STATUS REPORT OF THE PSI ACCELERATOR CONTROL SYSTEM

T. Blumer, D. Anicic, I. Jirousek, A. Mezger and L. Sekolec

Paul Scherrer Institute, Villigen, Switzerland

Abstract

The upgrade of the control system for the three accelerators and their associated beam lines is now near completion. Some of the major improvements of the last upgrade include the replacement of 16 bit computers PDP11 with RISC machines in VME. Aspects of performance, timing and realtime response are analysed and results measured in the real system are presented.

HISTORY AND STATUS OF THE CONTROL SYSTEM

The control system history shows continuous improvements and several major upgrades which reflect the continuous development of the accelerators. The present state now represents the third generation in a long line of evolution of the control system.

The first system started with one "big blue", a very precious central computer acting as an intelligent operator. Settings could be saved and restored, probe measurements analysed and beam profiles captured for later analysis of the beam line, using the data processing centre of the institute. At this first stage manual control was performed directly through dedicated hardware. The second generation introduced 16-bit minicomputers with CAMAC for communication and input/output to the process. Three such minicomputers were used, one for each of the three accelerators.

The proposal for the present upgrade of the PSI accelerator control system was first outlined in 1989 in Vancouver [1]. In 1993 in Berlin [2] more details followed as the project advanced and first results were presented.

Limitations of the minicomputer system, the development in computer hardware and the ever present demand for extension in size and quality of the system created the motivation to undertake the latest upgrade, where three distinct areas of change and improvement can be identified.

Firstly, the minicomputers, old and obsolete 16-bit machines (PDP11/44 with RSX11M+) had obviously to be replaced. At the same time we intended to increase the I/O bandwidth by one order of magnitude and restructure the front end software for better maintainability and more flexibility.

Secondly, the introduction of a distributed system based on messages for communication. This would allow global access to all parameters throughout the whole system, independent of the hardware configuration and the computer where an application is executed. It would improve reliability by separation of applications from the input/output on the front end computers. This is a solution where the demand for additional I/O performance can be satisfied, independent of whether the number of connected devices is increased or the update frequency is raised.

Thirdly, workstations with graphical user interfaces for the operator consoles were to be introduced. These would replace the old hardware panels with more flexible workstation displays and provide modern platforms for large and complex applications including a common standardised application process interface for the user.

STRUCTURE OF THE PRESENT CONTROL SYSTEM

The basic system upgrade is nearly finished, if measured by the degree of introduction of the upgraded frontend computers. It is now completed to 80%. The system is now fully scaleable in size and performance.

The upgrade of the front end computers to the new machines was done in steps. The first step consisted of a reconfiguration of CAMAC crates in hardware and in the central database and the second of a test with the new front end. This was done in one service day. The new configuration was then taken into operation on a subsequent service day. No additional interventions were needed for the applications, except some bug fixes.

The last residual effects of the move will be cleaned up by late spring 1996.

HARDWARE DESCRIPTION

The front end computers are formed of HPrt743 (64MHz) RISC machines in VME. The accelerator devices are connected via CAMAC and local fieldbusses. The CAMAC crates are accessed via bit-serial loops (5MHz). The serial loop is controlled by a CERN serial CAMAC controller [3].

Workstations provide the operator interface at the control room console.

