THE DESIGN OF THE CONTROL SYSTEM FOR THE SPS

M.C. Crowley-Milling

GENEVA
1975
Propriété littéraire et scientifique réservée pour tous les pays du monde. Ce document ne peut être reproduit ou traduit en tout ou en partie sans l'autorisation écrite du Directeur général du CERN, titulaire du droit d'auteur. Dans les cas appropriés, et s'il s'agit d'utiliser le document à des fins non commerciales, cette autorisation sera volontiers accordée.

Le CERN ne revendique pas la propriété des inventions brevetables et dessins ou modèles susceptibles de dépôt qui pourraient être décrits dans le présent document; ceux-ci peuvent être librement utilisés par les instituts de recherche, les industriels et autres intéressés. Cependant, le CERN se réserve le droit de s'opposer à toute revendication qu'un usager pourrait faire de la propriété scientifique ou industrielle de toute invention et tout dessin ou modèle décrits dans le présent document.

© Copyright CERN, Genève, 1975

 Literary and scientific copyrights reserved in all countries of the world. This report, or any part of it, may not be reprinted or translated without written permission of the copyright holder, the Director-General of CERN. However, permission will be freely granted for appropriate non-commercial use. If any patentable invention or registrable design is described in the report, CERN makes no claim to property rights in it but offers it for the free use of research institutions, manufacturers and others. CERN, however, may oppose any attempt by a user to claim any proprietary or patent rights in such inventions or designs as may be described in the present document.
ABSTRACT

After some general remarks about the requirements for the control system of large accelerators, a brief description is given of the CERN Super Proton Synchrotron. The major part of the report surveys the evolution of the hardware and software, with particular emphasis on the reasons for the choice made. The message-transfer system, the timing system, the operator interface, the command language and data modules are treated in detail, followed by some considerations about the protection and reliability. Finally, a preliminary assessment is made of the operation of the system for testing and commissioning parts of the accelerator equipment.
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>2. GENERAL REQUIREMENTS</td>
<td>1</td>
</tr>
<tr>
<td>3. THE ACCELERATOR</td>
<td>2</td>
</tr>
<tr>
<td>4. PRELIMINARY CONSIDERATIONS</td>
<td>4</td>
</tr>
<tr>
<td>5. THE FIRST DESIGN</td>
<td>5</td>
</tr>
<tr>
<td>6. THE HARDWARE INTERFACE</td>
<td>6</td>
</tr>
<tr>
<td>7. THE COMPUTER SYSTEM</td>
<td>8</td>
</tr>
<tr>
<td>8. THE MESSAGE-TRANSFER SYSTEM</td>
<td>13</td>
</tr>
<tr>
<td>9. THE TIMING SYSTEM</td>
<td>18</td>
</tr>
<tr>
<td>10. THE SOFTWARE SYSTEM</td>
<td>20</td>
</tr>
<tr>
<td>11. THE COMMAND LANGUAGE</td>
<td>21</td>
</tr>
<tr>
<td>12. DATA MODULES</td>
<td>23</td>
</tr>
<tr>
<td>13. THE OPERATING SYSTEM</td>
<td>25</td>
</tr>
<tr>
<td>14. THE OPERATOR INTERFACE</td>
<td>26</td>
</tr>
<tr>
<td>15. APPLICATIONS SOFTWARE</td>
<td>29</td>
</tr>
<tr>
<td>16. PROTECTION</td>
<td>30</td>
</tr>
<tr>
<td>17. RELIABILITY</td>
<td>31</td>
</tr>
<tr>
<td>18. PRELIMINARY ASSESSMENT</td>
<td>32</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>35</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

The control system for the large Proton Synchrotron (SPS) being built at CERN, near Geneva, has aroused considerable interest, both because of its size and of the unconventional approach to some of the problems.

Many requests have been made to explain the reasons behind the choices made in the design of the system, and this report has been written in an attempt to answer some of these questions.

It is being written at a time when the first parts of the system are only just coming into operation, and so the judgement as to the correctness or otherwise of the various choices taken can only be a very preliminary one. However, it was thought better to put down the reasons for the various choices now, while they are still fresh in the mind and not coloured by too much hindsight, and to issue a further report when the system has been in use for some time and a balance sheet can be drawn up.

A major difficulty in writing about the reasons for decisions is that in most cases decisions are not made in isolation -- the results of one decision influence other decisions -- and so the subject cannot be cut up into neat little self-contained chapters. In practice, of course, there are parallel streams of investigations going on during the development of a system, and to achieve success there must be adequate interaction between these streams. Frequently decisions have to be taken where the information is still incomplete, to ensure that the system will be available when required.

In this case, it seems best to keep to an approximately chronological division, as far as the major subsystems are concerned, so that the interactions can be brought out without asking the reader to make too many cross-references. Firstly, the general problems of a control system for a large accelerator are discussed and a brief description of the SPS is given. This is followed by the major part of the report in which the evolution of the hardware and software system is described, supported by sections devoted to particular developments. The report is concluded by a preliminary assessment of the operation of the system for the testing and commissioning of parts of the accelerator.

2. GENERAL REQUIREMENTS

The control system for an accelerator has to satisfy many requirements, from the conventional to the unconventional, and has to combine in one system the attributes of many. In control system language, it can be classified under the general category of supervisory control, but it often has elements of direct digital control and stored cycle optimization.

At one extreme, it can just provide for the remote control and indication of all the accelerator equipment at a single place, on a one-device/one-function basis, as was the
case for some of the earliest systems. At the other extreme, it could provide for the complete automatic run-up of the accelerator to pre-set conditions, continuous optimization of operation, supervision of all equipment and complete analysis of fault conditions with automatic switch-over to standby equipment when required, finishing with the production of an analysed log of the whole operation without the intervention of an operator at any stage. Such extreme automation is not warranted or even desirable for a particle accelerator, partly on the obvious grounds of complication and cost, but mainly because the exact performance and mode of operation of a particle accelerator cannot be predicted during the design stage with sufficient certainty to allow complete automatic control to be incorporated ab initio. During the design of the accelerator, the number of controls, status and alarm signals, and measurements that need to be interfaced to the system can be determined, and the simple algorithms for routine setting and surveillance can be defined. However, this is only a fraction of the duties required of a control system for a large accelerator. Experience has shown that, when such a machine comes into operation, it is quite common either for unexpected effects to appear or for known effects to show themselves in an unusual fashion. This is not strange, since such machines are usually built to the limits of our theoretical or technological knowledge. It does mean, however, that the control system has to be unusually flexible, to allow trials of control algorithms involving complex measurements, various computations, and the resultant multiple control operations. Any time taken for machine development investigations is time lost from the experimental physics program, and it is clear that the investigational work can be speeded up considerably if provision is made for direct interactive development of suitable algorithms and this must influence the choice of the system. In addition, a particle accelerator is not a fixed piece of apparatus, but undergoes development throughout its life to improve the performance in many ways and in some cases even to accelerate different particles. Again this requires flexibility, together with simplicity of basic concept, because however much flexibility may be built into a system, advantage will not be taken of it if the system is too complicated.

