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

Shebli Anvar, Olivier Gachelin, Pierre Kestener, Herve Le Provost, and 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 field-programmable gate array (FPGA) devices with embedded reduced instruction set computing (RISC) processor cores, high-speed low voltage differential signaling (LVDS) links and ready-to-use multigigabit transceivers allow development of compact systems with a substantial number of input-output (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 the 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.

Index Terms—Embedded processor, platform FPGA, real-time system, system-on-chip design, trigger and data acquisition system.

I. INTRODUCTION

With an ever growing number of electronic channels and interaction rates, particle physics experiments place a 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 amounts 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 field-programmable gate arrays (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, multigigabit transceivers (MGTs), steadily increasing quantities of programmable logic cells and supporting various differential or single-ended input-output (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 of meeting 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. Implemented as hard or soft IP cores, together these functionalities make 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 made.

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 RocketI/O multigigabit transceivers (MGT). The processors are able to run at 300-MHz clock speeds. The bandwidth between the instruction/data caches and adjacent RAM blocks, 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 with up to 3-Gbit/s transmission rates [5]. 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, etc.) 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 banks supporting most of the popular single-ended and differential I/O standards.

The 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 an arrangement, the IBM CoreConnect [6] 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 (OPB) handles connections with peripherals and memories of different width and transaction timings and assures minimal performance impact on the PLB activity.

A variety of pre-developed, verified and highly parameterizable IP cores of peripheral devices, such as UARTs, Ethernet controllers, PCI interfaces, etc., is readily available for inclusion in the design. General 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—makes the adaptation of user designs to both PLB and OPB busses easy.

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

For fast prototyping and development, we use 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 of 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 the 2VP7 FPGA and two in the 2VP30 and 2VP50 devices). Four and eight RocketIO MGTs are accessible, respectively, 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.

For illustration purposes, we give the example of a minimal system composed of a PowerPC processor, 32-kbyte program memory, an RS232 console and a user core attached to the OPB via the slave IPIF. A 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 read/write (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 design 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 RESsearch) 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 as means for the study of the violent exotic phenomena of the Universe. It consists of a large three-dimensional 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 a variety of electronic boards. The entire detector is linked to shore through a submarine electro-optical cable [7].

A test bench has been designed to ensure 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 LVTTL 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 an FPGA for ARS data processing coupled to an on-board PowerPC processor running the VxWorks real-time OS.

The test bench architecture consists of three interacting systems: the LCM-DAQ board under test, the SoC-based tester board and a Control PC that defines and guides the test execution sequence (Fig. 2). The testing board, emulating the LCM-DAQ environment, is actually a Memec development kit with the 2VP30 Virtex-II Pro FPGA. The versatility of the
FPGA IO banks and readily available IP cores, especially for Ethernet and RS485 / RS232 serial links, greatly simplified the development of interfaces with the LCM-DAQ board.

A PowerPC processor, embedded in the 2VP30 FPGA on the tester board, is clocked at 200 MHz and runs the Linux operating system with an NFS root file system on the Control PC. At power-up, the processor executes free boot-loader software U-Boot [8]. The boot-loader issues a DHCP request to identify the NFS server and to acquire an IP address and the filename of the Linux kernel image. The latter is then downloaded through TFTP and executed. The root file system, residing on the disk of the Control PC, is based on the BusyBox project [9]. It provides tiny versions of many common UNIX utilities in a single size-optimized executable (e.g., a lightweight shell ash, a telnet client/server service, etc). Both the Linux kernel image and the root file system are generated using a gcc-based x86 to PowerPC cross-compiler [10]. The Xilinx development environment provides OS interface layers specific to the target platform peripherals, which are added to the Linux kernel sources prior to the cross-compilation step. We have found that having a downloadable Linux kernel image and an NFS root file system was much handier for debugging than the solution where both the kernel image and the root file system reside in a flash memory of the target platform. In our particular configuration, running the Linux OS on the embedded processor requires not more than 5 Mbytes of on-board memory and 30 Mbytes of the NFS disk resources.

The tests are divided into three 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 other physical sensors (thermal, acoustic). For each of the LCM-DAQ functionality under test we have developed a pair of corresponding IP cores and specialized C++ class implementing the desired test functionality. The obtained software/firmware architecture is very modular and allows smooth incremental addition of new test functionalities. The organization of the tester firmware is shown in Fig. 3. The OPB with attached peripherals and user logic IPIF interfaces is cadenced at 66 MHz.

The adopted SoC approach has allowed us to trade the complexity of IP cores against the complexity of the corresponding software. Most of these IP cores became very simple to design because we only had to access FPGA registers in read/write mode executing test sequences in software. In addition, in such simple cases, the development of dedicated Linux kernel-space drivers was not needed as the hardware registers were mapped directly in the user space. A good example would be the tests of the LCM-DAQ RS232 and RS485 links. For these tests, the complexity of firmware development was considerably reduced by the use of a standard UART IP core and the software emulation of the MODBUS communication protocol [11]. Similarly, the IP developed for the trigger interface test is basically a hardware register mapped in the memory space of the test application, which allows us to drive the trigger signals of the LCM-DAQ board following an entirely software programmable sequence.

