MODERN TECHNOLOGY IN DISGUISE

Timo Korhonen#, Damir Anicic, Guido Janser, Mario Jurcevic, Ernst Johansen, Babak Kalantari, Roger Kalt, Michael Laznovsky, Lionel Schebacher, Thomas Schilcher, Dirk Zimoch,
PSI, Villigen, Switzerland

Abstract

A modern embedded system for fast systems has to incorporate technologies like multicore CPUs, fast serial links and FPGAs for interfaces and local processing. Those technologies are still relatively new and integrating them in a control system infrastructure that either exists already or has to be planned for long-term maintainability is a challenge that needs to be addressed. At PSI we have, in collaboration with an industrial company (IOxOS SA) [1], built a board and infrastructure around it solving issues like scalability and modularization of systems that are based on FPGAs and the FMC standard, simplicity in taking such a board in operation and re-using parts of the source code base for FPGA. In addition the board has several state-of-the-art features that are typically found in the newer bus systems like MicroTCA, but can still easily be incorporated in our VME64x-based infrastructure. In the presentation we will describe the system architecture, its technical features and how it enables us to effectively develop our different user applications and fast front-end systems.

INTRODUCTION

Each large research facility will at some point face the question of maintainability. On one hand the facilities are expected to be state of the art in technology to be scientifically competitive, on the other hand the investment in structures that are built is usually prohibitively large to enable it to be updated in step with progress in technology. Control systems are particularly affected by this, as the progress in electronics and computers, the core technologies in control systems is particularly rapid.

The existing control systems at PSI have been built mainly around the VME bus [2] system. The VME standard was first introduced 30 years ago and, although it has evolved and withstood the test of time exceptionally well, the technology has reached its limits. On the other hand, it is still the optimal choice for many applications, and it would be very difficult to replace the entire existing infrastructure, that still does its job extremely well. We need to find a way to bring in new technology, but avoid developing a second parallel infrastructure that would cost a lot of money and manpower and also require many changes in the surrounding infrastructure.

Evolution of the control system requires flexible ways to integrate new components and methods for processing and transporting the large data volumes the modern systems produce. Implementing the front-end electronic systems is cost and labour intensive. They also have a long lifetime, which makes it practically impossible to keep up with the continuously evolving computing equipment. A better strategy is to focus on data transfer infrastructure and connectivity.

However, several devices provide data too rapidly for even the fastest media available. The modern way to handle this is to use field programmable gate arrays (FPGA) to process the data stream right at the source and compress it to a level that the infrastructure can handle.

Introducing capabilities of fast front-end processing, data transport and industry-standard interfaces in our existing infrastructure enables us to build new types of instruments and embed them in the computing infrastructure, to utilize the full possibilities of modern data processing.

To achieve this, we cooperated with the company IOxOS who proposed a board design that matched our needs. The design was refined in a series of review meetings and numerous discussions, and an agreement about the work division was made [3].

SYSTEM STRUCTURE

The board, named IFC_1210, is built around the industry standard PCI Express [4], found in practically any modern computer. A 24-port, 32-lane PCI express generation 2 switch connects the main components (CPU, FPGA, XMC, external links) together. With its capability of non-transparent bridging it also allows us to connect the board directly to a powerful remote computing server as if it were an internal peripheral, and this way build effective parallel processing solutions.

A powerful FPGA (Xilinx Virtex-6) is attached to the PCI infrastructure, to serve two major purposes. It connects to the VME bus and to two FMC (FPGA Mezzanine Card) interface slots [5]. FMC is an industry standard for input/output cards that can be directly attached to an FPGA. The FPGA is also used for fast processing of data directly at the source. 512 Mbytes of DDR3 RAM is attached to the FPGA for fast storage. This RAM is accessible from the CPU via PCI express.

A small, but very important, technical detail is the use of a special connector (UHM, Ultra-Hard Metric) as the middle (P0) connector. That allows high-speed links such as the PCI express to be routed through the existing VME backplane. This allows us to use this new technology in all our existing facilities, old and new, and take full advantage of the existing infrastructure.