Thus it can be seen that the art of designing a control system for a particle accelerator lies in finding a suitable compromise between too little and too much automation; providing great flexibility and yet keeping the basic concept simple; guessing what the control requirements will be for equipment still in the design stage and, of course, doing this for the minimum cost and to a strict time schedule.

3. THE ACCELERATOR

The SPS is 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. To ensure that the protons can circulate many times round this
orbit without loss, the position of the guiding magnets has to be set and maintained to within fractions of a millimetre. This can only be achieved if the accelerator has stable foundations. To achieve this on the CERN site it has been necessary to build the machine in a tunnel, about 40 m underground, where it can be wholly beneath the unstable moraine and in the stable molasse rock.

Building the machine in a tunnel so far underground has the advantage that more than adequate shielding is provided for the maximum possible level of radiation that may be produced by the accelerator, but has the disadvantage that it would be very expensive to provide a large number of access points to the underground tunnel. 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.

A schematic view of the layout is shown in Fig. 1. 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.

Fig. 1 SPS site layout
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.

4. **PRELIMINARY CONSIDERATIONS**

Before the construction of the SPS was approved, a "500 GeV Machine Committee" was set up by Dr. J.B. Adams, the Director-General Designate. The members of this committee were accelerator specialists from CERN and outside laboratories, and subcommittees were set up to look into the detail design of the various subsystems. One of these subcommittees considered the control and instrumentation for the proposed accelerator.

From the start it was accepted without question that for such an accelerator, covering such a large area, extensive use of multiplexing would have to be made to cut down the number

![Diagram of computer system proposed by the Machine Committee](image-url)
of cables required, and that computers would have to be used to control this multiplexing and simplify the work of the operator. At that time there were several large accelerators which used computers in their control systems, and others were under construction. It was then conventional to have a computer, as large as could be afforded, in or near the control room, and to transfer as much surveillance and control as possible to this computer. Some laboratories were beginning to experiment with the use of minicomputers, acting as satellites to the main computer, to perform specific jobs. It was thus no surprise that the subcommittee should recommend the use of a multicomputer system, consisting of a central large control computer connected to a number of satellites, as shown in Fig. 2.

Some major components of the accelerator, such as the RF system, the injection and each of the two extraction systems, would be concentrated in a single building, and so it was logical to provide a single large minicomputer for each of these systems. Other equipment, such as the vacuum system, the beam monitoring system, and the general services, would be distributed fairly uniformly round the ring. Therefore it was proposed to provide a very small computer, called a process controller, in each sextant to control a multiplex system which would route control signals and acquire data from the equipment. These process controllers, marked "C" in the diagram, were joined to a fifth large minicomputer (MPX), which was joined directly to the main computer. Some of the other minicomputers also had process controllers joined to them, for particular duties.

Another proposal of the subcommittee was that the use of an interpreter for a command language should be considered, to provide flexibility and simplicity of program modification. A small system using an interpreter for the control of a beam line was in successful use at the Rutherford Laboratory, and experiments with an interpreter were being carried out at NAL. As will be seen below, it was decided later to base the software system on a command language interpreter.

There was also a proposal to use the CAMAC system as the interface between the computers and the hardware to be controlled. This was mainly on the grounds of standardization, as CAMAC was becoming the standard for experimental equipment, although it was recognized that there were some difficulties in its use for control applications.

Most of the other proposals were extensions of conventional design1).

5. THE FIRST DESIGN

When the SPS project was approved, the Group Leaders who would be responsible for the subsystems were appointed and one of their first jobs was to assist in the production of a Design Report2). Since, in many cases, the Group Leaders had taken part in the working parties, it is not surprising that the Design Report did not depart in essentials from the report of the 300 GeV Design Committee.

In the case of the control system there was some change in the proposed network of computers. It was recognized that the hierarchy of computers which the committee had recommended would involve multiple handling of data, and it seemed better to use a system such as that shown in Fig. 3, which is taken from the Design Report. In this, all satellite computers, both the larger system controlling a major function and the smaller process controllers with a more restricted task, would be joined to an "interface computer" which
had the double task of sorting out the intercommunication between the satellite computers and the centre and providing the major data and program storage for the system. The further evolution of this will be discussed later.

6. THE HARDWARE INTERFACE

Quite early on, it had been decided to use the computer-independent CAMAC system for the primary interface between the computers and the hardware. The main reason for this was that the design of the interface equipment could go on before the make of computer to be used was known. It was clear that the choice of computer would take some time, as so many
considerations had to be taken into account, and it was undesirable to delay the start of
the interface design until this work had been completed.

CAMAC has been adopted as a standard for nuclear physics and is rapidly being adopted
in other fields. It is primarily a system for the rapid build-up of interface systems be-
tween experimental apparatus and computers from standard crates and plug-in modules. There
are considerable advantages in such a system where the apparatus is grouped close to the
computer, but in applications such as the control of the SPS, the CAMAC system has some
serious disadvantages.

The standard CAMAC system uses a parallel highway, which is either an extension of the
computer I/O bus, if separate crate controllers are used, or the CAMAC branch highway, when
type A controllers are used. This means that the CAMAC crate cannot be far from the com-
puter, and individual wiring has to be run from every unit to be controlled to an area near
to the computer. This was a serious disadvantage for the SPS, as the equipment to be con-
trolled can be up to 100 m away from the computer in an Auxiliary Building, and in some
cases it is necessary to control equipment more than 1 km away from the computer. A serial
highway system for CAMAC, which would reduce this disadvantage, was under discussion at the
time but it was clear that even if standards could be agreed, fully tested equipment would
not be available in the time scale required.

Another disadvantage became evident when investigating the use of standard CAMAC
modules. Although there was a large choice of modules available for data acquisition,
which was the first use of CAMAC, there were few modules available for control applications
and many of these were unsuitable for driving equipment at any great distance. In addition,
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 mea-
surement 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 serious difficulty is that the
common earth normal to the CAMAC system can cause coupling between different pieces of
equipment connected to the same computer.

In view of the above disadvantages, and others such as the necessity to switch off a
complete CAMAC system before a module could be inserted or removed, it was decided to de-
sign a special serial-highway multiplex system for all control and acquisition where high
speed was not essential, and to use CAMAC for the beam instrumentation data acquisition,
where higher speed was required and suitable CAMAC modules were available.

It was expected that, in addition to the advantages that a specially designed multiple
system could provide, it would also be cheaper than CAMAC. This is mainly because a CAMAC
system has to provide all the facilities called for by the specification, even if they are
not used in a particular system. It has been estimated that, even if serial CAMAC had
been available, with special modules suited to the various tasks, the over-all cost per piece of equipment controlled would have been more than double that of the system being used.

