/afcs

MPS/CCI Note 74-27
Rev. - 17.9.1974 *)

THE NEW CONTROL SYSTEM FOR THE MPS

An Interim Report

by

E. Asséo, A. Daneels, H.E. Davies, C. Faraut, G. Jennings,
J. Gamble, J.H.B. Madsen, P. Mead, W. Remmer, J. Stark,
T. Streater, A. Silverman

Contents

1. Introduction
2. Assumptions
3. Standards
4. The hardware configuration for the control system
5. List of users' facilities
6. Lab. computers
7. Front End computers
8. User access to consoles
9. Operating systems and languages
10. Program development facilities
11. Data Base System
12. Transition from IBM 1800
13. System performance
14. The analogue function generator
15. User education

* * *

*) A draft version was dated 24th June 1974

1. INTRODUCTION

1.1. The purpose of this report

We wish to present in this report an overall view of the future MPS control system, emphasizing the users' aspects. The draft of this report has been presented to the CUC and the MAC*). Observations raised during the discussions are incorporated in this version.

Major design changes have been made recently : the centralized data base system described in this report has been abandoned in favour of a distributed system.**) The tasks of the various Front End Computers are still under considerable elaboration. Therefore, this report is an interim one and will be updated as soon as further progress has been made.

It contains, however, the external specifications for the system and this is the main reason to replace the draft by this revised version - although being outdated already !

We are aware that this report reflects the way it has been written : in two weeks time and by several authors ; no major effort could be spend on the editing. Too much design work is awaiting urgent attention.

1.2. Aims of the control system

From the start of our studies on the future MPS control system, we have aimed to provide an integrated computer control system for the MPS Division.

Studies in 1971 had as objectives for the control system :

a) overall control during normal running of the accelerator complex by a small crew of operators working in a central control room ;

b) to provide facilities to the accelerator researchers to perform their MD-type of work under favourable conditions (implies :

*) CUC : Computer Users Committee - see Minutes Meeting No. 8.
MAC : Machine and Area Committee - see Minutes Meeting No. 30, but also of Meeting No. 28 at which major design options were approved.

**) MPS/CCI/Note 74-44, Outline of the system software for the PS Control System, J. Gamble.

program development facilities, man - computer interfaces accessible to the non-expert).

In 1972 we inscribed an additional objective :

c) to provide good maintenance and testing facilities for equipment.

We also listed the main features the system should support :

d)
- rapid changes in machine conditions,
- production of stable and reproducible beams,
- rapid localization of faults or sources of instabilities,
- good communication with the SPS and other users.

The aims mentioned above are still valid. Aim c) has been emphasized, and we specify :

c') the system should allow - to a large extent - individual groups to set up and test their own hardware with the computer facilities.

1.3. Evolution in the design of the control system

From the beginning of our work - summer 1971 - we based our design on a centralized configuration : a central computer with satellite computers attached to it. For the first phase of the new control system we still intended to assign the central role to the IBM 1800. However, we were aware that the final solution should look different and we therefore aimed to build up the system with possibilities for a smooth continuation of developments. In fact, early in 1972 we had in mind to move towards a "twin configuration" : two equivalent main computers with a number of satellites attached to it.

The system configuration presented in this report - derived from earlier versions - reflects the facts which came up in 1973 and this year :

- construction of a new Linac,
- the intensity modulation involved a larger amount of hardware than estimated earlier (for PSB and new Linac),
- the policy decision to use CAMAC for new control applications.

An important consequence of these facts is the need to advance the introduction of satellites - now called Front End Computers (FEC). These FEC's are to be interfaced with CAMAC systems.

1.4. Medium term objectives

The re-orientation in the control project has introduced delays. To meet our first objectives :

- to provide the necessary support for operating at higher beam intensities and on complex cycles (including intensity modulation),
- to support the continuous ejection system,
- to support the new Linac project,
- to establish the link with the SPS and the ISR-TT_2 control system,

we concentrate on the elaboration of the basic software and continue in close collaboration with the future users to construct the system.

