29th International Cosmic Ray Conference Pune (2005) **8**, 295–298

Development of a FADC Data Acquisition System for Cosmic Ray Detection at 10^{18} eV

S. BenZvi and S. Westerhoff
Department of Physics, Columbia University, New York, NY 10027, USA
Presenter: Andrew O'neill (sybenzvi@phys.columbia.edu), usa-benzvi-S-abs1-he15-poster

A partial flash analog-to-digital readout system for cosmic ray detection at 10^{18} eV has been designed and tested at Columbia University Nevis Laboratories. The electronics consist of a FADC module that digitizes 16 phototubes at 40 MHz with 14-bit dynamic range. The module is read out to a PC through a PCI interface, also designed at Nevis. Taking advantage of the large bandwidth provided by the PCI bus, we have implemented a software-based data acquisition system.

We describe preliminary phototube and electronics tests carried out using a prototype FADC module, and discuss future plans for a full air fluorescence telescope-sized readout system.

1. Introduction

In the past few years, several proposals have been made to extend the stereo air fluorescence technique of ultra-high energy cosmic ray observation to the energy range between 10^{17} and 10^{18} eV [1]. To accommodate such observations, we have designed and partially implemented a fully digital readout system for an air fluorescence telescope.

In its current form, the readout system, which has been tested at Columbia University Nevis Laboratories, contains three basic components: a subcluster of sixteen Photonis XP3062 photomultipliers; an "FADC module," which is simply a printed circuit board responsible for digitizing the PMT outputs and making basic trigger decisions; and a compact PCI board that handles two-way communication between the FADC electronics and a data acquisition PC. The phototube outputs are digitized at a high rate (40 MHz) to observe relatively close air showers at the low end of the desired energy range.

In the design of the readout electronics, we attempted to follow two guiding principles. First, so as to limit analog noise and ease the signal processing requirements, the electronics are set up to digitize the PMT outputs immediately after integration and shaping by two preamplifiers. All subsequent monitoring and triggering tasks are performed on the digitized waveforms. Second, the system offloads most trigger decisions to the data acquisition (DAQ) software, taking advantage of the speed of the DAQ PC. The large bandwidth of the PCI bus (130 MB s^{-1} nominal) easily accommodates the large flow of data from the FADC module.

We will now discuss the components of the electronics system in detail, and describe the results of basic calibrations.

2. FADC Module

The FADC electronics are composed of a single digital signal processing (DSP) board that accepts differential analog signals from sixteen phototubes, and a backplane that receives i/o and clock input from a PC via the PCI interface. Analog data from the phototubes are transported to the DSP board by two high-bandwidth Mini D ribbon cables, while the DSP and backplane communicate via three MZP board-to-board PCB plugs.

After arriving at the DSP board, the integrated and shaped analog output from each tube in the PMT subcluster

is processed by two high-speed differential amplifiers and digitized by a 40 MHz, 14 bit, 300 mW flash analog-to-digital converter (FADC). The particular converter used in the FADC module (Analog Devices AD9244) was selected for its good balance between high digitization rate and low power consumption [2].

Following digitization by the sixteen FADC's, each input channel splits the digitized data along two paths: a trigger path for signal processing by an Altera Stratix FPGA; and a deep memory path to store the data while the trigger decision occurs. The memory path is implemented by a 1k×14b FIFO, which is sufficient to store the data for 25 μs during the trigger stage. The trigger path is shown schematically in Fig. 1.

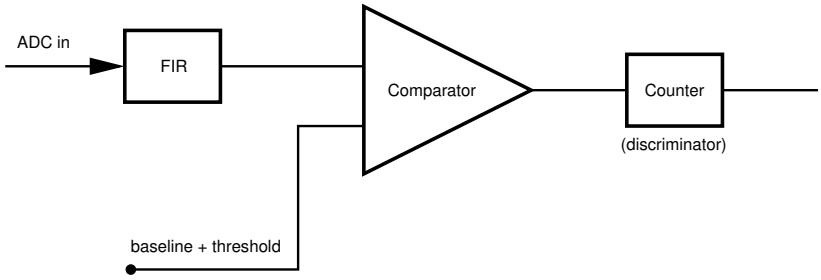

Figure 1. Flow diagram of one of the sixteen trigger channels in the FADC module. The trigger channel is primarily responsible for simple baseline and threshold comparisons.