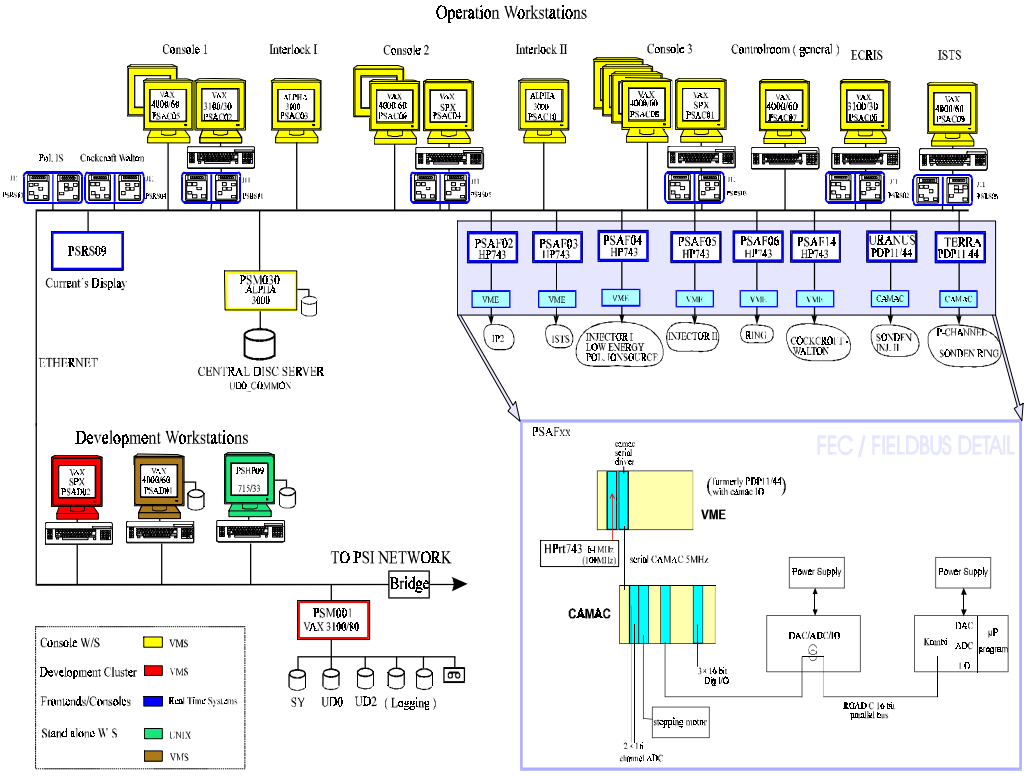
Dedicated nodes for knobs and touch panels are CAMAC-based PDP11 microprocessors (CES Starburst) connected directly to Ethernet.

A DEC AXP Alpha provides disk space for global data used in the system.

Additional nodes are used for auxiliary applications.

Development machines are separated from the machines needed for operation.

FIG 1: Present Control System Structure



COMMUNICATION AND SOFTWARE

The communication used in our distributed system is based on connectionless messages using the basic Ethernet protocol. This makes a loose coupling between FECs and workstations and ensures that no deadlock can