EVOLUTION OF CONTROL SYSTEMS FOR ACCELERATORS

M.C. Crowley-Milling
CERN, Geneva

1. HISTORICAL

The earliest accelerators were quite small, and their controls few, as can be seen from Fig. 1 which shows the 11-inch cyclotron built by Lawrence and Livingstone in 1932. Control was by switches and variable resistors and indication by a variety of meters, wired directly into the appropriate circuit. Where the control element had to be at high voltage, a loop of string was often used to operate it.

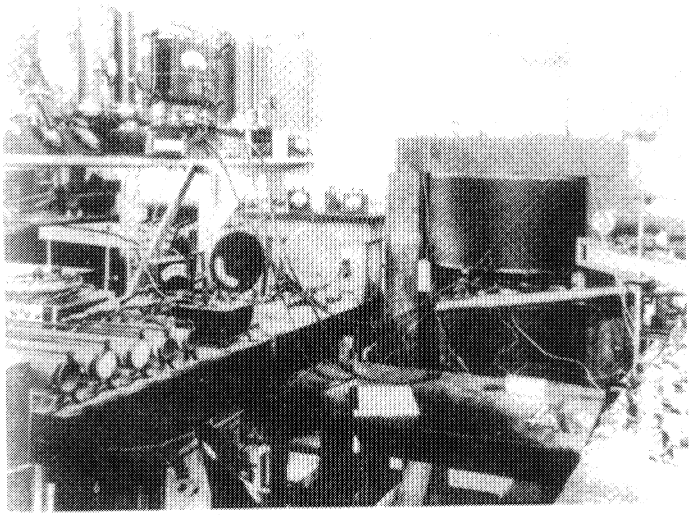


Figure 1 : The 11-inch Cyclotron at Berkeley

As the size and power of the accelerators increased, they had to be surrounded by heavier and heavier shielding, and the controls and indications had to be taken away from the accelerator itself and transferred to a separate control room. At first this was done by just extending the cables, keeping each control and indication separate, but the increasing numbers of values to be indicated led to some manual switching of instruments between different circuits to save panel space. In the 1950's, even though the cyclic accelerators had grown up to 200 m in diameter, the expense of taking cables for every control element to the control room was not excessive.

However, in the 1960's, projects of much larger size were being considered, where this would no longer be true. The first of these was the "2-mile" electron linac at SLAC. This is composed of 240 almost identical modules, each module having very many controls and indications of interest to the operators. The enormous number of cables which would be required to take all these to the control room was reduced considerably by remote multiplexing; that is switching one particular control or indication from each module in turn onto the same set of cables. Even with this, there was a cable termination room resembling a telephone exchange next to the control room, and a 1967 report told of a board of 1200 indicator lamps to show the status of various parts of the accelerator! Another project to make good use of remote multiplexing was the 10 GeV electron synchrotron at Cornell, which first operated in 1967.

Meanwhile, small computers had become sufficiently powerful to start finding uses in process control, where savings arising from the more accurate control possible were enough to offset the still quite considerable cost of the computers and associated equipment. This led to proposals for computer control of accelerators being put forward in the early 1960's. Some of these were quite ambitious, predicting automatic closed orbit control, completely automatic start-up and optimization, etc. to be only a few steps away. An example of this was the Russian proposal for a "cybernetic" machine. However, the reality was a little different, and most of the early applications of computers to accelerator control were restricted to data logging and multiplexing. The reasons for this are not far to seek. Process control uses a number of closed servo loops, and many years experience had been obtained with analogue multi-term controllers. The introduction of computers gave the possibility of supervising and linking these controllers to provide an overall optimization of the plant operation. What was needed was fairly well known, and once the system was operating satisfactorily, there were only infrequent needs for changes. On the other hand, much of the simple closed-loop control in accelerators, especially anything connected with the RF system, needs time constants of the order of microseconds, and the longer term, more complicated, stability problems were then not well understood. These factors, complicated by the constant changes and additions made to almost all accelerators, make it not surprising that it took many years to gain the benefits of full computer control for accelerators.

Pioneers in this field were Argonne, where a monitoring system was fitted to the ZGS in 1964, and Los Alamos, where plans for computer control for the projected big proton linac, LAMPF, were put forward in 1965.

This was the first big accelerator which was designed to be completely computer controlled from the start; no provision being made for overall manual control. By 1967 a "mock-up" of the system was in operation, before the start of construction of the accelerator, and much was learnt from this. In the same year, experiments with computer control were reported by Brookhaven and SLAC (for the beamlines). The following two or three years saw computers added to existing control systems at the CERN PS and at SLAC, and proposals for computer control systems for new machines such as the ISR and the big proton synchrotron to be built at what is now known as Fermilab. Since 1970 all large accelerators and storage rings have been designed to rely on computers for control from the start, although, as we will see later, there have been differences of opinion as to the best way of arranging the computers and their interconnection with the equipment to be controlled, both in the hardware and the software, and the interaction with the operators.

2. REQUIREMENTS FOR AN ACCELERATOR CONTROL SYSTEM

Although the requirements for a computer control system for an accelerator operating on a repetitive cycle for fixed target physics and for a storage ring where particles are injected, possibly accelerated and then stored for relatively long periods, may have detailed differences, this section applies to both types of machine except where a distinction is made.

The requirements can be divided into the following headings:

- Remote operation of equipment and acquisition of data;
- Open- and closed-loop control;
- Simulation and modelling;

- Surveillance and error reporting;
- Data logging and information distribution;
- Operator guidance and automation;
- Interlocks and access control;
- Interaction with experiments.

Let us look at each of these requirements in turn.

2.1 Remote operation and data acquisition

This is what I call the "remote arm and eye" requirement. It must be possible to operate every single revelant control, and read every relevant value, from the control position as if alongside the equipment. As the accelerator increases in size, the interpretation of what is relevant becomes wider. For example, with a small ring it may be satisfactory to leave some things which have to be operated infrequently, such as mains circuit breakers, under local control, but this could lead to considerable waste of time in a machine like LEP, where a journey of up to 10 km might be involved.

This facility uses the computer system as a remote multiplexer and is a pre-requisite for satisfying the other requirements, calling for the major part of the hardware investment. The advantage of using computers for this multiplexing is that it can provide the ability to set a single device, identified by name, to a value specified by the operator in arbitrary units, or to set a series of devices according to a mathematical function or stored table. It should also provide the ability to acquire the value of a variable, or an array of values, either "immediately", or at a specified time, and display the result in the form requested by the operator.

2.2 Open- and closed-loop control

Open-loop control involves the operator directly into the feedback loop: an observation is made, a calculation is carried out and, as a result, the operator can decide to perform some control action. The facilities required for the remote operation and data acquisition can be adequate for open-loop control, such as the observation of the beam position at a point, and the manipulation of a set of beam-bump magnets to minimize the deviation from the required orbit.

For closed-loop control, the operator should only be involved in setting the required conditions, the system then taking over to try to maintain them. As already mentioned, it was originally thought that this type of operation would be the main reason for adding computers to accelerator control systems, but in practice closed-loop control through the computer system is still not very common.

Closed-loop control systems can be divided into four main types. At one extreme, there are the very fast feedback systems, working in the timescale of the revolution period of the particles in the machine. They are almost always analogue systems which are only parametrized by the control system.

Next, there are the loops with time constants of the order of a millisecond and above, which can be handled by a single processor. In the past, many of these loops, such as power supply regulators, used analogue controllers, but with the advent of the microprocessors the trend is towards digital systems, which can also perform the switching, logic and supervisory functions for such a unit.

The third type, with time constants of a fraction of a second, can involve more than one processor, such as automatic closed-orbit correction or "Q" tracking.

The fourth type, with time constants of seconds or more, involves modelling, which is discussed in the next sub-section.

2.3 Simulation and modelling

With an accelerator, one can often arrive at an optimum operating condition by "knob-twiddling"; the successive variation of a number of parameters by small amounts, looking for local maxima. The partial loss of beam for a few cycles during this process is of little importance. The situation is quite different for storage rings, especially when the filling time is appreciable. It must be possible to change parameters such as the size of the beam at an intersection, involving simultaneous changes in a large number of parameters, without loss of circulating beam. The calculation of these changes is quite involved, and is best carried out using a computer simulation or model of the storage ring lattice. To give a response time for such changes which does not exceed a few minutes, the control system has to have sufficient "number-crunching" power, or a connection to a central computing system with a guaranteed response time. This method of using a mathematical model of the lattice within the control system was pioneered at the SPEAR storage ring at SLAC.

2.4 Surveillance and error reporting

The larger a system gets, the more important the surveillance becomes. All important parameters must be surveyed periodically, to detect changes in operating conditions. The main problem is to provide the operator with sufficient information to keep him aware of the state of the equipment, and make it clear when some action is required by him, without overwhelming him with information which is redundant or irrelevant.

In addition, as the equipment gets more complicated, self-test and diagnosis facilities must be provided, so that the operator is not only told what has gone wrong, but also how to rectify the situation.

2.5 Data recording and information distribution

The most important part of data-recording is the storing of "standard settings" at which satisfactory operation has been obtained, so that the same values can be set in the equipment when it is required to repeat the same operating conditions at a later date.

Data-recording is also required for producing operating statistics, etc., but a computer system makes it easy to log masses of data which is never looked at again, so there must be some careful selection, recording only that which is required for subsequent analysis. With a storage ring there is also the need for what might be called ephemeral data logging, the logging of parameters for a certain period in the past, which are then overwritten as time passes. This allows the "post mortem" analysis of the reasons for things such as a sudden loss of beam.

As well as recording parameters, it is necessary to distribute some of them to the experimenters and other persons interested, by television or other means, and to provide a connection to the experimental data acquisition computers, to allow the inclusion of machine operating conditions on the data tapes.

2.6 Operator guidance and automation

As operating procedures develop, it is important to record them in the control system, so that the operator can be given a clear statement what to do next at each step of a procedure. This can lead the way to automation or partial automation of the procedures, when they are clearly established and satisfactory. Because of the possibly serious results of a mistake, either in loss of beam or damage to equipment, it may be desirable to leave the initiation of automatic sequences to the operator, who may also be required to confirm whether an automatic sequence should continue at certain stages.

2.7 Interlocks and access control

Machine interlocks, those for the protection of the equipment, are being more and more carried out by software, but in the past there have always been some for which it has been considered essential to have a separate "hard-wired" circuit. This may still be so where very fast response times are required, but otherwise it may be possible to include them in the data transmission system, using modern redundant and error-checking circuits.

For a large machine the personnel access control system may not only have to cover the machine and experimental areas, but it could also have to provide facilities for controlling access to the remote sites and buildings, where it is not practicable to provide permanent guards at all times. The access control system must be "fail-safe", and up to now this has always called for a "hard-wired" interlock chain. The computer system should provide the control of the system and information on personnel movements.

2.8 Interaction with experiments

Until relatively recently, there has nearly always been a complete separation between the accelerator control system and the experimenters' data acquisition systems. However, the requirement to incorporate information concerning the machine operating conditions on the experimental data tapes, and for the experimenters to control some of the beam-line parameters, has led to some interconnections being provided between the two systems for the newer accelerators. In the case of storage rings, the experiments are closely integrated into the machine, and the requirements for inter-computer links are even greater.