In order to maintain computer independence and avoid delaying the start of the design, the controller for the multiplex system was designed to interface to the computer through the intermediary of a number of CAMAC modules. However, this put some complications into the system and, if it was being re-designed now, consideration would be given to making the controller plug directly into a CAMAC crate or even connecting it directly to the input/output bus of the computer.

Each controller manages a serial branch, and up to four controllers can be run from 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.

The output connections from all modules are fully isolated, using either reed-relays or opto-couplers, and modules can be removed from the system and reinserted without removing the power or causing any disturbance to the system. If the power fails, the system automatically initializes itself on restoration of power. To keep the system simple, interrupt facilities are not normally provided, but arrangements are made for a semi-autonomous scan of certain selected status bits, which can be performed at reasonably frequent intervals with the minimum involvement of the computer.

7. THE COMPUTER SYSTEM

At this point the development of the design of the hardware became inextricably mixed with that of the software. It had been decided, from the start, to use an interpreter for a high-level language as the keystone of the software system, and a small experimental system was set up using a PDP-11 and a CAMAC crate. The interpreter for BASIC was coupled into the DOS operating system and the necessary calls to CAMAC were added. The use of this system confirmed one of the advantages of using an interpreter -- the ease of interaction between the operator and the equipment connected to the CAMAC crate -- and demonstrated the power and simplicity of the language. The advantages and disadvantages of an interpreter will be discussed in another section, but the important thing to bring out at this point is that an interpreter is a re-entrant program, and that programs written in the high-level language are data for the interpreter program.
Recognition of this led to the proposal that all messages between computers should be in the form of data, i.e. statements or programs in the high-level language that can be interpreted, or data for such programs.

This would greatly simplify the organization of a multicomputer system, and the adoption of this proposal had repercussions on the hardware.

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. It is true that some accelerator control systems already had more than one computer at the control centre, but in these cases only one was actually controlling the accelerator, the other(s) being used for program development, or providing special displays, or standing by as a replacement for the active computer.

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 enquiry 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. The layout proposed for the latter system is shown in Fig. 4. The major difference between this and the earlier proposal is that the duties of the interface computer have been divided into two parts -- message switching and library facilities -- with a separate computer for each. In addition, the possibility of a connection to the CDC 7600 complex was introduced. This was done so that, if the multi-minicomputer alternative for the central system was chosen and it was later found out that some very complicated problems needed solving on-line, provision could be made for doing this work at the computer centre.

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 wanted to sell 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.
As a result of this enquiry, it was clear that the problems of the data transfer network would have to be separated from the supply of computers. Therefore a detailed specification was drawn up for the "ideal" minicomputer for this purpose, and work started on assessing the requirements of a message-transfer and data-transmission system.

This work took into account a number of fundamental and far-reaching decisions that were made at that time.

The first decision was that every computer would have a resident interpreter. This enabled the second decision (already mentioned above) to be taken, that all normal transfer of information between computers should be in the form of statements that could be interpreted or data to be used in the programs.

Just as important, this allows any computer to be used in an interactive mode from the teletype or other operator interaction device, for commissioning, testing and maintenance of the equipment connected to it.

The third decision was that no one computer would be the over-all master of the system. The message-transfer system would be designed so that any computer could pass a message to any other without a preset master-slave relationship. This implied that the system must be completely symmetrical and transparent.

The fourth decision was that a distributed data-base, with specialized data-base handlers for the different types of equipment, would be used. The advantages of this are discussed in Section 12.

These decisions were necessary before a fifth could be made -- to use separate minicomputers to perform the various duties required at the control centre. Without them, the organization of such a multi-minicomputer system would have required a formidable amount of software effort. With them, the system could be built up of modules which had clearly defined interfaces and could therefore be implemented independently.

The main penalty to be paid for such an approach is that the individual computers in the system have to be somewhat more powerful than would be needed in a system which was exactly tailored to the individual requirements at each point. In the latter case, however, not only would the software effort required be much greater, but the start of the work would have been considerably delayed, since at that time many of the requirements could only be guessed at.

The layout of the system arrived at as a result of these decisions is shown in Fig. 5. This does not differ fundamentally from that in the preliminary enquiry, except that the "process controllers" have become more powerful minicomputers, each having an interpreter and so giving the possibility of "stand-alone" operation. The layout has subsequently been modified by the addition of a second computer in the North Experimental Area, owing to increased requirements, the addition of a computer in BA7, and the deletion of the computer to interface with the CPS, as the data to be transferred between the two accelerators proved to be within the capability of a long-range extension to the multiplex system on the Injection computer in BA1.

A specification was prepared for the "ideal" computer for this network, in which as much stress was placed on the software needed as on the hardware, and tenders were requested
Fig. 5  Computer system for invitation to tender
from all manufacturers who had given a positive answer to the preliminary enquiry. The computer that came closest to the specification was the Norsk Data NORD-10, and 24 of these were ordered for the system.

8. THE MESSAGE-TRANSFER SYSTEM

The most difficult part of the whole equipment to specify was the message-transfer system, 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.

It has already been noted 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 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. For example, if the detection of an alarm condition caused the immediate transmission of a message to the alarm computer, certain faults could cause the generation of a very large number of separate messages, which could put a severe load on the links and the buffer storage of the message-transfer computer. This load would be significantly reduced if the satellite computer stored the alarm messages and sent them to the alarm computer at fixed intervals, or when it had a full block.

There are three main parameters which define the performance of a message-transfer system. These are: the maximum data transfer rate between any two computers when they have the full facilities available; the maximum total transfer rate when many computers are transmitting messages simultaneously; and the response time from the origination of a message from a process in one computer to its availability in another process in a second computer.

The data transfer rate necessary between two computers was largely determined by the decision to use the library to provide mass storage facilities for those computers that had no drum. This would not be worth while unless files could be transferred at the order of 10,000 words/second. This would also allow the reloading of a computer from the library in a few seconds.

The question of total transfer rate was more difficult to assess, but an attempt was made and this resulted in an estimated mean rate of 30,000 words/second, if uniformly distributed over the cycle.

The assessments of load were made by considering estimated peak conditions for any one computer and then taking the sum of these for the whole system. This would be unrealistically generous for a system where there could be confidence in the estimates. In this case, where much of the accelerator equipment was still in the design stage and the future developments could only be guessed at, it seemed prudent to plan the system on the basis of the sum-of-peaks of the estimated requirements, which should give some surplus capacity for the future.

As a result, a specification was drawn up calling for a message-switching system that could handle a total message transfer rate of 30,000 words/second between a maximum of 32 computers. Because of an expected system requirement, it was specified that it must be possible to send a short message, having the highest priority, from one computer to another in a total elapsed time not exceeding 5 msec, including all software overheads. This was required because, although it had been decided that any control loop that required correction during a cycle would only involve a single computer, it might be necessary to take a measurement, such as the betatron oscillation frequency, with equipment connected to one computer just before extraction and then set an element of the extraction system, controlled by another computer, to the appropriate value in a time of the order of 10 msec.
The NORD-10 computer was specified for the message-transfer system, to be consistent with the other computers. The data links had to be fully duplex and use the "POD" cable, which had been standardized for the SPS and was already being laid between some of the auxiliary buildings. This cable, originally developed for television relay use, is composed of a number of "quads", each quad being made up of a "video" pair and an "audio" pair, twisted together. Two quads could be used for each duplex link.