2. ASSUMPTIONS

During the review of the controls project in 1974, we made the following assumptions :

a) to make use of hardware existing or already developed (three PDP 11/45's, BIDUL and STAR),

b) no deadline on use of IBM 1800,

c) transformation from existing IBM 1800 based system must not cause delays or stoppage of normal PS operation,

d) the availability of the final system must be very high, back-up provided where possible,

e) to provide protection between users and within users' groups by working through a standard data base,

f) individual groups will accept to a large extent standardization (see chapter 3) in order to benefit from common experience and to reduce manpower required for system implementation and maintenance.

3. STANDARDS

a) The three main computers - PDP 11/45's - have identical configurations except for program development peripherals on a unibus extension.

b) Front End Computers and lab. computers have one of two standard configurations :

   i) PDP 11/10 (or 11/05) configuration which requires disk (11E10 or equivalent fixed head disk based system),

   ii) similar PDP 11/10 6or 11/05) configurations without disk.

   Applications requiring a more powerful FEC can be examined after realization of the system described here.

c) Languages :

i)	assembly language	: for programming only where other languages are not suitable
ii)	PL 11	: for systems and applications programming
iii)	ESAU interpreter	: applications - on main consoles and on lab. computers (a compiler version to be added later)
iv)	FORTRAN	: applications programming - on main computers

d) Operating systems :

i) RSX-11D on main computers

ii) RSX-11M on FEC's type b i)

iii) MINIMON on FEC's type b ii)

iv) users will follow any restrictions on the use of these operating systems or conventions for the general good of overall computer performance.

e) Consoles :

Each FEC will be equipped with a console which will be a subset of those planned for the new MCR.

f) CAMAC :

i) new systems will be interfaced via CAMAC

ii) FEC's will use a standard branch driver ; lab. computers can use the branch driver or a CC1

iii) crate controller, crate and supplies are standardized

iv) as far as possible CAMAC modules from a standard list will be used (list still needs to be established and will be frequently updated).

4. THE HARDWARE CONFIGURATION FOR THE CONTROL SYSTEM

Introduction

The basic guiding principle behind the proposed configuration is that it should be symmetric and modular. It must also follow as much as possible the relevant assumptions and standards as laid down in sections 2 and 3 of this report.

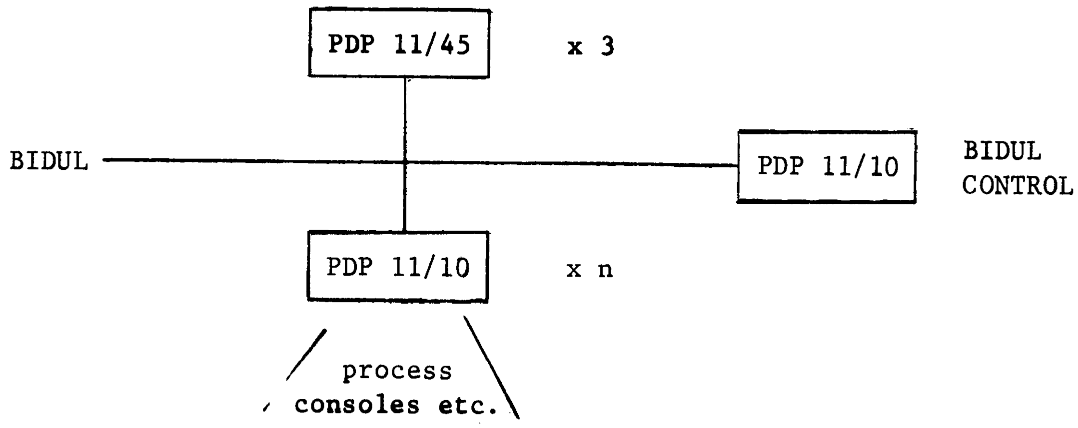
The main elements of the configuration are :

i) 3 PDP 11/45's - providing computer power,

ii) Front End Computers (FEC), generally PDP 11/10, which interface the computer system to various parts of the process,

iii) BIDUL, the transmission system which connects the various parts of the system together, and allows communication between them. BIDUL has its own PDP 11/10 to control it.

