IEEE Transactions on Nuclear Science, Vol. NS-26, No. 3, June 1979

SUGGESTED PRINCIPLES FOR THE CONTROL OF FUTURE ACCELERATORS

J. Altaber, F. Beck, M.C. Crowley-Milling, R. Rausch*)

Summary

The SPS at CERN has a control system which incorporates a number of features which were novel at the time of its design [1]. This system has been in use for the commissioning and operation of the accelerator and its experimental areas over the last three years, during which time considerable experience has been gained, and some modifications and changes have been made [2]. Some of the principles have been adopted for subsequent control systems for accelerators and other processes, with various modifications to suit the particular circumstances.

The purpose of this paper is to combine this experience with the predicted trends in microprocessor electronics to suggest a direction in which the design of control systems for large processes of this type may go. We will start with a brief description of the main features of the present system.

The SPS Control System

The SPS control system uses a number of similar minicomputers to perform all the duties required, some of them scattered round the various plant buildings, and others performing specialised duties at the control centre. Those concerned with the accelerator control are arranged in a simple star formation, with a store-and-forward message-switching computer at the node. The hardware to be controlled is joined to the computers through an interface consisting of a mixture of CAMAC and a specially developed serial-highway multiplex system (MPX).

Each computer is furnished with a re-entrant interpreter with text buffers at several interrupt levels, enabling any computer to run a program which becomes temporary master of as much of the system as it needs. The interpreter responds to the high-level command language NODAL, specially designed for programming a multicomputer network [3]. The hardware interface is driven by data-module sub-routines which render the individual accelerator elements available from the interpretive language in an obvious easy-to-use way. An individual data-module is responsible for handling each basic type of equipment and the data tables held it form the primary data base which is distributed over all the computers. Library facilities are available but are in general only used for holding programs and for back-up.

In operation, this system works by running control-programs written in NODAL in one of the console computers, with remote executes in whichever other computers are involved. In addition, most computers have scheduled surveillance programs, also in NODAL, which run at various intervals and report anomalies to an alarm computer.

The main reason for the adoption of the interpreter and data module concept was to try to overcome the so-called "software barrier" by opening up the possibility for the control programs to be written in a relatively simple way by the people who need them, the operations personnel and the members of the groups responsible for the hardware and services. Although this aim has been achieved with some considerable success,it has involved quite a large effort in systems programming, since each computer has to carry out many different tasks, some of which are extremely time-critical, and a satisfactory multi-tasking executive had to be developed.

The question now arises whether the advent of some of the new technologies could be used to improve the performance, minimise the software effort required, and reduce the cost of such an arrangement.

The Multi-multiprocessor Solution

One possibility,which is becoming attractive due to the continuing improvement in performance and reduction in price of microprocessors and their associated high density memory, is to make a multiprocessor system,each element of which is itself a multiprocessor. In other words,one could replace each of the multi-tasking minicomputers of the SPS system by an assembly of microprocessors, each of which performs one, single-stream,type of task. This would require a sufficient number of microprocessors, each with its own memory block, to carry out all the tasks of the minicomputer, including those of scheduling and communication between tasks, and communication with the rest of the system. A suitable physical arrangement would be to use a crate,into which the required number of micro-computers can be plugged, connected by a common bus which can be accessed for interprocessor communication under the control of an arbitration unit. Such a crate will be referred to as a 'compute' crate.

Communication with the rest of the system could be carried out by packet switching as in the SPS, but using a standard low-level protocol, such as HDLC, to take advantage of the special data-link driver chips being developed. A plug-in would be required in each compute crate having this hardware in addition to a microprocessor and memory to control and buffer the message flow. This could be done by an extended version of the normal microprocessor module. The message-switching computer at a node could consist of a crate of these units, and we will call this a 'cluster' crate. Incoming packets would be buffered in the memory of one microprocessor for transmission on the bus to the memory of another. Thus a truly autonomous operation could take place, occupying only those hardware elements involved in the transmission of a packet.

The somewhat complicated CAMAC + MPX interface system could also be replaced by similar crates, to take a family of plug-ins to interface to the individual types of equipment. These would be called 'control' crates, and would contain the same microprocessor plug-ins as used for the compute and cluster crates with an arbiter unit, where local autonomous action is required. They would also use the same standard unit for data link communication. The main difference is that, while a compute crate can be either master or slave, a control crate can only be a slave, and therefore cannot initiate inter-crate transactions. This difference is one of software.

*) CERN, F-01631 CERN CEDEX, FRANCE

0018-9499/79/0600-3278$00.75 © 1979 IEEE

Investigations which have been made show that it would be possible to make up a complete control network using one type of standard crate and the following plug-in units:

- microprocessor with memory and data-link driver,
- bus arbiter unit,
- microprocessor with memory (compute unit)
- mass storage unit,

plus the family of plug-ins to interface to the individual items of equipment.

By concentrating the design and manufacturing effort onto these few units, which would be made in large quantities, it is expected that the overall cost would be appreciably lower than for the present-day systems.

System software requirements in a scheme like this are limited to little more than the writing of an interpreter and bus driver subroutines, and to the writing of a free-standing driver for each kind of interface plug-in. Note that the operating system, which has a plug-in of its own, is reduced to little more than a scheduler, being aided by the hardware bus arbiter which queues requests for information transfers on the bus according to hardwired priorities.

Each of the three kinds of crate (cluster, compute and control) would use the same basic designs. Their functions would be defined by the choice of plug-ins and the programs in them.

