

ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Milano

INFN/TC-98/32
23 Novembre 1998

Chimera Collaboration:

PROPOSED ARCHITECTURE OF A FLEXIBLE SMALL-SCALE PARALLEL SYSTEM FOR THE CONTROL OF A DETECTOR ARRAY

Keywords: Small-scale parallel system, Digital Signal Processor, Real-Time

*Oral presentation selected for the Session Parallel/Distributed Architecture of PDPTA '98:
International Conference on Parallel and Distributed Processing Techniques and Applications
Las Vegas, Nevada, USA, July 13-16 1998*

*Published by SIS-Pubblicazioni
Laboratori Nazionali di Frascati*

Proposed architecture of a flexible small-scale parallel system for the control of a detector array

Chimera Collaboration

S.Aiello¹, A.Anzalone², M.Bartolucci³, G.Cardella¹, S.Cavallaro^{2,4}, E.De Filippo¹, S.Femino⁵,
C.Garusi³, M.Geraci^{1,4}, F.Giustolisi^{2,4}, P.Guazzoni³, M.Iacono Manno², G.Lanzalone⁴, G.Lanzano¹,
S.LoNigro^{1,4}, G.Manfredi³, A.Pagano¹, M.Papa¹, S.Pirrone¹, G.Politi^{1,4}, F.Porto^{2,4}, F.Rizzo^{2,4},
S.Sambataro^{1,4}, G.Savino³, L.Sperduto^{2,4}, C.Sutera¹, L.Zetta³

¹ Istituto Nazionale di Fisica Nucleare, corso Italia 57, I-95100 Catania, Italy

Tel. +39-95-7195111 = Fax +39-95-371600

E-mail: LastName@ct.infn.it

² Laboratorio Nazionale del Sud, via Santa Sofia, I-95100 Catania, Italy

Tel. +39-95-542111 = Fax +39-95-514430

E-mail: LastName@lns.infn.it

³ Dipartimento di Fisica dell'Universita` and I.N.F.N., via Celoria 16, I-20133 Milano, Italy

Tel. +39-2-2392296 = Fax +39-2-2392297

E-mail: Firstname.LastName@mi.infn.it

⁴ Dipartimento di Fisica dell'Universita`, corso Italia 57, I-95100 Catania, Italy

Tel. +39-95-7195111 = Fax +39-95-383023

E-mail: LastName@ct.infn.it

⁵ Gruppo collegato di Messina, I.N.F.N., via Santa Sofia, I-95100 Catania, Italy

Tel. +39-95-542111 = Fax +39-95-514430

E-mail: LastName@lns.infn.it

Presenting author: Paolo Guazzoni, Dipartimento di Fisica, via Celoria 16, I-20133 Milano,
Italy = Tel. +39-2-2392249 = Fax +39-2-2392297 = E-mail: Paolo.Guazzoni@mi.infn.it

Abstract

A commercial board, based on two ADSP-21060-SHARC Digital Signal Processor and installed in the PCI bus of a host computer, has been used such as the core of the proposed architecture of a small-scale parallel system. This system has to be designed for the control of an array of 1192 detection cells, under construction for Nuclear Physics experiments at intermediate energy.

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) a 4π detector for charged particles. This new multi-element detector array for charged particles and fragments will begin 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 to be 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 evolution of our previous system, realized with only one DSP [6, 7].

2. 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 more than 4000 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 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 output 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 inputs).

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 (Fig.2)

To develop our parallel computational system we decided to use DSPs [8] and in particular SHARCs 21060 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); eventually in both environments.

In any case, either the host computer, or one of DSPs has to manage the whole system. For sake of simplicity, we suppose to use a host computer and to use it 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.

