

FPGA-based System-on-Chip Designs for Real-Time Applications in Particle Physics

Shebli Anvar, Olivier Gachelin, Pierre Kestener, Herve Le Provost, Irakli Mandjavidze

Abstract—In this paper we describe our experience in designing real-time hardware/software systems for data acquisition and analysis applications in Particle Physics, which are based on a system-on-chip (SoC) approach. Modern FPGA devices with embedded RISC processor cores, high-speed differential LVDS links and ready-to-use multi-gigabit transceivers allow developments of compact systems with substantial number of IO channels, where required performance is obtained by a subtle separation of tasks among closely cooperating programmable hardware logic and user-friendly software environment. We report on the implementation of two such systems to illustrate these advantages of the SoC architectures. One is a flexible test bench for the off-shore read-out system of the ANTARES neutrino experiment. Another is a selective read-out processor device for the CMS electromagnetic calorimeter at LHC.

I. INTRODUCTION

With ever growing number of electronic channels and interaction rates Particle Physics experiments place heavy burden on their DAQ systems in terms of quantity and throughput of produced raw data and required processing power. Real-time needs might be extremely hard at early stages of data acquisition when few microsecond reaction times are essential for rapid data selection. Substantial amount of high speed communication links are necessary to move data from detectors to trigger electronics and high level read-out systems. Frequently, stringent space and low power consumption requirements make the development of such systems even more complicated.

In this context, the use of platform FPGAs becomes extremely attractive. They are designed to achieve higher integration levels in low-power low-cost electronic systems by embedding RISC and DSP cores, on-chip memory blocks, busses and peripheral devices, multi-gigabit transceivers, steadily increasing quantities of programmable logic cells and supporting various differential or single-ended IO standards. The FPGA-based system-on-chip (SoC) architectures with their support of synthesizable IP blocks combined with readily available software drivers and libraries allow for rapid design of hardware and its prompt adaptation to a variety of applications. Integration of application specific logic blocks

designed by users is facilitated by well-defined master and/or slave interfaces to peripheral busses of embedded processors.

The paper reports on FPGA-based SoC designs for two data acquisition applications: a flexible test bench for the off-shore read-out system of the ANTARES neutrino experiment [1] and a selective read-out processor (SRP) for the CMS electromagnetic calorimeter at LHC [2]. While the first application illustrates rapid development of a reliable functional test bench system, the second application demonstrates the feasibility to satisfy hard real-time performance requirements. Though these two applications differ substantially, the platform FPGA approach allowed us to use the same hardware – development kits with Xilinx Virtex-II Pro FPGAs – for prototyping and/or development of the corresponding final systems.

II. SOC ARCHITECTURES BASED ON PLATFORM FPGAS

A platform FPGA can be defined as a device that in addition to the field programmable logic cells integrates a predetermined collection of resources such as embedded CPUs, SRAM, versatile general purpose IO ports, high speed serial links, various standard peripherals and others. Collection of these functionalities that may be implemented as hard or soft IP cores makes the platform FPGAs extremely flexible reconfigurable SoC devices: they can be customised to a big variety of complex applications by adequately configuring and programming a needed set of available on-chip components.

Several programmable logic manufacturers offer platform FPGAs. Altera, Atmel, Xilinx and others propose devices that integrate hardware cores of ARM, MIPS and PowerPC CPUs and/or allow for instantiation of soft processor, DSP and microcontroller cores. For our developments we opted for devices from the Xilinx Virtex-II Pro FPGA family because their features and evolution path seemed most adequate for our needs when technology choices had to be done.

The Virtex-II Pro FPGAs [3] are based on the proven Virtex-II architecture, extended by embedded hardware cores of up to two PowerPC405 processors and up to twenty RocketIO multi-gigabit transceivers (MGT). The processors are able to run at 300 MHz clock speeds. The bandwidth between the instruction/data caches and adjacent block RAMs, where code software and data are stored, can be as high as 6.4 Gbit/s. Depending on their size, the devices integrate up to 8 Mbit of SRAM storage organized in 18 kbit true dual ported memory blocks of configurable depth and width.