occur. The format and the content of these messages are standardised on the basis of logical I/O access.

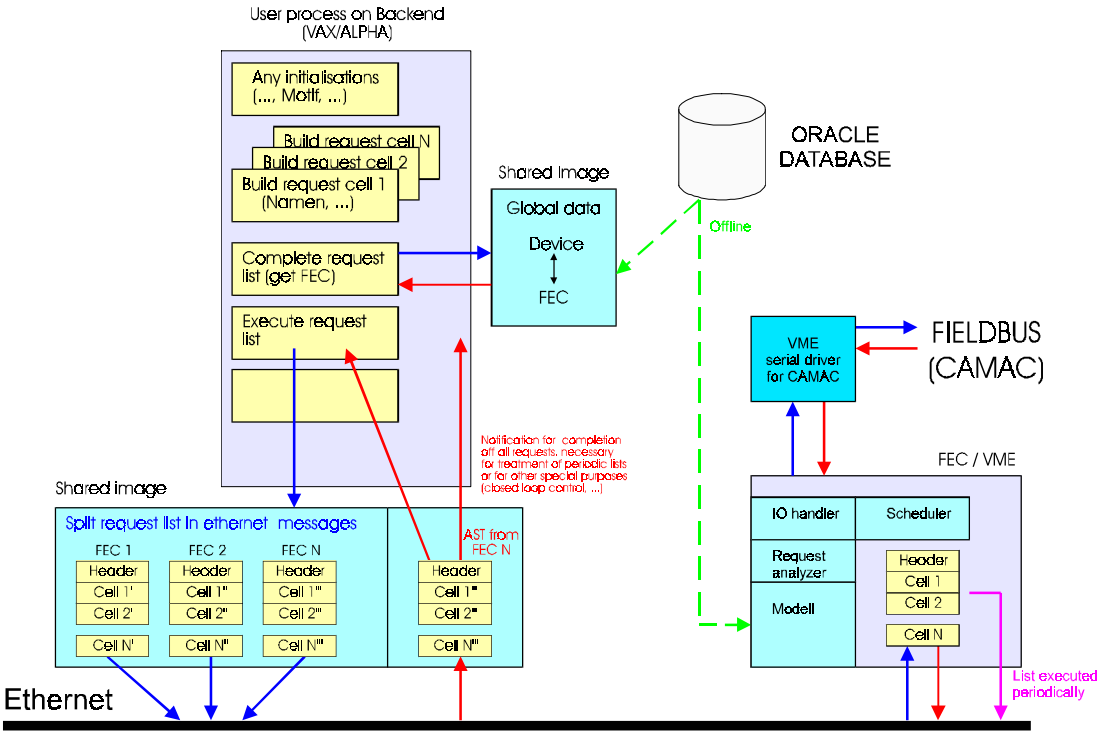
The messages on the network form an operating-system-independent interface between front end and requester, this was a pre-requisite for the replacement of the front end machines.
The front end software transforms logical I/O requests from the network into the appropriate hardware operations needed. This is done in a model representing the connected hardware. To construct this model the object oriented language Sather is used. The information for building the model is extracted from an Oracle database.