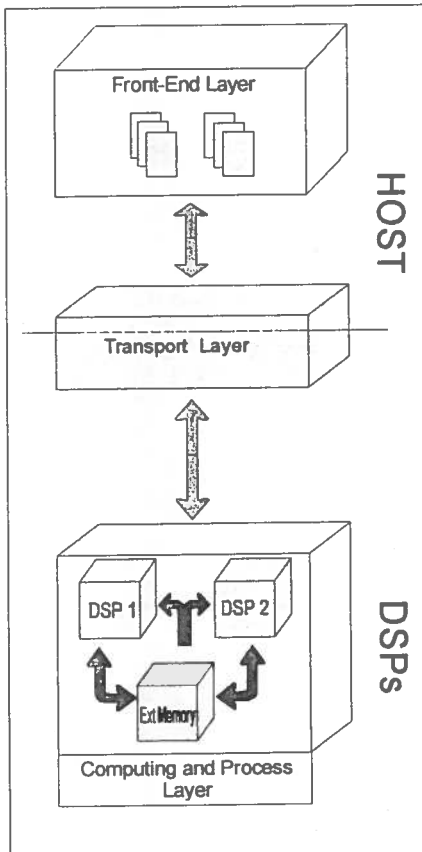


Fig. 1: System working mode

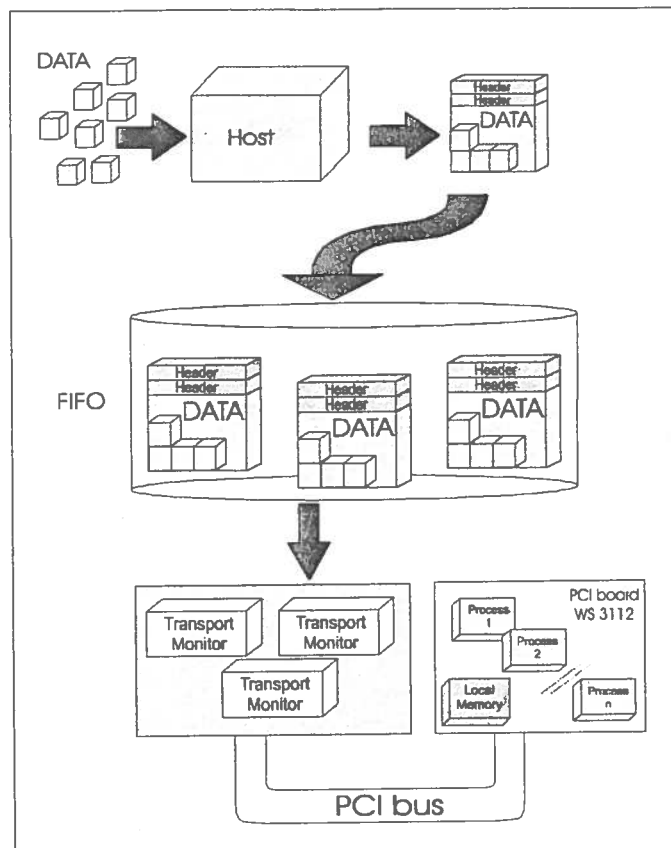


Fig. 2: Parallel architecture structure

3. HARDWARE AND SOFTWARE DESCRIPTION

In our prototypical study a simple version of the hardware platform (Fig.3) is used to test the implementation of the parallel architecture shown in Fig.2.

The host module is a Personal Computer based on a single Intel Pentium MMX Processor with the operative system Microsoft Windows NT 4.0. Two SHARCs mounted on the WS3112 board represent the computational units.

The two SHARCs have a peak rate of (120+120) MFlops at a clock frequency of 40 MHz (25ns execution-time per instruction).

It is important to point out that WS3112 is not a custom board but a commercial one; this produces some benefits such as low-cost and platform portability.

The WS3112 communicates via the PCI bus with the host system and is provided with 1,5 Mb of SRAM and a PCI compliant Master-Slave linking. All SHARC local bus devices are transparent to both the SHARC and PCI bus: for example the SHARC internal registers and memory spaces are transparent to the host PCI. The possibility of using more boards with a single host represents also an important characteristic for the implementation of our system design.

Any job for the control runs on hardware platforms: we can model our system such as a collection of tasks that compete with hardware resources. The software design follows the structure shown in Fig.1 and will be developed using different techniques based on the job model (computing and data-transport). The whole system will be developed using the C and C++, because of the compactness and modifiability of this programming language.

