

DESIGN OF THE DATA ACQUISITION SYSTEM FOR THE NUCLEAR PHYSICS EXPERIMENTS AT VECC

P. Dhara, A. Roy, P. Maity, P. Singhai, P. S. Roy, VECC, Kolkata, India

Abstract

The beam from K130 room temperature cyclotron is being extensively used for nuclear physics experiments for last three decades. The typical beam energy for the experiments is approximately 7-10MeV/nucleon for heavy ions and 8-20MeV/nucleon for light ions. The number of detectors used, may vary from one channel to few hundreds of detector channels. The proposed detector system for experiments with the superconducting cyclotron may have more than 1200 detector channels, and may be generating more than one million parameters per second. The VME (Versa Module Europa) and CAMAC (Computer Automated Measurement and Control) based data acquisition system (DAQ) is being used to cater the experimental needs. The current system has been designed based on various commercially available modules in NIM (Nuclear Instrumentation Module), CAMAC and VME form factor. This type of setup becomes very complicated to maintain for large number of detectors. Alternatively, the distributed DAQ system based on embedded technology is proposed. The traditional analog processing may be replaced by digital filters based FPGA (Field Programmable Gate Array) boards. This paper describes the design of current DAQ system and the status of the proposed scheme for distributed DAQ system with capability of handling heterogeneous detector systems.

INTRODUCTION

The data acquisition system at VECC (Variable Energy Cyclotron Centre) has been designed using standard commercially available hardware modules suitable for low energy nuclear physics experiments. The software has been indigenously developed at VECC for VME [1] and CAMAC [2] based system. Though the system is well proven for all the experimental needs; but the setup and management of this system for large number of detector channels becomes difficult job. The digital filter based FPGA boards may be very conveniently used instead of large number of analog modules. Different types of detector system may be used independently in the same experiment. This type of heterogeneous detector system needs a high precision time-stamping mechanism to synchronise the data.

CAMAC SYSTEM

The CAMAC is an old standard developed by ESONE (European Standards on Nuclear Electronics) committee in 1972. It is a 19" crate system with 25 slots to attach the modules. The slot 24 and 25 is used by the CAMAC crate controller. The CAMAC modules are independent functional units that can be accessed through crate

controller. The crate controller is generally interfaced with the computer through a PCI card. The Ethernet based crate controller are also available, and popularly being used for slow control of different devices. The CAMAC solution consists of a Hytec 5331 CAMAC controller with PCI interface and a Hytec 1341 List Processor.

Each CAMAC module is addressed by a group of three signals N , A and F . The N is the slot number. Each module can host sixteen sub-units; which are selected by 4-bit signal A . Each sub-unit can perform 32 functions; which are selected by 5-bit signal F . There are separate 24bit *Read* and *Write* bus. The crate controller also issues six control signals, $S1$ (strobe), $S2$ (strobe), Z (initialize), C (clear), B (busy) and I (inhibit). The CAMAC module responds with three signals; L (Look At Me, LAM), X (command status) and Q (operation status).

The typical CAMAC read cycle can be executed by the controller in 1 μ sec. In effect, a 24bit data read will result maximum data transfer rate of 3MB/sec. But the actual data transfer rate from the application software is determined by the software overhead. The single read takes 10 μ sec per word. The faster data rate is achieved by using List Processor module. This module can perform CAMAC operations as auxiliary controller and store the data in local buffer. The data is then read from the computer in block transfer. It takes average 2.14 μ sec for single read.

CAMAC Software

The CAMAC system is being used over decades at VECC. The CAMAC applications are built using the ESONE standard API. This makes the CAMAC system highly portable on different controller and operating systems. There are two versions of software:

- **T32:** It is the 32bit version of CAMAC developed using Windows SDK. The development life-cycle is over and only a limited support is offered for some old installations.
- **CAMACDAQ:** This is the currently supported version of the CAMAC system. It runs on both 32bit Linux and Windows machine. The software is the common framework for different DAQ hardware both VME and CAMAC system. A CAMAC crate is viewed as a single hardware module with multiple channels and inbuilt buffer. The list processor acquires the data independently from various modules inside the crate and stores them into the buffer. The software reads data from the list processor. So, the individual modules are isolated from the software perspective; thus, a wide variety of hardware modules are easily supported.