The user support in the workstations (open VMS) consists mainly of a general user application interface providing the logical access for I/O to the accelerator devices. This is realised in the form of a shared image.

X-window-based MOTIF is used as a graphical user interface for animated displays. Graphx, a PSI developed graphic package is used for static graphic applications.

Workstations, some of them forming a VMS-cluster, are used for the development of the control system software. The Oracle database resides on the VMS-cluster server. It centralises all the information for the description of the system.

FIG 2: Functional Layout of Control System HW and SW

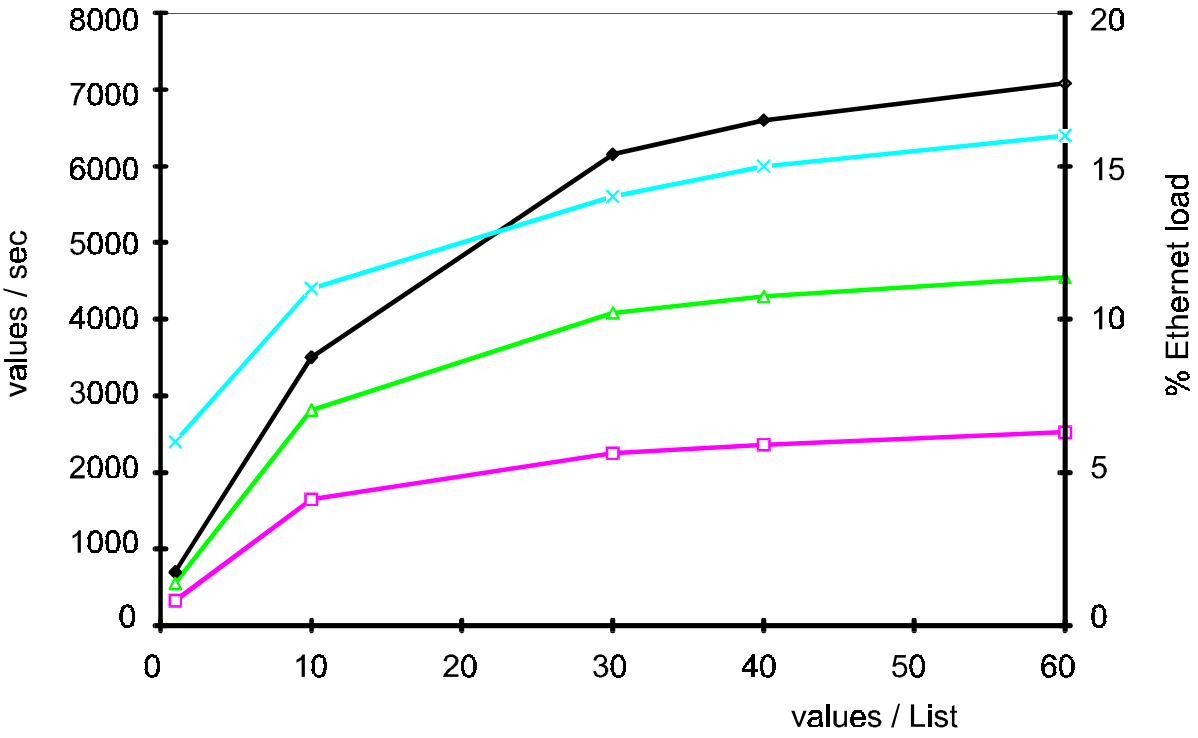


PERFORMANCE AND RESPONSE TIME IN THE SYSTEM

In our system the I/O performance of one front end is one of the dominant factors for the global system I/O bandwidth. Peripherals can only be connected to one front end machine, and if this machine is saturated its peripherals have to be distributed over more than one front end.

The maximum available I/O performance for requests from Ethernet was measured. This was done by loading one front end up to saturation using different forms of characteristic requests (cases 1,3 and 4). For efficiency reasons I/O requests are grouped in lists of up to 60 cells, in these lists one device or value is associated with one cell. Measurements were therefore made for 1, 10, 30, 40 & 60 cells per list. The front end computers measured are the HPrt743 operating at 64 MHz.

FIG 3: I/O Performance of one FEC over the Network



1) Closed diamonds in Figure 3. Shows the most efficient mode, used in displays for fast operator feedback. Here lists are dynamically predefined, a repetition rate specifies execution and I/O is performed without additional requests.

2) Crosses in the figure. For writing data or for I/O with varying destinations every requested list has to be sent to the front end. These figures are measured with one VAX 4000/60 acting as requester. The loop time is determined by the sum of the execution times in the two machines.

3) Open triangles. This is the same as case 2, but with four simultaneous requesters. To measure the maximum performance of the front end it is now loaded up to saturation.

4) Open squares. This is the Ethernet load corresponding to case 1. To decrease the Ethernet load a shortened protocol for reply messages with one third of the message length is also implemented.

From the above data the performance of the front end, as seen by the network, can be described approximately as the sum of a fixed overhead time per list and the time needed for the treatment of each cell within this list.

These figures now are:
- Previously defined list, Overhead 1.2 ms and 0.12 ms per cell.
- List defined at each request, Overhead 1.6 ms and 0.20 ms per cell.

The accelerators at PSI are operated in continuous wave mode, apart from the inherent 50 MHz structure there is no other time structure in operation, therefore the timing restraints mainly originate from control loops including magnet power supplies and from fast operator feedback and interactions. Therefore a response time between 40 and 100 msec seems adequate. Timing granularity for timed requests in the frontend is 20 ms.