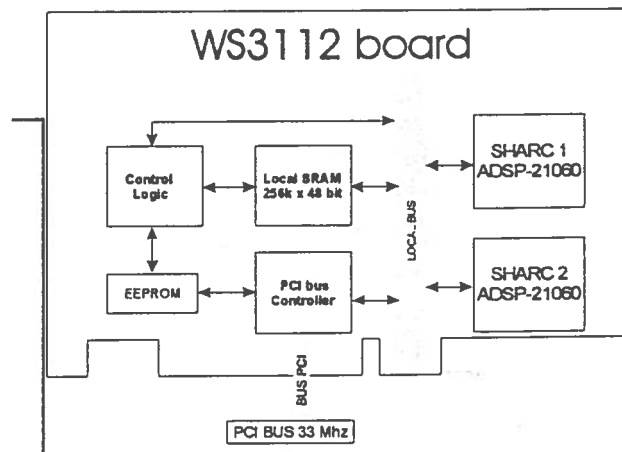
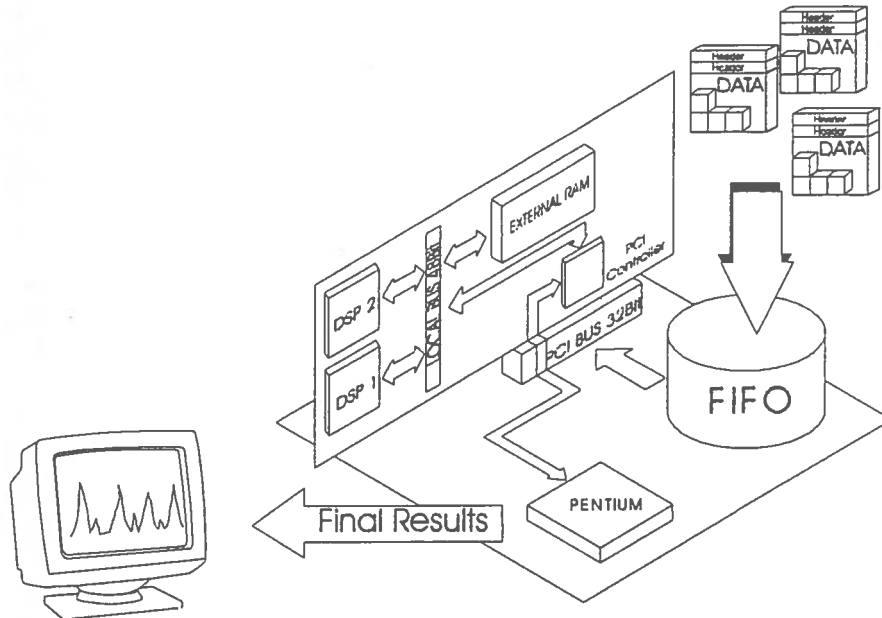


Fig. 3: The Hardware platform.

In the host, the tasks have to collect data from the acquisition system and manage the communication to present them to DSP tasks for the computation. There is also a job load distribution that determines the share of host computational resources. This happens when a system overload happens: in this case we generate a special task in the host.

To optimize the data transmission, the necessity arises of a transfer data protocol, to allow fast data exchange between the two systems. The use of a data packing system can be a valid solution; it is a flexible structure and may support any type and length of data.

So we have designed a packet composed of two sections:

1. Header, used to store some general information: Identification Label (necessary to characterize contiguous block) - Length of Data Block – Data Processing Mode – SHARC Identification Label (for a special type computation that requests the use of a specific DSP).
2. Data, used to store an array of data elements to be processed.

Once defined a common protocol, we need a queue to collect data packets. In our project, we use a pre-emptive FIFO allocated in the host dynamic memory. The host, to share a computation with the DSPs, has only to construct a data packet with the program instruction and add it to the FIFO. Special task will fetch it and route to the SHARC. Moreover pre-emption allows immediate execution for time critical jobs.

Transport monitors are tasks that manage software communication between the two systems. Hardware supports monitors by means of EPLD ALTERA and the SHARC PCI bus controller that allow packet routing on the 48 bit local SHARC bus through the 32 bit PCI bus. This hardware feature represents an arbiter among the host module and the computational units.

Programming library functions allow SHARC to access the PCI host bus and the local bus: only one request at a time can be satisfied. To handle communications, we need a semaphore locking the bus until an access is terminated. We used, in the software design, the eight 32-bit registers of the PLX Mailbox, provided by the PCI bus controller because they do not need to master the SHARC local bus.

One of registers is used to semaphore the bus access while two others are used to retrieve the SHARC data request the first for download and the second for upload. In particular the bit corresponding to the SHARC identification number is set to 1 in these registers.

The transport monitor tests the DSP status and when it is ready to receive or transmit data, the bus semaphore is locked and the transmission begins. Otherwise it waits until the bus is free.

Data communications are not only between the two systems. It is necessary to foreseen the exchange of data also between the two DSPs or between each DSP and the local memory of the board. The only possible solution is to manage a local bus master. A semaphore can be set in a DSP special memory location mirrored in all the DSPs. In this

way every one of the tasks running on a DSP sets a bit in this location. The same happens in all the other DSPs, avoiding the necessity of a local transport monitor.

4. PERFORMANCES AND CONCLUSIONS

As a benchmark algorithm the power-law formula [9] for charged particle identification has been employed:

$$\text{CIF} = (E + \Delta E)^X - E^X \quad (\text{CIF} - \text{Charge Identification Function})$$

where X is a real number depending on the kind of the reaction and of the produced ions (generally $1.5 < X < 1.8$). CIF is a function with a frequency distribution function characterized by separate frequency peaks, each corresponding to a differently charged reaction product.