A much simplified diagram thus shows :



We may now consider, from the point of view of the configuration, the following aspects of the system :

a) the PDP 11/45 complex

b) BIDUL

c) the IBM 1800 - STAR system

d) the new console system

e) CAMAC and new applications

f) other front end computers.

a) <u>The PDP 11/45 complex</u>

The PDP 11/45 has been chosen as the basic computing element to replace the IBM 1800. This computer has a memory capacity which may go to 128 Kwords, and which for the configurations ordered, will be much larger than that on the IBM 1800. Besides this, it has a central processor which is several times faster. Thus in principle a single 11/45 should be capable of handling the current load with no problems. There are however several considerations which, taken together, cause the use of more than one PDP 11/45. With more than one 11/45, we are able to provide back-up in the case of the failure of one 11/45. We will also be able to do program development during normal PS operation. Also, as the new control system will be more heavily computer-controlled than at present, the capacity necessary to handle the load will be greater. A final consideration is the fact that, due to the time-scale for the construction of the new Linac, some computing facilities will need to be available well before the final system is finished. In order to satisfy this last constraint, a PDP 11/45 with a PDP 11/10 will be used in stand-alone mode, separated from the rest of the system. These two machines will then be re-integrated from the rest of the system at a later date and the 11/45 housed in the computer room.

To satisfy the other criteria, we have two other PDP 11/45's, one used for PS/PSB applications, the other for back-up and program development.

Each machine will have the same configurarions anticipated as follows :

96 Kwords memory
2 x RK05 cartridge drive (in future perhaps 1 x RP04 disk pack)
1 fixed head disk for the data base
1 Versatec printer-plotter
1 Decwriter

There will also be the following peripherals connected on a Unibus extension. They will be switchable to any PDP 11/45 via Unibus switches and normally attached to the machine used for program development.

2 xRP04*) disk packs for permanent files
1 x LP line printer
1 x CR card reader
8 x VT05 displays
1 x cassette drive

---

*) or equivalent

1 x DEC tape drive

1 x paper tape reader / punch

1 x RK05 cartridge drive

b) BIDUL

BIDUL is the transmission system designed to allow communication between any two computers attached to the network. As BIDUL has been extensively described in other notes*), we confine ourselves here to aspects concerning the configuration.

The main element of BIDUL is the BUS, which acts as the transmission path. We plan currently to have two Buses (for those people familiar with previous ideas on the configuration, see section 4.c), which will be identical. This gives us a back-up path in case of a bus failure, and also enables the transmission load to be shared. For symmetry, all computers are connected to all buses. BIDUL is controlled by a PDP 11/10, however if the transmission load saturates the CP of this machine, it may be upgraded to an 11/40. A maximum of 31 computers can be connected to one BUS.