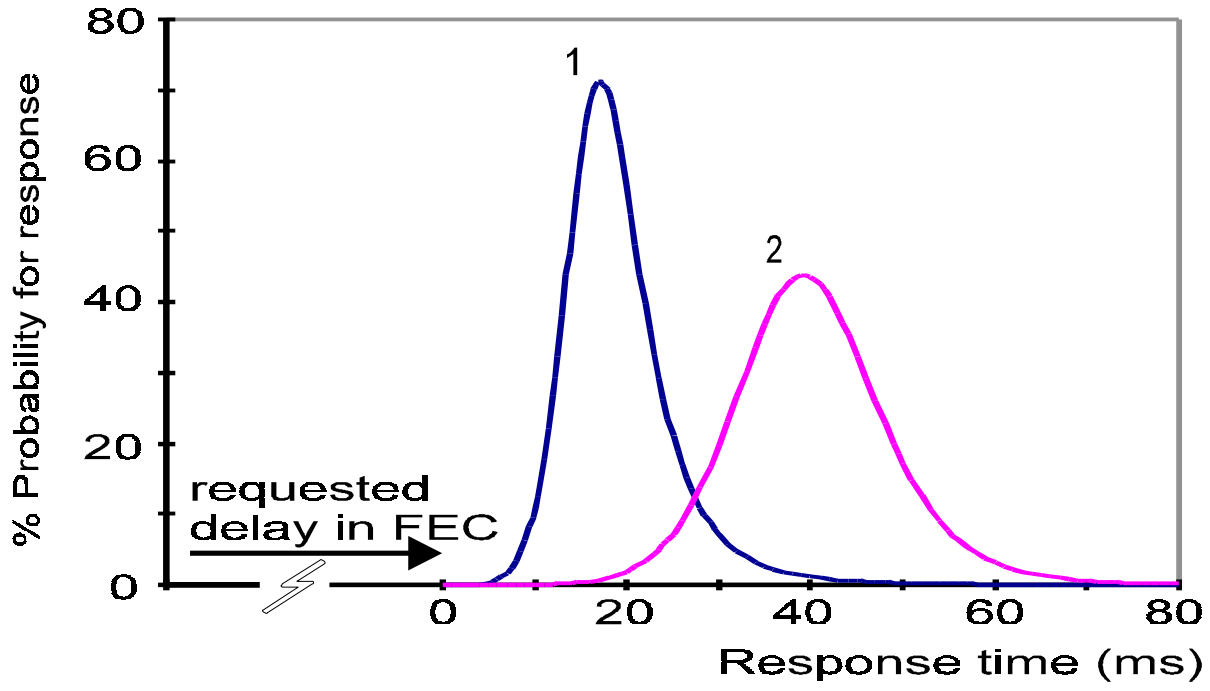
The variation of the turnaround time was measured for two typical MOTIF applications with repetitive execution in a loop. In both cases the workstation application sends one compound request (write, wait and read) to two FECs. The timing between write and read is done in the FEC. The loop restarts after reception of the read data and some processing in the workstation. This is a typical case of a control loop for multiply-coupled parameters.

In case 1 (see Fig. 4) all time-critical code was executed at AST-level. It showed that 90% of the response time lay within a window of 20 ms and that only 0.1% lay outside 40 ms.

In case 2 most of the time-critical code was executed at user level. It included signalling through X-windows. Here the respective figures are 30 ms for 90% and 50 ms for 0.1%. An additional delay from the increased code can also be noticed.

Both curves show a very small tail towards longer delays. This is caused by the non-deterministic nature of the workstation operating system, the network and the variation in load of the FEC.

FIG 4: Variation of Turnaround Time for repetitive MOTIF-Tasks on an Open VMS-Workstation addressing two FECs simultaneously