In Fig. 4 the ratio between the total (i.e. computation plus transmission) time and algorithm complexity vs. algorithm complexity is shown. As it is possible to see, with complex algorithms (more than 10.000 machine cycles) the trend is constant, while the ratio increases when the algorithm complexity decreases. For our benchmark algorithm (about 1.200.000 machine cycles per packet), the used transmission protocol allows to optimize the system performances.

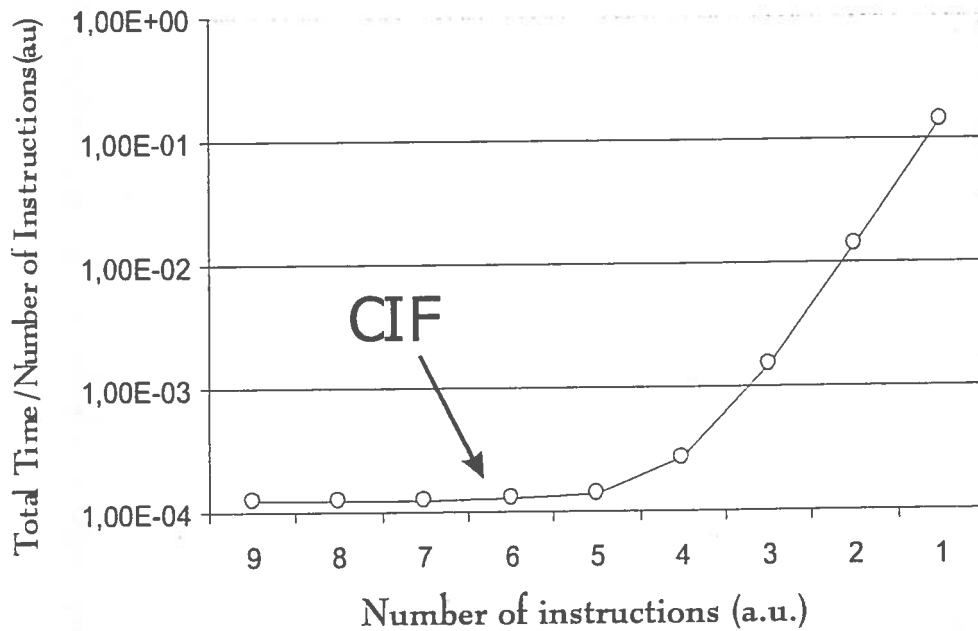


Fig. 4: Ratio between computation plus transmission time and algorithm complexity vs. algorithm complexity.

The architecture presented in this paper is nothing but a simplified version of that to be used for the system to design. In fact we have developed the present architecture for a two DSP board, while for the final release we foresee to use DSP cluster hardware. Anyhow the achieved structure can be easily extended up to a large number of processors, because of the expandability and modularity criteria used in the design.

REFERENCES

- [1] S.Aiello et al., "Chimera: a project of a 4π detector for heavy reaction studies at intermediate energy", Nucl. Phys. A 583, 1995, pp. 461-464.
- [2] S.Aiello et al., "Timing performances and edge effects of detectors worked from 6 in silicon slices", Nucl. Instr. and Meth. A385, 1997, pp. 306-310.
- [3] S.Aiello et al., "Light response and particle identification with large CsI(Tl) crystals coupled to photodiodes", Nucl. Instr. and Meth. A369, 1996, pp 50-54.
- [4] Wiese – Signal Verarbeitung GmbH Seelandstrasse 3 – 23569 Lübeck – Germany.
- [5] Analog Devices: ADSP 2106x SHARC™ User's Manual – Second Edition 1997.
- [6] S.Aiello et al., "Real Time Computing of Special Algorithms with a DSP₄ based board", Proceedings of the Eighth Euromicro Workshop on Real-Time Systems, L'Aquila, Italy, June 12-14, 1996, pp. 57-63, IEEE Computer Society Press.
- [7] S.Aiello et al., "Detector array control and Triggering" IEEE-RT 97, Beaune, September 22-27, 1997, IEEE Trans. Nucl. Sci. 45, 1998, pp. 1798-1803
- [8] S.Aiello et al., "Comparing Different Architecture for Real-Time Computing of Special Algorithms: a case study", Microprocessors and Microsystems 22, 1998, pp. 111-120.
- [9] F.S. Goulding and B.G. Harvey, "Identification of Nuclear Particles", Ann. Rev. Nucl. Sci. 25, 1970, pp. 167-240.