The ARS data transfer tests, however, required development of high speed IP cores. Fig. 3 schematically describes the IP core designed to simulate the 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 six types of events with sizes ranging from 4 to 519 bytes. The IP core includes a 16 × 256 on-chip memory where predefined event data are stored. A finite state machine uses a 50-MHz readout clock from the LCM-DAQ board to serialize the data and to send them at 350 Mbit/s. The three-state LVDS buffers of the FPGA drive the data readout links. Clearly, in the data flow tests the burden of the high speed operation is entirely laid on the firmware, while the task of the embedded processor is limited only to the data transfer initiations.

Running Linux on the embedded platform made it possible to reuse the ANTARES DAQ software framework, via a simple cross-compilation step. The control PC, the LCM-DAQ board and the test board run C++ applications implementing a control state machine from the ANTARES RunControl software shown in Fig. 4.

For both the LCM-DAQ and SoC-based tester boards, we have developed a set of test-specific C++ classes with member functions implementing the actions performed during the state machine transitions. The PC sends commands to the LCM-DAQ and tester boards through the ControlHost publish/subscribe message passing interface [12] keeping the boards perfectly synchronous with the state machine. The test bench operations are launched through a C++ GTK-based GUI which sends
Fig. 4. The ANTARES run control finite state machine as implemented in the test bench software.

Fig. 5. Partial view of the test bench crate with the LCM-DAQ board under test (bottom) and the SoC-based tester board (top).

Fig. 6. SRP made of 12 VME64x Algorithm Boards.

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. Such a complex distributed scheme could not have been developed as easily without the software programming capabilities provided by the SoC approach.

A partial view of the test bench crate is presented in Fig. 5 showing at the bottom an LCM-DAQ board with its backplane and the SoC-based tester board on the top. The test bench allows hot-swapping of LCM-DAQ boards. It takes \(15\) minutes to accomplish a complete test. The production test bench has already been in use since March 2005. Up to July 2005, 150 LCM-DAQ boards will be manufactured and tested.

**IV. SELECTIVE READ-OUT PROCESSOR**

The SRP is part of the CMS ECAL read-out electronics [13]. It contributes to the 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 the energy deposits satisfying certain programmable criteria. It then directs the ECAL read-out electronics to apply predefined zero suppression levels to the crystal data, depending on whether the crystals fall within these regions or not.

The SRP is housed in a single 6U VME crate. It is composed of 12 identical single-slot VME64x compliant algorithm boards (ABs). They are organized following the four partitions of the ECAL (Fig. 6). 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 the 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 the 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 \(\gamma\) and \(\phi\) directions and execute a 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 electronics to keep all energy samples for further levels of event selection. To achieve the necessary data reduction factor of \(\sim 20\), 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 any significant degradation in energy resolution, nor any perceptible energy scale non-linearity [14].

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 \(\sim 5\) \(\mu\)s [13]. In addition, some flexibility is needed to allow for limited changes in 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 Virtex-II Pro FPGA which includes two embedded PowerPC cores and 20 bidirectional RocketIO MGT. The RocketIO serial inputs and outputs are connected to two pairs of pluggable 12-channel parallel optical transmitter and receiver modules [15]. These optical links provide the necessary connectivity with the trigger and DAQ systems and between the ABs. One of the PowerPC cores is used to configure and monitor functionalities 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 fits into 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. 7). TT flags received from trigger electronics and neighboring ABs are placed in multiport memories and processed by a 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 an 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 (TCS), 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 LB to a remote run control system via the VME bus or to the embedded processor via IPIC.

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 used 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. An RS232 console provides a simple alphanumeric menu based user interface to the applications. In absence of the VME interface on the development kits, 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.

The scheme adopted for addressing the configuration and status space of application IP cores is shown on Fig. 8. Only 32-bit read/write accesses are supported by the LB. 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, and yet another for the spy memory of modules.

For test purposes, a dedicated firmware and software package for the CMS TCS emulator has been developed. It runs on the 2VP7 development kit. Again, the SoC approach has been used. Controlled from an RS232 console, the TCS emulator reproduces the LHC bunch structure, generates level-1 accept triggers and commands that closely follow the LHC run control state diagram [16]. All these signals are distributed via flat cables to the AB and the AB tester (Fig. 9). The TCS emulator 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 [13]. The SRP latency has been assessed showing that it is feasible to satisfy the stringent 5 μs timing budget imposed to the SRP by the CMS ECAL read-out system. The development of the interface with the CMS TCS 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 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: The readily available development environment allows for relatively short learning phase followed by a rapid progress in the implementation of a target design. The developments are facilitated by the 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 the implementation of proprietary modules with the needed functionality. Well defined master/slave interface cores make way for a straightforward integration of the user designed modules within the system. The SoC approach may simplify developments for real-time applications even further, implementing in hardware only a 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 popular operating systems.

Facility of Debugging and Testing: Debugging and testing a complex design is a tedious job. It is even more complicated when the simulation of an entire system remains impractical due to prohibitive computing times. This is the case, for example, when a large number of RocketIO MGTs has to be instantiated in a design such as the CMS SRP. Modeling a single RocketIO MGT is rapid enough so that the debugging of the MGT-based communication channel module and the validation of proposed protocols is fairly simple. A detailed simulation of the entire AB firmware with up to 17 communication channels is, however, out of question, especially if the 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 the functionality of the entire system was checked by test applications running on the embedded processor. Such a 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 FPGAs.

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

REFERENCES