The RocketIO MGTs deliver high speed serial channels

S. Anvar (e-mail: anvar@dapnia.cea.fr), O. Gachelin (e-mail: gachelin@hep.saclay.cea.fr), P. Kestener (e-mail: Pierre.KESTENER@cea.fr), H. Le-Provost (e-mail: Herve.LEPROVOST@cea.fr) and I. Mandjavidze (corresponding author, phone: +33-1-69-08-69-09; fax: +33-1-69-08-31-47; e-mail: Irakli.MANDJAVIDZE@cea.fr) are with the DAPNIA, CEA Saclay, 91191 Gif-sur-Yvette Cedex, France.

with up to 3 Gbit/s transmission rates. They perform data framing, CRC calculation, 8b/10b encoding/decoding and data serialization/de-serialization. The transceivers support several standard communication protocols (Ethernet, Fiber Channel, Infiniband...) and allow custom protocols to be implemented.

The processor and transceiver cores are connected to the familiar programmable elements and routing resources of the Virtex-II FPGA fabric: configurable logic blocks with combinatorial, synchronous and distributed storage elements; digital clock managers, dual-port memories, multiplier blocks and general purpose IO bancs supporting most popular single-ended and differential I/O standards.

Development environment for the Virtex-II Pro FPGAs ranges from the low level VHDL description to the high level operating system layers and lets the hardware/software functionality of produced system-on-chip design to be tailored to the specific needs of applications. A typical SoC architecture based on a Virtex-II Pro FPGA is shown on Fig. 1. In such arrangement the IBM CoreConnect [4] technology busses connect a RISC PowerPC CPU with an on-chip instruction and data memory and several peripheral devices. The Processor Local Bus (PLB) provides a high-bandwidth, low-latency path between the CPU, the memory and fast peripherals. The On-chip Peripheral Bus handles connections with peripherals and memories of different width and transaction timings and assures minimal performance impact on the activity going on PLB.

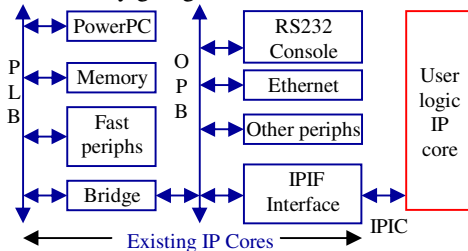


Fig. 1. A typical SoC architecture based on a Virtex-II Pro FPGA

A variety of pre-developed, verified and highly parameterisable IP cores of peripheral devices, such as UARTs, Ethernet controllers, PCI interfaces, etc, is readily available for inclusion in the design. Generic purpose interface IP cores (IPIF) with their master/slave, interrupt handling and DMA capabilities facilitate integration of user developed logic in the system. In addition, Xilinx IP InterConnect (IPIC) – a standard user logic interface of these cores – eases the adaptation of user designs to the both PLB and OPB busses.

Development tools proposed for the Virtex-II Pro FPGAs create desired configuration of the integrated on-chip platform using existing hardware and software IP cores. They also prepare software libraries and drivers corresponding to the instantiated peripherals and components. Thus the application developers concentrate their efforts on implementation of specific hardware/software functionality of user cores and on their integration in the overall design.

For fast prototyping and development of our designs, we are using one FG456 and two FF1152 development kits from Memec Inc. with respectively 2VP7, 2VP30 and 2VP50

Virtex-II Pro devices. The boards include several programmable clock sources, 32 or 64 Mbyte external memory, an interface to P160 standard expansion modules, an Ethernet interface and one or two RS232 interfaces depending on the number of embedded PowerPC processors (one in 2VP7 FPGA and two in 2VP30 and 2VP50 devices). Four and eight RocketIO MGTs are accessible on the FG456 and FF1152 boards. In addition, the latter provide several high-speed 16-bit LVDS interfaces. Both types of development kits allow operating systems (Linux, VxWorks) to be run on the embedded PowerPC processors.