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.

An invitation to tender against this specification was sent to a large number of firms, but only three offers were received. All concerned proposals for a packet-switching system for both short and long messages, but none of them met the specification in all respects, and there were defects in the methods of interfacing the data links and in the protocol for the exchange of messages. Anticipating this possibility, some design work had continued at CERN, and an amendment to the specification was produced in which the detailed protocol was laid down.

Since it was expected that most of the messages would be relatively short, it was decided that the blocks should contain a maximum of 64 words, with a four-word header, making a packet of 68 words. The packets would be transmitted over the "video" pair of the quad and input to the computer through Direct Memory Access (DMA), while the control would be carried out by use of the "audio" pair, using programmed input-output (PIO). By the use of special hardware in the audio input interface, this pair could be treated as providing up to 16 separate single-bit control channels.

In a "store-and-forward" system, the message-transfer computer acts first as destination and then as a source for messages, and if it is "transparent", i.e. no modification is made to the message in passing it on, the protocol required for the message transfer is not affected by its presence. Therefore, to simplify the explanation, the basic case of communication between two computers can be considered, either a satellite to the message-transfer computer, or vice versa.

With most transmission systems, it is necessary to make an enquiry "can you accept a message" to the recipient, followed by an affirmative reply, before the message can be transmitted. Because of the requirement for the transmission of a short message in 5 msec, it was necessary to keep this "handshake" overhead on the exchange of a packet to a minimum. Therefore, it was decided to design a system in which the acknowledgement, which is normally made at the end of each message, is also used to prepare for the next. This also has the advantage of reducing the number of interrupts that have to be serviced. With this system, after a packet has been received, the computer initializes a DMA input for 68 words from the data link and sets a bit in the control register of the other computer to say both "the packet has been received" and "ready for the next packet". This is the normal idle state. When one computer has a packet for the other, it can check the "ready" bit in its control register and, if set, output the packet immediately. To allow for short messages of less than one block, the interrupt in the receiving computer is not made on completion of DMA,
but from the control word register. When the sending computer has transmitted the packet, which may consist of from 5 to 68 words, it sends a word on the control line which sets the bit in the control register to denote "you have a message" and this raises an interrupt.

This method of operation requires the message-transfer computer to be able to have at least 25 DMA channels open at the same time, and to have a high probability that a word can be read into memory from any input channel register before the next word arrives 30 usec later. If this does not happen, an error detection circuit causes the retransmission of the block. The earlier designs, before the computer was chosen, assumed that 68-word input registers would be required, but these could be dispensed with when using the NORD-10 computer, which can deal with multiple interlaced DMAs at high speed.

Serious consideration was given to the security and error checking in the message-transfer system. A parity bit is added to each word of the packet, and an extra word is added at the end, giving the longitudinal parity or "check-sum". In the case of the control word, where it is even more important to detect any error, only 12 bits are used for control purposes and the spare 4 bits are used for a multiple check-sum. All these tests are carried out by the hardware. In addition, the data links were designed to reject a high level of common mode interference. Any error detected causes a bit to be set in the control word register of the originating computer, which calls for a retransmission of the last block or control word. A system which keeps a DMA channel open for long periods when no messages are being transmitted is more susceptible to interference, and so it was arranged that only signals that formed a sequence at the expected bit-rate were accepted.

Since the traffic estimates which had been made had a large factor of uncertainty, it was decided that the message-transfer computer should have some special programs which could be called on demand, in addition to the programs for the exchange of messages and for checking the operation of the data links periodically. These special programs would provide a log of the traffic between the various computers, and a "snapshot", at a given time, of the size of the various queues in the system. These would allow the traffic estimates that had been made to be checked, and the source of any possible congestion to be identified.

Very serious thought was given to the question of reliability of the message-transfer system. Reliability covers two aspects: the error checking and recovery in normal operation, which has already been discussed, and the effects of abnormal operation.

Abnormal operation includes overload conditions and failures. When the system is overloaded, it should "fail soft" by shedding some of the load rather than coming to a complete halt or corrupting the messages. This has been achieved by having a bit in the control word which, when set, instructs the satellites to stop sending long messages. When the free buffer space in the message-transfer computer falls to a certain level, it can set this bit in some or all of the satellites while it digests the temporary overload. Short messages of up to one block can still be sent, and so normal housekeeping operations continue, but long programs and large blocks of data, such as those required for complicated displays, will be delayed until the congestion is eased.

It is much more difficult to make effective provisions to deal with failures automatically. Since the computer is the heart of the system, the first thought is to provide a standby computer with change-over facilities, as a provision against hardware faults in the
computer. Whether the change-over is automatic or manual, the arrangements for this cause complication and introduce additional equipment that can fail. It has been the experience with some automatic systems that there have been more failures in the change-over system than in the system it was intended to protect.

A second possibility is to have a second computer, either sharing the load or acting as a "hot standby", and switching by software when one computer fails. One of the difficulties of such a system is that often failures do not just cause the computer to stop working, but make it work in an abnormal way, and are often intermittent. To sort all this out requires a very complicated software system, which will certainly take a long time to design, write, and debug, and will probably slow the system down appreciably.

Another thing to be taken into account is that the hardware components in the data-link terminals, including those which perform the parity checking, etc., are the same types as are used in the computers, and so there is no reason to believe that there would be any significant difference in reliability. Granted, such a failure might result in the loss of a single data link rather than the whole system, but this could be almost as serious for the SPS system.

Taking all this into account, it was thought that the only system which would give increased reliability and yet be simple to implement would be to have two complete message-transfer systems, with duplicate data links and interfaces to the satellite computers, and for each satellite to send alternate messages on each system. If any satellite gets consistent error returns from one link, it routes all its traffic on to the other link, informs the message-transfer computer that its partner is in difficulties, and sends an alarm message to the operator. The additional software required for such a system is minimal.

However, the SPS control system is designed so that nothing disastrous should happen if the message-transfer system goes out of action and, in any case, the operation of other parts of the system -- such as the main power supplies -- relies on the operation of a single computer. Therefore it was decided to fit a single system for the initial operation, but to make provision in the layout for a duplicate message-transfer and data-link system to be added later if it should prove desirable. In addition, the single system was designed to be as modular as possible, and it is planned to have sufficient diagnostic software to enable a faulty module to be found rapidly and replaced. With the software kept as simple as possible, once it has been debugged thoroughly it should be very stable, but nevertheless a drum has been added to the message-transfer computer to provide for rapid reload as an additional precaution.