VME SYSTEM

The VME is a computer bus system, consists of address bus, data bus, control bus, utility bus and user I/O bus. The VME Bus can be efficiently utilised by a microprocessor based VME controller board. The VME is also a 19" crate based system with 21 slots and a common backplane for all the connectivity. The VME modules are available in different form factors 3U, 6U or 9U height (1U=1.75inch). The VME DAQ system uses 6U form factor VME64/VME64x crates. Following are the salient features of VME64X crate

- VME is asynchronous bus system, working in master/slave mode. Data transfer is done in handshaking mode. The bus can be used in both multiplexed and non-multiplexed mode.
- The backplane supports 32bit data bus and 32 bit address bus. The data and address bus may be multiplexed to provide 64 bit data and 64bit address. The VME system supports different addressing mode A16, A24, A32, A40 and A64. Data can also be transferred in D8, D16, D32 and D64 bits.
- The minimum read cycle time is 10ns, provided the slave could operate at the same speed as master. So, D32 mode may have maximum data transfer rate of 40MB/sec. The D64 mode may have maximum transfer rate of 80MB/sec.
- The VME Bus supports multiprocessing and four levels of interrupt. There must be one system controller at slot 1 to perform the bus arbitration. Multiple masters may request for bus, and the bus arbitration logic grants the bus access to masters.
- A typical VME master initiates bus access in Read/Write cycle, Read-modify-write cycle, block transfer cycle (BLT) or multiplexed block transfer (MBLT) cycle. The block transfer cycle may be in FIFO mode also, where an address cycle is followed by a number of data cycles.
- VME offers Chained Block Transfer mode (CBLT) of operation. In this mode, the block transfer call may span across a number of VME modules in the crate. The modules configured in this mode send acquired data one after another against a single block transfer call. The first module starts sending data, after it is purged out, the IACK signal is passed to next module for sending data.

VMEDAQ System

The VMEDAQ system has been implemented using CAEN V2718 Controller and SIS3100 controller. None of these controllers have processors for user application programming. They are driven from a connecting PC through PCI card and fibre optics cable. System supports CAEN V785 peak sensing ADC (analog to digital converter), V775 time to digital converter, V792 charge to digital converter, Mesytec MADC32 peak sensing ADC, MDI2 peak sensing ADC, SIS3820 scalar etc.

VMEDAQ Software

The VMEDAQ software is a multi-threaded C++ application developed on Linux. The software has been designed in layered architecture for easy maintenance and support. There are three layers *front-end*, *associator* and *back-end*.

- The *front-end* layer contains the graphical user interface, using QT-3 toolkit. It has the main control window, histogram windows and configuration management etc.
- The *associator* layer establishes the data and control communication between front-end and back-end. It stores common information shared by all layers.
- The *back-end* layer is also written in C++ with STL (standard template library), which reads data from the DAQ hardware. It also sorts the data and generates the global event. Afterward, the events are stored and processed.

The back-end layer implements three independent threads *read-out*, *event-generator* and *process*, which are working in producer-consumer configuration (see Fig. 1). The *read-out* thread reads the from VME or CAMAC hardware in block transfer mode (CBLT for multi-module VME setup). The data is stored in one of a dual buffer system. The buffers are shared by read-out and event-generator threads using two pairs of read/write semaphores. Same mechanism of data sharing exists between event-generator and the process threads. The event-generator thread generates the global event by organizing the fragments of event registered by different modules. Next, the process-thread stores the data into file and generates the one-dimensional and two-dimensional histograms as per function list.

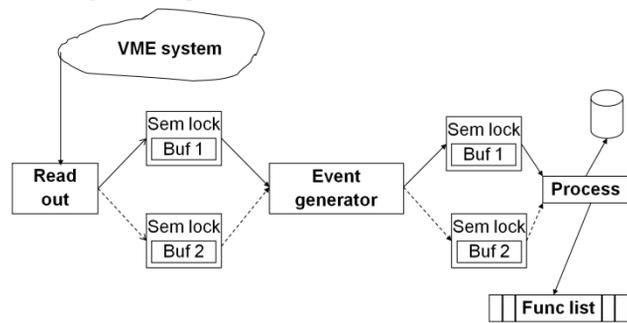


Figure 1: Event processing in threads.

DAQ Setup

The DAQ software works in common dead time mode. The detector output signal first goes through the front end electronics (FEE) circuit, which does the signal conditioning suitable for particular type of acquisition. Then the signal is connected to an ADC channel. The VME785 peak sensing ADC has 32 input channels, which can digitize the input data within 5.7µsec for all channels. There is a gate generator logic which generates a logical GATE signal to validate an event. The ADC modules are enabled by the GATE signal and the input coming within the GATE signal are digitized and stored as valid events.

