

DEVELOPMENT OF A VERSATILE READOUT SYSTEM FOR HIGH RATE DETECTOR ELECTRONICS

M.C. Mertens^{*},¹, F. Hügging^{*}, J. Ritman^{*}, T. Stockmanns^{*}

^{*}Institut für Kernphysik, Forschungszentrum Jülich, D-52425 Jülich,
Germany

Abstract

During the research and design phase of new detector electronics, the development of a suitable test environment takes significant amount of time. Most existing systems are specifically designed for certain frontend electronics and cannot be reused for future developments.

Thus, our approach is to build a flexible test environment with state-of-the-art hardware, which can be reconfigured to support various frontend electronics. This is achieved by deploying a modular design concept, which is followed in both hardware and software. A key feature of the hardware platform is the modern FPGA (Virtex 4) and consequent separation of analog and digital parts of the readout. This is accompanied by a modular software framework written in C++ which declares different communication layers for easy hardware access. These levels of abstraction make it easy to add support for changed or completely new devices.

The system is presented and its key features are explained in detail.

1 Introduction

The availability of a suitable readout system for debugging is mandatory during the research and design of new detectors. The setup of such a readout system is required for all new detector developments. Due to the lack of a generic, easy to use readout solution, a custom system is usually designed specific to the projected needs.

Since no suitable readout system already existed for our purpose, this task also came up during development of the *Micro Vertex Detector* (MVD) for the upcoming $\overline{\text{P}}\text{ANDA}$ experiment (See Ref. [1]).

¹E-mail address: m.mertens@fz-juelich.de

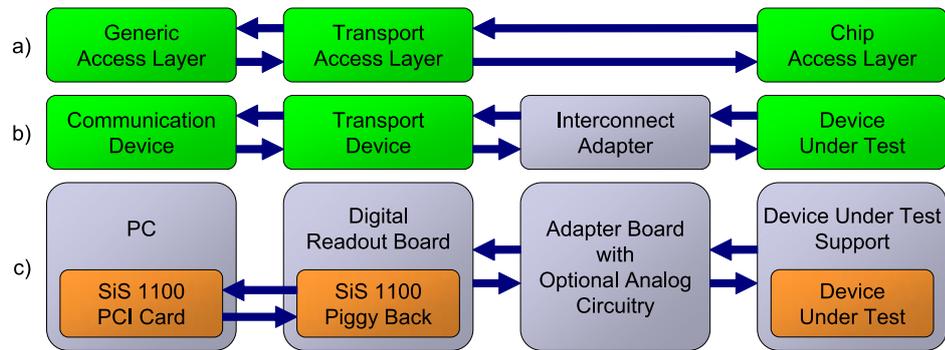


Figure 1: Readout chain. a) shows the abstract model of the software modules, b) the abstract model of the hardware devices and c) shows the actual setup.

One of our main goals is to design the readout system in a way which is generic enough to make it easily reusable, yet simple enough to be easy to use.

2 System Overview

The complete readout system features a modular design of hardware with a corresponding software framework which comprises the following parts (normal text indicates generic parts of the system, while *italics* denote application specific parts):

- Hardware
 - SiS 1100 PCI card (gbps optical link).
 - Digital readout board featuring Xilinx Virtex 4 LX60.
 - *Pinout adapter board.*
 - *Device under test support.*
- Software
 - SiS 1100 device driver (for Linux kernels 2.4 and 2.6).
 - C++ communication layer declarations and base definitions.
 - *Device specific communication protocol.*

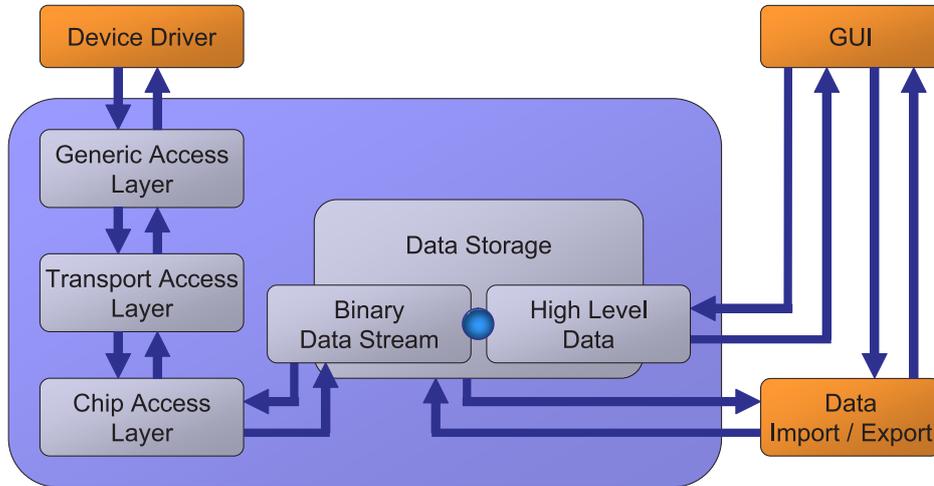


Figure 2: MVD readout framework.

