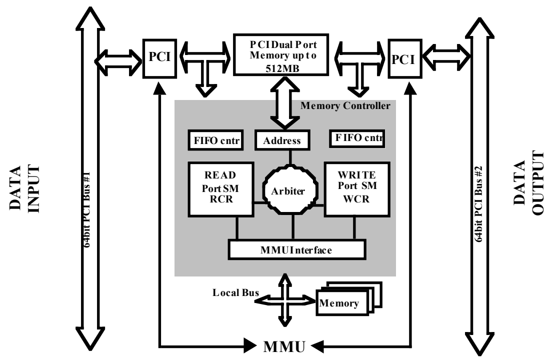

Fig.8. Memory Controller

4.4 Readout Unit Control

Readout Unit Control is done basically by device after the fourth PCI bus in the RUM and RUIO designs. There are two different schemes of implementations. First of them is using commercial available I/O processor as IOP480 from PLX Corp. with supporting around the processor components. Second is to use programmable logic devices and replace with simple protocol the control of the RUM. The first solution is already successfully implemented in RUIO prototype. Experience with IOP and I2O protocol will be accumulated due to the prototype developing. Block diagram of the Readout Unit Control implementations is shown on Fig.9.

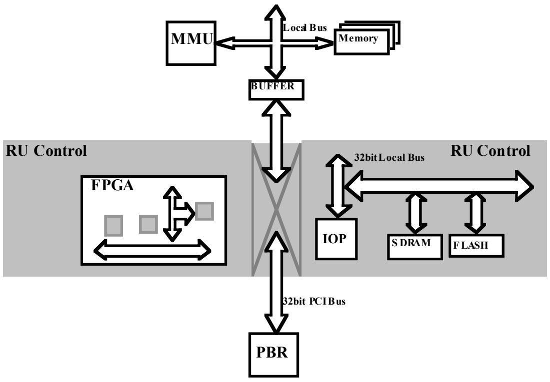

Fig.9. Readout Unit Control implementations

5. RU CONFIGURATIONS

As was written above the Readout unit is a set of two physical PCI devices RUM and RUIO connected together. Complexity of each of those devices provides the possibility of building RU by choosing one RUM and two RUIO, or one RUM and one RUIO devices. First configuration is shown on Fig.10, when the functions of RUI and RUO are implemented inside the RUM using IOP processor.

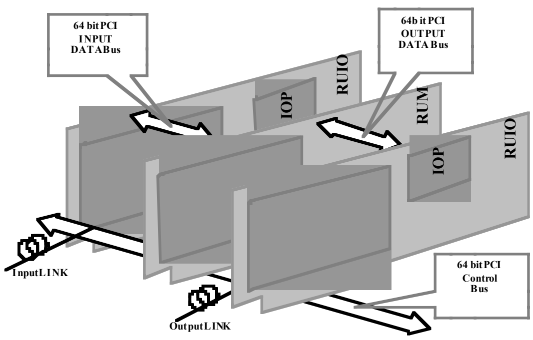

Fig.10. RUM and two RUIO

The second configuration is shown on Fig.11, where one RUM and only one RUIO are using. The functions of RUI and RUO are realised by RUIO devices. For control and data are using two different PCI busses.

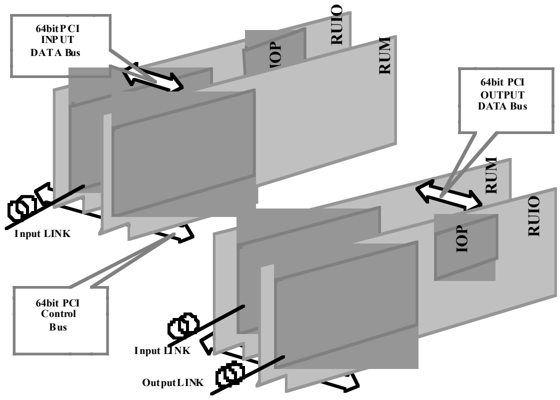

Fig.11. RUM and one RUIO

6. CONCLUTIONS

The RU prototype follow more closely the needs of CMS Data Acquisition and can be considered as a stand alone firmware DAQ that can be used in test beam, in mini data acquisition systems and as a fundamental element for testing switched systems in realistic conditions. Using FPGA components latest generation is a flexible way to implement new and improve the existing functions of the RU unit. Future available on the market 64bit at 66MHz PCI machines (PC, Workstations, etc.) and PCI data link devices are base for implementing RU in real DAQ systems. The time scale for evaluation of the RU prototypes is about one year.

7. 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. CMS FPGA dual port memory prototypes. D.Gigi Third Workshop on Electronics for LHC Experiments. London, September 22-26,1997.