c) The IBM 1800 - STAR system

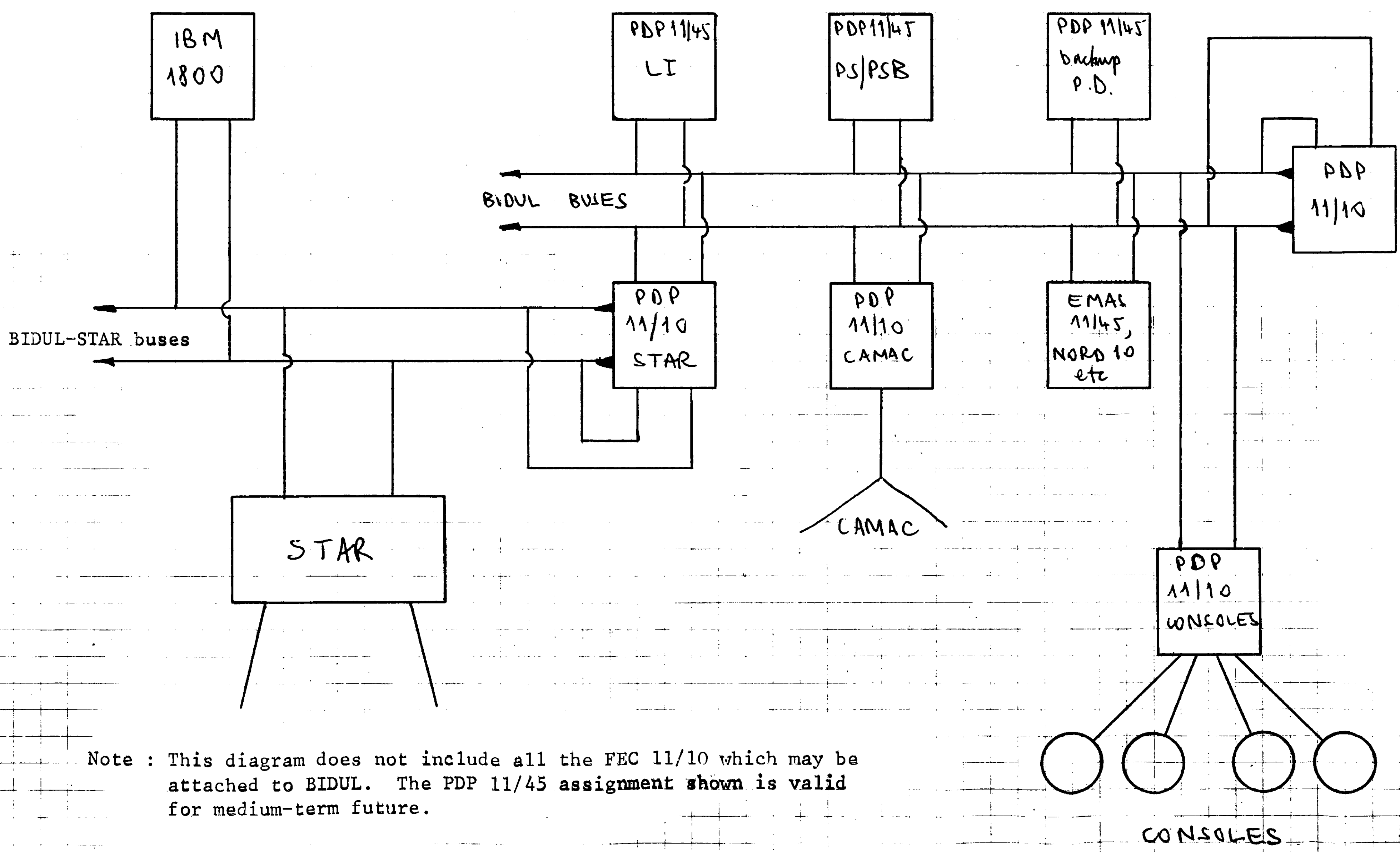
The existing control system consists of an IBM 1800 interfaced to the process through STAR. As the STAR represents a large investment, and more importantly has a lot of equipment attached to it, it is not feasible merely to transfer this hardware to the new system. Thus it is decided to keep the STAR. It is quite probable that equipment will be taken off the STAR little by little, once the new system is in operation, as and when necessary, particularly if new equipment of the same type is added to the new system.

It was originally proposed that the STAR be connected directly to BIDUL, and having the 1800 driving STAR through BIDUL. This approach has a number of disadvantages, however, which are :

---

*) Technology note D12
Technical notes can be consulted in E. Asséo's office.

MPS CONTROL SYSTEM - COMPUTER CONFIGURATION



Note : This diagram does not include all the FEC 11/10 which may be attached to BIDUL. The PDP 11/45 assignment shown is valid for medium-term future.

i) it would involve a great software effort in the IBM 1800,

ii) it would require two BIDUL buses (because of STAR transmission polarization). For these there would be no back-up,

iii) transition between the old and new systems would become more complicated.