3. HARDWARE CONFIGURATIONS

The requirements described in the previous section can be satisfied in many different ways, by different hardware and software configurations, and it is the balancing of the advantages and disadvantages of the different approaches that can make the design of control systems so fascinating, and can often lead to strongly held and divergent views between different teams as to the best solution to the problems. In this section we shall look at the different hardware configurations, although they cannot be considered entirely alone from the software considerations.

3.1 Computers

The first experiments in computer control of accelerators used a single computer, usually as large as the project could afford, but still not very powerful by today's standards. To start with, the computer just performed one or two tasks, but as more and more of the operation of the accelerator became dependent on the computer, it was sometimes found that a repetitive task would occupy the CPU for too much of the time. The relatively cheap minicomputers were then becoming available, and in some cases the problem was solved by adding a minicomputer to the system to take on that particular task. The minicomputer acted as a slave to the main computer and the interaction between the two was specially programmed. As time went on, the number of attached slave minicomputers grew, and typical examples are shown in Figs. 2 and 3.

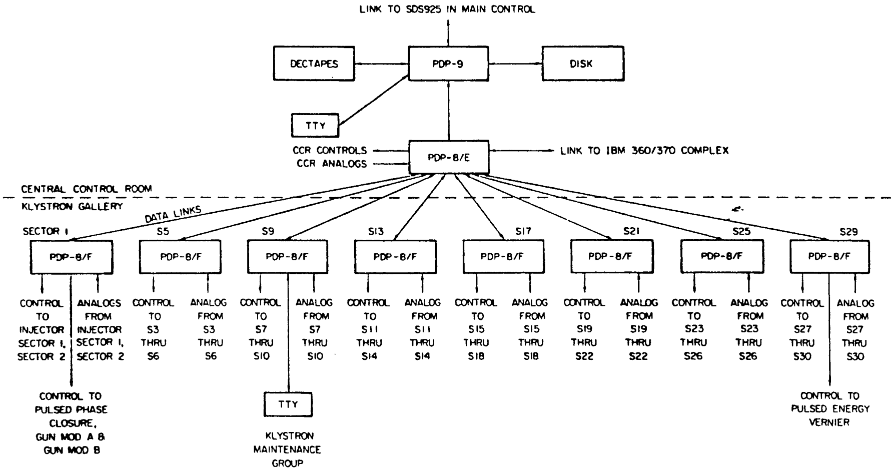


Figure 2 : Multicomputer system at SLAC (1974)

Fig. 2 shows the arrangement at SLAC in the mid-1970's. A PDP-9 had been fitted some years earlier to supervise the running of the accelerator, but, working through the original multiplex system, it could only operate on one section of the linac at a time. To overcome this, eight PDP-8 minicomputers were added as slaves, each dealing with four sections of the linac, and they were connected to the PDP-9 by using another PDP-8 as a communications controller. The PDP-8's had simple, fixed programs, to perform commands and acquire data, and all the computation was done by the PDP-9.

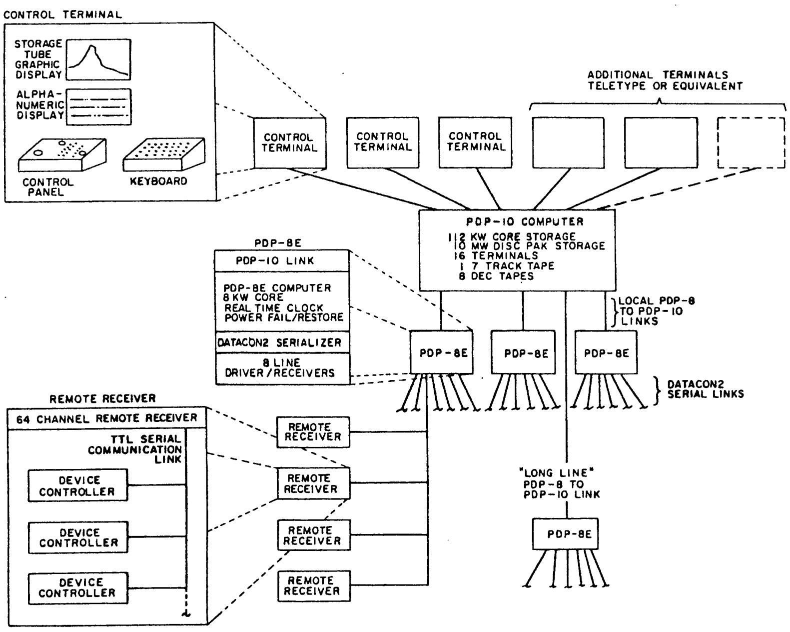


Figure 3 : Multicomputer system at Brookhaven (1974)

Fig. 3 shows the situation at Brookhaven at about the same time. Here the general rule of starting with a main computer and adding satellites was reversed. In 1965, two PDP-8's were added to the control system to do specific jobs as an experiment, and later others were added to do different jobs in an independent fashion. It was only later that a large central computer, in this case a PDP-10, was added to coordinate the whole.

Many other systems of this hierarchical type, where one or more large central computers have been used to carry out the control computations and interact with the operators, using a number of smaller slave minicomputers distributed around the equipment, acting largely as data concentrators, have been used for accelerator and storage ring control, the largest being that for the Fermilab Synchrotron, which originally used two Sigma-2 computers at the centre, controlling 12 MAC-16 minicomputers. This system was expanded as a result of demands for increased facilities, and is now being extensively redesigned to use more modern computers.

A different configuration was adopted for the SPS control system, and the reasons for this are discussed in a later section. The jobs that are normally carried out by a large central computer were split between a number of mini-computers, using an interconnecting message transfer system that allowed any computer to become temporary master, and provided for portions of a program running in one computer to be exported for execution in another one.

The advent of the microprocessor has resulted in even greater distribution of processing power, first into the interface and then into the equipment itself.

3.2 The interface to the equipment

The use of computers for process control requires some connections between the computer(s) and the equipment to be controlled, and the means of making these connections is generally known as the interface system, which involves both hardware and software.

The connections are of two basic types, analogue and digital, and they may be required for input and for output. Analogue inputs are used where measurements of voltage, current, temperature, etc. are required, and analogue outputs for power supply set points, displays, etc. Digital inputs can be used for sensing status (where individual bits can be used independently), or for reading values that have already been digitized, and digital outputs can be used bit-by-bit for on-off controls, or for setting values into equipment that can take them in digital form.

In the early days of process control, individual computer manufacturers made interface equipment to suit their particular computers, and the range of products available was small and they were usually relatively expensive. This applied particularly to analogue-to-digital and digital-to-analogue converters (ADC's and DAC's), so that often a single unit was used, the different inputs being connected to it in turn using a multiplexor, as shown in Fig. 4. Some signal conditioning is usually needed, to bring the input values within range. In the cases where analogue outputs were required, often stepping motors connected to potentiometers were driven from the digital outputs were often used, rather than DAC's, where either a separate one is required for each output, or long time-constant sample-and-hold circuits have to be used with a single DAC multiplexer for a number of outputs.

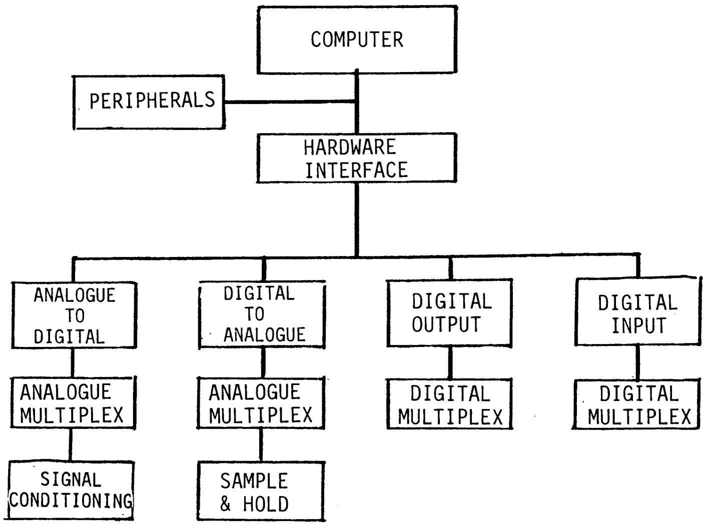
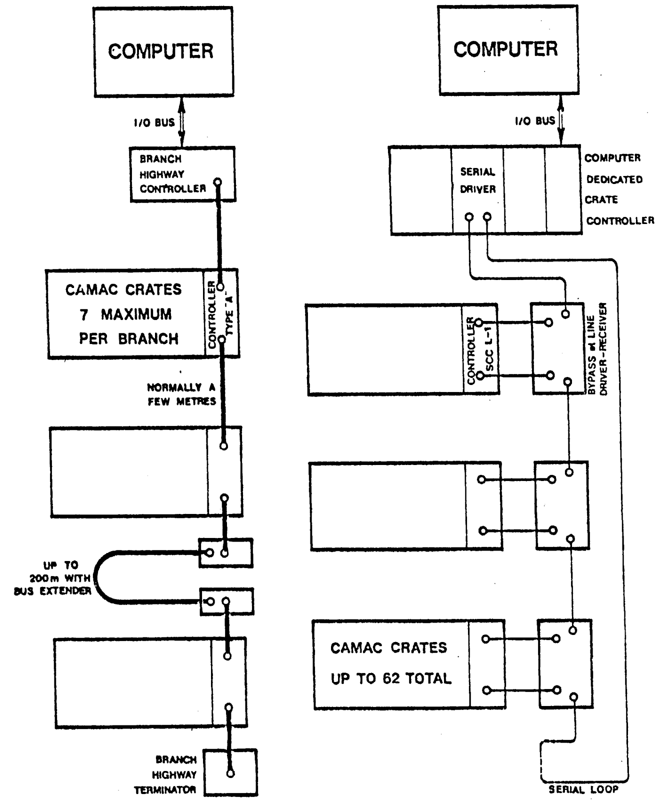


Figure 4 : A typical Process Control System layout

It is to the manufacturers' advantage to get his customers tied to using equipment that only operates with his computers, but it is to the customers' advantage to be able to build up his system from a wide range of modules that are not specific to one manufacturer. One such system has been devised for nuclear electronics, CAMAC, and this was soon taken as a means of interfacing for accelerator control. The original CAMAC system consisted of a crate and bus system, with a controller in each crate which interfaces between the bus and a CAMAC branch highway. As shown in Fig. 5(a), up to seven crates can be connected into a highway, and the only computer-specific item is the Branch Highway Controller, which interfaces with the computer I/O bus. Various analogue and digital input and output units have been available for many years, as well as more sophisticated units such as stepping motor controllers and display generators.

