# Re-configurable I/O interfaces for modern Data Acquisition Systems

*G. Antchev[1],M.A. Bellato[2],L. Berti[3],M. Gulmini[3],G. Maron[3],L. Pollet[1],R. Ponchia[3],A. Racz[1],N. Toniolo[3],G.Vedovato[3],S. Ventura[2]*

[1] CERN, Ch
[2] INFN - sez. Padova, Italy
[3] INFN – Lab. Naz. Legnaro, Italy

**Abstract**

A modern approach to high speed I/O interfacing is proposed through the use of a flexible architecture based upon Field Programmable Gate Arrays (FPGA) and the PCI bus. Three distinct solutions are reviewed, all manufactured according to PMC (PCI Mezzanine Card) standard: one solution is CPU-based and the other two are CPU-less. The reconfigurability of the architectures is emphasized, showing that, by means of modern design techniques, one can easily partition between hardware and software some CPU-demanding or high latency tasks that are typical of modern Data Acquisition Systems.

keywords I/O interfaces, FPGA, vxWorks, PMC

## 1. Introduction

In modern data acquisition systems (DAQ) for physics experiments, flexible and uniform interfaces between subsystems are a key feature to achieve system integration, maintainability and a clear upgrade policy. A block diagram of a typical DAQ is shown in fig. 1. Its constitutive elements are: detector front-end, trigger logic, readout units (RU), event builder, event filter units (EFU), computing services, and event manager. The front-end electronics acquires data corresponding to physical events and the trigger logic selects the data to send to the readout units. A readout unit consists of a dual-port memory[1] (DPM) and several Detector Dependent Units (DDU). A DDU[3] controls the associated front-end electronics and receives the trigger selected data. As a DDU is receiving data on several data links, a packet is built locally (DDU event). The DDU event is kept available for the DPM. As a DPM is reading out several DDUs, a DPM event is also built. Then, the DPM event is sent through the event builder to the requesting EFU. The association of the RU with an EFU is the task of the event manager.

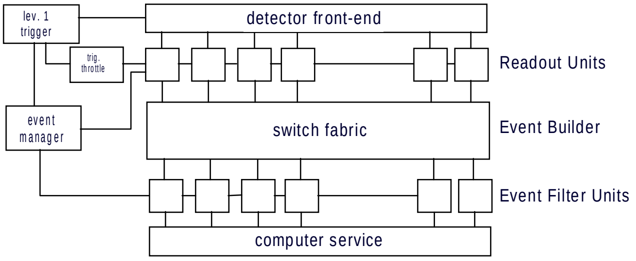

**Figure 1 - A typical modern DAQ block diagram**

## 2. Interfaces and standardization

The DAQ is the natural sink of the data produced by the sub-detectors which constitute the experimental apparatus. Interfacing these sources is a critical matter that impacts directly on the complexity of the system. Reducing the diversity in the electronic devices is a valuable approach regarding system integration, maintenance and upgrade operations. Looking at hardware interfaces between heterogeneous systems, one can find a generic architecture often encountered in DAQ applications[2] (see figure 2). As soon as off-the-shelf hardware drives (or is driven by) a specialized one, this generic architecture is used with a suitable customization. This bridging function plays a major role in system integration and may relate to common problems such as performance bottlenecks, reduced functionality or proliferation of specialized hardware. Some of the relevant aspects to be taken into account are: adoption of standards to ease integration, hardware reconfigurability for customization, speed, local processing and temporary storage.

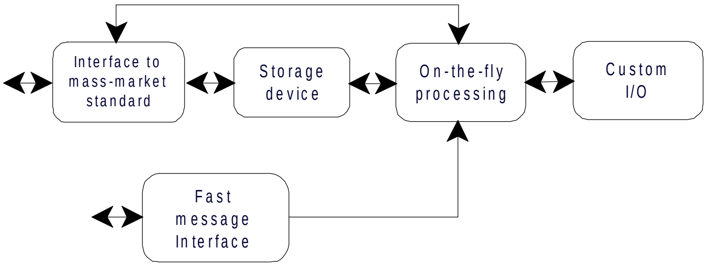

**Figure 2 – Generic interface architecture**

Three different designs tried to meet these criteria, each one with its own interpretation of the expected functionalities of each block of the generic interface architecture, as shown in fig.2.Two of them have been realized by the CMS DAQ group[3] and the third one by ATENIX[4].

## 3. Generic DAQ platforms

### 3.1. The first CMS DAQ board

The first DAQ hardware platform was designed and developed by CMS DAQ group in order to provide a prototyping facility for testing other DAQ boards and evaluate architectural options. The board features total reconfigurability thanks to the on-board FPGA (50Kgate-equivalent) 100% available to user-application. The FPGA is linked to a commercial PCI interface (initiator, target and DMA controller) and to a 77 pins user port through bi-directional TTL drivers. The drivers are fully controlled from the FPGA. The FPGA has also access to 32 KBytes of nvRAM.

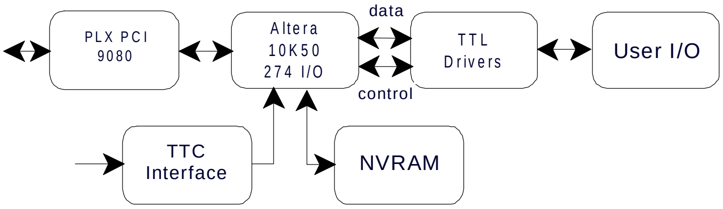

**Figure 3 - DAQ generic platform block diagram**

#### 3.1.1. Some applications

- DPM exerciser: the board reads data through the output port (the board is initiator of the DMA) and to re-inject these data into the input port. The data cycle at 128 MB/s sustained (97% of the maximum possible performances of PCI bus at 32bit/33MHz).

