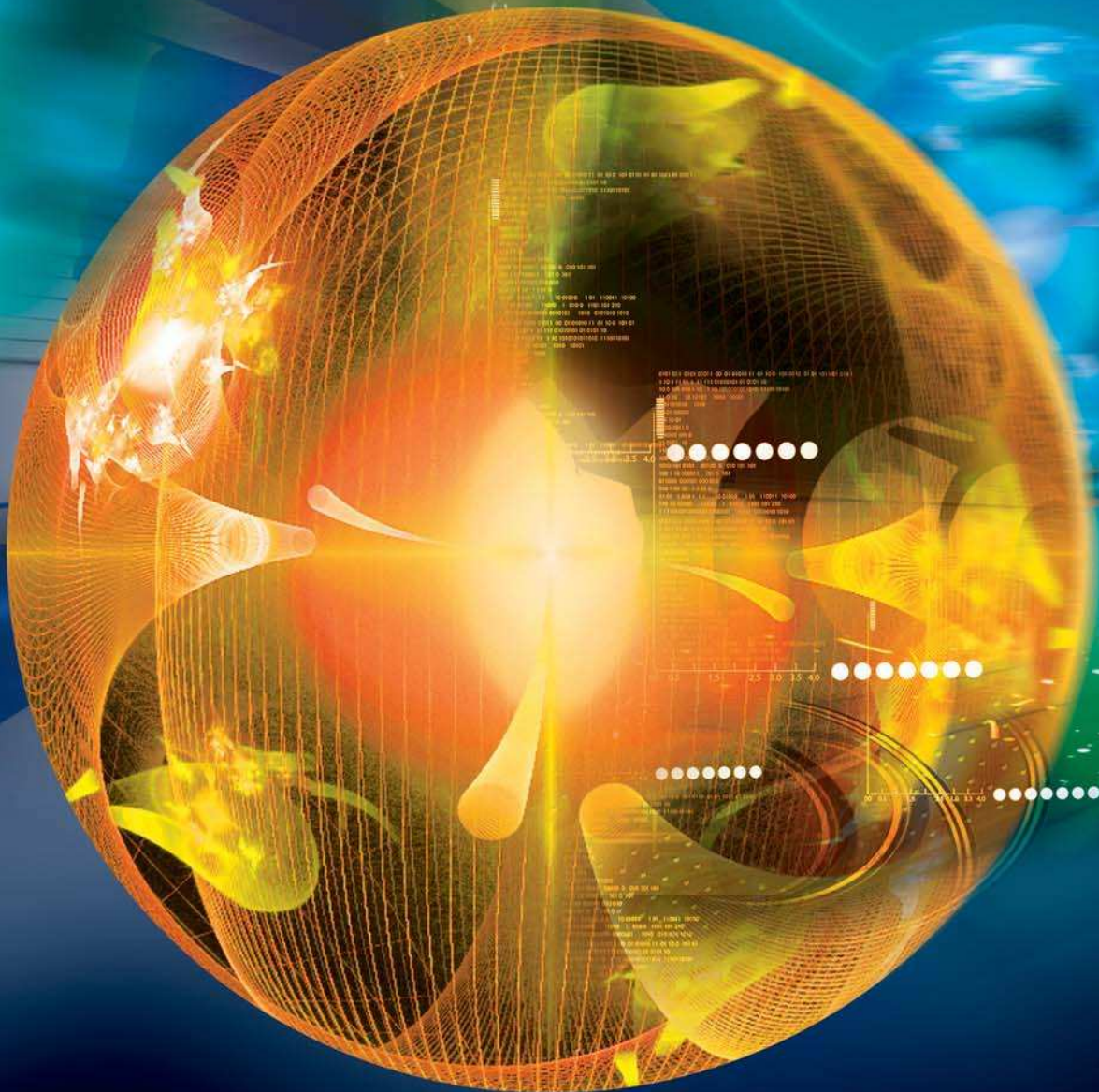# FPGAs Help Measure Trajectory of Particles in CERN's Proton Synchrotron

System processes 15 billion analog samples per second to track the path of particles traveling close to the speed of light.