While this entire system could be constructed out of CAMAC crates (for which most of the bricks alreadv exist), to make it economicallv interestinr it would reauire a norm which is both cheaoer and more suitable for the solution of the multi-processor arbitration Droblem. A norm of this kind, based on the CERN instrumentation module (CIM) and on a bus comDromise suitable for several microDrocessors on the market, has been devised at the SPS and is alreadv beinr used for several intelliRent control processes at the edee of our current network. Ooportunities therefore exist for develoDinr most of the necessarv modules as a bv-Droduct of the norrhal development of an existinr accelerator control svstem.

Network Topology

The SPS network started as a simple star formation, with a single node. A second node was added later to accommodate the experimental area controls. Although a star network looks simple as it is usually illustrated, the actual layout is no longer simple when applied to a large accelerator, as can be seen from Fig.1, with many of the data links following parallel paths and requiring several repeaters on the longer links.

For such an application, it would seem more logical to base the network on a ring formation, rather than a star, leading to the layout shown in fig. 2, in which a number of star clusters are connected together to form a ring. By suitable placing of these clusters, the need for repeaters may be avoided. In such a scheme, the traffic on the inter-cluster links would be the sum of the traffic on the parallel links of fig.1 which they replace, if one kept to the present 'central library' philosophy. Considerable flow reductions would occur if local node libraries were provided. The new technologies will certainly allow a mass-storage plug-in for use in the cluster crate. The datalinks on the SPS are not heavily loaded, so the present speed of transmission, 750 k-baud, would probably be sufficient. It would not be difficult to provide higher speed by the use of optical links, if they could be routed to avoid high radiation areas.

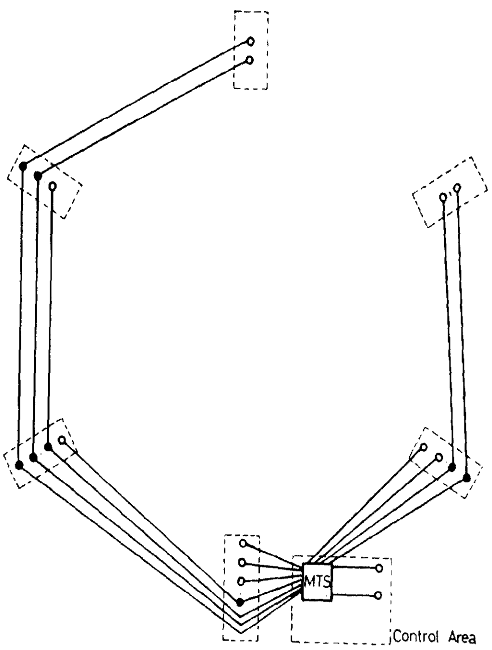

Fig. 1 Present 'Star network for SPS accelerator (there is a second star with another message transfer computer, MTX for the experimental areas).

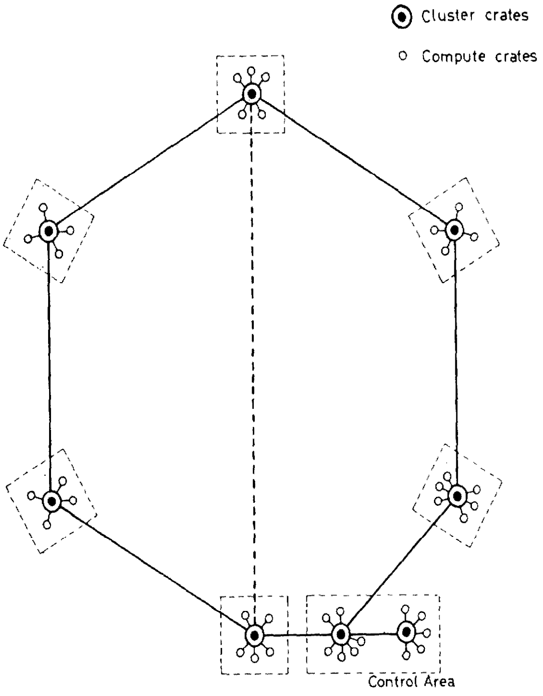

Fig. 2 'Ring' layout of clusters of computers

Such a ring structure already contains some redundancy, and other lines might be run across the ring, as shown dotted, if suitable paths are available, to increase the safety factor. However, for simplicity of the software in the cluster crates, the routing of individual messages between any two computer crates should be unique: only if a link breaks down should alternative paths be used, by reloading routing tables in the cluster computers concerned.

Conclusions

A control network composed of the three types of crates described can give full computer control and surveillance of a large process without using any mini or large computers. By using interpreters to provide programming facilities for applications, and a compiler with the same source language for writing data-module interface routines, the software effort needed, while considerable, is well-defined and independent of the accelerator design and layout. The operating system is a free-standing program in a dedicated plug-in and is considerably simpler than those on modern minicomputers despite the fact that true parallel processing can be carried out.

Provision can be made for the fullest use of local autonomous operations, thus reducing problems of timing and information flow. Many of the modules in a crate can work simultaneously, and therefore computing speed problems should be reduced.

An important advantage of this kind of control system is the commercial one. Microprocessors are beginning to be available from multiple sources, and single-stream applications of this kind require no complicated proprietary operating systems. Since in any case the microprocessors constitute a small proportion of the total price, one can design a system secure in the knowledge that the majority of components can be bought in a competitive market and that obsolescence is to a very much lesser extent than at present at the whim of the manufacturer.

References

1. J.T. Hyman, The Computer Control System for the SPS, Proc. IXth Intern. Conf. on High Energy Accelerators, 503, May 1974.
2. F. Beck et al, Operational Experience with the SPS Computer Control System, IEEE Trans.Nucl.Sci., 24(3), 1674 (1977).
3. M.C. Crowley-Milling, G.C. Shering, The NODAL System for the SPS, CERN 78-07, SPS Division, September 1978.