One of the duties of the message-transfer system that has not been mentioned so far is to provide facilities for remote loading of the satellite computers. Although all computers are fitted with a mains-failure protection system, the accelerator conditions will not be the same after a mains interruption, and the simplest recovery procedure may be to reload with a new system. In addition, there will be times when it is desirable to be able to change the system in a satellite computer from the centre, since some of the satellites are up to 4 km away by road, and not all the satellite computers will have means for loading the system locally.
The remote loading is carried out as follows: a command given to the message-transfer computer, with the number of the computer concerned, causes a certain control word to be sent repeatedly to the data-link hardware connected to that computer. Reception of these words, at the correct timing, causes the bootstrap line of the computer internal bus to be energized, forcing the computer to jump to the starting address of the bootstrap program, which will later be held in read-only memory. The message-transfer computer also sends a message to the library computer requesting it to send the appropriate core image, block by block, to the computer concerned. The bootstrap program in that computer takes each block as it arrives and assembles it into the core.

9. THE TIMING SYSTEM

Many parts of the SPS equipment require timing pulses to trigger them at particular points in the cycle and in a few cases these points are fixed with respect to the cycle, but in others the required timing will change with operating conditions of the accelerator.

Some earlier accelerators, such as the CPS, use a motor-generator set for the main power supply and, as the speed of this can vary, the machine cycle is not fixed with respect to clock time. This complicates the timing system and requires the distribution of a number of different trains of pulses. The situation is simpler in the case of the SPS, since the whole magnet cycle is fixed with respect to clock time, and almost all the timing requirements can be provided by a single train of pulses. However, since the start of each cycle is dependent on the time of injection of a beam of protons from the CPS, which does not have a fixed repetition rate, there must at least be another pulse to start the SPS clock each cycle. If all the SPS cycles had to be identical this would be sufficient and all timings could be specified as so many clock pulses after injection. However, it is planned to use the SPS in a double cycle mode, e.g. alternating between a 200 GeV cycle for the West Area and a 400 GeV cycle for the North Area. In such a situation, the timing of some parts will have to be the same for both cycles, but for others the timing will have to be different, or the triggers suppressed.

To cater for this type of operation, an "event" train was introduced. For each "event" in the cycle, such as the start of magnetic field rise, fast extraction, etc., a coded pulse is generated and distributed, together with the clock pulses, round all the equipment. A trigger pulse can then be generated at such an event, or a given number of clock pulses afterwards. In the case of the double cycle, some of the event pulses are repeated at the appropriate time in the two cycles, and others occur in only one of the cycles. An additional advantage of this scheme is that a number of associated triggers can be timed with respect to an "event", and the timing of all of them moved with respect to the cycle by just varying the timing of the event, rather than having to readjust them all individually.

It was decided to provide a series of standard timing units to be used throughout the SPS, which could be preset to provide trigger pulses at a given number of clock pulses after a given event. Since it was necessary to set these parameters remotely through the computer system, and the most convenient way of doing this is through the MFX system, it was decided to incorporate the timing system into the MFX, as shown in Fig. 6. This means that timing pulses can be obtained anywhere in the system where the multiplex is available and that no separate timing distribution system has to be provided, except in the experimental areas where the multiplex system is not being used.
Fig. 6 General purpose multiplex system with timing
Timing modules can be plugged into any of the multiplex crates that have the special station controller. Registers in a timing module can be set by instruction from the computer to select the event required and the delay after that event for the production of a pulse to trigger external equipment. A second pulse can be produced with a variable delay after the first, and an external clock can be used instead of the internal 1 msec intervals for providing the delays.

10. THE SOFTWARE SYSTEM

It has already been mentioned that it was decided to provide a command language interpreter for the main control programs for the accelerator, and the reasons for this should be explained.

In the majority of computer control systems in the past, the control programs have been written in the assembly language of the computer concerned. Each program, after being written, has to be assembled, and the resulting machine code linked into the rest of the system and loaded. Any error made in writing the program can only be corrected by reassembling, relinking, and reloading, which usually requires taking the computer off-line. A more serious disadvantage is that an error in a program can not only cause that program to do something unexpected, but it can also cause the whole computer to come to a halt, from which the only recovery is to reload the complete system. This is known as a "software crash". Since the assembly language is particular to a computer, and the syntax reflects the detailed method by which the computer performs an operation rather than the basic logical sequence of operations, a skilled programmer is needed to write all but the simplest assembly language programs, and not many accelerator experts have such a skill. Thus, apart from assembly language programming being long and tedious, it also requires extremely close co-operation between the programmers and the accelerator experts.

To avoid both the large number of programmers and the necessity for the very close co-operation which would be needed for a system as complex as the SPS, 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 doing this, 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.

It has been asked why a widely known language such as FORTRAN is not used more frequently for control purposes. FORTRAN is well suited for solving mathematical problems, but in a control system this type of operation is only a small fraction of the total work. Where FORTRAN has been used for real-time control purposes, it has been necessary to add special input/output instructions or functions. Associated with these functions there is usually a special process data structure, which has to be different from the number data structure used in the mathematical part of the program owing to the restrictions of FORTRAN. This has led to systems that are complex, inflexible, and difficult to learn. In addition, a FORTRAN program has to be compiled, linked, and loaded into the system before it can be run, as in the case of an assembly language program, and is difficult to debug without more sophisticated facilities than are usually provided in even the larger minicomputers.
Apart from the failure to provide the interactive facilities desired, the use of FORTRAN as the main programming language would only have been possible in the SPS system by increasing the size of the satellite computers very considerably, and it would not have reduced the amount of assembly language programming needed to provide the links to the hardware.

The use of an interpreter solves many of the problems, and although it has some disadvantages, these are far outweighed by the advantages in most control situations. An interpreter 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 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 where higher speed is necessary.

The decisions, mentioned earlier, 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. This was at a time before the computer was chosen, when it seemed likely that any manufacturer's real-time operating system that could be offered would require a large amount of core store. However, since the operating system SINTRAN II, which only occupies about 3K of core store, was available with the NORD-10 computer, the full development of the "virtual computer" concept was not justified.

The SINTRAN II operating system and its development are discussed later.

11. 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.

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. A test system was set up using a PDP-11 computer with a CAMAC controller and a single crate. A number of functions were added to the DEC BASIC interpreter, to provide the interface with CAMAC, etc., and also the changes and additions necessary to use BASIC under the disk operating system (DOS) were made.

Although only a few simple devices were attached to the CAMAC to demonstrate the elementary control operations, and the only output device for the operator was a storage display tube, the system was sufficient to demonstrate the advantages of the interpretive approach. Should there have been an interpreter for BASIC available for the computer chosen for the system, this would probably have been used as the control language for the SPS. However 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 TELECOMP (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 SNOROL 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. A NODAL interpreter which operates under DOS has also been written for the PDP-11 series of computers, but this does not have the string-handling facilities.

A manual is available which describes the NODAL language and its properties\(^3\).

