PRAMANA — journal of physics © Indian Academy of Sciences Vol. 57, No. 1 July 2001 pp. 135–139

Data acquisition for experiments with multi-detector arrays

A CHATTERJEE, SUSHIL KAMERKAR, A K JETHRA, S PADMINI, M P DIWAKAR, S S PANDE and M D GHODGAONKAR
Nuclear Physics Division and Electronics Division, Bhabha Atomic Research Centre, Trombay, Mumbai 400 085, India

Abstract. Experiments with multi-detector arrays have special requirements and place higher demands on computer data acquisition systems. In this contribution we discuss data acquisition systems with special emphasis on multi-detector arrays and in particular we describe a new data acquisition system, AMPS which we have developed recently which is in regular use in experiments at the Pelletron Laboratory, Mumbai. This includes the in-house development of a dedicated crate controller, PC interface card and software.

Keywords. Data acquisition; multi-detector arrays; CAMAC.

PACS Nos 29.50.+v; 07.05.Hd; 07.05.Wr

1. Introduction

The main characteristic of data generated in experiments with multi-detector arrays is a large number of parameters grouped into many clusters. In terms of requirements for a data acquisition system this translates into (a) multi-crate architecture, (b) complex trigger logic and (c) faster readout.

2. Possible configurations

The instrumentation platforms available today for data acquisition are: Fastbus, VME/VXI, Fast-CAMAC, CAMAC-FERA and CAMAC. Fastbus is used mainly in high energy physics experiments while for experiments at lower energies one uses CAMAC, with the option of fast encoding and read out (FERA). Some systems additionally employ memory and processing units in VME/VXI. Fast-CAMAC is a proposed extension to the CAMAC data acquisition standard, which substantially increases the data transfer rate of CAMAC systems while retaining full compatibility with standard CAMAC. Use of Fast-CAMAC is not yet prevalent, however it may eventually become the optimal choice.

Recent reports of data acquisition systems in the literature indicate the use of CAMAC as the front end. LeCroy FERA and FERA compatible modules are commonly used with readout and transfer to VME memory modules. A visible trend is that large mainframe computers have given way to PCs. By and large, the operating system of choice seems to be Linux. Optical links provide a fast connect from VME to PC and where several PCs are to be interconnected, ethernet with TCP/IP protocol is used.

A data acquisition system for a target multi-fragmentation experiment with large solid angle detectors [1] utilizes 2 CAMAC crates with kinetic systems crate controllers K3922 and auxiliary crate controller K3976, interfaced using K2915 PC interface card to a PC running Linux and networked over ethernet to a second PC. Data storage is on writeable CDs and 8 mm tapes. The data acquisition system for the spectrometer SMART at RIKEN [2] uses FERA readout to HSM 8170 VME memory. An optical link (up to 2 km) is made to PCs which are connected over ethernet.

3. Zero suppression

An important feature of data acquired with multi-detector arrays is that the data sparsity is high, i.e. only a limited number of the detectors in the array actually fire. An event has a large number of zeroes. In this context, zero-data refers to digitized data not meeting pre-defined threshold criteria. Lower thresholds are useful to ensure cut offs below noise levels while upper thresholds could be used e.g. when a time-of-flight exceeds a useful value.

We define the average value of the fraction, F of zero data in an event as the sparsity. The data density, $D = 1 - F$, is defined as the average fraction of useful (i.e. non-zero) data in an event. Data with F as high as 80–99 % is common with multi-detector arrays.

Data acquisition could deal with sparse data at two levels. The first level is zero-suppressed ADC readout. Several models of ADCs have the possibility to skip readout of channels that do not satisfy a preset threshold condition. Using such a feature results in saving readout time in addition to data compaction. Unfortunately, different schemes of zero-suppression are in use making things rather complicated when ADCs of different manufacturers are used. At the second level, zero suppression could be done by software. This results in large reduction in storage of event-mode data. With present technologies, the capacities of storage media (especially hard disks) are increasing exponentially. There are also possibilities for data compressed to disk on the fly by the operating system itself. In spite of these possibilities, zero-suppression of sparse data by the data acquisition system remains very important. Running large experiments results in huge data volumes. No matter how large the storage medium, it is worthwhile to have compact files. Moreover compact files are faster to read during analysis. Leaving the data compression task to the operating system is not a good idea, because the general compression algorithms used would cause a greater software overhead as compared to an algorithm devised for the specific task at hand. The most common methods for zero-suppression are the hit pattern method and the addressed method. The former is useful when the number of parameters and/or the data sparsity is relatively small. The addressed method results in very good compression when there are a large number of parameters with a large fraction of zeroes. We also describe another method, which we call the group suppression method that is better suited for a wider range of sparsity values and is therefore a better choice for data acquisition with multi-detector arrays.

In the hit pattern method, the data for each event is preceded by a binary pattern of *zeroes* and *ones*, with *ones* indicating that a data value meets a pre-set threshold condition.

The bit-pattern is followed by only non-zero data. The basic limitation of this scheme is that when the number of parameters is large, a large amount of space is required for the hit-pattern itself. This means that there are possibilities for further compression as the hit pattern itself contains long runs of *zeroes*.

In the addressed method, each non-zero data is preceded by an address indicating the parameter number while zero-data are skipped. Since this amounts to using double the space for each non-zero data, the method starts to be useful at a sparsity, $F = 0.5$. However it wins over the hit pattern method only when the data density, $D < 1/16$ (sparsity $>$ 93.75%).