One of the disadvantages of the original CAMAC system, which showed up as soon as the distances involved in accelerator control became large, was that the parallel branch highway, involving some 130 wires, could normally only be a few metres long, so all connections had to be brought close to the computer. This disadvantage was overcome by the development of serial CAMAC, which uses a single loop round which the information is sent in serial form. As shown in Fig. 5(b), up to 62 crates can be connected into the loop, which is driven by a special serial driver unit, normally housed in a CAMAC crate which has a computer dedicated crate controller. The only limit on the size of the loop is set by the speed and response requirement, and slow speed loops have been operated intercontinentally.



a-BRANCH HIGHWAY b-SERIAL LOOP

Figure 5 : CAMAC systems

While CAMAC has been used as the interface medium for many accelerator systems, it has some disadvantages that have led to some other systems being developed, particularly for the larger accelerators. One disadvantage is the cost; almost any general purpose standard system will be at a cost disadvantage compared with a system specially designed for a restricted range of users, when large numbers are required. At the time of the design of the SPS, before serial CAMAC was standardized and developed, the restrictions on distance between the computer and crates would also have had an influence on the cost of cables.

Another disadvantage was that the modules available were not well matched with the requirements for the equipment to be controlled. For example, the control of a power supply might require four on/off contacts, a digital word input to set the level, six status bits to be acquired, and an analogue measurement to be made. With the available CAMAC units, this would require one quarter of a 16-contact output module, half a double digital word output module, one third of a digital word input module and one sixteenth of an analogue multiplexer and ADC module. This means that a junction box must be provided for the cross-connections between modules and cables to the equipment. Some form of junction box would be needed even if modules had been available which were better matched to the equipment, as the miniature plugs and sockets which have to be used on CAMAC modules, owing to the narrow panel width, require miniature cables which are only suitable for short links. Another disadvantage is that the common earth normal to the CAMAC system can cause coupling between different pieces of equipment connected to the same computer.

Therefore it was decided to design a special serial multiplex system for the SPS. As explained below, the controller for this system is a CAMAC module, and up to four controllers, each managing a serial branch, can be fitted to a CAMAC crate. A normal branch can be up to 300 m long, but this can be extended to 1.5 km with special drivers, and stations can be connected in at any point in the branch up to a maximum of 32. Each station consists of a crate with its controller and space for up to 11 modules. These modules are of the 3H NIM size, which normally have panels twice the width of CAMAC modules, and so allow the use of full-size plugs and sockets. Where the quantity required makes it economical, special modules are tailored to the requirements of the equipment to be controlled. For example, special modules for the 1000 ion-pump power supplies, for the 400 closed-orbit and stop-band correction supplies, for the nearly 300 function generators, etc., have the requisite numbers of control contacts, status-bit sensing, digital commands and acquisitions, and analogue acquisitions to satisfy all the control requirements of the particular equipment. This allows a one-to-one correspondence and a single cable between module and equipment. For the rest of the cases where the design of a special-purpose module is not justified, a few "general-purpose" modules have been provided, with varying mixes of the control and acquisition functions. Simple requirements can almost always be satisfied by a single module, and it is very rare to require more than two to control any type of apparatus.

Another machine to use a special purpose interface system is PETRA, for which the serial-highway system SEDAC was developed. This also has modules which match the equipment requirements where large quantities are involved.

While these special systems can be cost-effective for large control systems, small systems cannot justify the development cost, and then it is often best to use a well supported standard interface system.

The incorporation of microprocessors into the interface equipment has helped to reduce the load on the control computers. The first major implementation has been the Auxilliary Crate Controller, ACC, in CAMAC. In many cases, a single control action on a piece of equipment can involve a number of CAMAC cycles, each of which involves the process computer. The ACC can take over some of this load, which can become considerable when repetitive events or the successive setting of equipments to tables of values are involved. Another typical use for an ACC is to run surveillance programs, so that it is only necessary to interrupt the process computer when a fault occurs.

The stage beyond this, the one which has already been tried experimentally at a number of places, and one that will certainly be used extensively for future machines, is to fit microprocessors into the equipment itself, to deal with a large part of the fixed program operations, such as start-up, stabilization, hysteresis correction and surveillance for a power supply. In such a case, the communication between the power supply and the process computer can be in the form of messages, rather than in the elementary command/response mode used by such interface systems as CAMAC. Such messages can be transferred by a relatively simple multi-drop communications highway system, and such systems will be considered later, when we discuss the proposals for the LEP control system.

3.3 The interface to the operators

A user tends to judge a control system by the ease and convenience with which it enables the required operations to be carried out. Therefore the interface between the control system and the operator is an extremely important part of the over-all design.

There are many ways of presenting information and providing for operator actions, and it is possible to design different systems, using various combinations of these, all of which will satisfy the basic requirements of the control system. However, the convenience of operation will vary with the different combinations and, although some assistance can be obtained from the science of ergonomics, this is a region where personal preferences and prejudices exert a strong influence.

As shown earlier, before the introduction of computers, most accelerator control rooms had rows of racks with separate controls and indications for each piece of equipment, with some of the most frequently used controls duplicated on a control desk.

Even with a computer system, the control room can be made to look much the same, by using separate panels with matrices of parameter-insertion and program-request buttons, and separate indicating devices for each major piece of equipment, etc. This approach is both inflexible and expensive for a large accelerator, but it does have the advantage that an operator soon gets to know the physical position of each control, and can scan the state of various parts of the accelerator by a quick glance round the racks.

The first applications of computers to accelerator control systems were for data logging and the performance of a few control actions, and the operator interaction was often limited to the use of a teletype to enter a program name and a few parameters, which then churned out metres of printed columns which were hardly ever looked at. The next stage was to add program call buttons and display devices based on the cathode-ray tube, the most versatile of all the output devices presently available, which can also be used as an input device, if equipped with a light-pen, or other means of sensing operator actions. Values were sometimes fed in by means of thumb-wheels, but often knobs connected to incremental encoders were used to give the operator the feel of varying an analogue valve.

As the number of applications programs increased, the number of program call buttons became uncomfortably large, and a number of different ways have been used to overcome this difficulty. At Los Alamos, a single set of buttons was used with a set of interchangeable cards which fitted over them. Each card had a different set of program titles, and a coded cut-out at the bottom of the card informed the computer which card was in place. At SLAC and the SPS, button images are drawn on the face of a cathode-ray tube by software, and means used to detect the presence of a finger touching the button. Such "touch-buttons" or "soft-buttons" are becoming more and more popular, and a number of different means are used to detect the position of the touching finger; ultrasonic and infrared light beams crossing the screen, cross-wires which touch under finger pressure, transparent capacitive and resistive networks over the screen, etc. The capacitive system is used in the example shown in Fig. 6.

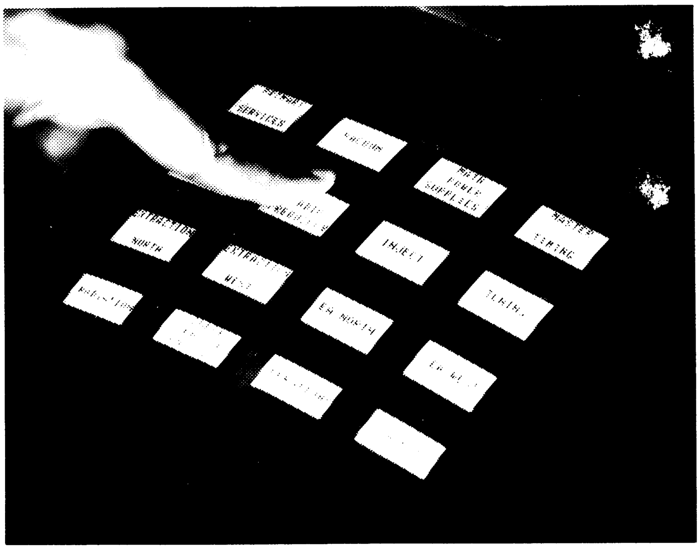


Figure 6 : The SPS touch-button screen

The quasi-infinite number of different button titles possible are usually organized in the form of a tree, with a trunk program giving a first choice of system to be acted on. On making this choice, the button labels change to give a choice of subsystem, and so on. The final choice is then sometimes made on a display list or mimic diagram, using a tracker-ball or joystick to position a cursor on the screen. This latter method was not always popular when a single central multiprocessing computer was used, as the cursor's progression across the screen was sometimes irregular as the processor jumped between tasks of varying priority. This problem has been overcome with separation of tasks between different processors and the availability of sophisticated display controllers which do not require frequent interaction with the main computer.

4. THE SOFTWARE SYSTEMS

The software for almost all the earlier process control systems were written in the assembly language of the computer concerned and this practice was followed for the early accelerator control systems. Computers were relatively slow and storage expensive, so it was important to produce as compact a code as possible.

4.1 Operating Systems

At first, operating systems for small computers were relatively simple and often had no provision for "real-time" operation, so in most cases special executive programs were written to provide either a time-sliced or interrupt-driven multi-tasking environment for the applications programs.

When slave processors were added, these executives were extended to cater for the elementary communication between them and the master computer(s), and simplified executives were written for the satellites, which in many cases were just performing I/O operations, with some treatment of the data acquired. In the 1970's the manufacturer's operating systems for minicomputers improved, and started to provide for "real-time" foreground tasks as well as batch-like background operations. However these multipurpose operating systems have to make compromises between the range of facilities they can offer and the speed of response to a change, since usually the bigger the operating system the more frequent the disc accesses have to be. This becomes even worse when networking software for a multicomputer system is involved. As a result of this, we can still find "home-made" executives and communications software in use at a number of accelerators, even though manufacturer's systems are available for the mini-computers used.

4.2 Applications Software

As mentioned above, the early control programs were written in assembly language, usually by professional programmers. It wasn't until about 1970 that there was mention of the need for some easier method of programming, particularly for machine physics investigations, so that the physicists and engineers concerned could make changes as required, rather than have to specify everything to a programmer. The successful demonstration of the use of an interpreter for programming a beam-line control system at the Rutherford Laboratory triggered off interest in this method of programming and part of the Fermilab system used an interpreter initially. However, it was not until the SPS came into operation that an interpretive programming system was fully integrated into a multicomputer control network, as described below. Meanwhile, the more conventionally inclined were recognizing the advantages of using a high level language, even if it is compiled, and a number of systems began to be programmed in FORTRAN, chosen probably because it was the most popular language for scientific computation, even though it had shortcomings in the process I/O area. One of the few exceptions to the choice of FORTRAN as a high level language was the ISR control system, which adopted CORAL, a process control language developed for the Ministry of Defense in Britain. Nowadays, PASCAL and its variants are being proposed for accelerator control, and the PS new control system is partly programmed in P+, a PASCAL-like language specially designed for the purpose.

4.3 Data Bases

In the earliest control systems, which had relatively few programs, information as to the I/O addresses of the various connections to the equipment, conversion factors, etc., were written into the programs, but it was soon found more convenient to gather this sort of information into a data base, where it could be accessed by any program, so that modifications to the equipment could be catered for by changing the entries in the database, rather than having to modify the programs.

With single computer systems it was logical to have a single central database, and this could contain not only the semi-permanent data of the type mentioned above, but also such things as the last demanded value for a variable, maximum and minimum allowable values and tolerances, etc. The inclusion of the latest measured values into the database was the subject of some controversy. If a program requires a measured value, the normal way is for it to call a subroutine which calls for the measurement to be taken, the reading to be processed and the results to be read into the program. This takes some time, especially for high accuracy measurements where ADC's have an appreciable settling time. The alternative approach is to call for repeated measurements of all values of interest and the reading of the processed values into the database. A program wanting a value has then to access the database, with negligible loss of time, but how close the value it gets is to the actual value at the time of request depends on the frequency of scan and the rapidity with which the variable changes. The overhead of constantly updating the database must also be taken into account when assessing the relative merits of the two methods.