12. DATA MODULES

In addition to a high-level language, it is necessary to have some lower-level modules to perform the actions necessary to access the hardware. These involve a knowledge of the hardware addresses, the desired state and how to reach it, calibration constants, etc.

In the earliest systems which used computers for process control, there was usually a single program in the computer and this program had written into it all the detailed information about the hardware. With more complicated systems, where it was required to have more than one program which could operate on the same variable, it was necessary to communicate between programs, and this was done by having a common "data-base", containing the values of variables, etc., which could be accessed by any program. The next stage was to remove the necessity for the individual programs to carry out the detailed work of accessing the hardware by having a common data-base handler program. Individual programs only need to pass requests for access to the hardware to this data-base handler, and do not then need to know the details of the method by which the access is performed.

This system of central data-base has worked well with many computer systems, but suffers from some disadvantages. The amount of data required to be stored in the data-base for different types of equipment may vary, so that a uniform system covering all equipment may be complicated or inefficient in storage requirements. A more serious disadvantage is that the data-base handler has to deal with all types of equipment, and so the full information must be available before work can start on designing and writing it. If additional equipment has to be added later, with different requirements for accessing the hardware, the data-base handler has to be modified, with the risk of introducing errors which can affect the operation of parts that have previously been debugged and operate satisfactorily.

Another problem arises if a central data bank is used with a distributed computer system. Each simple input or output action called for by a program may require several interactions between the data-base and the hardware: to set a value into a piece of equipment may require checks on the status of the equipment, the setting of the value, and then a check that the value has been attained. This causes a large amount of traffic on the data links between the central computer, which handles the data-bank, and the computers directly coupled to the equipment. In some cases, proposals have been made to reduce this extra traffic by having subsets of the central data-base in the satellite computers and to share the work between the central data-base handler and special handlers in the satellites. Apart from increasing the storage required, it is always dangerous to have two lots of tables accessible to the system which are supposed to have the same entries, but which may differ during the execution of a program.

For the SPS, it was decided to use a fully distributed data-base system. In this the equipment, as far as its interaction with the control system is concerned, is broken down into the smallest separately controllable items, such as power supplies, stepping motors, etc. All the items of one type, irrespective of the system in which they are used, are
handled by a specialized handler, called a data module, which manages a data table containing information on all the items of that type connected to a given computer.

This gives several advantages. Since each handler and data table is concerned with only 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.

Provision has been made for 64 different types of data module, and so far it looks as if just over 50 will be needed. In any one computer, of course, only the data modules needed for the equipment connected to that computer will be installed, and the number of data modules in the individual computers will vary from zero (in the console computers) to about 30 in the general-purpose computer in Auxiliary Building 4.

It might be argued that a number of separate specialized data table handlers in one computer use more space than a single general-purpose handler. This is not necessarily true. The separate specialized handlers can share many re-entrant subroutines, so the advantage to be gained from closer integration would be small. To gain even this small advantage would necessitate having a special version of the general-purpose handler for each computer. This would involve extra work, and the advantage of being able to implement the system progressively would be lost.

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

\[ \text{SET MAGNET (4, #CUR) = 125.4} \] .

In this case, the property denoted 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. A description of the data module concept is given elsewhere*).

It is of interest to note that the British firm ICI Ltd., after some years of experience with many different types of process control computer systems, have just recently published the decision to base future systems on a very similar two-level basis. At the lower level, there is a series of data sets defining the plant input-output and the sequence operations, called Plant Sequences. They are particular to a piece of plant such as a pump, reactor, etc. At the higher level, programs called Process Sequences written in the Plant Control Language are interpreted, and the required calls to the Plant Sequences made. The Plant Sequences do not have to be rewritten for each new installation that uses the same type of plant, and programming in the interpretive Plant Control Language is simple, so it is hoped to reduce considerably the amount of time and effort needed to bring a new installation into service.
13. **THE OPERATING SYSTEM**

As explained earlier, it seemed that it would only be possible to provide a multi-computer 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.

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 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 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.

When the detailed operation of SINTRAN II was investigated to determine the best way to make the additions, it became clear that some changes were also necessary. One of the disadvantages of SINTRAN II is that if a program goes into a waiting state, it prevents the start of any other programs at that priority level or lower. This is because it uses a "stack" to provide working area for the programs, and space on this is dynamically allocated. To reduce the effect of this, the stack was divided into a number of fixed-length partitions, so that a program in a waiting state only prevented the start of programs within that partition. It was desirable to have a system which could access all files on all devices in a uniform way and, to simplify the provision of this, I/O system buffering was introduced, using dynamic allocation of 64-word blocks of core. In addition to the links for NODAL and the CAMAC and MPX I/O routines, a number of other small changes were also introduced and the resulting system was sufficiently different from SINTRAN II to be given a new name. It was called SYNTRON, since it was designed for synchrotron control, but unfortunately the name is not sufficiently different from SINTRAN to avoid confusion.

As soon as these changes had been made, SYNTRON was used for stand-alone operation of single computer systems while the work necessary to add the filing system and the handlers for the messages to and from other computers was being carried out.

When a program in any computer calls a file by name, the filing system is required to search the in-core files and then the local drum, where present. If the file is not found locally then a message must be generated calling for the file to be sent from the library. Thus the library can provide mass storage for the computers that have no drums, and an extension of the storage for those that have.

In making these additions to SYNTRON it was necessary to make some modifications, and the opportunity was taken to introduce a system of resource allocation and device reservation, and to introduce additional security measures. This resulted in SYNTRON II, which is intended to be the final operating system for all the computers except one, the service computer. Despite the considerable additional facilities, including the message handling, the core resident part of SYNTRON II occupies under 6K.
The service computer, which is the largest and has the greatest number of peripherals, is expected to be used for background work and program development, as well as providing the facilities for carrying out computations beyond the capability of the other computers, and so a more comprehensive operating system is required.

The Norsk Data SINTRAN III provides all the facilities required, but this is a completely different system rather than an enhancement of SINTRAN II. Therefore the decision had to be made whether to add a time-sharing system to SYNTRON II, or to write an interface to SINTRAN III so that, seen from the message-transfer system, it looks like SYNTRON II. The latter solution was chosen as it seemed to involve much less work, especially at the peripheral driver level. This also has the advantage that the NORD-10 has been adopted as one of the standard computers for use with the experimental apparatus, and SINTRAN III will also be used for this application.

At the time of writing, the only major system work still to be done is the filing system for the library computer. At present, a simplified system is in operation, with a limited number of fixed length files, but it is expected to have the full facilities available before the end of the year.

14. THE OPERATOR INTERFACE

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.

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.

At the other extreme, it is theoretically possible for a large accelerator such as the SPS to be controlled entirely from a teletype. While infinitely flexible, this would be intolerably slow and inconvenient. The requirement is to find a solution lying between these two extremes.

The most versatile computer output device is a cathode-ray tube, and this can also be an input device if equipped with light-pen or other means of sensing operator actions. The
other types of input devices used most frequently include a knob connected to an incremental encoder to change a single parameter and a rolling ball connected to two encoders to set two parameters, usually represented by the position of a cursor on a screen.

