DATA ACQUISITION ARCHITECTURE FOR A NUCLEAR PHYSICS EXPERIMENT AT A COLLIDER

P. CERELLO
INFN, Sez. di Torino, Torino, Italy

V. FILIPPINI
INFN, Sez. di Pavia, Pavia, Italy

L. FIORE
INFN, Sez. di Bari, Bari, Italy

P. GIANOTTI
INFN, Laboratori Nazionali di Frascati, Frascati, Italy

S. MARCELLO
INFN, Sez. di Torino, Torino, Italy
E-mail: Marcello@to.infn.it

B. MINETTI
Dip. di Fisica, Politecnico di Torino and INFN, Sez. di Torino, Torino, Italy

A. RAIMONDO
Dip. di Fisica Sperimentale, Università di Torino and INFN, Sez. di Torino, Torino, Italy

ABSTRACT

A parallel Data Acquisition Architecture, based on VME and using only a few CAMAC crates, has been developed to be used in FINUDA, an experiment on hypernuclear physics at DAΦNE, the Frascati Φ–factory. The acquisition software runs on embedded CES RTPC 8067 processors, using the LynxOS operating system and taking care of the Front-End Electronics read-out. The sub-events are transported, via a dedicated VIC bus, to a Global Event Builder program running on IBM, Power-PC machine. This process controls also event recording on DLT tape and sends, on the network, some events to the monitoring machines. The system and the performances are described.

1. Introduction

FINUDA[1,2] is a non focusing magnetic spectrometer approved for construction and continuous running at $DA\Phi NE$[3], an $e^+ - e^-$ collider with a center of mass energy of Φ meson, that will be operational at $INFN - LNF$ in 1997. The main goal of the experiment is to use the K^- coming from Φ decay in order to study the fine spectroscopy of $\Lambda-$hypernuclei produced after the K^- capture at rest into nuclear targets and the study of their non-mesonic decay.

The FINUDA apparatus is divided into two main regions: the interaction region used to identify the K^- beam and the tracking region, which has been designed to measure the momentum of charged particles. A Time Of Flight[4] (TOF) system of

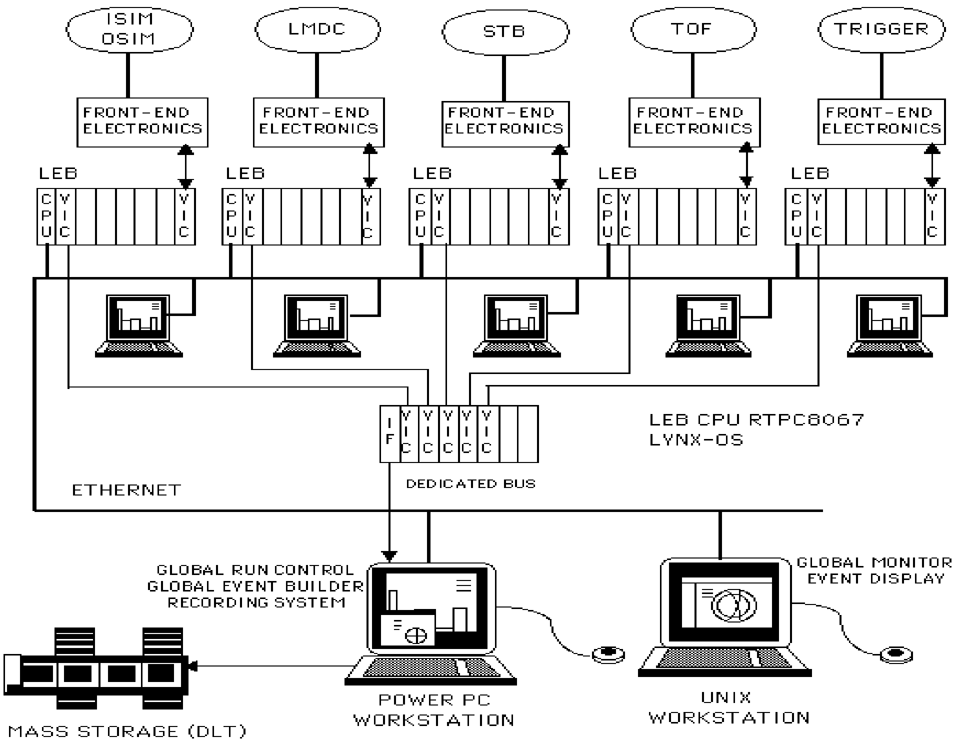

Fig. 1. The FDAQ hardware architecture.

plastic scintillator bars is used to perform energy loss discrimination, topology and timing selections for trigger purposes and for neutron energy measurements.