With a multicomputer system the balance can change, depending on whether the database is kept as a single entity, or distributed amongst the computers. With a central database, to update it continuously with new values, whether needed for a program or not, involves a considerable traffic on the data links connecting the satellites to the main computer, much greater than would be involved in transmitting only the values that were needed for the programs running at that time. If, however, the appropriate parts of the database are distributed amongst the satellite computers, they can be continuously updated at the expense of satellite CPU time only, the data links traffic being limited to calls from the central computer to access the satellite data bases.

Opinion is not uniform on this subject, and a recent large control system, that for the PEP storage ring, has been built with a single database which is constantly updated by messages from the satellite computers. The original message transfer system was found to have too high a software overhead, and some difficulties due to congestion were experienced, but a simpler software protocol has provided a system that can cope with the required traffic.

4.4 Equipment Drivers

Another area where there have been divergences of opinion, closely allied to the previous one, is the approach towards the I/O routines known as equipment drivers. Largely due to the early forms of interface equipment, the drivers are often written as a general purpose packet which performs digital and analogue input and output to a series of I/O multiplexors. The type of equipment concerned is only specified in the central data base, where, for instance, a magnet current is identified as, say, Device X; analogue input No. 2.

The alternative approach is to think in terms of the elementary items of equipment to be controlled; power supplies, pumps, etc. and design separate equipment drivers tailored to the requirements of each different type. For best effect, this also requires that the database be broken into separate modules, which are linked to the corresponding driver, making an entity known as a data module, or equipment module, which one addresses according to the action to be performed, and the form of the action, whether digital or analogue, is not of concern to the programmer. We will expand on this subject later.

5. RELIABILITY

When computers were first applied to accelerator control systems, the question of reliability was often to the fore. Computer hardware was not too reliable, and software was worse; primitive operating systems allowed programming errors to crash the system, and fault finding was time consuming. Despite this, the accelerator down-time attributed to the computer system was not excessive, and this was largely due to the rôle the computer system was asked to play; to supervise and to make changes. In most cases the hardware was designed to retain the last commanded values until fed with fresh ones, and cyclic devices to continue cycling until told to change. Thus the computer could stop for a short time, be reloaded and start again without a major effect on the operation of the accelerator. This no longer applies when closed-loop control through the computer is involved, or where computers are used as direct function generators.

As computers took over more jobs, their reliability became more important, but never to the level where extreme measures were justified. An accelerator normally has a down-time of about 10% of the scheduled running time, due to failures of different types of equipment, and if the down-time due to computer system failures is not greater than about 1% of the scheduled time, extra expense to reduce this figure is rarely justified.

Quite a few systems using a large central computer had a second one as a spare, with provision for switching the I/O connections to the spare when necessary. Although often obtained on the excuse of improving the reliability, in some cases the real reason for the second computer was to allow program development to be carried out without disturbance of the operation of the machine.

One of the difficulties with a changeover to a second central machine is that, with the loss of the data base in the failed computer, it may be necessary to go through a complete start-up procedure after switching over. This problem can be reduced by having the computers share a common database, but this only contributes to the reliability if the database hardware and software is more reliable than the computer.

This is the basic problem with duplication of equipment. It is only effective to duplicate a part of the chain of command between the operator and the equipment to be controlled if that part is clearly less reliable than the rest. Otherwise, it is necessary to duplicate the complete chain, and that is only effective for clear failures, where it is obvious which chain to retain in use. Where the two chains give different, but plausible, results, a third chain is needed, with two out of three voting, as is used for space satellites and aircraft landing systems. Even then, one has to rely on the single voting device!

6. A CASE HISTORY OF CONTROL SYSTEM DESIGN

In 1971, the construction of the Super Proton Synchrotron (SPS) was authorized at CERN, and it may be of interest to follow the evolution of the design of the control system for this machine.

6.1 The Accelerator

The SPS was designed to take protons from the existing Proton Synchrotron, the CPS, at an energy of 10 GeV, to accelerate them to a maximum of 400 GeV, and to provide beams of protons and other particles up to this energy for experiments.

In a proton synchrotron, the protons are constrained to circulate many times round a constant path by a system of magnets and are given an increase in energy by radio-frequency fields each time they go round. Thus very high energies can be given to the protons without requiring excessive voltages to be generated. Because of limitations in the magnetic fields that can be used, the higher the energy required, the larger the proton circulation path, or orbit, has to be. In the case of the SPS the orbit is approximately circular with a mean diameter of 2.2 km. From a number of considerations, it has been necessary to build the machine in a tunnel, about 40 m underground, in the stable molasse rock. Building the machine so far underground has the disadvantage that it would be very expensive to provide a large number of access points. The system of magnets used to guide the protons has a superperiodicity of six, and so six access pits have been provided. All the equipment needed to supply and control one sextant of the ring tunnel is housed in a building at the top of the pit and all the pipes, cables, etc., have to go down the pit and along the ring tunnel.

After acceleration, the protons have to be deflected out of their normal orbit by special equipment and directed upwards to experimental halls on the surface. The beams of protons have to be split and directed on to targets and the secondary particles thus formed have to be collected and directed towards the experiments. The supply and control equipment for these beam lines is housed in buildings near the experimental areas.

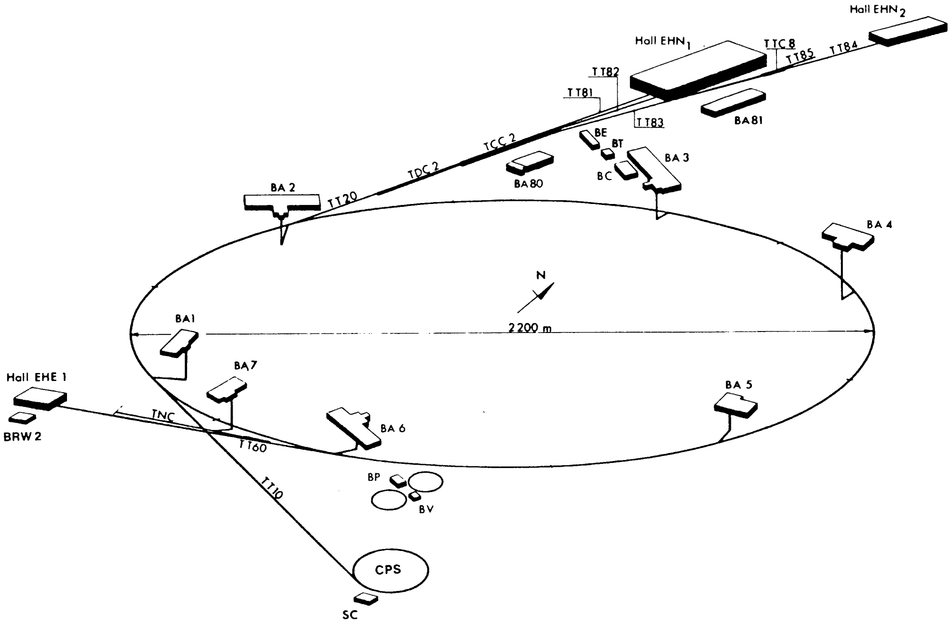


Figure 7 : SPS site layout

A schematic view of the layout is shown in Fig. 7. The main equipment to be controlled is situated in buildings BA1 to BA6 for the accelerator, BA7 and BRW2 for the West Experimental Area, BA80 and BA81 for the North Experimental Area, and BE for the substation. The main control room BC is situated near BA3 and the main laboratory buildings.

From a functional point of view, the accelerator can be divided into subsystems such as main magnets and power supplies, vacuum, radio frequency, injection, extraction, beam lines, etc., and the over-all control system must be designed to deal with them as such at the higher levels. However, each of these major subsystems is composed of parts which may be common to several subsystems, such as power supplies, water-cooling systems, beam instrumentation, positioning systems, etc., and so these are the items that the control system must deal with at the lower levels.

6.2 Preliminary Considerations

Before the construction of the SPS was approved, a Design Committee, consisting of accelerator specialists from CERN and outside laboratories, was set up. One of the subcommittees was concerned with the control and instrumentation.

From the start, it was accepted that for such an accelerator, covering such a large area, extensive use of multiplexing would have to be made to cut down the number of cables required, and that computers would have to be used to control this multiplexing and simplify the work of the operator. It is not surprising that the subcommittee, reflecting the situation at other laboratories at that time, proposed the use of a multicomputer system, with a large central computer controlling a number of minicomputers as satellites. However, as a result of the experiments at Rutherford Laboratory, it was suggested that an investigation should be made as to the possible use of an interpreter in such a system. The subcommittee also recommended that CAMAC should be used as the primary interface.

When the project was approved and the team to be responsible for the control system had been set up, these aspects of the design were looked at critically and it was seen that decisions made in one area would affect the choices to be made in another area, so that the problems had to be looked at as a whole. The most important and far-reaching decision taken was to use an interpreter for the main applications programming of the system.

6.3 The Interpreter

It was realized that the provision of adequate applications programs for the operation of the SPS would involve a very large effort in a relatively small time. The groups concerned with the construction of the various components of the SPS were not all convinced of the advantages of computer control, and specification for control programs would only be thought about after the equipment construction problems had been solved. Thus to have followed the usual pattern of the equipment experts having to write detailed specifications for professional programmers to turn into code, debug and then modify as the experts changed the specification, would have required an army of programmers. To get round this, and to allow the inevitable modifications to be introduced in the least painful way, it was thought essential to make provision for the accelerator experts to be able to write most of the application programs themselves. In addition, sufficient protection must be provided in order to avoid, as far as possible, the effects of an error in an application program spreading outside that program. Such facilities can be provided by the use of an interpreter, which is a program which executes instructions in a high-level language directly, rather than translating them into machine code and then executing the compiled program. It can be thought of as an on-line linker, linking together and executing small machine-code modules for each statement of the source program. Single line statements can be executed in the "immediate" mode for checking and testing, or a number of them can be strung together to form programs.

An interpreter can also provide the desired isolation between the application programs and the operating system of the computer. Errors, even run-time errors, can be detected, signalled, and then easily corrected in terms of the high-level source code. Even more important is the facility an interpreter can provide to allow the logic of the control programs to be checked or "debugged". The source code instruction at which things are going wrong can be detected easily, and the program changed in a simple fashion, or the immediate mode can be used to enter instructions individually until the problem is located and a solution found.

The main disadvantage of an interpreter is that it is rather slow, since each statement has to be examined, checked, the necessary tables scanned, and the appropriate linkages made each time a program is run. In many control applications this slow speed of execution is of no consequence, and it can be largely overcome by providing additional machine code modules as functions of the interpreter, to perform actions which are frequently required in which higher speed is necessary.

Experiments using interpreted BASIC, to which the necessary CAMAC calls had been added, on a PDP-11, controlling simple pieces of equipment connected to a CAMAC crate, confirmed the ease of interaction between an operator and the equipment, and helped to convince some of the sceptics.