Because of these problems it is now envisaged to connect the STAR to the new system via a PDP 11/10 FEC. A BIDUL switch will be inserted between the 11/10 and the STAR, to make STAR switchable between the 11/10 and the IBM 1800. This would enable the current system to function substantially unchanged, while development continues on the new system. Virtually no software effort on the IBM 1800 would be needed. See also Note MPS/CCI 74-24, Structure multiordinateur au PS, by E. Asséo, for a discussion at greater length.

d) The new console system

The new consoles will be connected to a console PDP 11/10. By this means we maintain the modularity of the system, all consoles being functionally equivalent.

e) CAMAC and new applications

It has been decided that new applications will be interfaced to the process via CAMAC. To simplify the interfacing of the CAMAC itself to the new computer complex, a PDP 11/10 FEC will be used, with the CAMAC attached to it. This avoids putting CAMAC directly on an 11/45 and thus avoids the question of which 11/45 to connect it to. If the CAMAC were connected to BIDUL directly, there would remain the problem of the LAM's*), thus we interface through a FEC.

f) Other Front End Computers

Other machines such as the EMAS PDP 11/45, and a NORD 10 used to communicate with the SPS will be interfaced to the new system via BIDUL.

---

*) Look At Me requests.

Conclusions

The proposed system should satisfy most of the necessary constraints. It is the most fail-safe that may be envisaged. If an 11/45 goes down, we lose only that machine, once the back-up path has been established, not any consoles or CAMAC attached to it. Since all computers are attached to all BIDUL buses, we have a back-up transmission path as well as load sharing. It should be possible to have a spare PDP 11/10 ready to replace any of the FEC's (for the EMAS 11/45 and the NORD 10, no back-up is possible). The one possible weak point is the communications controller 11/10, for which back-up is complicated as it has a lot of BIDUL hardware attached. Care will have to be taken with this machine to ensure reliability, for example by keeping the configuration simple.

The availability figures currently quoted for stand-alone minicomputers are not better than 0.96 to 0.97 (computer with peripherals, including disk). We do not yet know the figures for the PDP 11/45's but they are unlikely to be superior. The back-up facility we aim to introduce should provide an 0.99. availability on the 11/45 level.

No automatic back-up is availible on the FEC level. In case of breakdown, a repair or replacement must be made. Repairs will be easier on the FEC's which are located in the central computer room (spares on hand). Users may want - to obtain high availibility - to store control values required to drive elements in local independent memory instead of in the FEC.

5. LIST OF USERS' FACILITIES

a) Lab. computers for off-line hardware testing,

b) Front End computers for interfacing via CAMAC to the process ; functions performed : routine control and surveillance during normal operation, including support to multi-pulsing ; limited access from a local console.

B. Carpenter, Preliminary specifications for the communications monitor and message switcher software in the PS multi-computer system, MPS/CO/Note 73-9.

c) Access to consoles : on new MCR consoles for centralized operation of Linac, PSB and PS ; on midiconsoles connected to the Front End computers.

d) Program languages : high level and interpretive.

e) Program development facilities : a PDP 11/45 will be dedicated to this purpose.

f) Data Base System : providing access to and use of all process variables.

g) Transition from IBM 1800 : safeguard of present computer applications and facilities to implement new programs.

h) The analogue function generator : basic element in most of the controls application ; access via PDS-1 consoles.

The facilities listed are described in some detail in the following chapters.

6. LAB COMPUTERS [1)]

The purpose of acquiring Lab. computers is to allow hardware specialists and engineers to interactively test and debug their equipment, in particular CAMAC equipment, in their laboratories, prior to inserting it in the process. The Lab. computer is a PDP 11E10 which consists of :

- 11/10 CPU with 16 K memory (more memory will be required)
- dual drive magnetic tape cassette subsystem
- cartridge disk drive with RK05 disk (1.2 million words)
- DEC writer.

To this we propose to add :

- a display with alphanumeric and graphic capability (e.g. a Tektronix 4010 storage display)
- a number of shaft encoders

1) Handwritten memo by H. Davies of 29.4.74 presented to CUC.

- a number of push-buttons (perhaps in the form of a touch panel)
- up to three CC11 Crate Controllers to interface CAMAC crates to the unibus

We propose an 11E10 on the basis of low cost. It should provide sufficient speed and capability for testing equipment, although it does not provide floating point hardware and has a memory limitation of 28 Kwords. An option - the Extended Arithmetic Unit - may be purchased if fixed point arithmetic capability is needed.