by Terry Barnaby
System Engineer
Beam Ltd.
terry@beam.ltd.uk

A particle has a hard life at CERN. A proton starts its journey as a hydrogen atom in a small, common-looking red bottle the size of a soda siphon at the head of an imposing 80-meter linear accelerator. With its electron stripped, electric fields accelerate it to about one third of the speed of light. It soon enters the 200-meter-diameter Proton Synchrotron (PS) machine, where further acceleration and conditioning take place, with other protons, as a beam. After that, the particle is sent to a bigger machine such as the Large Hadron Collider or directly used in an experiment. Normally its journey will end in a near-speed-of-light collision with some other unsuspecting particle. At every stage along the way, various sensors and high-speed data-processing engines are spying on the proton, not least in the PS machine.

A versatile juggler of particles, the Proton Synchrotron (Figure 1) is a key component in the accelerator complex at CERN, the European Organization for Nuclear Physics in Geneva, Switzerland. The PS machine accelerates and manipulates protons or heavy ions for various experiments. Apart from system downtimes, the PS runs continuously, operating on a different particle beam, for the various ongoing experiments, every 1.2 seconds or so. During operation it all looks quiet around the PS ring's complex. All that an observer can notice of the particle motorway traffic within the evacuated tubes is a loud and heavy surging buzz from the multiple large power transformers outside every 1.2 seconds as megawatts of power surge to supply the system.

One of the many important instruments required for this machine is the Trajectory Measurement System. The purpose of the TMS is to measure the track of the particles as they race around the ring at nearly the speed of light. There are 40 sets of metal plate detectors spaced around the PS ring, providing 120 analog signal channels. Each of these signals, three per detector pickup,

needs to be digitized at 125 megasamples per second. That is an overall data rate of 15 billion samples, or 30 Gbytes of data, per second. Due to this high data rate, there is a need to process the data in real time, to reduce the amount of data and pick out the information of interest before storing it.

CERN's scientists and engineers had devised the basic processing algorithms suitable for FPGA implementation. For engineers at Alpha Data and at Beam Ltd., our job was to develop the idea and pro-

Figure 1 – CERN scientists accelerate and condition particles in the ring-shaped Proton Synchrotron.

duce a complete working system in a reasonable time frame and at reasonable cost to do the job.

Designing the TMS
From the early stages of design, we envisaged a very modular system that would provide fault tolerance and ensure easy live maintenance. It also had to be flexible and expandable. We took the concept of modularity down even into the FPGA fabric, where each FPGA implements three separate pickup-detector channel blocks.

Alpha Data and Beam have worked jointly on a number of real-time, FPGA-based instrumentation systems. These

generally employ a generic Linux-based computer as the main control and data-access module, with a number of FPGA-based modules at the front end to do the fast data capture and real-time processing work. The TMS is based on that system model. It comprises a Linux-based system controller that provides control, data postprocessing and external data access, and four, eight-slot, CompactPCI rack modules, each with a Concurrent Technologies Intel dual-core processing

card running Linux. Within the rack modules are specially designed, FPGA-based, Pickup Processing Engine, or PUPE, cards (Figure 2).

