

# A DSP-BASED CONTROL SYSTEM FOR THE MULTIDETECTOR CHIMERA

CHIMERA Collaboration\*  
Italy

## Abstract

A DSP-based control system has been designed for the detector array CHIMERA under construction and to be used with the Superconducting Cyclotron at Laboratorio Nazionale del SUD (LNS) in Catania. The present paper describes the upgrading of the system and the results obtained on beam.

## 1 INTRODUCTION

In Nuclear Physics at intermediate energies, experiments can be performed to study different phenomena, such as, e.g., multifragmentation.

To collect the reaction fragments that can be emitted by a target in all the directions, a very large number of detectors must be employed. This is necessary to assure both the covering of almost the totality of the solid angle around the target and a high granularity of the detector system, necessary to reach the highest spatial resolution in the event selection.

With this purpose in mind, we have designed and are constructing CHIMERA [1] (Charged Heavy Ions Mass and Energy Resolving Array). This new multi-element,  $4\pi$  detector-array for charged particles and fragments began to be operative at LNS in Catania in 1999. The main characteristics of the detector are not only the energy loss and residual energy measurement, to identify the reaction products in charge, but also a systematic measurement of the time-of-flight (ToF) allowing velocity and mass determinations, and a low multi-firing probability due to the adopted high granularity. When completed, it will cover a solid angle of  $(0.94 \cdot 4\pi)$  sr using 1192 cells, each one of two detectors (Silicon [2] and CsI [3]). So each cell outputs 4 signals (Si, fast-gate CsI, slow-gate CsI, time

and there are more than 2000 electronic chains to control and more than 4000 analog signals to handle.

In the present paper the architecture proposed for a small-scale parallel system designed for controlling the multidetector CHIMERA is described. The prototypical unit consists of a commercial board WS3112 [4] installed in the PCI bus of a host computer and based on two ADSP-21060-SHARC (Super Harvard ARChitecture) Digital Signal Processors (DSP) [5]. The use of a board with two DSPs allows a parallel approach and represents an important upgrading of our previous system, realized with only one old generation DSP [6].

## 2 GENERAL OVERVIEW

### 2.1 The user architecture

As a result of the complexity of CHIMERA and of the huge number of signals involved, it is highly evident that the real-time control of the stability of the multidetector strictly related to the necessity of a high-reliability data collection, is not a trivial problem.

Depending on the working mode of CHIMERA the control system can be subdivided in three different phases, correlated among them: data collection, data computation and result presentation.

Firstly it has to collect input words, deriving from the randomly acquired analog signals and belonging to different detector cells, each one identified by a proper pattern word. This pattern has to be kept into account for all the data-paths, to allow a correct correlation between output results and input data, together with an unambiguous identification of the fired detection cell. As second point, the acquired words have to be processed by means of different algorithms, each one chosen to perform a particular task of the full control operation. At the end

---

(\*) P.Guazzoni<sup>1</sup>, S.Aiello<sup>2</sup>, M. Alderighi<sup>3</sup>, A.Anzalone<sup>4</sup>, M.Bartolucci<sup>1</sup>, G.Cardella<sup>2</sup>, S.Cavallaro<sup>4,5</sup>, E.De Filippo<sup>2</sup>, S.Feminò<sup>6</sup>, E.Geraci<sup>4,5</sup>, M.Geraci<sup>4,5</sup>, F.Giustolisi<sup>4,5</sup>, M.Iacono Manno<sup>4</sup>, G.Lanzalone<sup>5</sup>, G.Lanzano<sup>2</sup>, S.LoNigro<sup>4,5</sup>, G.Manfredi<sup>1</sup>, A.Pagano<sup>2</sup>, M.Papa<sup>2</sup>, S.Pirrone<sup>2</sup>, G.Politi<sup>2,5</sup>, F.Porto<sup>4,5</sup>, D.Salvadori<sup>1</sup>, S.Sambataro<sup>4,5</sup>, G.Sechi<sup>3</sup>, L.Sperduto<sup>4,5</sup>, C.Sutera<sup>2</sup>, L.Zetta<sup>1</sup>

<sup>1</sup>Dipartimento di Fisica dell'Università and Istituto Nazionale di Fisica Nucleare, via Celoria 16, I-20133 Milano, Italy;

<sup>2</sup>Istituto Nazionale di Fisica Nucleare, corso Italia 57, I-95100 Catania, Italy;

<sup>3</sup>Istituto di Fisica Cosmica, CNR via Bassini 15, I-20133 Milano, Italy;

<sup>4</sup>Laboratorio Nazionale del Sud, via Santa Sofia I-95100 Catania, Italy;

<sup>5</sup>Dipartimento di Fisica dell'Università, corso Italia 57, I-95100 Catania, Italy;

<sup>6</sup>Gruppo collegato di Messina, Istituto Nazionale di Fisica Nucleare, corso Italia 57, I-95100 Catania, Italy.

the output data have to be presented on a proper display unit to allow, if required, a *by eye* checking.