The evolution of the language is discussed in a later section, but the important decision made at this point was to have a re-entrant interpreter in every computer, to allow programs at different priority levels to be interpreted independently. This led to the simplification of the problem of operating a network of computers.

6.4 The computer network

When the original proposal from the Working Group to have a central computer controlling a number of satellites was considered, it was realized that an appreciable part of the central computers' capacity would be taken in communications tasks, and the layout was modified to include a minicomputer as communications controller between the control computer and the satellites.

The next decisions that were taken were interconnected, and resulted from the decision to have a resident interpreter in each computer. One of the difficulties with a fully distributed computer system, in which both programs and data can be exchanged between computers, is to organize the correct linkages between incoming messages and the processes running in the computers. However, the source code is data for the interpreter program, and so if the intercomputer traffic is restricted to interpretable statements and data which can be directed into a buffer for the interpreter to act on, the problem is greatly simplified. It is simplified further if all computers appear the same to the network, so that instead of having one permanent master computer and many slaves, any computer can temporarily become master to and initiate communication with one or more of the others. Thus it was decided to have a single level of hierarchy for the computers joined by a data transfer system, and that all inter-computer messages would be in the form of interpretable statements or data.

Up to this point, the established practice of using a single large computer for the central control had not been seriously challenged, although some of the disadvantages, such as delayed response to the operator's actions, had been noted. However, once the simplified form of intercomputer communication had been adopted, it became possible to think of dividing the individual duties of a large central computer between separate minicomputers. This seemed to provide some advantages, one being that the operating system in a special-purpose minicomputer could be very much simpler than that which would be required to do everything in one large machine, especially if the latter had to provide services for several operators simultaneously. In addition, the separate special-function systems could be developed and tested separately, with much less interaction than in a single system. The only major disadvantage was that the whole computing power of a large machine could not be brought to bear on a single problem. An investigation into the size of the control programs likely to be required revealed that the only ones expected to require a powerful computer were those concerned with closed orbit correction, where manipulation of large matrices might be required.

At this stage, a preliminary inquiry was sent out to a large number of computer firms. This asked for proposals and budgetary prices for a system of computers with a suitable interconnecting data transfer network. The requirements for the individual satellite computers and for the central computer system were given in as great detail as could be determined at that time, but the firms were given freedom to propose either a single large central computer or a number of smaller computers to do the same job.

When the answers were received, it was found that no computer manufacturer proposed a system that met all the requirements; only one made a proposal for the data transfer system that showed that the problems had been considered and most of them merely offered computers; some even proposing systems made up of different sizes of minicomputers which were not compatible in either the hardware or the software! Most of the serious replies favoured the split-up of the central computer into separate minicomputers, and in some cases firms quoted prices for both single and multiple computer systems. In such cases, the single central computer was considerably more expensive than the multicomputer system.

Therefore the decision was taken to have a control network made up of identical minicomputers, only differing in the size of core store and the number and capacity of the peripherals. The resulting layout is shown in Fig. 8.

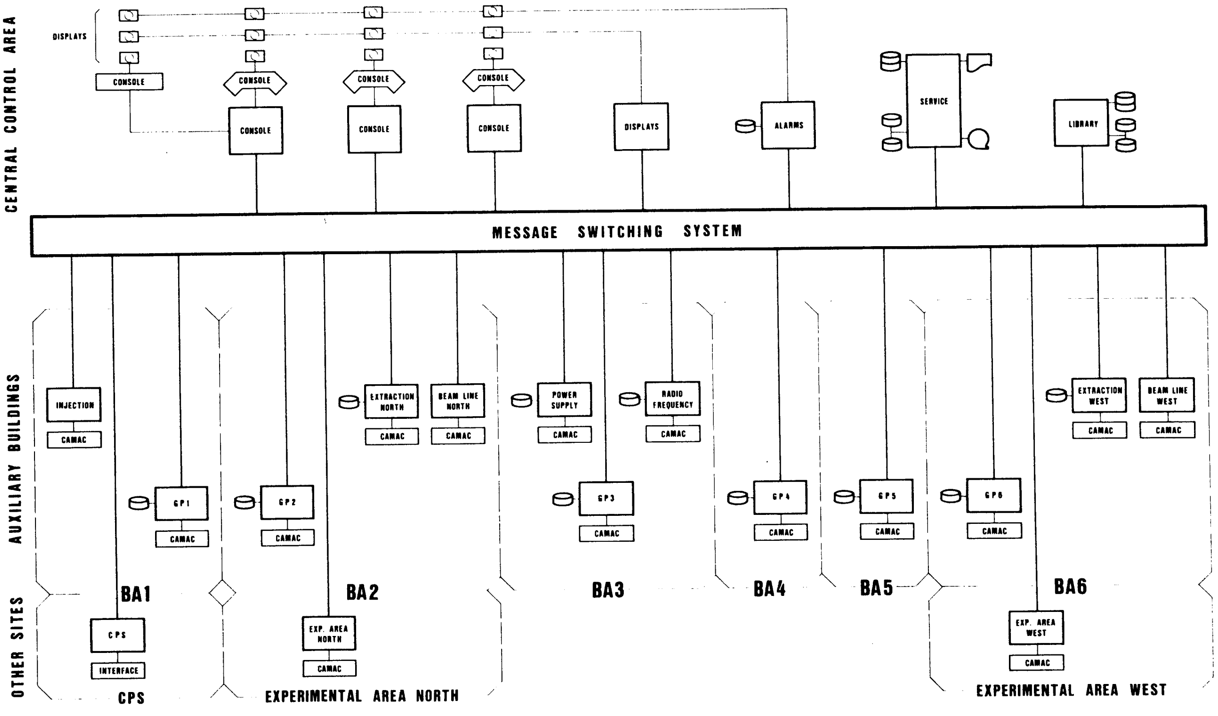


Figure 8 : Computer Layout for the SPS (1973)

6.5 The message transfer system

The Preliminary Inquiry having shown that no manufacturer could offer a standard computer communication network system at that time, it was clear that a special development was required. In fact, this was the most difficult part of the whole control system to specify, since the whole concept of a fully distributed computer system depended on its operation. There were no precedents to use as guides, and the amount of traffic to be carried could only be the subject of intelligent guesses, since so much of the accelerator equipment was still in the design stage.

One requirement was that the system must be fully transparent; that is to say, it must transfer messages from one computer to another without modifying them in any way. Other special requirements came largely from the decision that all normal messages would be statements or data for the interpreter, and from the way programs were to be split up amongst the computers. For example, a console computer could send a complete program down to one of the satellites for execution and the return of the results, or, as the result of a program running in the console computer, it could ask for single items of data from each of several satellites. In addition, it was intended that the library computer should be able to provide mass storage for those computers that lacked such facilities.

This meant that it was necessary to cater for long messages and short messages, to be able to interleave messages of different priorities to different computers, and to have several different channels open at the same time. There are two ways of carrying out such a transfer: circuit switching or store-and-forward packet switching. Circuit switching can be carried out by either hardware or software. The hardware switch is similar in concept to a telephone exchange. If one computer wishes to send a message to another, it has first to send a request for the necessary connection to be made, send the message, and then release the connection. The software switch is similar, except in this case the message-transfer system computer, instead of completing a direct hardware link between the two computers, creates a software link by establishing an automatic transfer between an input buffer and an output buffer. Systems using switches are efficient for long messages, but a link cannot be made until both the sender and receiver are ready for it, and it is difficult to allow for the interruption of one message by another of higher priority from another source.

In the case of the packet-switching scheme, all messages are broken up into blocks of a given maximum size and a header is added to each block before transmission. This header contains information on the source, destination, priority, etc., and the header and block together form the packet. A packet can be sent to the message-transfer computer at any time when the latter is ready to receive it. Packets from different sources can be interleaved and priorities can be respected. Although more storage has to be provided at the message-transfer computer than is needed in the case of a circuit switching, the amount of buffer storage needed at the satellite computers can be reduced, as it is not necessary to keep a message stored until the recipient can take it, and there is an over-all net gain with the packet system when there is an appreciable number of satellites.

The main disadvantage of a packet-switching system is that there is a certain overhead in handling each packet, and so, for a given data-link speed, the speed of transmission of a long message is lower than with circuit switching.

Considerable time was spent in trying to assess the amount of traffic that the message-transfer system would have to handle for various operating conditions of the SPS. It was found that this was very dependent upon the assumptions made about the way in which parts of the system would be designed, particularly the mechanism chosen for dealing with messages when major faults occurred. As a result of these deliberations, and some guesswork, a specification was drawn up for a message switching system that could handle a total message transfer rate of 30,000 16-bit words per second between a maximum of 32 computers. It must be possible to send a short message, at the highest priority, from a process in one computer to a process in another computer in less than 5 msec.

An invitation to tender against this specification was sent to a large number of firms and it was recommended that packet switching should be used, at least for the short messages, but the possibility was left open for the use of circuit switching for long messages, as it was recognized that the requirements for the SPS exceeded the capabilities of any existing or projected packet-switching system by a considerable margin. Only three offers were received, all proposing the use of packet switching for both short and long messages, but none of them was entirely satisfactory. Further work at CERN led to the development of a satisfactory protocol, and the chosen firm produced a system that met the specification fully. The work was made easier by the network using only one type of computer, and of course the same computer was used for the message switching.

6.6 The equipment interface

Since the choice of the make of minicomputer to be used would take some time, as it had to be the subject of competitive tendering after very detailed specifications had been drawn up, it was decided to use CAMAC for the primary interface to the computers, and for some equipment, such as beam instrumentation, where suitable CAMAC modules were readily available, since it is computer independent. The reasons for developing a special serial multiplex system for the SPS, where equipment could be distributed up to 200 m from the computer controlling it, have already been given in section 3.2.

In addition to the control and measurement signals, the multiplex highway cable also carries timing signals, in the form of clock signals and event markers. This is to solve the problems of exact timing and simultaneity in a multi-computer network. With multiprocessing, it cannot be guaranteed that an action will be carried out at exactly the right time, or that the same action will take exactly the same time to be carried out in different computers.

This difficulty can be overcome by designing the equipment so that it can be set up by the computer system to perform a certain action when triggered by the timing sytem - "Do such-and-such on the seventh clock pulse after event No. 6" - for example. Programmable timing modules, to select the event and then count the number of clock ticks before giving an output trigger, were designed to plug into the multiplex crates, as can be seen in Fig. 9.

6.7 The command language

A programming language consists of two main parts: the data structure, and the instruction set which manipulates the data. For the SPS control language, an integrated data structure was required, which included mathematical variables and arrays, process input and output data, subroutines, programs, etc., referred to by means of mnemonic names. The instruction set should reference these names as simply and naturally as possible so as to permit easy programming and efficient operation.