- DDU emulator: the board emulates a DDU. Under the control of the PCI bus , the FPGA translates on the fly a PCI data stream (DMA) to the input format of the DPM.
- FERA to PCI interface[5]: the board bridges the FERA bus to the PCI bus by means of DMA.
- Trigger interface for CMS muon chambers: the board interfaces the chamber readout system, based on PCI, with beam signals (spill, trigger). Busy and veto signals are also controlled.

### 3.2. The second CMS board

Technology improvements, the need for higher flexibility, more storage capacity and faster FPGA/PCI interface led to a new development. It features: PCI 66MHz, 64 bits capability, 32Mbytes SDRAM, 800MB/s. internal bandwidth, 300Kgate-equivalent FPGA,148 pins port for custom hardware, JTAG support for BSCAN and FPGA configuration.

The GT64120A from Galileo Tech. enables the fastest speed on PCI (524 MB/s peak) and the usage of inexpensive memory chips (100 MHz SDRAMs). The FPGA is a Xilinx Virtex 300 providing the equivalent of 300K logic gates. Due to BGA technology used for the two previous chip, IEEE-1149.1 (JTAG) had to be implemented for test/debug purposes. Bi-directional LVDS drivers wired to the FPGA have been implemented.

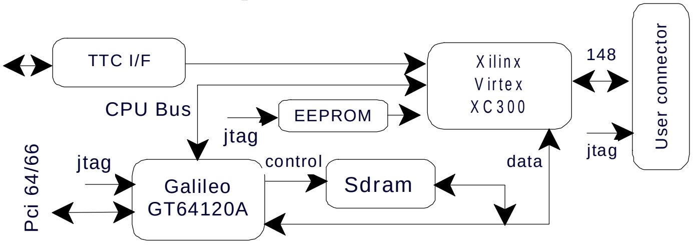

**Figure 4 – Core board block diagram**

#### 3.2.1. Some of the foreseen and potential applications

- A 200m-400MByte/sec data link from underground counting rooms up to surface buildings.
- Level 1 Accept generator/Control networks. As the DAQ setup evolves towards the final architecture, trigger generation devices and fast control networks are needed to exercise the setup under realistic conditions.
- Front-End data fan-in/fan-out. For low/high occupancy detectors, data may need to be multiplexed/de-multiplexed before/after the FEB. This is needed to use the FEB bandwidth in an efficient way or to reduce the load of the switch.

### 3.3. CPU based generic platform

A somewhat different approach has been undertaken in the third design by inserting a CPU in the generic interface architecture of fig. 2. Its main features are: CPU Motorola MPC860T, 64MB SDRAM dual ported, 32MB SDRAM dedicated to CPU use, 4MB flash memory for OS boot, 2 serial ports and a 10/100 Mb/s Ethernet port, 50Kgate-equivalent FPGA, PCI bridge 32bit/33MHZ, 80 pins I/O port for custom hardware.

The CPU based architecture expands the complexity of the tasks carried out by the “local processing” block of fig.2, at the expense of performance. To boost flexibility, a real time operating system (OS) (VxWorks) has been ported to this PMC board. It provides a convenient platform for a fully interactive remote develop-download-debug cycle through TCP/IP. The FPGA part is PCI-compliant, so that the custom connector may become a second PCI bus port.

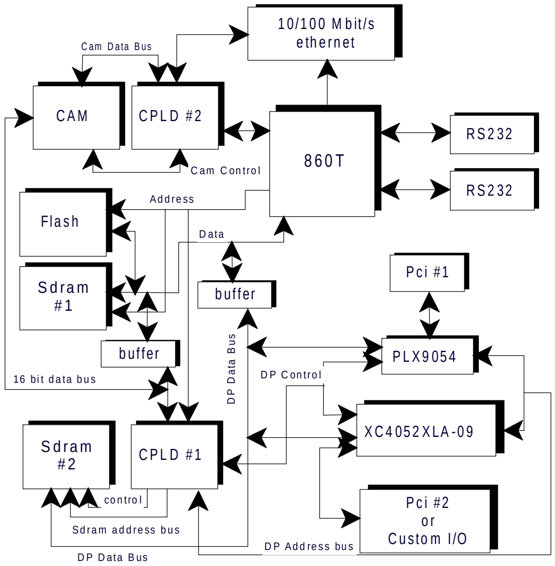

**Figure 5 – CPU based generic interface block diagram**

### 3.3.1. Applications

The range of applications of this board naturally overlaps with that of the first CMS generic DAQ board and somehow extends it. Its software approach to handle complexity and the presence of a TCP/IP connection let it become a very effective basis for value added products.

## 4. Conclusions

The key concept of a generic interface for DAQ applications has been introduced; it relates to some basic communication principles that are at the heart of an effective system integration policy. The main benefits of a standardized approach are ease of integration, debug and maintenance. Three different implementations of these requirements are reviewed, together with their applications. From the kinds of applications (developed, foreseen and possible), it comes to evidence that, this reconfigurable hardware approach, together with a careful architecture support, can be at the heart of a fair amount of solutions in the field of data acquisition systems.

## References

[1] D. Gigi, “CMS Dual Port Memory”, LEB99 Proceedings
[2] A. Racz et al., “Generic Hardware for DAQ applications”, LEB99 proceedings
[3] A. Racz, “CMS DDU Design Specifications”, CMS Note 1999/10
[4] ATX-800 User Manual, Atenix s.r.l.
[5] M. Bellato, R. Ponchia, “A Versatile FERA to PCI Interface for the GASP Experiment”, INFN Report