To illustrate we give an example of a minimal system composed of a PowerPC processor, 32 kbyte memory, a RS232 console and a user core attached to OPB via the slave IPIF. Very simple user logic contains a reset and identification register, a data register and a 256 byte memory. The user logic responds to 32-bit r/w accesses from the CPU. Instantiated in a 2VP30 device – a middle-range Virtex-II Pro FPGA – such a configuration needs only 12% of RAM blocks and 7% of programmable logic, leaving plenty of resources for development of much more sophisticated user cores.

III. TEST BENCH FOR THE ANTARES LCM-DAQ BOARD

The ANTARES (Astronomy with a Neutrino Telescope and Abyss environmental RESearch) experiment is a large neutrino telescope designed to operate at a depth of 2500 m in the Mediterranean Sea. This detector will search for high energy cosmic neutrinos coming from extra-galactic sources and will hopefully allow for discovering dark matter signatures. It consists of a large 3D array of photomultipliers, arranged in 12 lines of about 300 m high, each of them carrying 25 storeys. To each node of this detector corresponds a Local Control Module (LCM) containing variety of electronic boards. The entire detector is linked to shore through an electro-optical cable.

A test bench has been designed to assure production quality of some 350 LCM-DAQ boards located in LCMs. The LCM-DAQ board communicates through several interfaces. The RS485 and RS232 serial links send control commands to and gather information from instrumentation boards (temperature, compass, acoustics...). High speed LVDS links interact with four data acquisition boards populated by Analogue Ring Sampler (ARS) ASICs. An LVTTTL bus is used for slow control and a 100 Mbit/s Ethernet interface is used to send collected data to shore. The LCM-DAQ board contains a FPGA for ARS data processing coupled to a on-board PowerPC processor running the VxWorks real-time OS.

The test bench architecture consists of 3 interacting systems: a LCM-DAQ board under test, a SoC-based tester board and a PC controlling all the tests (Fig. 2). The testing board, emulating the LCM-DAQ environment, is actually a Memec development kit with the 2vp30 Virtex2Pro FPGA. The versatility of the FPGA IO banks and readily available IP cores for Ethernet and RS485 / RS232 serial links greatly simplified the development of interfaces with the LCM-DAQ

board. An embedded PowerPC processor runs Linux with a NFS root file system on the Control PC. We have used slightly modified U-Boot free software [6] to boot the OS, and the BusyBox [7] project, which provides tiny versions of many common UNIX utilities into a single small executable, as a size-optimized root file system. The PowerPC and the peripheral busses are cadenced at 200 and 66 MHz.

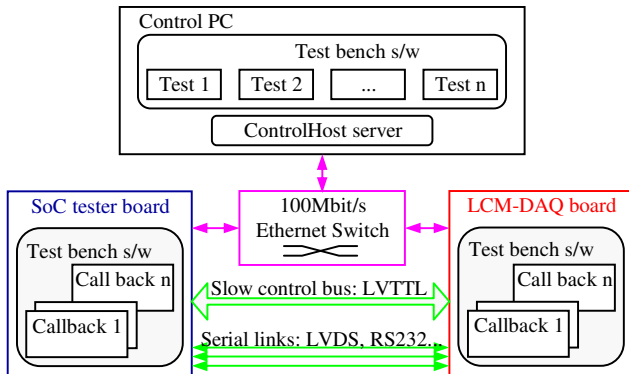


Fig. 2. Test bench architecture for the ANTARES LCM-DAQ board

The tests are divided into 3 sets: hardware tests of some components on the LCM-DAQ board (*e.g.* programming of the FPGA); emulation of ARS board (slow control communication and data flow with the ARS boards, simulation of trigger signaling); emulation of communications with others physical sensors (thermal, acoustical). For each of the LCM-DAQ functionality to test we have developed a corresponding IP core and a specialized C++ callback function.