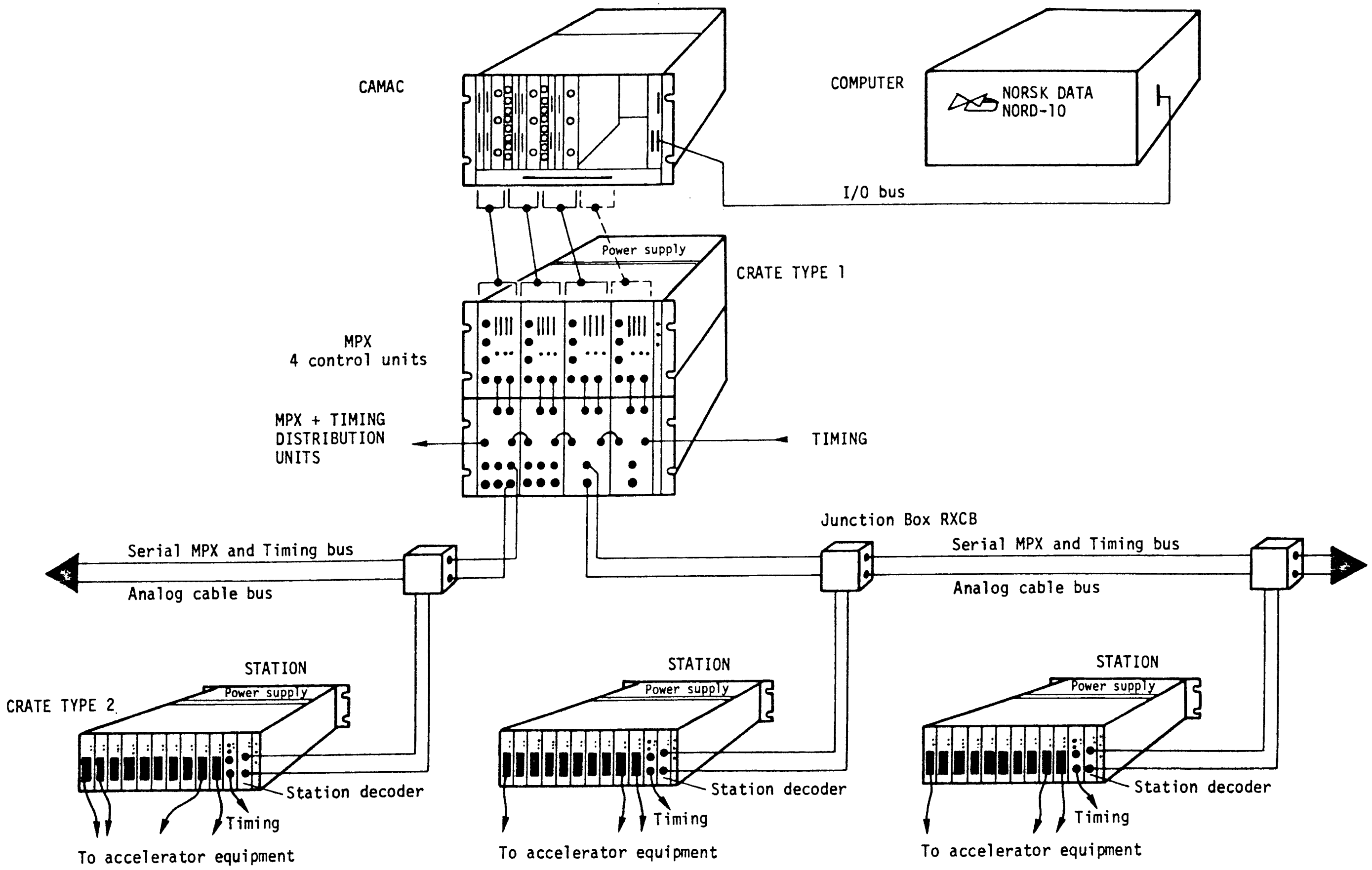


Figure 9 : The SPS Multiplex and Timing Distribution System

Most languages which have been designed for interpretation have a data structure directed towards simple mathematical computations and are somewhat limited in their facilities. Statements start with a command keyword and can be linked together into programs by means of line numbers. This approach was pioneered by the Rand Corporation with the language JOSS, followed by the Digital Equipment Corporation with FOCAL and Dartmouth College with BASIC. Since BASIC has the widest following, and interpreters are available for it on a considerable number of computers, the first idea was to use it for the SPS control system. However, the chosen computer, the NORD-10, although having excellent software in most other respects, had no interpreter, and since one had to be written specially, the question as to what was the most suitable language had to be examined.

The languages used for interpretation differ in the command words and in the details of the syntax, but can be divided into two main types, according to whether the lines are numbered sequentially as in BASIC, or are divided up into groups as in JOSS, FOCAL and TELCOMP (a language similar to FOCAL; used on a large time-sharing network in the USA). The big advantage of the latter type is that separate groups of a program can be executed as subprograms, and this is of particular interest in a multicomputer system, where it is often desirable to execute different groups of a program in different computers. It also leads to the writing of structured programs, which are simpler to understand and debug.

The decision to be made was whether to start from BASIC, since that was already fairly widely known, and put up with the disadvantages of its line structure, or invent a new language using the better group structure, which could be tailored more closely to fit the requirements for real-time control purposes. The existing group-structured languages could not be used directly, as the copyright is held by the originating firms. So much would have to be added to BASIC to make it suitable for real-time control use that even those who knew the language would have quite a bit to learn. Thus it was decided that the advantages of starting from a standard language were not sufficiently great to offset the disadvantages of the structure.

A new language was designed, using what seemed to be the best features of FOCAL, with some additions from BASIC, and also incorporating the string-handling facilities of SNOBOL 4. An interpreter was written for the NORD-10 and the language was given the name NODAL, to recognize its part-parent FOCAL and provide a contorted acronym of NOrD Accelerator Language.

What made NODAL different from previous interpretable languages were the commands EXECUTE, REMIT and WAIT. A program could call for a group or groups of the program to be EXECUTEd in another computer, the results to be REMITted back to the original program, which had to WAIT for the results. These commands provided the means for programming a multicomputer system.

Subsequently this language, or slight variations of it which have been given different names, has been adapted for control purposes at DESY, Rutherford Laboratory, BNL, KEK and JET.

6.8 The operating system

As explained earlier, it seemed that it would only be possible to provide a multicomputer operating system within a reasonable time scale and manpower limit by restricting inter-computer communication to NODAL statements and data for NODAL programs. This removed any necessity for direct real-time connections between the executives in the separate computers, but called for the facility to schedule programs to be run in one computer by means of NODAL statements sent from another computer.

The decisions, to use an interpreter in every computer and that the communication between computers should be in the form of interpretable statements or data, meant that each computer plus re-entrant interpreter could be considered as a virtual computer which obeys instructions in the control language directly and interactively. At one time it was thought that it might be possible to do without a real-time multiprogramming operating system in the satellite computers, by using multiple buffers for the interpreter, together with special functions which could provide for the scheduling requirements. However, one of the factors taken into account in choosing the NORD-10 computers was the availability of the executive SINTRAN II, which occupies only 3K words in the core-only version. This provides the facilities to schedule programs and to execute them according to priority and, in the drum version, enables the drum to be used as virtual memory by swapping a number of different "core-loads" in to and out of the same area of core.

It was thus possible to have a multiprogramming executive in each computer without excessive space requirements, but to this basic executive it was necessary to add the links to allow NODAL programs to be executed under the control of the monitor, and to interface with the message-transfer system. Subsequently, modifications were made to the executive to overcome some of its shortcomings, and, as the computer manufacturer standardized a larger, and slower, operating system, the present SPS executive has been almost completely written at CERN. The combination of this executive, the message transfer software and the NODAL interpreter form a multicomputer operating system, which can be programmed interactively.

6.9 Data modules

In the earlier sections on data bases and equipment drivers, the idea of the data module, consisting of part of a distributed data base and special purpose equipment driver, was introduced. This gives several advantages. Since each handler and data table is concerned with one elementary item of equipment, the table layout can be optimized and the handler can be designed to carry out many more specialized operations than would be practicable in a handler covering all types of equipment. The design of each data module can be carried out as soon as the method of operation of the individual item has been settled, and does not have to wait for information on the whole system to become available. Also, individual data modules can be loaded into the computers for testing and commissioning some of the equipment while design is still proceeding on other parts.

The access to the equipment via the data modules is carried out by statements involving an equipment name, a serial number, and a property, such as

SET MAGNET (4,#CUR) = 125.4 .

In this case, the property denotd by #CUR is the current in the magnet 4. Similarly, the magnet might be switched on by the property #SW1, or the multiplexer address inserted in the table by the use of the property #ADR.

6.10 The operator interface

Considerable thought was given to the operator interface. One of the first decisions was that the control consoles should be "general purpose". This means that any console should be able to be used to perform any action, the transformation for a particular use being made by the software. In practice, this ideal of generality could not be carried through with respect to the displays, since it would not have been economically justified to repeat the special high-frequency oscilloscopes needed for the RF equipment on every console.

The principles adopted were as follows:

- Three identical consoles would be provided to allow several things to go on simultaneously when setting up the machine or carrying out machine development. It was expected that normal operation of the machine would be carried out from one console, in which case the others would be available for beam-line operation, program development, or back-up in case of failure.
- Each console would have a separate computer. This was already decided when the multi-computer layout for the centre was chosen, to ensure sufficiently rapid response to operator actions such as cursor movements, to avoid interference between different tasks and thus allow control programs to go into the "waiting" stage, and to give the full back-up possibilities.
- All computer-generated displays would be of the television raster scan type, thus enabling displays to be reproduced anywhere on the site where they might be required, using cheap television monitors. Storage tube displays can give better resolution, but require erasing to make any modification, and "vector scopes" which use the computer memory were too expensive.
- A minimum of three colours should be available for the main interactive displays, since colour can be extremely effective in drawing the operator's attention to particular features of a display and can be used to indicate different states of a variable (selected, being set to a new value, arrived at new value, etc.).
- The main operator selection device would be a "touch screen", on which buttons with legends could be drawn and means provided for recognizing which button had been "pressed".
- The other operator interface devices would be a tracker ball (used normally to set a cursor to select some option displayed on a screen), a knob and a keyboard. It was thought that a single knob would be adequate for a slow-cycling machine, given the facility provided by the interpreter for coupling a number of parameters together in a mathematical relationship and using the knob to vary the resulting virtual parameter.

Experience has shown that these interactive devices are liked by the operators, and there has been little demand for change, except for the provision of some permanent displays. There are some things which the operator wants to be able to check by just turning his eye to a particular spot, rather than interrupting what he is doing and going down the touch-button tree to get the display on his console.

6.11 Assessment of the correctness of the choices made

It is clear that, for the SPS, the choice to provide a system whereby the users could program their equipment in a simple fashion was a key factor in the commissioning of the SPS in such a short time. The flexibility and modularity of the system has made it possible to change the operation of the SPS from a slow cycling, fixed target, operation to become a proton/antiproton storage ring without major reorganization of the control system. As described below in connection with the LEP system, technological advances have rendered many of the compromises made in the design of the SPS system to keep the memory requirements small now unnecessary, but it is not intended to depart from the basic principles.

7. CONTROL SYSTEMS OF THE FUTURE, WITH SPECIAL REFERENCE TO LEP

The main differences between control systems of the past and those being designed now are due to the dramatic reduction in the cost of processing power and of memory, as a result of the rapid progress in the art of large scale integration. Whereas previously a single processor looked after a number of equipments, now one can afford to have a separate microprocessor, with its own copy of the programs, for each equipment. However, once the microprocessor is incorporated into the equipment, the temptation to make it carry out other jobs cannot be resisted, and so, although the hardware costs may be small, the software cost may rise.