2. General Architecture

The FINUDA experimental setup consists of several sub-detectors, each of them running on independently VME CPUs: Inner/Outer Silicon Microstrip[5] detector (SIL), Low Mass Drift Chambers[6] (LMD), Straw Tubes[7] array (STB), Internal and External TOF detectors and finally Global Trigger Supervisor (GTS), which is considered as a sub-detector.

We developed a general purpose Data Acquisition system with high flexibility and able to manage the standard operations of a DAQ system. That system can be considered a multiprocessor, tree structured architecture controlled by a Run Control and syncronized for data building by an Event Builder. All the pieces of information coming from the sub-detectors Front End Electronics (FEE) are read-out in a parallel mode by several RTPC8067[8] CPUs (the Local Event Builders), each of them assembles

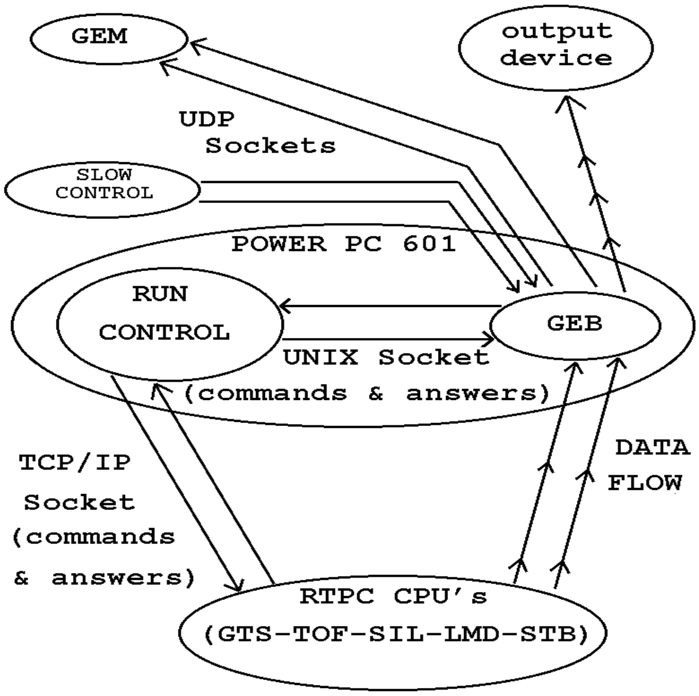

Fig. 2. The processes involved in the FDAQ system and their communication scheme.

a single sub-event and sends it via a dedicated bus to an IBM Power PC running the Global Event Builder process.

Fig. 1 and fig. 2 respectively show the scheme of the FINUDA Data Acquisition (FDAQ) architecture and of the communications among all the processes involved in the system.

At present FDAQ can read out all the VME and CAMAC modules used by the FINUDA sub-detectors, but it can be easily extended to other similar situations.

It is a modular program consisting of some main blocks, each providing a well defined function, communicating with each other through socket connections: the Finuda Run Control (FRC), the Global Event Builder (GEB), the Local Event Builder (LEB), the Global Event Monitor (GEM), the Local Event Monitor (LEM) and the Slow Control System (SCS).

2.1. The FINUDA Run Control

The FRC drives all the DAQ system by sending commands, through TCP sockets, to the VME CPU of each sub-detector and, through an internal UNIX socket, to the Global Event Builder, taking advantage of a Graphic User Interface based on the *Tcl/Tk*[9] package.

A *Tcl/Tk* window corresponding to each stable state (SLEEP, READY, ACTIVE) is available to the operator for sending commands to the sub-detectors and the GEB. Then the operator can easly perform standard operations such as changing global state of sub-detectors, choosing trigger configuration, etc.