When developing such real-time systems, especially with FPGAs, it is important to decide what work the various system modules will handle. FPGA hardware is excellent for doing simple, real-time, repetitive things in parallel and at relatively high speed. Software running on a conventional processor is good for control and data access as well as flexible data postprocessing. Getting the partitioning of this work right, together with the protocols used among the system's various hardware
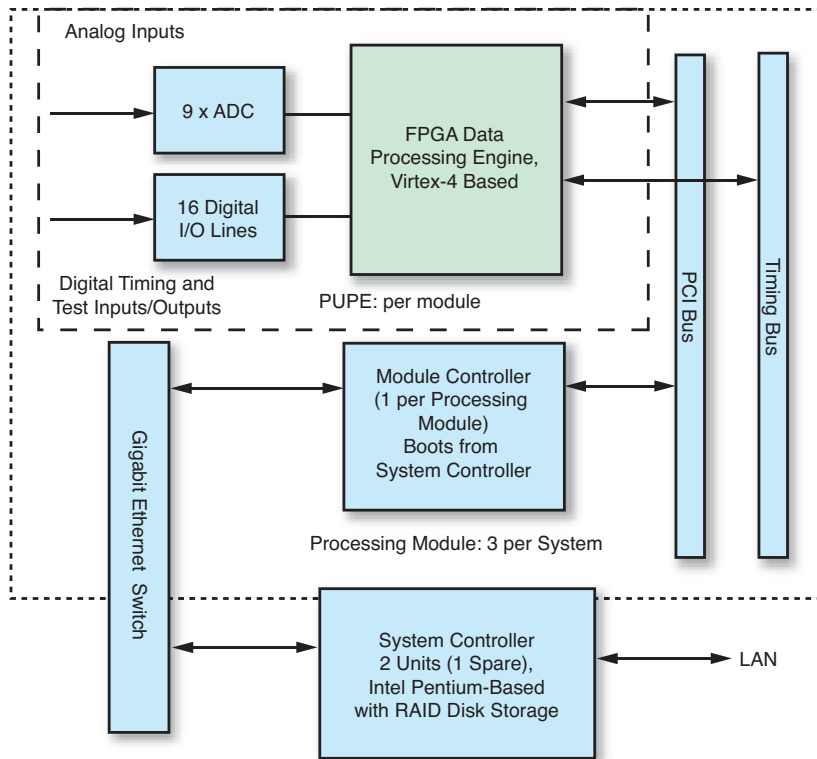
Figure 2 – The Trajectory Measurement System

and software modules, is essential in projects of this kind.

Communication between the TMS modules uses Gigabit Ethernet for both external links and internal communications between the system controller and module controllers. The Beam Object Access Protocol (BOAP) is a simple and efficient protocol for this work. The BOAP system employs quality-of-service protocols to tighten message delivery times. We implemented communications with the PUPE FPGA cards as a register and shared memory interface over a CompactPCI bus.

Along with the 120 analog channels, the CERN PS machine produces a number of digital system-synchronization signals. The main one is a systemwide 10-MHz clock that is used to synchronize the complete system. A digital timing bus links these signals to each of the PUPE boards.

PUPE FPGA Board Design

To reduce project costs and development time, we decided to use commercial off-

the-shelf components wherever possible. For the Pickup Processing Engine, we initially considered using off-the-shelf FPGA modules and A/D converter modules that Alpha Data produces. However, in this case, we decided instead to design and produce a custom CompactPCI board based on an existing PMC design. The size and performance of FPGAs had reached the

stage where one could easily support nine ADC channels with the appropriate data-processing algorithms. Sharing the FPGA and associated communications hardware among nine ADC channels allowed us to reduce the system cost considerably, trim the system's physical size, more tightly couple the ADC clocking system and support extra features. Alpha Data's experience in producing Xilinx FPGA boards allowed us to produce the new PUPE board within seven months, from conception to first working boards.

These PUPE CompactPCI cards (designated ACP-FX-N2/125) are the heart of the Trajectory Measurement System (Figure 3). They utilize a Xilinx® Virtex®-4 FX100 FPGA, 1 Gbyte of DDR2 SDRAM and nine 125-MHz, 14-bit ADCs. The board's core design is based on Alpha Data's ADM-XRC/FX100-10/1G FPGA PMC module, providing a high degree of FPGA firmware compatibility with this hardware. The design employs a low-jitter, phase-locked loop (PLL) synchronized clock source for the ADCs. One of the design issues was to distribute this clock to all nine ADCs without increasing the clock jitter appreciably. The board's PCB layout was also crucial to achieving higher performance from the A/D converters as well as low on-board noise from the board's components.