In each trigger channel, the digitized photomultiplier waveforms are integrated over one, two, four, or eight successive samples by a FIR filter implemented in the Stratix FPGA. The integration time of the filter is adjustable within the DAQ software. The filter output is compared to the sum of external baseline and threshold parameters, and then is further discriminated by a counter. The external channel baseline, trigger threshold, and counter discriminator window are unique to each channel, and may also be adjusted by software.

At each ADC clock, the counter increments when the PMT signal is above threshold, and decrements when the signal drops below threshold. A discriminator monitors the counter level, and when it exceeds a predetermined base value for a predetermined number of clock cycles (both adjustable by the user), the discriminator will enable the trigger channel. Hence, the counter can effectively double the length of time available for time coincidence between channels when compared to using the signal alone.

Once the trigger channel has been enabled, either by the counter discriminator or by an explicit request from the DAQ software, data stored in the memory paths are packed into 18-bit words and moved to the PC. This decision can occur in two ways. First, if an adjustable number of channels are triggered in time coincidence, the stored data from all channels will read out to the PC. Second, the DAQ software may make an explicit data request — as is often useful during testing and baseline monitoring — at which point all channels will read out for an adjustable period of time, independent of their trigger states.

As mentioned earlier, the channel baselines, thresholds, discriminator settings, and the time coincidence number may be set dynamically from the host PC. In this way, the DAQ software can monitor each channel "offline" and adjust the trigger parameters for the subcluster to raise or lower trigger rates depending on drifts in background light levels. However, the Stratix FPGA aboard the FADC module also monitors the signals "online," tracking an internal baseline and variance for each channel by averaging the FADC outputs over 256 clock cycles. If the user chooses, this fast "online" internal baseline can be used for trigger decisions rather than the relatively slow "offline" external baseline.

3. PCI Readout Board

When a trigger occurs in the FADC module, the data are sent to the DAQ host computer, an Intel PC running Linux. The host communicates with the FADC module through a compact board that plugs into its PCI bus. This board, a 32 bit, 33 MHz PCI accelerator, is driven by a PCI 9056 chip from PLX [3]. It has three connections to the FADC module i/o backplane: two high speed Fiber Channel cables for control and data, and a USB 2.0 link for clock.

The PCI card itself contains two major components: the PLX PCI 9056 chip, which sets up data transfers over the PCI bus; and a Xilinx Spartan FPGA, which codes and decodes control and event data as they move between the PC and FADC module. The PCI 9056 chip is used in direct slave C-Mode [3], meaning that PCI memory addresses are mapped directly into the virtual address space of the PC operating system. Hence, software access to registers and memory spaces in the PCI 9056 and Spartan chips does not require the overhead of a function call, making most operations extremely fast. Moreover, the card contains two channels for direct memory access (DMA) transfers to the PC, allowing it to move large volumes of data into PC main memory without taxing the host CPU.

The PCI card is nominally capable of very large data transfers, up to 8 MB in a single transfer at a rate of 130 MB s^{-1}. In benchmarking tests conducted on a Windows PC at Nevis, we have observed sustained transfer rates of $\sim$ 80 MB s^{-1}.

4. Software DAQ

The operation of the card and data acquisition system in the PC is fairly straightforward. When data arrive from the FADC module, they are decoded and repackaged by the Spartan FPGA into two lists: an event list of 18-bit data words containing online baselines, variances, and digitized waveforms; and a list of addresses that point to events in the event list. Both of these lists are moved through the DMA channels to PC main memory, where the data are then accessed by the offline DAQ software.