There may be any number of modules involved, depending upon the number of signal channels. All modules work with the same GATE signal. In this common dead time configuration, the system dead time τ is the maximum dead time of all individual components (see Fig. 2).

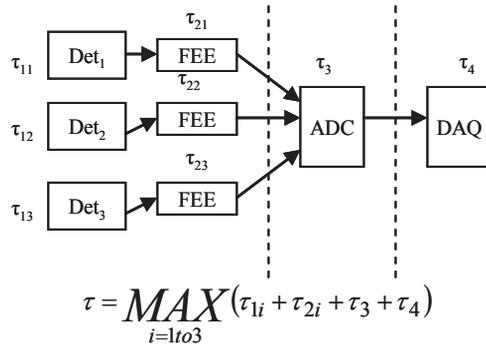


Figure 2: Common dead time configuration.

The DAQ software is configured by a single text based configuration file. There are four directives to be declared. The *System* directive defines the overall system parameters like transfer mode, event trigger etc. The *module* directive is used to declare hardware modules with parameters. The *function* directive declares the processing functions like histograms, add etc. The *gate* directive declares conditions. The software supports all the necessary functionality to manipulate histograms for physics research.

The Multi-crate DAQ Setup

The CAEN V2718 VME crate controllers can be connected in a daisy chain, for maximum eight crates per PCI card (see Fig. 3). The optical fibre interface works at a speed of 70MB/sec. The address space of each crate is isolated and modules in each crate can be read independently. The GATE signal is common to all the modules. A synchronization module has been designed in-house, to block all the successive GATE signals, until the current GATE signal is processed by all crates.

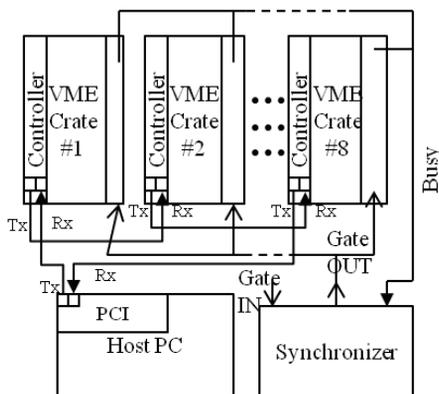


Figure 3: Multi-crate system.

The DAQ system is capable of handling 1.2Mparameters/sec in a dual-crate configuration.

FUTURE DAQ

The modern experiments are using a large number of detectors of various kinds into a single experiment. The module based systems are not convenient any more. The response time of different detectors may render the common dead mode configuration impractical. The following development works are under progress.

The Digital Processing

The detector signal processing with digital filter in FPGA is under development. A 16-channel DPP (digital pulse processing) board has been planned for spectroscopic application.

The ASIC Board

The availability of ASIC (Application Specific Integrated Circuit) chip for pulse shaping & processing is also being explored. The traditional FEE chain may be replaced by the ASIC board.

The Heterogeneous DAQ with Timestamp

The heterogeneous DAQ setup uses different detector arrays working together. They have different response time and resolution. There may be independent DAQ for each detector array. The events are marked with a global unique timestamp. The data may later merge together using the timestamps.

The work is going on to design the timestamp distribution module. The offline sorting and merging algorithm has already been developed.

CONCLUSION

The DAQ system has been intensively used by user communities at VECC in their experiments with K130 cyclotron beam. The system is ready to do experiments with larger detector arrays with K500 beam. There are some improvement proposed for the software to make it efficient in handling large number of histograms.

The DPP or ASIC based DAQ board will make the system more sophisticated for future use.

ACKNOWLEDGMENT

We thankfully acknowledge the help and contribution of all our colleagues of DAQ & Dev Section, VECC. Also we thankfully acknowledge the encouragement and feedback from Experimental Nuclear Physics Division of VECC.

REFERENCES

- [1] P. Dhara et al., "Multi-threaded object-oriented VME data acquisition system on Linux", DAE-BRNS Symp on Nuclear Physics, Vol. 46B, p. 530-531 (2003).
- [2] P. Dhara et al., "Integration of CAMAC system in VME based DAQ Software", DAE-BRNS Symp on Nuclear Physics, Vol 50, p. 454 (2005).