The principles adopted for the design of the system for the SPS were as follows:

- The control console hardware should be as completely general purpose as possible, the transformation for a particular use being made by the software.

- 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, 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 are too expensive. The original idea was to use a fixed-head disk to store the information for the raster scan, as a commercial system for this was available. However, it was found that this system could not provide all the features required and would involve complicated programming. The solution adopted is discussed below.

- 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 to be "pressed". Touch screens using a matrix of wires on the surface of a cathode-ray tube had been used before, as well as systems using infrared light and ultrasonics, but all of these suffered from some defect and so a new type using the change of capacitance between transparent electrodes placed in front of a cathode-ray tube was developed.

- The other operator interface devices would be a rolling 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.
Since only one knob was to be provided, it seemed desirable to have this as versatile as possible, and so a mechanism was developed whereby the knob could have a variable resistance to turning, or have a spring-loaded return-to-zero action, or be indexed like a rotary switch\(^5\). The keyboard provides the full facilities of NODAL, but it is expected it will be used mainly for issuing immediate commands. The resulting console layout is shown in Fig. 7. In addition to the facilities described above, provision is made for the observation of waveform signals on conventional oscilloscopes; the choice of signals being made through the computer system, using one of the touch screens described above.

Fig. 7 One of the three control desks
The television monitors can show either "live" pictures from cameras on beam monitors or in the auxiliary buildings, or displays. The displays are of two types: characters and graphics. The character displays are generated by special hardware and can be in black-and-white or three colours. Upper and lower case characters are available together with a number of line symbols which can be used to form simple mimic diagrams. The graphic displays are provided by scan-converters. The display computer can draw vectors on the screen of a scan converter, which stores the resulting display and provides the repetitive raster scan output. Since the scan converters are relatively expensive, a pool of five is provided for the use of the three consoles. Multicolour graphics would require the reservation of three of the scan converters for one display, so normally graphic displays are in a single colour, but a video signal-switching network allows the superposition of a multicolour character pattern on a graphic display.

15. APPLICATIONS SOFTWARE

The applications programs can be divided into two main types: surveillance programs that are scheduled to run in the satellite computers without action from the operator, and control programs that run in the console computers, as called for by the operator. To simplify the organization of error returns, and to allow the eventual analysis of faults and elimination of consequentials, all error messages from the surveillance programs are routed to the Alarm computer, which decodes and presents them on a special display screen on each of the consoles.

The control programs are selected by "buttons" on the touch screen. At the start, the operator is presented with the choice between the main systems of the accelerator: main power supplies, radio frequency system, vacuum system, etc. On touching the appropriate button, the legends change to give the subsystems which can be controlled and, after a further choice, the legends change again to give the actions that can be carried out on that subsystem. On the choice of an action, a program is started which may provide a display to allow further choices by setting a cursor, or ask for input parameters through the use of the other console facilities.

This program will be in NODAL and may call for files to be sent from the library, for small subsections to be executed in one or more of the satellite computers and the results returned, for displays to be provided by the Display computer, or possibly for some calculations to be performed in the Service computer.

The method of choosing the program to be run has been likened to a tree, where the trunk has branches on it representing the major systems; these have twigs, or subsystems; and the twigs have leaves, which are the programs. A special button is provided on the console to go back to the previous choice, and repeated operation of this will take the operator back to the trunk, from which it is possible to choose another branch. However, in normal operation it should not be necessary to go right back to the trunk frequently, as the same leaf program can be attached to more than one twig, so a branch can have attached to it all that is necessary for the operation of that system, even if it means using subsystems of another branch.
16. PROTECTION

One disadvantage of a system as flexible as that developed for the SPS, in which any variable can be accessed by name from any of the computers, is that it is easy to perform an incorrect action, either by accident or design, so some form of protection must be incorporated.

Secrecy is not involved in this case, so it should be possible for anyone to read a file or the status of the equipment, or make a measurement; but protection is needed to prevent unauthorized operation of the equipment or modification of the data tables or program files. Some differential protection is also needed. Operators obviously need to be able to operate all parts of the equipment, but they may be prevented from making changes to limit values, etc., and they may not be allowed to make changes to program files; experimenters may be allowed to change some beam-line elements but must not be able to alter anything else, and so on.

Since it had been decided to use plastic identity cards, with the name and CERN number of the owner coded in punched holes, for control of access into the accelerator, it was logical to require the use of the same card for "logging" into the system before any protected action could be carried out at the control centre. When the card is inserted into the reader on the console, the CERN number is read and a program is called which looks up a table to find the correct password. For local operation at one of the satellites, where there is no reader for the password card, the user must type in his CERN number, followed by a check word, chosen by him, as a protection against someone else typing in his CERN number. In some cases, as for the experimenters, a particular input device, such as a teletype, will have a fixed password.

With this procedure, even if someone knows the correct password for gaining access to some apparatus, he has no means of inserting it into the system, except by borrowing the card from someone who has the appropriate capability.

The protection for operation of equipment is at the data module level. Anyone can call for a program to be run, but when this requires an access to the equipment, the system sends the password with the request, and the data module checks if that particular password covers the requested action. If not, it returns the error message "unauthorized action". In the case of writing into protected files, the password is checked against the file header by the file handler.

To provide the required flexibility the password has three parts, one of which, the index, is only used for the division of files into groups. The other two, the section number and the capability word, are used in the protection scheme. The section number refers to a small group of people with responsibility for a particular type of equipment, and the correct section number might be required to perform certain special actions on the equipment or to set such things as maximum allowed value or multiplexer address into the data tables. The capability word determines what types of equipment may be controlled in the normal manner, each bit referring to a broad class of equipment, such as RF, vacuum, main power supplies, etc.

The ability is provided in the data module to specify whether a "write" property requires a particular section and/or capability. Such protection usually applies to all units of that type, but protection at the individual unit level can also be provided where necessary.
In the case of files, the "owner" will be able to specify whether protection is needed and, if so, what section and/or capability is required, or whether the file is "locked" for read-only.

17. RELIABILITY

Reliability has already been mentioned in connection with the message-transfer system; but this is only one part, and the possibilities and consequences of failures in all parts of the control system must be taken into account in the over-all design.

First of all, the level of reliability that is required has to be established. An accelerator is a complex collection of individual pieces of equipment, many of them working under high stress of one sort or another, and often going to the limits of technological achievement. It is therefore surprising, not that accelerators suffer from breakdowns, but that many of them are running satisfactorily for such a large proportion of the scheduled time. For the last two years, the CPS has provided beams for the experiments for over 90% of the scheduled time, and other accelerators have come close to this achievement. Obviously the SPS cannot do as well as the CPS (which will supply the protons for the SPS to accelerate), since the SPS will have its own breakdowns, but hopefully, after the running-in period, these will not more than double the down-time, giving an over-all availability of between 80% and 90%. In such a situation, it would be annoying but not too serious if the control system made a contribution of 1% to the down-time, but it would be serious if it contributed 5%.