#timo.korhonen@psi.ch

ISBN 978-3-95450-139-7

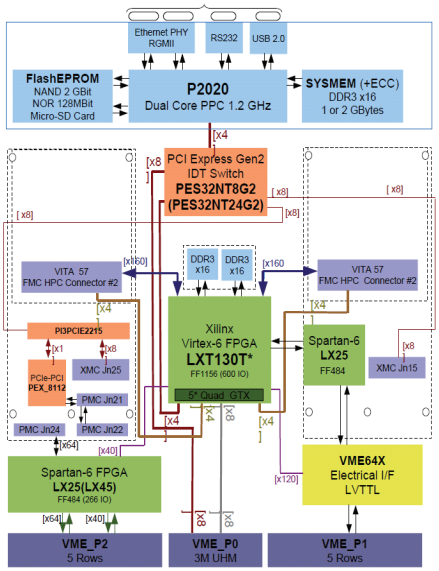

Figure 1: Block diagram of the IFC_1210 card.

To complete the board as an embedded computer a Freescale [6] dual-core QorIQ P2020 CPU is included on the board. On this CPU we can run an operating system such as Linux, and on top of it the control system software EPICS [7]. The board can thus be used both as a master controller on the VME bus in the traditional sense and as a standalone computer. A regular CPU is accessible for a wider range of programmers and the availability of many software tools make it very flexible. Much effort has been invested in software support for the board, to make it easy to embed in the control system environment.

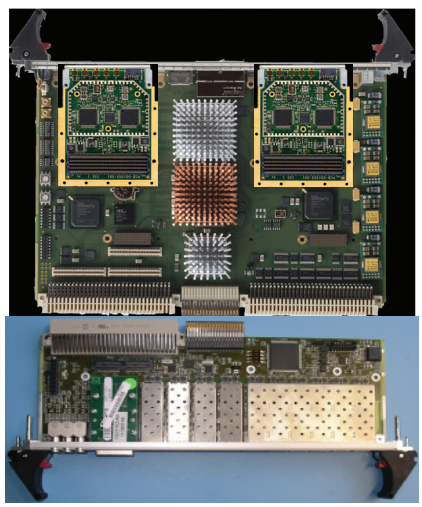

Figure 2: An example combination of the main board and a transition board providing high-speed links.

With a transition board that connects to the P0 and houses the necessary multi-gigabit transceivers for either PCI Express External Cabling standard connections or Multi-Gigabit Transceivers connecting to the FPGA, a number of fast connections can be achieved, theoretically capable of transferring 14 Gbytes/second in total. (Figure 2.)

FPGA Development

A central part of the board is the FPGA, a component found more and more in high-speed computing devices. On the IFC board the FPGA provides access to many peripherals, above all to the FMC slots and to the VME bus and enables connecting custom hardware to the board. It is also an important computing component as many algorithms can be accelerated by implementing them in hardware and taking advantage of the parallelism that the FPGA offers.

However, programming the FPGA new for each application would be prohibitively time-consuming. IOxOS provides a firmware package called TOSCA-II which gives a lot of pre-programmed functionality and interfaces to user-defined blocks. The TOSCA can be understood as kind of an “operating system” for the FPGA. It provides a PCI express “network on chip” infrastructure, fully switched to provide a better scalability than a shared bus. For user-specific firmware, standard interfaces (“User Block”) to connect to the board infrastructure are provided.

FPGA Mezzanine Card (FMC)

FMC is a standard by the VITA organization [5] that defines a connector family, mechanical and electrical properties for add-on cards that interface directly to an FPGA. The goal is to enable development of cards that can be plugged into a carrier board with an FPGA and accessed by board-specific firmware. By decoupling the base board and the application-specific front-end circuit, the baseboard can be re-used for multiple applications and also the design updated with a different front-end without having to replace the whole infrastructure.

However, each FMC board requires a board-specific firmware before it can be used. For access from software applications, the resources of the board have to be mapped to the processor's address space. The FMC standard does not define any firmware structures but leaves the firmware implementation fully open for the developer. Thus, while being compatible on the hardware level, each vendor's firmware cannot readily be used in different baseboards, thus limiting the interoperability of FMC boards.

To maximize the re usability of our firmware, we defined structure for user-defined firmware within the TOSCA infrastructure so that it is divided in three blocks: interface to the board infrastructure with a pre-defined register layout, application-specific logic and interface to the hardware on the FMC card. This way, we can use the same FMC in different applications by changing the application-specific part, or use the same application-specific logic with different FMC hardware. Although the register functions can be different for each application and FMC card combination, the address ranges are predefined

ISBN 978-3-95450-139-7

and there are a number of standardized registers containing for example the hard- and firmware signatures.