The readout hardware strictly separates analog and digital parts of the readout. While the digital readout board itself contains purely digital components, any (optional) analog circuitry is implemented on an additional adapter board which connects the frontend support with the digital readout board. Its embedded FPGA can be reconfigured to support arbitrary communication protocols and I/O pinouts. The connection to the PC is realized via two SiS 1100 optical gigabit links, one is installed as PCI card in the PC, the other integrated as piggy back on the readout board. The interconnects of the actual setup are shown in Fig. 1c.

2.1 Software Environment

The software framework, in the following referred to as MRF², follows a modular design approach by implementing an abstract communication model which defines a hierarchy of communication layers, each layer corresponding to a single module in the software framework (compare Fig. 1a) respectively a single hardware device within the readout chain (Fig. 1b).

In addition to the definition of an abstract model, the MRF implements the base functionality for each communication layer where possible, so only device specific code has to be added. Another module of the MRF is a special data storage class which provides a common interface for transfer, storage

²MVD Readout Framework. This name originates from the initial motivation of building a generic readout system which is primarily intended to be used during development of the PANDA MVD.

and display of both configuration data and data returned by the device. Fig. 2 shows the interaction of individual MRF modules.

The MRF's communication model defines a hierarchy of three communication layers. Direct interaction is only allowed between adjacent layers, thus each layer uses only the functions provided by its direct lower layer for communication. This, in combination with the definition of the minimum functionality each layer has to provide, ensures that the specific implementations of all three layers are independent from each other and can be exchanged in a modular way.

The three modules implementing the communication layers are complemented by an additional module for data storage. All modules along with a functional description are summarized below.

- Generic Access Layer (GAL)
 - Lowest level communication layer.
 - Directly interacts with the communication device installed in the PC.
 - Provides register based read and write access.
- Transport Access Layer (TAL)
 - Uses GAL to communicate with the transport device.
 - Provides access to transport device features (*e.g.*, raw data transfer to the device under test (DUT), built-in diagnostic routines).
- Chip Access Layer (CAL)
 - Provides access to the connected device under test.
 - Functions in this layer implement DUT related commands (*e.g.*, configuration, initialization of data transfer).
 - Uses TAL to implement data transfer between control software and DUT.
- Data Storage Class
 - Display and configuration of device parameters as well as retrieval and display of data returned by the device.
 - Conversion between logical data entities (*e.g.*, integral values) and a device specific data stream.
 - Common interface to store data in a standard format for transmission via MRF functions.

2.2 Hardware Platform

The central component of the readout system is the *digital readout board* featuring a Xilinx XC4VLX60 Virtex 4 FPGA, DMA enabled data transfer at a rate of one gigabit per second to the PC via a SiS 1100 optical link, 32 LVDS I/O lines, four separately adjustable voltage lines and three differential clock lines. Its flexibility, resulting from the powerful FPGA, the freely configurable output lines, clock sources and supply voltages, allows to easily adapt this board for a wide variety of readout tasks.

Next part within the chain is the *pinout adapter board*. Its main purpose is to interface between the connectors of the digital readout board and those on the DUT support, but it can also be used for placing possibly needed analog circuitry (*e.g.* in order to connect to frontend chips with analog output lines). This board has to be designed specific to the actual application.

The *device under test* (typically a frontend chip) is then mounted on a corresponding support board (*DUT support*) which connects it to the pinout adapter board.

Compare Fig. 1c for an overview on the setup of the readout chain.

3 Conclusion

We have introduced a versatile readout system comprising reconfigurable hardware and a modular software framework. This approach makes it possible to deploy the readout system in various environments by reconfiguring the FPGA based hardware and adding the necessary application specific code to the generic software framework.

4 Acknowledgments

We would like to thank:

H. Kleines, P. Kämmerling, M. Ramm, P. Wüstner *et al.* (ZEL, FZ Jülich) for development and production of the digital readout board hardware and Linux kernel module driver for the SiS 1100 PCI card.

The $\bar{\text{P}}$ ANDA MVD subgroup at TU Dresden (K.-T. Brinkmann, R. Kliemt, T. Würschig, H.-G. Zaunick *et al.*) for fruitful discussions.

References

- [1] $\bar{\text{P}}$ ANDA Collab., *Technical Progress Report*, www-panda.gsi.de.