There will inevitably be failures in the control system, and the problem must be attacked on two fronts, one to minimize the failures, and the other to reduce the effect of failures on the operation of the accelerator. To help with the latter, the policy was adopted, as far as possible, that once a command was given for some part of the accelerator equipment to operate in a certain way, it should continue to do so, cycle by cycle, until commanded to do something different. This is not possible in some cases -- for example the main magnet power supply requires the active interaction of a computer throughout each cycle -- but in others special provisions can be made, such as the fitting of local memories to the function generators for the pulsed power supplies.

In a control system using computers, failures can be of two types: hardware and software. The ability for the accelerator equipment to carry on for a while without intervention from the control system is most important in the latter case.

In a working system, software failures are usually caused by the occurrence of a combination of circumstances that was not foreseen in the design of the software; this causes one or more of the computers to go into a loop or otherwise lock out the subsequent executions of the required programs, and the usual method of recovery is to reload the system.

This action can be carried out in a time of the order of seconds rather than minutes, and so the ability for most of the accelerator equipment to continue to operate for a few cycles without intervention from the control system is clearly an advantage.

The situation is different for hardware faults, since the recovery is likely to take much longer, unless duplicate equipment can be switched in automatically.
For a particular piece of equipment, the criterion of unavailability can be defined as the mean time to repair (MTTR) divided by the mean time before failure (MTBF), and the smaller this figure the smaller the contribution this equipment will make to the total down-time. The MTBF can be maximized by choice of the most reliable components, and the MTTR can be minimized by modular construction and the provision of means for rapid diagnosis.

One of the difficulties in choosing the components is that the development is so rapid that reliable MTBF figures are not usually available for electronic equipment until it is considered to be obsolete, and so it is generally necessary to rely more on the past reputation of a manufacturer than on the actual figures for the equipment it is planned to use. During the initial operation of the equipment, it will be found that some components will have a higher than average rate of failure, and it may be necessary to consider changes in order to improve reliability.

Components can also change their characteristics without actual failure, and reliability can also be improved by avoiding circuits that depend on critical values of components for their operation. However, to make a circuit less dependent on the properties of the components may require the use of a greater total number of components, so a balance must be struck.

These means of increasing the MTBF have been borne in mind in the selection of components for the SPS, but the major effort has been spent in trying to minimize the MTTR. Wherever possible, plug-in modules or printed circuit cards have been used, and those used in any appreciable number have been tested in computer-controlled test stands that have reproduced the expected conditions of use. The computer programs employed for these tests are being used in modified form to find faults in the system, and can form the basis for the rapid diagnosis programs that will be available for the accelerator operator to call when faults occur.

The case for duplication of equipment with automatic change-over is difficult to justify in this type of application, and brings in additional problems, as mentioned above in the discussion on the message-transfer system. However, it is intended to apply the "hot standby" philosophy in the case of the Library computer, since it provides the majority of the mass storage facilities for the whole system. None of the facilities to be provided by the Service computer will be vital to the normal routine operation of the accelerator, and so it will be possible to use it as a standby for the Library computer, if the latter fails, by exchanging the cartridge disk packs and changing the computer number in the message-transfer system. The Service computer can also act as replacement for the Alarm and Display computers, but this requires the change-over of some cables in addition.

18. PRELIMINARY ASSESSMENT

So far, experience has only been obtained in the testing and commissioning of various parts of the accelerator equipment. As the first of the computers arrived, they were installed in laboratories and used for development and testing of interface hardware and software, and the Service computer was set up to provide time-sharing facilities for about 10 users.

The first experience of using the interpreter for testing user equipment was obtained by installing a NORD-10 in the assembly hall, to be used for detail testing of the vacuum
pump and orbit correction dipole power supplies on reception from the manufacturer, and for the control of prototype fast kicker pulsers.

Direct calls to CAMAC and the multiplex system from the interpreter were used for this application, since the data modules were not available, and three interactive buffers were provided, so the three activities could go on simultaneously without interference.

The test programs were written by the engineers or technicians concerned and the software system proved to be very reliable; the interpreter forming an effective filter between the user and the system. It has also given little trouble on the hardware side, with only one component failure in over 18 months of operation.

A control console with computer was set up at an early stage of the project, so that operator interface equipment could be tried out, routines to simplify the writing of display programs could be developed, and operation of subsystems of the accelerator could be simulated. This has proved invaluable for these purposes, and has also helped to "indoctrinate" the engineers and physicists in the use of the NOADL language for writing their own application programs, rather than having professional programmers do it for them.

The situation at the time of writing is that the central control room is nearly complete, one console being used for the simulation and another being available for use with the message-transfer system and the other central computers to control equipment in several of the auxiliary buildings. About half the data modules required are available in usable form, and the application programs to control equipment from the centre are gradually coming into use. This is enabling the "bugs" in the system to be found and rectified.

With this relatively small amount of experience to go on, it is only possible to make a very preliminary assessment of the correctness or otherwise of the choices made. However, one thing has become quite evident: it would have been impossible to provide the facilities that are expected to be available at the time of the start-up of the machine without the use of the interpreter and data module concept. Even if an army of programmers had been available, and the other advantages of the interpreter approach had not been required, it would not have been possible to provide conventional compiled and assembled programs to do all the things required on the time scale, since details of the control requirements for many parts of the equipment were not available until very late in the program.

With the system chosen, programs could be written in small modules as the information became available, and the protection provided by the interpreter has minimized the difficulties due to the interaction or interference between different programs. This causes the greatest headaches and consumes most of the debugging time with conventional systems even when a single computer is used, and the difficulties are increased enormously with a multicomputer system. Once the data modules are complete, the writing of the application programs is relatively simple, as the present users have found, and this should help to prevent the inevitable last-minute rush from delaying the start-up of the accelerator.

The decision to use separate computers at the centre rather than a single large one has enabled parts of the system to be developed independently, and no significant disadvantages have shown up.

The multiplex system is working well, and a recent comparison made in connection with another application has confirmed the considerable saving in cost over CAMAC, the particular
configuration considered showing a cost ratio of more than 2.5, and the absence of direct interrupt facilities have not proved a disadvantage so far. An additional module to provide interrupts is being designed and will be available for future application of the system.

A report giving a detailed assessment of the performance of the control system, the difficulties and failures experienced, and changes that seem desirable in the light of such experience will be issued when sufficient evidence is available.

References to papers giving detailed information on various parts of the control system can be found elsewhere⁵).
REFERENCES

1) A design of the European 300 GeV research facilities, MC/60 (CERN, Geneva, 2 December, 1970).


4) M.C. Crowley-Milling, Data modules, CERN Internal Report, Lab. II-CO/Int/00/75-1 (1975).

5) F. Beck and B. Stumpe, Two devices for operator interaction in the central control of the new CERN accelerator, CERN 73-6 (1973).