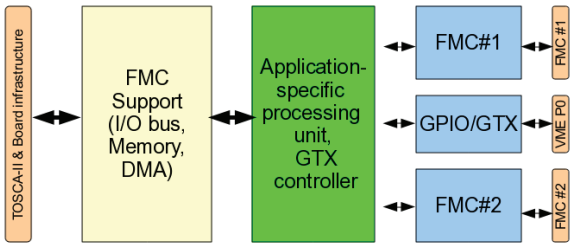

Figure 3: Modules for TOSCA-II user defined blocks.

When firmware adheres to this structure, a software driver can read from known addresses what is plugged in the FMC slots – or are they unconnected. It could even auto-configure the system with help of applications that link the correct software elements to the device signatures.

FMC Cards

For our applications in PSI we have evaluated a number of commercial off-the-shelf cards and also co-developed an ADC-FMC, ADC_3110 [1] together with IOxOS. In the evaluation process we have learned a number of things. First, good support from the FMC vendor is important as the interface is very low-level and having somebody to answer technical questions makes the board integration much easier. Important is also good documentation and availability of sample VHDL code to help getting started with the board. It takes 1-2 months of work from an experienced FPGA engineer to fully support a FMC board; without a standard framework it would be much longer.

The small form factor of the FMC was long a source of worries. However, our experience with the high-speed, high precision cards has shown that with careful design it is possible to achieve a very good performance that is close to the theoretically achievable limits, like ADC performance that approaches the specifications of the A/D converter and the form factor is not a limiting factor for the performance.

SOFTWARE

As the operating system on the board we decided to use embedded Linux with Preempt-RT extensions. It would also be possible to run a full real-time kernel like Xenomai [8] but that would involve quite a lot of work in adapting our control system software (EPICS) to take advantage of the features. On the other hand the real-time requirements for the planned applications can be met without having to use Xenomai, so that path is not pursued actively unless a definite need appears.

The company supplies us with a library that gives access to the board resources. On top of that we developed a user-configurable memory-mapped driver for EPICS that can be configured to access registers and access hardware without having to have a dedicated driver for each piece of hardware. Not all but many application requirements can be covered with such a driver.

INTEROPERABILITY

The IFC_1210 board has the VME form factor and can thus be used in a VME bus system as a master or slave card. It also implements all the functionality according to the VME64x specification, including the 2eSST transfer mode. However, the board is not limited to VME bus. Using the PCI express and MGT connections through the backplane, the board can be connected to other computing infrastructure in several ways, and also be used as a standalone computer board (Figure 4.)

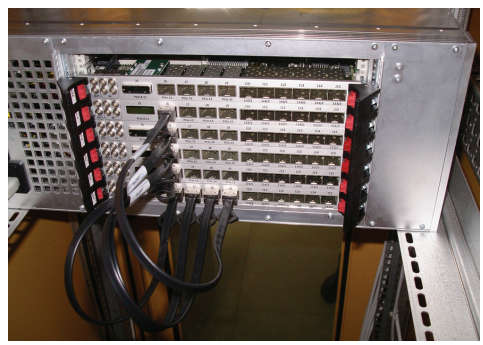

Figure 4: PCI express connections over the backplane.

APPLICATIONS

Low-level RF

One of the first planned applications and a driving factor behind all this development has been the application of this board in our low-level RF applications. These systems need a large number of fast analog I/O, primarily ADCs, fast acquisition and processing, much of which can and has to be implemented on the FPGA, plus the possibility to run feedback algorithms on the board. For that, the on-board CPU provides an easy programmer access with a fast development cycle. The code, or parts of it can if necessary also be implemented on the FPGA when the requirements and constraints are known.

This application requires a large number of ADC channels, too many to fit on one board. Thus, a large number of data has to be transferred between boards. For that VME bus would be too slow and provide no parallelism. Because of this the data is transferred between boards using several PCI express links (Figure 3). This way the data can be transferred simultaneously from several boards to one master board.

ISBN 978-3-95450-139-7

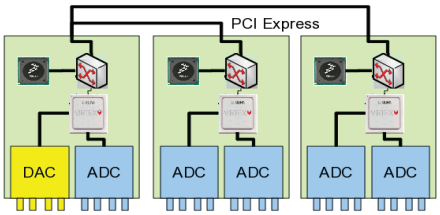

Figure 4: Simplified schematic of an LLRF system with three IFC_1210 boards.