2.2. The Global Event Builder

The main task of the GEB process is to collect the event pieces coming from different sub-detectors and to build up the complete event to be written on the output device and to be sent, through UDP sockets, to the GEM and the LEMs. Moreover, the GEB periodically stores the buffer from the SCS together with the physics event data to be written on tape. The GEB takes data from the VIC8251 mirrored memories, bufferized to minimize the dead time, and after having checked the sub-events alignment, it formats the global event and sends it to the writing procedure.

The GEB process makes possible the monitoring of the most important variables and distributions related to the data acquisition performances, such as event length and rate, through a memory shared with the FRC.

2.3. The Local Event Builder

The main task of the LEB process is to collect data, through VIC8250 bus, from the FEE of each sub-detector and to make the information available to the GEB by writing on the mirrored memory of a VIC8251 module. Each read-out starts when a trigger signal coming from GTS is present on the Corbo module.

Data buffers are structured with a header part, containing general information as buffer length, run number, event number, sub-detector identifier, etc., followed by the data, structured into equipments, one for each module type.

In order to allow a dynamic change of the VME and CAMAC read-out configuration for each sub-detector, a suitable system has been developed. The list of the modules to be read is not fixed, but it can be chosen before each run changing a configuration file shared between the acquisition VME CPU and the monitoring machine through NFS. In that way it is not necessary to recompile the DAQ program if the list of modules to be read-out has been changed.

At present 11 different types of VME modules and 11 CAMAC modules, by means of a CAMAC branch driver CBD8210, can be read-out, but it is not difficult to add new modules thanks to the flexibility of the structure.

2.4. Monitor and Slow Control systems

The GEM and the LEM get events from GEB through UDP sockets. The GEM allows a complete overall data quality control through a full reconstruction of event and besides the LEM performs a local, fast monitoring task, in order to check the performance stability of each sub-detector.

The SCS, based on DANTE[10] architecture, controls and monitors the sub-detector power supplies, gas flows, magnet performances etc. Periodically it sends a buffer of information, concerning the general conditions of the data taking, to GEB.

The commercial system Mini Structured Query Language[11] has been chosen as Database and it is available to GEM, LEM and SCS providing access to data during run time. The Database will store calibration and monitoring data generated online and offline.

3. Conclusions and Performances

A DAQ system prototype was assembled and tested. The acquisition rate with a simulated buffer of 2 $kbytes$, corresponding to a tipical FINUDA event, reaches 300 Hz.

The described FDAQ system is based on commercial hardware standards like VME and CAMAC. The consistent use of the standard VMV and VIC busses as data path, between FEE and EB, has led to a high degree of flexibility in the DAQ architecture.

The use of an external module list, read as data card, allows the change of the hardware configuration avoiding a new compilation of the program.

Furthermore the modular run control software, with its high level man/machine interface, provided by the Tcl/Tk package, makes the overall DAQ extremely user friendly.

Thanks to its open structure the system can grow with new demands and can be adapted to new technologies.

1. The FINUDA Collaboration, INFN Report LNF-93/021 (1993).
2. The FINUDA Collaboration, Technical Report, INFN Report LNF-95/024 (1995).
3. L. Maiani *et al.*, *The Second* DAΦNE *HandBook* ***I-II***, LNF-INFN (1995)
4. L. Celano *et al.*, *Perform. of a long plastic scintill. det. for neutron studies* (Proc. of 4^{th} Int. Conf. on Position-Sensitive Det., Manchester, UK, 9-13 Sep. 1996), accepted by *Nucl. Phys. A*.
5. L. Celano *et al.*, CERN-PPE/95-106, accepted by *Nucl. Instr. Meth.*.
6. M. Agnello *et al.*, *Nucl. Instr. Meth.*, **A367**(1995)100.
7. L. Benussi *et al.*, *Nucl. Instr. Meth.*, **A361**(1995)180.
8. C.E.S., *RTPC 8067LK Technical Manual* (1995).
9. John K. Osterhout, *Tcl and the Tk Toolkit* (Addison-Wesley, 1994).
10. G. Di Pirro *et al.*, Proc. of Real Time 93 (1993), p. 250.
11. D.J. Hughes, *MiniSQL a lightweight Database server* (ver. 1.1).