Microprocessors have up to now been mainly used in the interface equipment, to take over some of the repetitive tasks, as mentioned previously in Section 3.2, and the auxiliary crate controller in CAMAC has probably been one of the most popular applications of microprocessors so far. However, microprocessors are starting to be incorporated lower down the chain, in the equipment itself, at many laboratories and this will become the rule in the future.

7.1 Microprocessors in the equipment

Microprocessors incorporated into the equipment will take over the duties of sequencing, local surveillance, local servo-loops, function generation and testing, thus reducing considerably the load on the control computer. There are other advantages as well, as we can see by taking as an example a magnet power supply unit. Previously, such a power supply would need a number of control actions and status acquisitions to be provided by the interface equipment, as well as connections to a multiplexed analogue measurement system, and, if the power supply had to be ramped, there would be the need for an external function generator. Because of the cost of interfacing, the possibilities of remote fault-finding were rather limited, and all the external connections to the power supply were potential sources for interference pick-up. With microprocessors the situation changes considerably. The large-scale integration has also brought down the price of precision analogue-to-digital converters so that a separate one can be fitted to each power supply. Microprocessors can be programmed to act as function generators, operating on a stored table of values previously loaded down from the process computer, using the timing system for synchronization. Diagnostic and test programs can be run when something goes wrong, to provide the operators with full information as to what to do. A command interpreter can be provided so that the interaction with the control system can be in the form of simple messages, instead of a succession of command/response actions. All this can be provided within the power supply unit itself, with just the connection for messages and the timing as links to the outside world, reducing the cost and possibilities of interference pick-up.

7.2 The computer network

There are some arguments as to the number of levels of hierarchy there should be in a computer control system. On one hand, there are the local area network enthusiasts, who see one level only; every piece of equipment is connected to a network, or a series of branching networks, and every device can talk to any of the others. The main difficulty of such a system, apart from the problems of contention on a single network, is that each device, however small, must either carry a directory of addresses, etc., of all the devices it may want to communicate with, or interrogate a central directory and get an answer back each time a communication is required. In the first case, changing one piece of equipment requires a change to a multitude of directories, and in the second there is an additional overhead in response time and network occupancy. On the other hand, there are the advocates of a three-level hierarchy, involving a main central computer which controls a number of satellite computers, which in turn control microprocessors in the equipment.

For LEP, we propose to use a two-level system, having a network of computers all at the same level, with each of the computers controlling equipment having one or more lower level networks to which the individual items of equipment are connected. The intercomputer network will take the form of a number of interconnected stars, and the lower level networks will be multidrop highways, as shown in diagramatic form in Fig. 10.

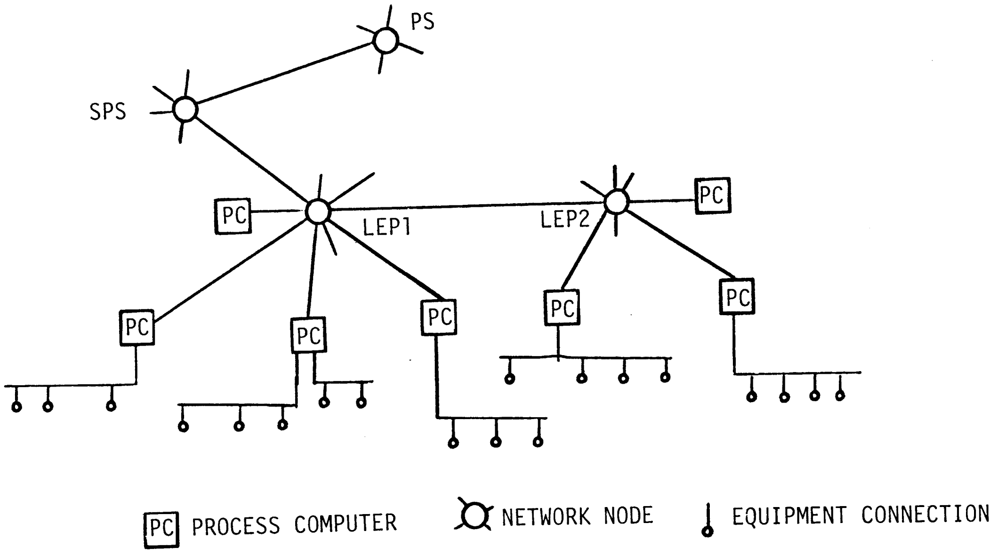


Figure 10 : The Two Network Levels for the LEP Control System

The intercomputer network should be directly connectable to the SPS network, which has a similar topography, since the two machines will be run together, from the same control room, and could therefore share some resources. However, as explained earlier, the SPS message transfer system was designed before international standards for packet-switching networks were agreed, so that use cannot be made of the specialized integrated circuits now available to carry out the lower-level protocols for the standard systems. Therefore it is proposed to use one of the standards, HDLC (High-level Data Link Control), for LEP data links, and provide a gateway to provide the necessary conversion at the SPS/LEP interconnection, which can be done in hardware with negligible time penalty. The higher level protocol must remain the same, since a change here would involve an appreciable software penalty. An idea of the size of the network needed can be obtained from the preliminary design drawing shown in Fig. 11.

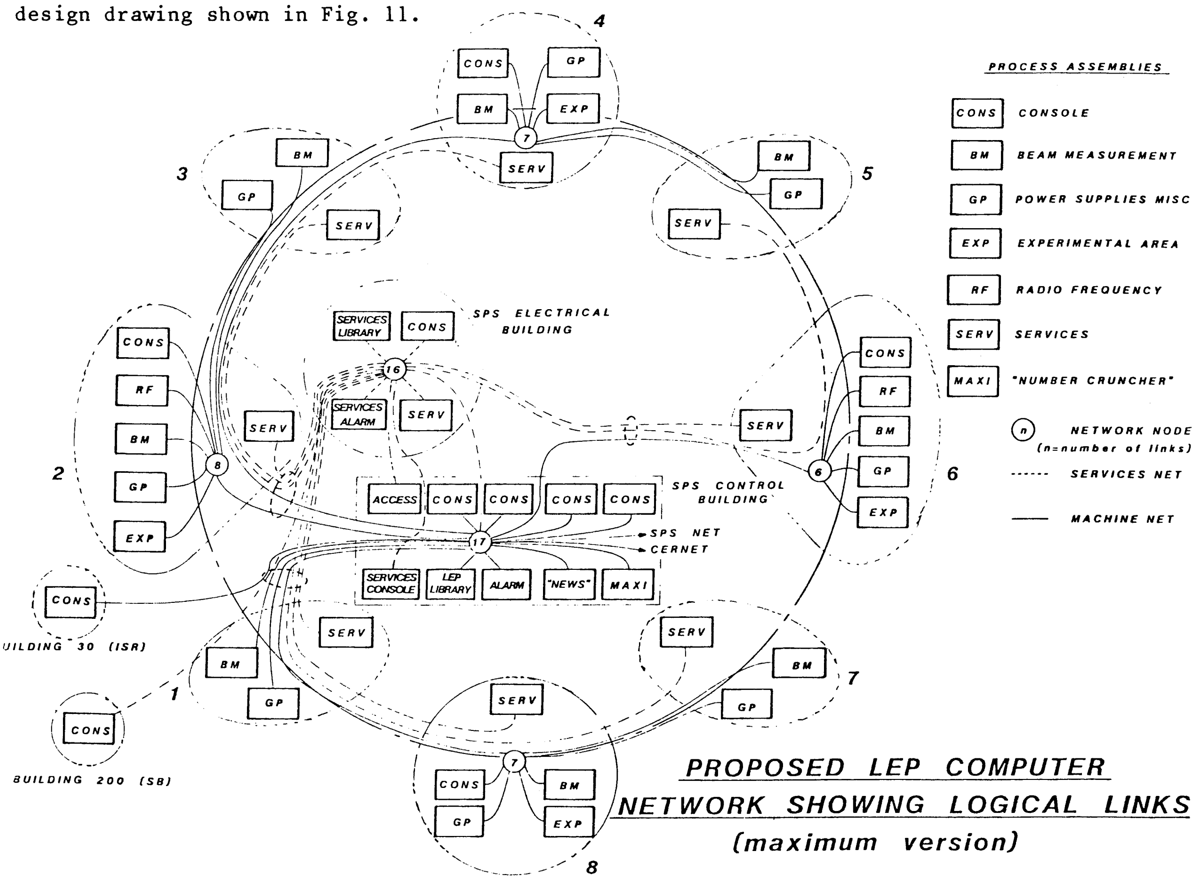


Figure 11 : Proposed LEP Computer Network Showing Logical Links (maximum version)

There will be two main interconnected stars, one for the control of services such as electrical distribution, water cooling, ventilation, personnel access, radiation monitoring and vacuum, where continuous service is required independent of whether the accelerator is running or shut down, and the second for control of the various components of the machine. The second star will have subsidiary stars, where the saving in data links justifies the addition of a node. The links shown in the diagram are logical ones, the physical means of producing them is discussed below.

7.3 The computers

One of the difficulties experienced with computer control systems is that the life of an accelerator or storage ring is normally longer than the life of a computer. At the PS, at Fermilab and at Los Alamos, amongst others, the original computer(s) installed have become obsolete, are expensive to maintain, and in some cases the original manufacturer has given up making computers, or even ceased to exist. Even when the computer, or a more modern replacement, is still supported by the manufacturer, sometimes the old operating system is not and the new operating system requires extensive modifications to the existing software.

With a system like that of the SPS, using a central library, the process computers (those connected to the equipment) are fairly simple, using a real-time executive, and they do not require all the facilities of a comprehensive operating system. It was suggested some time ago that such a computer, where the processor has to be shared between a number of concurrent processes, could be replaced by an assembly of microprocessors, each dealing with a single process. The executive would then be reduced to little more than a scheduler and arbiter. Compared with a minicomputer, such an assembly should be cheaper initially, but the main advantage would be on maintenance, which could be carried out by replacing a complete microprocessor board, rather than relying on an expensive, and sometimes unsatisfactory, maintenance service from the computer manufacturer. Also, additional requirements could be met by plugging in an extra module, or replacing one by a later, more powerful one.

This idea of using an assembly of many microprocessors, each performing a single stream process, has only been worth considering since the price of memory has fallen so low, because, to preserve the basic simplicity of the concept, each microprocessor should have enough memory to carry out its normal tasks, and should communicate with the others by passing messages, rather than using shared memory. Critics of this proposal point out that very soon powerful microprocessors will be available with a comprehensive multi-tasking operating system included on the chip, which will be able to replace the present day minicomputer directly, without going to the complication of the multi-microprocessor assembly. This may be true for such things as personal computers, scientific work stations, etc., but the situation is different for a real-time process control application, where the simplicity of the software, the ability to debug and test the different processes independently, and the possibility of tailoring the hardware configuration to fit the requirements, make the multiprocessor assembly sufficiently attractive that it is proposed to use this scheme for the process computers for LEP. It is also interesting to note that Texas Instruments have based their latest designs on the principle of performing specialized functions in separate processors. They call it "Function to Function" architecture. An experimental set-up has been made at CERN using a number of CAMAC auxilliary crate controllers to replace the majority of the tasks performed by a NORD-100 minicomputer in the SPS system, to see what difficulties might be experienced, before building prototypes suitable for use with the LEP system.