These phases reflect the multidetector-working mode: data collection, data computation, and result presentation. In Fig. 1 it is possible to identify the phases one and three with the Front-End Layer and the phase two with the Computing & Process Layer. The Transport Layer is devoted to transfer data and information from and to the other two layers.

The design of this intermediate layer is the crucial point of the software architecture. In fact it will permit dynamic workload redistribution to the different computational units without any reduction of time performances of the system (i.e. without any loss of data).

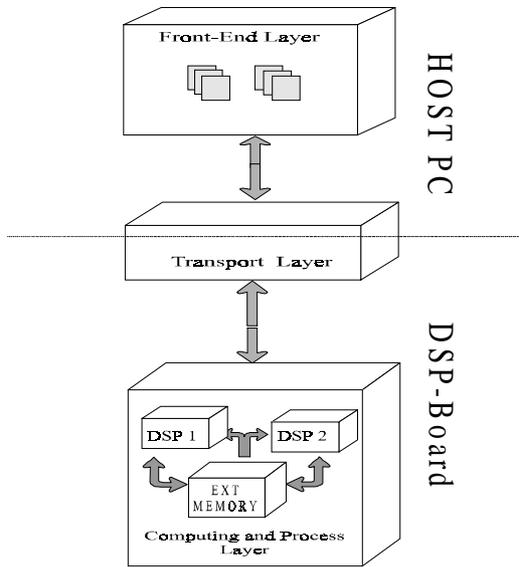


Figure1: Schematic of the hardware platform

With a view to satisfy the previous constraints we designed a model of flexible parallel architecture that preserves its validity independently of the needed number of processes, of the algorithms used and of the type of inputs and outputs.

To develop our parallel computational system we decided to use DSPs [6] as computational units. DSPs can be grouped in one or more boards. Depending on the necessities, they can be installed inside either a VME bus (in a VME crate) or a PCI bus (in a host computer) or in both.

In any case, either the host computer, or one of the DSPs has to manage the whole system. For sake of simplicity, we started using a host computer as a process manager.

We designed the software architecture with some purposes in mind. First aim is to create a model of parallel processes independent of the number of computational units (processors).

Second aim is to obtain a dynamic redistribution of the workload (data and processes) to solve the computational bottlenecks that sometimes grow in a system with random input distribution.

Last aim is to maintain the transparency of the data path and computational processes to guarantee the performance predictability, fundamental feature in a real-time environment.

## 2.2 The Hardware Platform

The host module is a Pentium MMX processor under the operative system Microsoft Windows NT 4.0. The computational unit is a commercial board WS3112 [4], mounting two ADSP 21060-SHARC DSPs [5] with a peak rate of (120+120) MFlops at a clock frequency of 40 MHz (25 ns execution time per instruction). The control subsystem is linked to the data acquisition system [7] via an Ethernet 10 Mbit/s network, through which retrieves data packets, broadcast by the CPU of the data acquisition system.

## 2.3 The Implemented Software

One of the most important requirements of the real-time software design is not only to guarantee the algorithm correctness, but also to allow timely completion of the computation. Our data source (CHIMERA) can be modeled as a Poisson process with some average time distance between successive events. The total dead time of the data system acquisition is about 100  $\mu$ s per event, which states a worst case for the event frequency.

As shown in Fig. 2, to satisfy the timing constraint of our real time application we decided to collect the physical events, coming from the acquisition system, in packet form and store them in a FIFO queue.

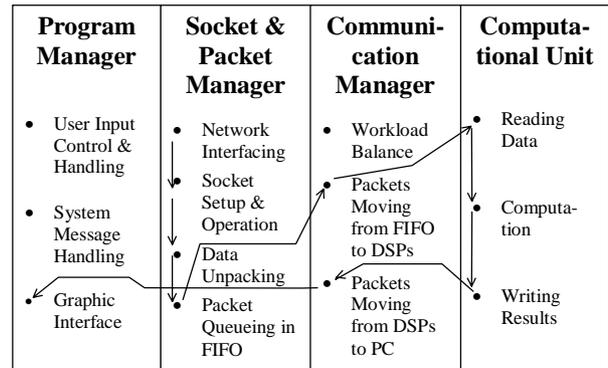


Figure2: Implemented software general diagram

So we designed a preemptive FIFO to comply with the dynamical system, because of the presence of time dependent physical events.