Adopted SoC approach allowed us to trade complexity of IP cores against the complexity of corresponding software. Most of these IP cores became very simple to design because we only had to access FPGA registers in the read/write mode executing test sequences in software. The ARS data transfer tests, however, required development of high speed IP cores.

Fig. 3 schematically describes the IP core designed to simulate data flow from the tester to the LCM-DAQ board. The ARS ASICs behave like data generators using a readout signal to directly write data into a memory located in the LCM-DAQ board FPGA. Each ARS ASIC can generate 6 types of events with sizes ranging from 4 to 519 bytes. The IP core includes a 16x256 on-chip memory where predefined event data are stored. A finite state machine uses a 50 MHz readout clock from LCM-DAQ board to serialize the data and to send them at 350 Mbit/s. Three-state LVDS buffers of the FPGA emulate the data readout links in the tester board.

While in the data flow tests the task of the embedded processor is limited to data transfer initiations, in the slow control interface tests it brings the flexibility in defining the test actions. The slow control communications between the LCM-DAQ board and ARS boards are based on a protocol derived from the Motorola Serial Peripheral Interface (SPI) and imply reading/writing operations on the 239-bit parameter register of the ARS ASIC. A LCM-DAQ board can configure four ARS boards containing three ARS chips each. Data are formatted in frames transmitted from and to the ARS chips on

a bi-directional shared serial line at a slow 50 kHz clock generated by the LCM-DAQ board FPGA. Developed IP core emulates the presence of the 12 ARS ASICs and handles the SPI protocol. The embedded processor on its turn receives and interprets the slow control command and eventually responds on them according to the developed C++ software.

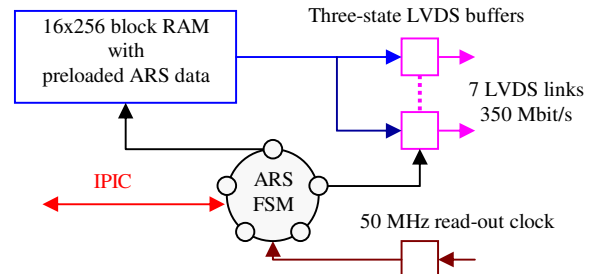


Fig. 3. ARS Data flow emulation IP

Running Linux on the embedded platform made possible to reuse the ANTARES DAQ software framework, via a simple cross-compilation step. The control PC, the LCM-DAQ board and the tester board run C++ applications implementing a control state machine from the ANTARES RunControl software. The PC sends commands to LCM-DAQ and tester boards through the ControlHost message passing interface [5] keeping the boards perfectly synchronous with the state machine. The test bench operations are launched through a C++ GTK-based GUI which sends commands through a message passing server to each of the 3 interacting machines and generates for each LCM-DAQ board a test bench report used to populate the ANTARES quality control database.

The test bench allows for hot-swapping of the LCM-DAQ boards. It takes ~15 minutes to accomplish a complete test. The production test bench is already in use since March 2005. Up to July 2005, 150 LCM-DAQ boards will be manufactured, tested and, thus, ready for integration.

IV. THE SELECTIVE READ-OUT PROCESSOR

The SRP is part of the CMS ECAL read-out electronics [8]. It assists to on-line reduction of raw ECAL data to a level acceptable by the CMS DAQ system. For each positive level 1 trigger, the SRP is guided by trigger primitive generation electronics to identify ECAL regions with energy deposit satisfying certain programmable criteria. It then directs the ECAL read-out electronics to apply predefined zero suppression levels to the crystal data, depending whether the crystals fall within these regions or not.