The operating system we intend to use is RSX 11-M, which is a compatible subset of RSX 11-D and which is due to be released by DEC later this year. Under this scheme, and following certain rules, such as no FORTRAN, programs can be developed on an 11/45 for intended use on an 11/10 (see chapter 10).

There will be a subset of the ESAU interpreter which does not rely on a data bank plus a library of standard subroutines to drive CAMAC ; also console driver software. The user may add to the library his own PL 11 or ESAU routines.

In normal operation we expect that there will be one or perhaps two tasks at a time. In the foreground will be a user at the display interacting via ESAU and CAMAC subroutine calls with some CAMAC equipment under test. In addition there may be in the background another task, written in PL 11 or ESAU, which is performing long-term stability or endurance tests on other CAMAC equipment. During the latter test, interventions on the CAMAC system are limited due to possible interferences.

For two tasks under ESAU we estimate about 10 K core, for the console driver 2 to 3 K core. The complete RSX-11M requires 8 K core, we expect to occupy less core by selecting necessary parts only.

Lab. computers have been acquired by the Linac and AE groups already. The purchase of others will follow : one for CCI to test BIDUL and CAMAC equipment (to be used by PSB as well).

7. FRONT-END COMPUTERS (FEC's)[1]

The reasons for adding FEC's to our original configuration are described in Ref. 1 : they include CAMAC driving, surveillance tasks and multiple-pulsing. In addition they will allow some restricted local access to the process via a new type of midiconsole and in the event of the link to the central complex failing, they will allow the process to continue via local control.

Like the Lab. computers the FEC can be based on a PDP 11E10 with a cartridge disk and magnetic tape cassettes, although we will add more memory to bring it up to 28 K. Such a configuration suits the development of applications on the FEC. However, high availability will not be achieved due to the cartridge disk. An alternative solution with a fixed head disk or core only is still under investigation. In the future, depending on load, we could imagine upgrading a FEC to an 11/40.

To drive the CAMAC we will use a Branch Driver with some sub-systems adding some form of Serial Driver[2].

The midiconsole, which will be interfaced through CAMAC will consist of :

- display unit - alphanumerical TV or refresh display
- control unit - 4 shaft encoders with perhaps some function buttons
- selection unit - touch panel with 16 "buttons"

Each FEC will be linked to the central complex via 2 entirely independent BIDUL buses.

Again like the Lab. computers, the operating system will be RSX 11-M. For any FEC's without a disk it may be possible to adapt RSX 11-M but failing this MINIMON could be adapted for use.

---

1) CCI/note 74-21 by J.H.B. Madsen, presented to the MAC No. 28.
2) CCI/Note 74-25.

As the FEC's are the computers which actually drive the process, they must be extremely reliable, which implies stable software. They contain during normal mode (see 7.4 a)) only fixed tasks, which will include :

- BIDUL communication tasks
- CAMAC driver with fixed data tables[1)]
- simple surveillance tasks
- console and simple data base software

and on a number of FEC's as well :

- multiple pulsing task with associated data tables
- special tasks such as closed loop control.

With the simple surveillance, individual parameters can be checked on tolerances, and equipment status verified. The special tasks are introduced if due to constraints in the overall system, the FEC can provide a possible solution for a particular application. Introduction of special tasks is subjects always to reliability considerations and, of course, core space.

Estimates for core utilisation of a FEC used for multipulsing showed that an 11/10 will suffice. More detailed analysis are under way to verify our preliminary conslusions.

A copy of a subset of the data base for that process - as defined by the 11'45's and without directories - will reside on the disk of each FEC.