In our group pattern method, the parameters are considered to be in groups of 4. Each event is preceded by a pattern, similar to the hit-pattern but requiring only $1/4$ the number of bits. A *one* in the group pattern signifies that at least one parameter in the group of 4 satisfied the threshold criteria. The parameter number within a group is a number, 0–3, which can be encoded in 2 bits which we write in bits 14–15 of the data word itself (the ADCs in use have a maximum of 13 bits). We also utilize the 16th bit of the data word to signify the end of the event. Compared to the hit pattern method, this is better, since the pattern requires 1/4th the space while extra information is coded in the higher unused bits of the ADC data.

For data with n parameters and data density, D, unsuppressed data needs n words of storage per event. The addressed method requires $2nD$ words/event, the hit pattern method requires $n/16+nD$ words/event and the group pattern method requires $n/64+nD$ words/event.

4. Need for development of a new system at our laboratory

At our laboratory we were using a parallel processing data acquisition system [3] with 4 transputers along with a DOS based PC. We decided to develop a new data acquisition system [4] with the following motivations. For the operating system we considered Windows 98, Windows NT/2000 and Linux and selected Windows NT/2000 considering its real time features, its graphical environment, and scalability. We designed a new interface card compatible with PCI bus (our earlier PC interface card was designed for the ISA bus). With the advent of much higher processor speeds and availabilty of multi-processor PCs and with the decreased popularity and availability of transputers, we decided to employ the PC for the software tasks. We however retained 1 transputer for CAMAC i/o as part of the crate controller CC-10.

5. Software design

The software makes use of the symmetrical multi-processing (SMP) feature of Windows NT. It is scalable to multi-processor PCs. Multiple threads are used for the following tasks:

1. Readout of CAMAC buffered data from CC-10
2. Zero-suppression and output of list data to disk
3. Build gated spectra
4. Display and user interface
5. Computation threads (peak analysis etc.)

A dedicated kernel mode device driver with DMA interupt facility under Windows NT/2000 was written for the first task. Obviously this runs at the highest priority. Multiple buffering scheme was implemented through inter-thread synchronization mechanisms of Win32.

Multi-threading architecture permits overlapping of operations of data acquisition, data saving and spectrum building, resulting in an improved system performance even on a single processor system. Use of multiple threads allows taking advantage of multi-processor support in Windows NT/2000.

The software was developed using Visual C++. The development was component based, using the established component standard, Microsoft Active X and the development framework, Microsoft foundation classes, MFC.

The new software components built included, $1d/2d$ graphs control and a cell editing user interface for data input.

6. Main functional features

The software is designed in one integrated offline and online version. When loaded (without the driver) it can be run offline for data analysis under Windows 95/98 as well.

The graphical user interface (GUI) is consistent with the multiple document interface standard (MDI), and the buttons/icons etc. give the program the look and feel of most modern Windows programs.

Multiple 1D/2D spectra windows operate independently. All the usual features relevant for processing spectra have been incorporated with many enhancements. The banana gates are tunable: after drawing a polygonal boundary over a two-dimensional spectrum, it is possible to use the mouse to adjust the verticies of the polygon and view the effect on a projection.

The system was initially implemented on a single CAMAC crate, however multi-crate functionality is planned. The data format of all files [5] is compatibile with the previous system (spectra, zero-suppressed spectra, list mode, zero-suppressed list mode).

A user programmable facility has been included. This allows the user to include his own code for generation of pseudo parameters etc. The user code can be written comfortably since a *wizard* has been provided for the C++ source file, resulting in the generation of a user DLL (dynamic link library).

Built-in file safety features are provided. Whenever the buffer count reaches a user settable value all the spectra are written to disk. Similarly, list-mode file is closed and reopened in append mode periodically. These features allow recovery of data after a power failiure or system crash.

7. Performance

The system with its new PCI card can acquire and write list data to disk with a throughput of 1 Mbyte/s. The speed is limited by the speed of the serial link from the transputer crate controller to the PC interface card (which is 20 Mbits/s).

When the data rate is 500 Kbyte/s, on a low-end Pentium-II PC with a clock speed of 266 MHz, 64 spectra $(1d/2d)$ can be built simultaneously with no additional dead time. On a dual processor Pentium-II, the corresponding figure is 250 spectra.

8. Future enhancements

We intend to replace the present CC-10 transputer crate controller with a device built using field programmable gate arrays (FPGA). This should result in a more flexible design and will also remove the present bottleneck on the throughput.

References

[1] Y Tanaka *et al*, *Nucl. Instrum. Methods* **A425**, 323 (1999)
[2] H Okamura *et al*, *Nucl. Intrum. Methods* **A443**, 194 (2000)
[3] R D Patil, M S Dhamange, S P Borkar, A K Jethra, A Chatterjee, M D Ghodgaonkar and K R Gopalkrishnan, *Int. Nucl. Phys. Symp.* (1995) I-15
[4] S Padmini, M P Diwakar, A K Jethra, S Kamerkar, S Pande, M D Ghodgaonkar and A Chatterjee, *DAE Symp. Nucl. Phys.* **B42**, 362 (1999)
[5] A Chatterjee, *Int. Nucl. Phys. Symp.* (1995) I-30