The SRP is housed in a single 6U VME crate. It is composed of 12 conceptually identical single-slot VME64x compliant Algorithm Boards (AB). They are organized following the four partitions of the ECAL (Fig. 4). For each level 1 trigger, the ABs receive data characterizing energy deposits in calorimeter trigger towers (TT). This information, called TT flags, is sent by trigger electronics via 108 optical links at 1.6 Gbit/s. A given AB serves a certain region of the ECAL. To assess energy deposits on the edges of their regions, adjacent ABs exchange flags of their frontier TTs. This is done

via a passive optical cross-connect providing 39 bidirectional communication channels at up to 2.4 Gbit/s each. After all necessary data are collected, the ABs scan the calorimeter in η and ϕ directions and execute sliding window algorithm to determine zones with energy deposits of a certain value and pattern. They then derive the so called selective read-out (SR) flags and deliver them to the ECAL read-out electronics via 54 optical links at 1.6 Gbit/s. For crystals that form the identified zones the SR flags instruct the read-out electronic to keep all energy samples for further levels of event selection. To achieve the necessary data reduction factor of 20, the rest of the crystals are optionally read out, if their energy is above a certain zero suppression threshold. Physics performance studies show that the adopted selective read-out algorithm does not introduce either any significant degradation in energy resolution, nor a perceptible non-linearity in energy scale [9].

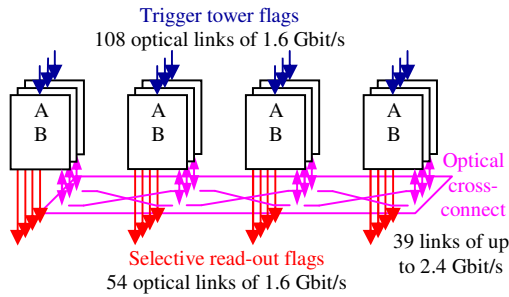


Fig. 4. The SRP made of 12 VME64x Algorithm Boards

The SRP is a hard real-time asynchronous system operating at up to 100 kHz level 1 trigger rate. For each event it has to deliver the SR flags before the corresponding front-end data arrives at the ECAL read-out electronics. This fixes its timing budget to $\sim 5\mu\text{s}$. In addition, certain flexibility is needed to change to some extent the selection algorithms. These requirements could be satisfied bringing together the latest advances of optical communication technologies and of modern FPGA devices with SoC architecture.

The ABs contain a high integration Xilinx 2VP70 Virtex2Pro FPGA. It includes two embedded PowerPC cores and 20 bi-directional RocketIO MGTs. The RocketIO serial inputs and outputs are connected to two pairs of pluggable 12-channel parallel optical transmitter and receiver modules. These optical links assure necessary connectivity with the trigger and DAQ systems and among the ABs. One of the PowerPC cores is used to configure and monitor functionality of the card and to communicate with the ECAL local DAQ and the CMS run control systems.

An application IP core for the ABs, which seamlessly integrates in the SoC architecture on Fig. 1, is under development. The IP core deploys a programmable number of communication channels consisting from MGT hard cores and associated framer modules (Fig. 5). TT flags received from trigger electronics and neighbor ABs are placed in multi-port memories and processed by pipelined selective read-out algorithm logic. Derived SR flags are then sent through the communication channels to the ECAL read-out electronics. These actions that have to be taken on event-per-event basis

are scheduled by the selective read-out state machine (SR FSM). The trigger interface module receives trigger, timing and control information from the CMS trigger control system, interprets commands and notifies the run control state machine (RC FSM) that governs AB operations at the CMS trigger/DAQ system level. Access to programmable parameters, status information and accumulated statistics, as well as contents of spy memories of each module is available through a local bus (LB). The arbiter module grants accesses to the local bus to a remote run control system via the VME bus or to the embedded processor via IPIC.