The use of data packets not only avoids loss of data in case of a burst acquiring but also guarantees an efficient communication between the host and the computational units.

To carry out the system job scheduling (crucial factor in case of timeliness requirements) a host main task has been developed, that allows the support of dynamical service communication towards the computational units. Its design is the central point of the software architecture, as

it permits dynamic workload redistribution to the different computational units without any reduction of time performances of the system (i.e. without any loss of data inputs).

Three main tasks run in the host. The first is devoted to receive the input data. The second is a set of “n” threads, where “n” is the number of the DSP-based boards installed on the system (1 in the current prototype), able to manage the DSP communication, while the last one prepares the data for graphic outputs and memory mass storage. This subdivision permits to modify one or more of these tasks saving the rest of system architecture.

Another set of tasks runs on the DSPs: on each of those a process handles communication with the host while a separated one performs computations on the incoming data. Each one of the DSPs involved in the control system can perform a different part of the computation needed, running one of the many complex mathematical functions which will be used to monitor the behavior of CHIMERA.

All the software running on the host is written in C++, in particular using Microsoft Visual C++ 5.0 along with the Microsoft Foundation Classes (MFC) libraries. This allows the use of all the tools of a modern object oriented development environment, such as correctness features, code portability and reusability and easier documentation and maintenance of the programs. It permits also to maintain a high-level approach, avoiding dealing with problems closely related with the particular implementation chosen for the hardware.

As well as on the host, also on the DSP side C/C++ has been adopted as programming language, in its standard ANSI version. We used the optimizing compiler by Analog Devices, specially written for the Sharc DSP. The usage of more powerful DSPs in a number as large as needed will allow a major improvement of the previous prototype tested, in which a single TMS-C30 DSP and assembly language were used instead [6].

### 3 PERFORMANCES AND TESTS

To evaluate the system performances we have used data collected in on-beam experiments and the power law for particle identification as benchmark algorithm.

The power-law algorithm for charge identification can be applied to the signals coming from two detector telescopes. If  $\Delta E$  and  $E$  are the energies lost in the thin and thick detector, respectively, the Charge Identification Function derived from this algorithm can be expressed as:  $CIF = (E + \Delta E)^x - E^x$ , where  $x$  is a real number ( $x \approx 1.73$ ) experimentally found [8].

For a two-silicon-detector telescope CIF gives a monodimensional frequency distributions, characterized by different peaks, each one corresponding to the different charges of the reaction products. So we have tried, in a heuristic approach, to compute CIF using the light ( $L$ ) and the energy loss ( $\Delta E$ ) signals coming from a CHIMERA detector cell, without any preliminary calibration, in order to have an on-line algorithm for controlling the multidetector. And this also if it is well known that the light signal from a CsI(Tl) detector strictly depends not

only from the energy lost but also from the type of the reaction product. The results obtained for the function  $CIF = (L + \Delta E)^x - L^x$  are quite satisfactory as shown in Fig. 3 where a CIF frequency distribution for the reaction products of the  $Ni^{58} + {}^{27}Al$  at 30 AMeV is shown. We have obtained a CIF computation time of about 10  $\mu s$  per pair of data ( $E$  and  $\Delta E$ ).

The possibility to use more boards with a single host represents also an important characteristic for the implementation of our system design: future developments will involve a distributed architecture based on several computers linked via a particular network, each one equipped with two or three boards.

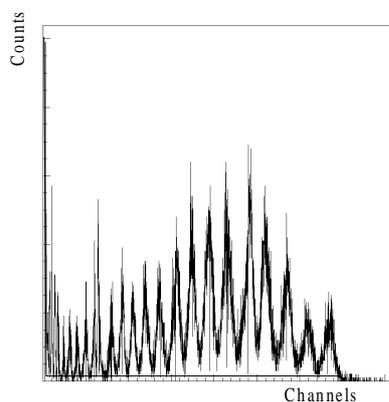


Figure3: CIF spectrum (see text)

### REFERENCES

- [1] S. Aiello *et al.*, Nucl. Phys. A583 (1995) 461
- [2] S. Aiello *et al.*, Nucl. Instr. Meth. A385 (1997) 306
- [3] S. Aiello *et al.*, Nucl. Instr. Meth. A369 (1996) 50
- [4] Wiese GmbH – Lübeck – Germany
- [5] Analog Devices – USA
- [6] S. Aiello *et al.*, IEEE Trans. Nucl. Sci. 45 (1998) 1798
- [7] S. Aiello *et al.*, IEEE Trans. Nucl. Sci. 45 (1998) 1877
- [8] F.S. Goulding and B.G. Harvey – Ann. Rev. Nucl. Sci. 25 (1970) 167