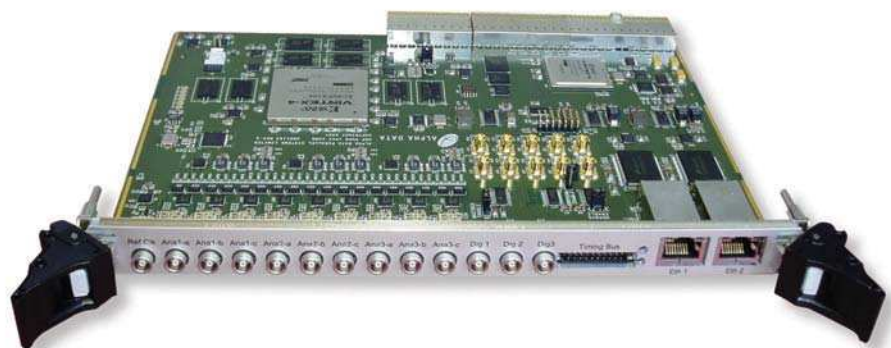The board employs a second Xilinx Virtex-4 LX25 device for CompactPCI interface duties. This uses the PCI bus'

Figure 3 – Alpha Data's Pickup Processing Engine (PUPE) board
is based on Xilinx FPGAs.

The core module is the particle beam synchronized phase-locked loop. The FPGA allowed us to implement this real-time structure together with the DSP processing algorithms side-by-side in the same chip rather than in dedicated hardware.

FPGA firmware as developed by Alpha Data for its existing PMC boards. The PUPE also has two Gigabit Ethernet PHYs with the associated RJ45 connectors on the front panel, connected directly to the FPGA. The large number of I/O pins available on the FX100 allowed us to connect all of the ADCs and all other on-board components directly to the FPGA with minimal glue logic. We use quick-switch devices for interfacing to external 5-volt digital signals.

CERN is a primarily a research institution. So throughout the TMS design process, we tried to add features that could be useful in the future as needs changed. The FX100, for example, could be swapped for an FX140 if a CERN project required more hardware resources. Since five PUPE boards share the CompactPCI bus, there is a communications limit of about 100 Mbytes/s (20 Mbytes/s per board). The Gigabit Ethernet channels can provide up to 500 Mbytes/s, per five PUPE modules, for future needs. The PUPE is thus capable of employing the internal PowerPC processors with the Gigabit Ethernet interfaces to run a local Linux control process if required.

Design and development of the PUPE progressed relatively smoothly, with only a few minor issues to sort out before the first production board run. The on-board JTAG chain and ChipScope™ interface helped enormously with initial board debugging, allowing us to configure the FPGA fabric for test purposes (Figure 4).

PUPE Firmware

The PUPE FPGA firmware is written in VHDL. We used the ISE® 9.2 tool set for development purposes. It was helpful using a readily available development package; CERN was using this same tool set, so we

could exchange FPGA firmware designs relatively easily without the necessary changes for different vendors' tools. We also used Alpha Data's standard SDK to simplify development and deployment.

The design makes use of a number of key core modules that we had developed for other projects. The use of these standard, proven cores helped us to produce the FPGA firmware within the project time

scale with relative ease. The most difficult and complex block is the SDRAM access block. This has a time-shared, dual-port access scheme so that the host computer can read the acquired data while it is being captured in real time. This block caused us

the most issues during development, mainly due to the tight timing requirements, and thus increased code compilation times. Thankfully, our modular approach helped out here, allowing us to develop and test the firmware using just one of the three pickup channels. This reduced compilation times significantly, with the lower gate count used, and thus improved the rate of firmware development.
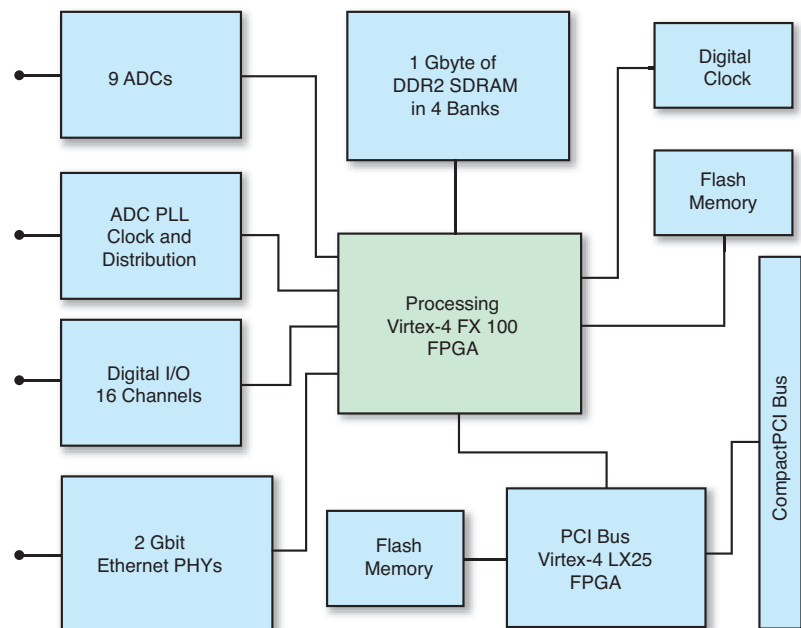