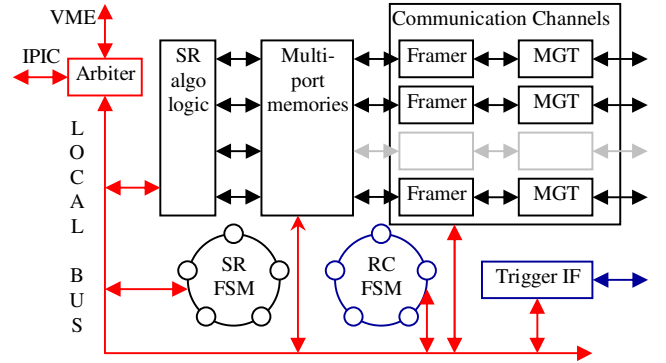


Fig. 5. Application IP core for the Algorithm Boards

By modifying the IP core firmware, an Algorithm Board can be transformed into a tester device for the ABs. The tester can send preprogrammed TT flags to an AB under test via the parallel optic transmitters. Similarly, it can receive selective read-out flags from the tested AB and verify their integrity and correctness by comparing them with preprogrammed SR flags.

Even though the AB hardware is under development, the adopted SoC approach allowed for fast prototyping of the SRP to validate its architectural principles, to study the feasibility of its implementation, to develop and test AB firmware and software. For these purposes we are using the three development kits described in Section II. In all our designs only one embedded PowerPC is operational running standalone “C” applications at 100 MHz. The applications are resident in the internal 32 kbyte memory. The PLB, OPB and consequently LB busses are clocked at 50 MHz. A RS232 console provides a simple alphanumeric menu based user interface to the applications. In absence of VME interface on the development kits, all configuration and monitoring tasks are performed by the embedded CPU. The PLB-to-OPB bridge (Fig. 1) maps addressable resources of the application cores within the address space of the active PowerPC.

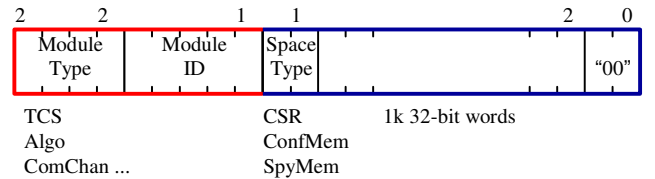


Fig. 6. Local bus addressing scheme of application IP cores

The scheme adopted for addressing the configuration and status space of application IP cores is shown on Fig. 6. Only

32-bit read/write accesses are supported by the Local Bus. Together the “Module Type” and the “Module ID” fields uniquely identify an addressed module within the application core. The rest of the address space is divided into four 1k pages with only three of them currently in use: one page – for configuration and status registers (CSR), another page – for configuration memories such as various lookup tables, yet another – for the spy memory of modules.

For test purposes, a dedicated firmware and software for the CMS trigger control system emulator has been developed and run on the 2VP7 development kit. Again, the SoC approach has been used. Controlled from a RS232 console, the TCS emulator reproduces LHC bunch structure, generates L1 accept triggers and commands that closely follow LHC run control state diagram [10]. All these signals are distributed via flat cables to an AB and an AB tester. It also receives and interprets synchronous trigger throttle signals from them. An AB tester firmware, low level software libraries and a test application run on the 2VP30 development kit, while the kit with the 2VP50 FPGA is used for AB developments.

With this setup and designed firmware and software we could operate the optical communication channels at up to 2.5 Gbit/s rates and validate a common protocol proposed for all SRP communications [8]. The SRP latency has been assessed and possibility to satisfy the stringent 5 μ s timing budget has been shown. Development of the interface with the CMS trigger control system and of the corresponding run control state machine is substantially advanced. The software is also well in progress as the low level software libraries developed for the embedded applications will be reused for integration of the SRP into the CMS Trigger/DAQ software environment.

V. CONCLUSION

Working on our respective projects we have gained some experience with platform FPGA-based SoC developments. Based on this experience we would like to make some conclusive remarks.