The DAQ PC is an Intel machine running Linux kernel 2.4, chosen for its compatibility with the PLX PCI device driver. The DAQ software is responsible for allocating large, contiguous blocks of main memory — in sections of up to 8 MB of physical RAM — for DMA transfers. As data are recorded in the DMA buffers, the software makes rapid trigger decisions (for instance, geometrical triggers on phototubes) and moves surviving events out of RAM and onto a RAID-1 disk.

The large volume of data moving over the PCI bus into main memory constrains the DAQ software in several ways. First, there must be sufficient physical RAM available so that arriving data can be stored while the CPU makes trigger decisions about older data. Second, if the available memory is exhausted, the DMA transfer must be halted until sufficient space again becomes available. And third, DMA transfers must stop and start as rapidly as possible to avoid loss of valid data.

Regarding the first constraint, Linux turns out to be a superior choice for the host PC; several excellent noncommercial patches for the Linux kernel allow users to allocate hundreds of MB of contiguous physical memory at boot time [4]. However, in order to leave time to construct triggers, it makes sense to divide allocated memory into small blocks and allow the CPU to analyze a reasonable amount of data. A common solution to this problem is to split available memory into blocks and then create a linked list of memory regions [5]. Once the memory has been allocated, it is extremely easy to construct such a memory structure, for instance a "virtual ring buffer," in software.

In our implementation, we have divided the contiguous memory space of about 100 MB into blocks of 4 MB

to 8 MB, matching the maximum DMA transfer size of the PCI interface and containing about 3 to 6 ms of FADC data. These blocks are linked together into a ring structure, and the DAQ operates by iterating through the ring, analyzing the data block-by-block. We split the DAQ process phase into two threads: one to start and stop DMA transfers into every memory block, and one to iterate through the ring and read data after it has been written. This scheme automatically satisfies the second constraint; if no empty memory blocks are available, the DMA "write" thread will simply wait until memory is cleared by the data processing "read" thread. Note also that the read thread can be made to process several blocks at once so as to avoid splitting up longer events.

As for the last point regarding the speed of setting up and halting transfers, these operations essentially involve setting and clearing bits in four PCI 9056 registers. Data are not recorded for the duration of the start and stop operations, but thanks to the direct slave address mapping allowed by the PCI card, the time required by both operations is a negligible fraction of the total recording time.

5. Discussion

Using the partial readout system of sixteen photomultipliers, the prototype FADC module, the PCI board, and a host PC, we have carried out several simple light calibration tests at Nevis. Placing the PMT's in a dark box and pulsing them with a blue LED (attenuated with neutral density filters), we observed the single-electron response of the subcluster.

At the typical operating gain of the phototubes (5×10^4) we found that one photoelectron corresponds to approximately one ADC count. This suggests that the dynamic range of the FADC's is sufficient to view showers in the desired energy range without the need for additional low-gain overflow channels. Changes in the background light level can be accounted for by dropping and raising the threshold and discriminator constraints in the DAQ software.

Within the DAQ software itself, we have implemented further simple threshold triggers to analyze the data, but we have yet to build a geometrical trigger for a full camera (sixteen pixels is not sufficient). In order to use this readout in a full air fluorescence detector, we will design an intermediate readout board to collect data from sixteen or more FADC modules.

6. Acknowledgments

The purchase and development of the electronics is supported by the National Science Foundation under grant NSF-PHY-0134007.

References

[1] T. Adams et al., e-print astro-ph/0303484 (2003)
[2] Analog Devices AD9244 A/D Converter Data Sheet, Rev. B (2004)
[3] PLX, PCI 9056BA Data Book 1.1 (Oct 2003)
[4] J.P.M. Middelink, http://www.polyware.nl/ middelink/En/hob-v4l.html
[5] S. Ludwick, Eval. Eng. 41, 12 (2002)