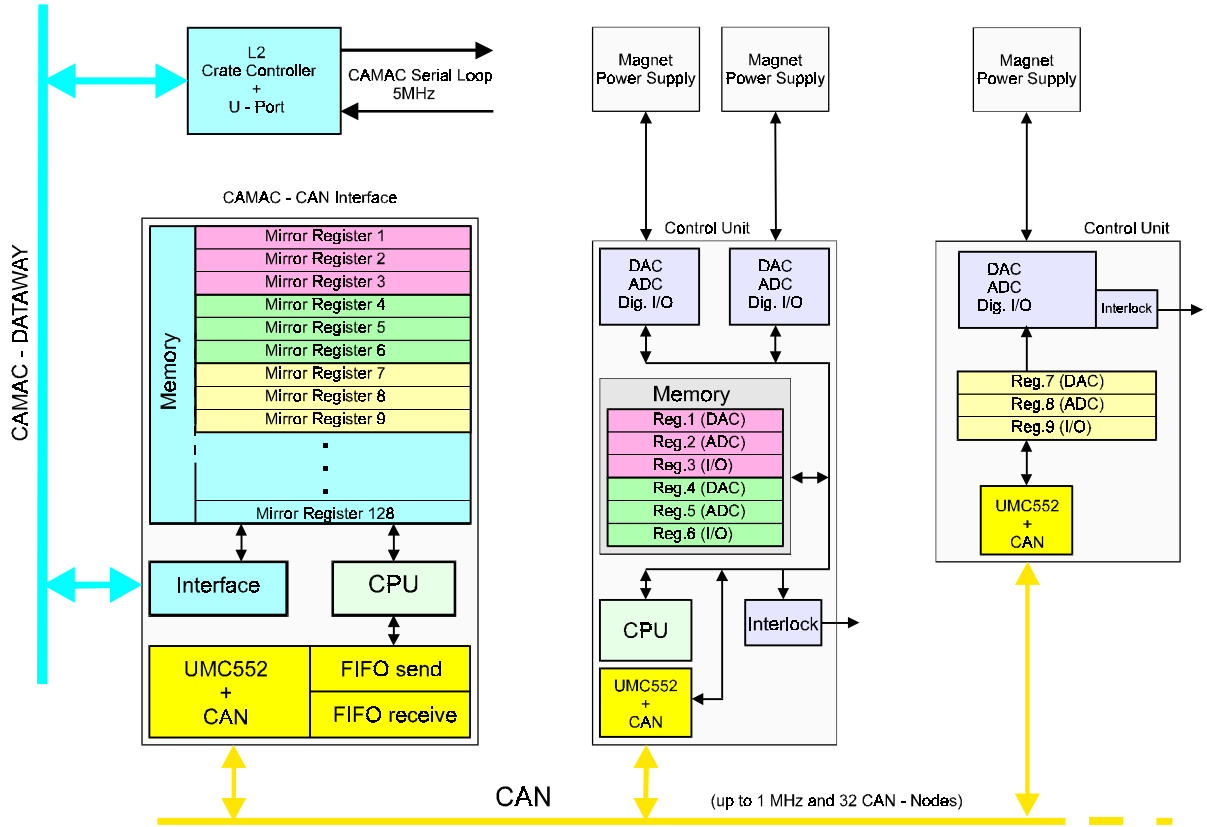
UPGRADE OF THE FIELDBUS

We still use a secondary fieldbus of a design that dates from the origin of the PSI accelerator complex. This parallel bus features 16 bit data, 8 bit function and address space. There is no error check, unless read after write is specified within an application. Galvanic insulation is achieved by transformer coupling of the peripherals. The bus is capable of 50 kHz, this corresponds to 100 kByte/s or 20 µsec per 16 bit data transfer for random addressing. Since errors are difficult to detect and the cabling is delicate and expensive, we are replacing this fieldbus in order to get increased reliability.

The present prototype for evaluation uses the CAN bus. Other modern fieldbusses using programmable microcontrollers in the remote stations could be used instead. With the CAN bus it is not possible and also not expected to achieve the same bandwidth we have now. The high bandwidth available now is mainly used for filtering. We estimate that this task is anyway better done at the source, prior to any multiplexing mechanism. So for our applications we intend to overcome the handicap of a lower bandwidth by normally pre-treating, usually filtering, the data at the source.

The cascading of two different bus systems is always time critical. In order not to impede on the specific CAMAC timing, a solution with a dual-port memory in the CAMAC/CAN interface is adopted. The registers of the control units are cyclicly read and then refreshed in the dual-port memory. For write functions the data is directly transferred to the control units using a write-through mechanism.

FIG. 5: Prototype for Fieldbus Upgrade



ACKNOWLEDGEMENT

We would like to take this opportunity to remember Z. Sostaric for his contribution to our system. He died this summer in a tragic accident.

REFERENCES

[1] T. Blumer et al., (ICALEPCS '89, Vancouver BC, Canada 1989) Nucl. Instr. and Meth. A 293 (1990) 54

[2] T. Blumer et al., (ICALEPCS '93, Berlin, Germany 1993) Nucl. Instr. and Meth. A 352 (1994) 150

[3] "A CAMAC Serial Highway Driver in VME", L. Antonov, V. Dimitrov, W. Heinze, CERN PS/CO/Note 91-22