Flexibility of platform FPGA based designs: The SoC approach allowed us to avoid dedicated hardware developments for prototyping, to adapt the commercially available hardware for implementation of various functionalities needed for these purposes and to progress in firmware/software design for the final systems.

Simplification of developments: Available development environment allows for relatively short learning phase followed with a rapid progress in implementation of a target design. The developments are facilitated by availability of a large variety of synthesizable IP cores with supporting software libraries and drivers. Basically, the SoC development tools provide the system designer with a hardware/software kernel including embedded CPU, an associated bus, memory and various peripherals such as RS232 console and a network interface. The designer mostly concentrates efforts on implementation of proprietary modules with needed functionality. Well defined master/slave interface cores let for

straightforward integration of the user designed modules within the system. The SoC approach may simplify development for real-time applications even further, implementing in hardware only restricted set of functionalities that require high rate operation and cannot be coped in software. Design flexibility is usually improved by delegating to the embedded software the rest of the functionalities with relatively slow response times. In addition the software engineering may be carried out within the comfortable environment of embedded popular operating systems.

Facility of debugging and testing: Debugging and testing of a complex design is a tedious task. It is even more complicated when simulation of entire system is not possible as it may last impractically long. This is the case, for example, when a large number of RocketIO MGTs has to be instantiated in the design such as the CMS SRP. Modeling of a single RocketIO MGT is rapid enough so that debugging of the MGT-based communication channel module and validation of proposed protocols is fairly simple. Detailed simulation of entire AB firmware with up to 17 communication channels is, however, out of question, especially if post place-and-route model has to be used. The SoC design greatly facilitated the testing of our application cores. Individually simulated, debugged and tested modules were directly integrated in the SoC design and functionality of the entire system was checked by test applications running on the embedded processor. Such testing procedure is even more practical as the development environment provides embedded software debugger tools that connect to the processor via the JTAG interface of the FPGA.

We foresee further evolution of our SoC developments. The work performed for the ANTARES test bench lays a good ground for design of a new version of the LCM-DAQ boards where the ARS FPGA and the on-board PowerPC will be integrated within a single platform FPGA. In the final version of the CMS SRP, to be commissioned in early 2006, we do not exclude to run embedded software applications under Linux.

REFERENCES

- [1] The ANTARES Collaboration, “Technical Design Report of the ANTARES 0.1 km² project”, V1.0, July 2, 2001.
- [2] The CMS Collaboration, “The Electromagnetic Calorimeter Project – Technical Design Report,” CERN/LHCC 97-33, 15 December 1997.
- [3] Xilinx Inc., “Virtex-II Pro Platform FPGAs: Complete Data Sheet”, April 2004, <http://direct.xilinx.com/bvdocs/publications/ds083.pdf>.
- [4] IBM, « The CoreConnect Bus Architecture », White Paper, September 1999, <http://www.chips.ibm.com/products/coreconnect>.
- [5] R. Gurin, A. Maslennikov, “ControlHost - Distributed Data Handling Package”, http://www.nikhef.nl/~ruud/HTML/choo_manual.html.
- [6] W. Denk, “U-Boot - Open Source Firmware for Embedded PowerPC, MIPS, ARM and x86 Systems”, <http://sourceforge.net/projects/u-boot>.
- [7] BusyBox Documentation, <http://www.busybox.net/about.html>.
- [8] N. Almeida et al., “The Selective Read-Out Processor for the CMS Electromagnetic Calorimeter”, *In Proc. IEEE Nucl. Sci. Sym.*, Rome, Italy, October 2004.
- [9] P. Paganini, “Data Volume Reduction Strategies in the CMS Electromagnetic Calorimeter”, *In Proc. 10th Int. Conf. Calorimetry In HEP*, Pasadena, USA, March 2002.
- [10] J. Varela, “CMS L1 Trigger Control System”, CMS Note 2002-033, 13 September 2002, http://cmsdoc.cern.ch/documents/02/note02_033.pdf.