CAMAC has its limitations as a basis for such a system, as it was not designed for multi-master operation, and there is no provision for an arbitration scheme. There are a multitude of card, crate and bus systems for microprocessor use in existence, mostly designed by microprocessor manufacturers to suit their particular components. Although some of these have become de facto standards, and have been recognized as such by the American IEEE, this institution set up a Working Party to try to propose a standard for a comprehensive, microcomputer-independent, bus system, to which it gave the project number P896. However, there have been difficulties in reaching agreement over some of the details, and little progress seems to have been made recently. Concurrently in Europe, a Sub-Committee of ESONE (The European Standards Organization for Nuclear Electronics) was studying the same problem, and has come up with a specification for an integrated, microprocessor-independent, system, using the Eurocrate mechanical assembly and the Eurobus for interconnecting the modules. It is hoped that this standard, known as E3S, will be adopted by ESONE at its General Assembly next year. It is intended to use this system for the construction of the prototype units for the multi-microprocessor assemblies for LEP.

7.4 The interface network

Earlier it was pointed out that, with microprocessors in the equipment, communication with the process computers could be in the form of messages. For the LEP system, it is proposed that these messages should be in the form of ASCII characters, which can be interpreted by the microprocessor. This has the advantage that the messages can be in a printable and easily recognizable form, and that the equipment can be operated for test purposes from a very simple terminal. For the transfer of these messages, from a process computer to the equipment it is to control, it is proposed to use a multidrop serial highway system, as this topography tends to give the minimum length of cable for the expected distribution of equipment and lends itself well to providing galvanic isolation between units, and the failure of one unit does not cause the whole system to go down, as happens with some loop or ring systems.

Two systems are being evaluated for the LEP multidrop highway; a system standardized by the American aviation industry, known as MIL-STD-1553B, and an in-house development using a modified HDLC frame and a highway based on RS422, which defines the connections to a computer interface socket.

The decision to limit the communication on the multidrop highway to ASCII messages means that special steps have to be taken to connect to any equipment that does not contain a microprocessor. For example, if it is wished to use some CAMAC modules, because they are particularly suited to a certain job, it will be necessary to provide a CAMAC crate controller which incorporates a microprocessor, to take the messages and turn them into the CNAF command/response mode required by the CAMAC protocol. Similarly, adaptors will be needed to control test equipment which is already interfaced to the GP-IB (IEEE 488) standard, which is becoming quite popular.

7.5 Databases and data modules

The incorporation of microprocessors into the equipment means that part of the database also has to be in the equipment, otherwise many of the advantages of having the microprocessors there would be lost. The parts that are common to all equipments joined to a process computer should be in that computer, and parts that are particular to an equipment should be in its microprocessor memory.

In the case of LEP, the data module, as described in the chapter on the SPS control system, will be divided into two parts; the "Equipment Directory" in the process computer and the "Property Module" in the equipment.

The equipment directory will be entered by a call to a data module with name, property, equipment number and possibly value or array. For that name, the directory will give the valid properties, the password requirements, the interface address of the property module, and any translation necessary to call the required property. This directory will contain semi-fixed data, which only needs to be changed if any changes are made to the equipment or to the property modules. The master copy of the data for each equipment directory will be kept in the library, and this master copy will be incorporated automatically into the equipment directory each time it is assembled. The exception to this could be some status indications for the equipments which would be read in from the property modules.

The property modules code should be in PROM with only the data table in RAM. This data table will have semi-fixed data, such as conversion factors, tolerances, maximum allowable values and interface addresses, and variable data, such as demanded values, status, measured values, etc. These distributed data tables form the operational data base for the control system; any program wishing to interact with the equipment or obtain information on its status, will use the property modules and their data tables. There will be no automatic update of a central equipment data base, as with some systems, although copies of the property module data tables will be taken and held in the library periodically. These copies will be solely for archives and for reloading into property modules if, for any reason, the working tables become corrupted.

In addition to this distributed equipment database, it is necessary to have a central part which is concerned with the operation of the machine as a whole. For a given operating condition, a large number of equipments must be set to certain values, and the ability to record and recall these standard settings must be provided. This will be done by programs creating and using files of standard settings kept in the library.

One of the major advantages of a computer control system is the ability to control derived or virtual parameters, such as Q, beta, etc., rather than individual quadrupole currents. In setting up LEP it will be necessary to have files of a number of this type of parameters from which the settings of individual elements can be calculated, and with which the measured values can be compared. These files will form the central part of the distributed database, and should be partitioned in such a way that a control program will only need to load a subset to provide the environment in which a certain series of actions can be performed.

Where modelling or simulation is involved, requiring greater computing power or capacity than can be provided by the console computers, the control program can send a request to a larger computer to obtain the appropriate file, perform the calculation and remit the results, either direct to the requestor or into a library file.

7.6 The operator interface

While it is certain that the existing operator interface equipment will be subject to a continuous series of improvements, for example visual displays will have higher definition, full range of colours and be simpler to program, and CRT's may be replaced by flat screens, it is not clear whether any basically new new devices will be adopted. Some people prefer the "mouse" to the tracker-ball, but this is only a variant on the same theme. More striking would be the use of the type of eye movement sensors being introduced into aircraft "head-up" displays, which could be used to input to the computer the chosen position on a display, instead of using a cursor or light pen - a blink could then take the place of the interrupt button!

There is a tremendous interest in voice input and output devices in other fields. For accelerator control, voice output to warn the operator of fault conditions will probably come, since it can now be provided so easily, even to providing a choice of language to be used in a multi-lingual environment such as CERN. On the other hand, voice input seems a little dangerous in an accelerator or storage ring context - an inadvertent word might dump the beam!

Hand-written input instead of typing, for situations not covered by the touch buttons, does not seem to offer any great advantages, but to be able to sketch a display or mimic diagram and have it produced in tidy form on the screen, as with some CAD systems, could be interesting.

For LEP, the existing devices used on the SPS consoles would be adequate, but a choice does not have to be made for several years, and progress in this field will be watched.

7.7 Integrated communications systems

In any large accelerator, there is a number of services that require cables round or across the ring, and between the equipment buildings, the control centre and the experimental areas. These include inter-computer data links, computer to hardware connections, audio communications, television pictures, waveform signals, access control and interlock signals, etc.

Since one wide-band cable is often cheaper than a number of narrow band ones, it is useful to see how a number of different signals can be put on a single cable. There are two main ways of doing this, frequency division multiplexing and time division multiplex. Frequency division multiplexing has been used mainly with analogue signals, which are used to modulate an RF carrier to produce an "up-converted" version of the analogue signal, occupying a band of frequencies adjacent to the carrier frequency. By modulating each analogue signal by a different carrier frequency, the signals can exist together without interference on a common cable. Most of the present long distance telephone trunk lines use this method of compressing a number of telephone circuits onto a single cable. Cable television is another form of frequency division multiplexing, the different television programmes using different frequency bands on the same cable. This same TV cable system is being used in some broad-band local-area network systems to transmit digital data as well as analogue signals.

On the other hand, time division multiplexing requires the information to be in digital form, the bits from different sources being interleaved to provide a single signal at a higher bit rate for transmission, which must then be broken down into its original components at the receiving end. This system has been adopted by the PTT's of many countries as the future multiplexing system for long distance telephone connections, the analogue signal from the telephone being sampled at a sufficiently high rate (8 kHz) to be able to reconstruct the highest frequency normally transmitted (about 3.5 kHz) and the amplitude of each sample digitized to an 8-bit level, giving a 64 k-bit/sec result. Thirty-two of these signals can be interleaved to give a 2048 k-bit/sec signal, and four of these combined in an 8.5 M-bit/sec channel. Further concentrations of four channels into one give rates of 35 M-bits/sec and 140 M-bits/sec which are already in operational use and 580 M-bits/sec links are being experimented with, using optical fibres for transmission.

This system could be of interest to reduce the numbers of cables required round a large accelerator, since advantage could be taken of the specially reliable equipment developed for the PTT's, and experiments are being carried out to see how it best could be applied for LEP, not only for the machine itself, but also for data links between the experimenters' computers and the CERN computer centre.

8. SOFTWARE

As was indicated earlier, there is some divergence of opinion as to the best way to provide the software, and in particular the applications programs for an accelerator control system. Sufficient laboratories have adopted the interpretive approach for it to be considered conventional these days, although there are those that think it lends itself too easily to the production of badly constructed programs with poor documentation and difficulties in the long term maintenance of programs when the writers have moved on to other work. The American Department of Defense, faced with the same sort of problems, has sponsored the production of the language ADA and a software environment for it intended to enforce an orderly approach to programming computer systems embedded in military equipment, and provide tools to make it easy for them to do so.

Despite this, it is intended to use NODAL as the main applications programming language for LEP, since the advantages seem to heavily outweigh the disadvantages. The main question to be settled is what language to use for writing the interpreter, the operating system and the functions, etc. that the interpreter uses. For the SPS, these were written in the assembly language of the minicomputer, but for LEP they should be written in a higher-level language, both to save time in writing them and to increase the portability, so that a change of microprocessor in the future will not need extensive rewriting. ADA should be suitable for this purpose, but the time scale for its availability with full environmental support is still not clear. It may be that we will have to settle on one of the PASCAL derivatives now available, in order to be sure of meeting the LEP requirements. One thing is sure, however, and that is that FORTRAN will be with us for many years to come, and many of the more complicated functions, such as those used for simulation and modelling, will be written in FORTRAN.

* * * * * * *

Further reading

The major developments in accelerator control systems have been reported in the Proceedings of two series of conferences. The first of these, the International Conferences on High Energy Accelerators, has been held periodically since 1956. A list is given below of the dates, where they were held, and the publishers of the proceedings.

1st	1956	CERN	CERN Geneva
2nd	1959	CERN	CERN Geneva
3rd	1961	Brookhaven	USAEC Washington
4th	1963	Dubna	USAEC (English translation)
5th	1966	Frascati	CNEN Rome
6th	1967	CEA	Cambridge University, Mass.
7th	1969	Yerevan	Academy of Sciences, Armenia
8th	1971	CERN	CERN Geneva
9th	1974	SLAC	Stanford University, California
10th	1977	Protvino	IHEP Serpukhov
11th	1980	CERN	Birkhauser Verlag. Basel

The second series of conferences on particle accelerators, held every two years since 1965 in the United States, concentrates on the engineering aspects. The proceedings of these appear in the IEEE Transactions on Nuclear Science. The volume and issue references are as follows:

1965	NS-12. 3
1967	NS-14. 3
1969	NS-16. 3
1971	NS-18. 3
1973	NS-20. 3
1975	NS-22. 3
1977	NS-24. 3
1979	NS-26. 3
1981	NS-28. 3

Information on the accelerator control language NODAL is given in CERN Yellow Report CERN 78-07, and some parts of chapter 6 of this paper have been taken from CERN Yellow Reports CERN 75-20 and CERN 78-09.