Power supply controller

PSI has developed an architecture for digital power supply controller [9,10] that has also been adopted for use in several other facilities. These controllers communicate with a higher level control system via optical point-to-point links and a dedicated communications protocol. To interface this to the global control system (EPICS), we have implemented an application on the board that communicates with a number of power supply controllers on one side and provides EPICS channels for accessing the power supplies from the control system. The protocol to communicate with the power supplies using 8B/10B coded data has been implemented within the board FPGA, and the EPICS software (drivers, databases) run on the board CPU. The optical links to the power supplies go via the backplane using a custom-built transition card. This application can be used standalone on the board without requiring any use of the VME bus, that only provides the housing for the system. In this system a global feedback system can communicate with the system directly via the FPGA and bypass the CPU and EPICS software, thus giving a very fast direct access to the power supplies.

Digitizer Applications

A commonly requested application in a control system is to provide a way to digitize a number of signals and present them to the user in a way that an off-the-shelf oscilloscope does. Using regular oscilloscopes as embedded devices in a control system has a number of drawbacks. An oscilloscope is typically not capable of real-time operation and delivering large amounts of data. It is difficult to integrate an oscilloscope with devices like an event system, as the oscilloscope hardware is typically not meant to be extended by the user.

For this reason we are developing a generic waveform digitizer application for the board. In this case, the software should be able to cope with different kinds of FMC cards, namely with ADCs of different bandwidths and data widths, but still provide a uniform user interface. Using the firmware structure explained above, we are able to make the end application to a big degree generic, just changing the parts of the firmware and the corresponding software that talks to the particular ADC card in question.

CONCLUSIONS

Together with industry, we have designed a board that enables us to inject state-of-the art technology in our VME-based infrastructure, and even so that it can be used in any of the systems we have now. Most of them were never foreseen to be upgraded in such a way. This not only saves us the effort and money to replace the systems but enables us to use all the existing infrastructure and earlier developments where they still are appropriate.

Although attractive from the performance and flexibility point of view, FPGA-based solutions are usually very labour-intensive and the development cycle can be rather long. With a strategy to define reusable blocks and libraries that can be combined with each other we have been able to achieve a high degree of modularity and VHDL code reuse. Availability of the TOSCA-II firmware source code has also helped in the integration, even if we do not modify the base code ourselves.

With this development, the fact that this card happens to be in VME form factor does not limit its performance in any way. The essential technology (hardware, firmware and software) and our application-specific know-how is independent of the form factor and can easily be reused in other platforms that use similar technologies.

ACKNOWLEDGEMENTS

This project would not have been possible without the collaboration with the company IOxOS SA. They provided us the board design and have provided an exceptionally good support and lots of ideas to the project. We also want to thank the SwissFEL project of PSI for supporting this development.

REFERENCES

[1] http://www.ioxos.ch
[2] “American National Standard for VME64 Extensions,” ANSI/VITA 1.1-1997 (R2003), American National Standards Institute, Inc. ,1998.
[3] “The New Control Systems Backbone”, T. Korhonen, R. Kalt, B. Kalantari, D. Anicic, E. Johansen, M. Jurcevic, M. Laznovsky, L. Schebacher, S. Scherrer, D. Zimoch, PSI Scientific Highlights 2012, p. 52.
[4] “PCI Express Base Specification 2.0”, PCI-SIG, available from http://www.pcisig.com
[5] “ANSI/VITA 57.1 FPGA Mezzanine Card (FMC) Standard”, VMEbus International Trade Association, 2008 (revised 2010), ISBN 1-885731-49-3.
[6] http://www.freescale.com
[7] http://www.aps.anl.gov/epics/
[8] http://www.xenomai.org/
[9] “Digital control for highest precision accelerator power supplies,” L. Tanner, F. Jenni, PSI, Villigen, Switzerland, FPAH010, in Proc. 2001 Particle Accelerator Conference, p. 3681; www.JACoW.org
[10] “A new generation of digital power supply controllers,” M. Emmenegger, H. Jäckle, R. Künzi and S. Richner, WEPD071, 2010 International Particle Accelerator Conference, pp. 3263-5; www.JACoW.org

ISBN 978-3-95450-139-7