Figure 4 – The PUPE Board Structure

We implemented the interface to the host computer as a register plus shared memory interface. The firmware implements a set of state machines that are software programmable to carry out most of the data-capture and data-processing func-

tions with minimal host software interaction. The FPGA hardware can thus be programmed to implement the appropriate real-time data capture and data processing for the PS machine's cycle.

The core module is the particle beam synchronized phase-locked loop. This structure allows the processing algorithms to synchronize to the incoming analog beam signals. Implementing this real-time structure would have probably required dedicated hardware. The FPGA allowed us to implement it together with the DSP processing algorithms side-by-side in the same chip.

The firmware design called for a number of clock domains. The ADC clock, the PCI bus clock, the external system clock as well as the lower-level clocks were all needed. The Virtex-4's DLLs and clock structure handle the requirements admirably.

We developed the firmware in parallel with the hardware and system software, using Alpha Data's FPGA boards as a test and development platform prior to having the real hardware available. This helped to tighten up the project development cycle and ensured that any prospective issues emerged at an early stage.

The FPGA utilization was about 75 percent. During development we were conscious of and targeted the power levels the system would use. One thing we noticed, during development, was the need to turn off the "keep alive" for the Multi-Gigabit Transceivers. This saved 7.5 watts of power usage by each FPGA.

TMS System and Software

We used a reduced but standard Fedora Core 6 Linux distribution for the system controller and our own simplified Busybox-based Linux system for each of the three module controllers. The open-source nature of Linux, as well as its stability, is a real boon when used in scientific research. The real-time software we developed provides control, data postprocessing and access, system monitoring and test.

The FPGA's firmware bit file resides on the system controller and is downloaded to the FPGAs on initialization. This method

makes it easy to test out and use different processing algorithms within the FPGAs.

We built test blocks into the FPGA firmware and system software. This allowed testing of the system during development when there were no ADCs available, and later when system testing. (It is difficult to find an old Proton Synchrotron lying around for test purposes!) Again, the FPGA helped enormously here. We could easily add test hooks and module emulation hardware within the FPGA that could be switched in and out under system control or just included during development. In fact, the PUPE firmware has a complete data and logic analyzer built in to aid with diagnostics in the running system.

FPGAs in Use

In scientific research projects like this one, FPGAs really do excel. The nature of scientific work entails a degree of algorithm development as ideas and experiments change. The use of an FPGA makes it easy to tightly integrate parallel DSP functions together with digital PLL, logic state machines and other structures into one chip. The programmable nature of the FPGA allows the engineers to modify the fast real-time data-processing algorithms as required to meet the needs of the experiments. In our case, it also made the overall TMS and PUPE board itself quite flexible and suitable for many other application areas.

The resulting power usage of the complete system is around 700 W in full use (one system controller, three Linux module controllers and 15 PUPE boards). Each PUPE board takes about 35 W—a very reasonable figure considering the processing work involved.

The TMS is now in use at CERN. It is providing more detailed and complete information on the trajectory of the particle beams, giving CERN scientists more valuable data in their never-ending search to discover the workings of the universe we live in.

More information on this system is available at *http://portal.beam.ltd.uk/support/cern.* Alpha Data's FPGA product range is detailed at *http://www.alpha-data.com/product_comp.php.*