The FEC's may be run in one or three modes. They are described as follows :

a) Normal mode, i.e. the link to the central computers and MCR is maintained by at least 1 BIDUL bus. The FEC controls the process as directed by MCR and the 11/45's. At the midi-console

1) CCI/Note 74-15

full access is provided to all acquisition parameters plus access to a small number of control parameters with MCR permission - see chapter 8.

b) Stand-alone mode, i.e. the link to the central facilities is broken. The FEC continues to drive the process but now an operator at the midiconsole has full access to all parameters and is able to manipulate any 4 control parameters simultaneously, for example to correct for any drift. This requires minimal switching of fixed tasks.

c) Lab. computer mode - see section 6. In this mode the operator or engineer uses the FEC as a lab. computer, making no use of BIDUL. This requires a different set of tasks in core, particularly ESAU.

During shut-down and maintenance periods, the user of a FEC may select one of the above modes. Only if the central facility is not available must he choose between working with the midi-console or the interpreter.

Recovery procedures must be foreseen. When in normal mode communication is lost with the central computers, the operator will be warned and the FEC will enter stand-alone mode. When communucation is re-established, the reversion to normal mode should follow a fixed procedure, not interfering with the process.

When a FEC crashes, the process will stop until the error is corrected (for hardware this may involve inserting a replacement). System reload must be done in an orderly fashion which should include initialization of the console and updating of the data base both locally and in the central computers.

Initially the FEC's will be positioned close to the processes being controlled but later, depending on experience with Serial CAMAC, one could envisage that the computers (but not the midi-consoles) may be grouped centrally, thus allowing for much easier restart after a system crash.

The following FEC's are under consideration :

a) new Linac
b) PSB
c) PS-beam measurements
d) PS-auxiliary magnets
e) PS-program generator
f) new MCR console system
g) STAR
h) continuous transfer
i) PFW
j) PS-power house

8. USER ACCESS TO CONSOLES*)

We shall first give a short description of those sections of the consoles which relate to operator interaction with the computers. For a fuller description of consoles see MPS/CCI Note 74-6 by M. Bouthéon.

Each new MCR consoles will include :

- a high resolution refresh display with alphanumeric and graphic capabilities and some means of interaction such as a tracker ball,

- one or more TV or refresh alphanumeric displays : one would be used for keyboard interaction (e.g. with the interpreter), others for data. Alarms would be posted on one of these displays and some colour facilities may be useful,

- possibly a storage display for high quality graphics without interaction,

- shaft encoders with perhaps some function buttons and means of identification of the currently associated parameter,

---

*) For console use during transition period : see chapter 12.

- touch panels,
- other equipment for selecting and displaying analogue signals and for entering absolute values (e.g. "contraves").

The other consoles attached to the process - the new midiconsoles and any local mobile consoles - should be subsets of this MCR console. In particular, the midi will include an alphanumeric display, 4 shaft encoders and a touch panel - see section of FEC's.

The consoles in the MCR will be interfaced by CAMAC, except perhaps for the main graphic display, to PDP 11's referred to as console computers. The number and size of these console computers will be chosen when the final design of the MCR is fixed and a proper computer load can be calculated. The midiconsoles will be attached to their FEC's.

Considering first the MCR consoles, interaction will be mainly via the ESAU interpreter and the new touch panels - used either separately or in conjunction - together with the shaft encoders and the data base. More detailed descriptions of ESAU and the data base are given in later chapters of this report, and so here we shall give only brief descriptions of their use.

The touch panel is a very general tool, able to be used in different ways. For example, it will take the place of today's PRU (program request unit) with the program calls arranged in a "tree structure" so that is presents to the operator successive pages of options relating more and more closely to the problem he wishes to consider. When running a program the touch panel could be used to select options or pass parameters. It will be used to gain access to a set of data base parameters such as PSBooter - INJection set ; and so on.

ESAU will also be used in a variety of ways - for example to display acquisition data, set control parameters, hook parameters to shaft encoders, call for execution of programs in the 11/45's and so on.

A short note on data base access. Because of the possibility of control of the process from several consoles, but NOT the same parameters simultaneously of course, see below - and because of the need for internal self-consistency, whereas the master data base for a process will be in the 11/45's, and set of current control values for that process will be stored (in core if possible) in the FEC and referred to by the master data base if needed.

When an operator at MCR first accesses a parameters, for example to hook it to a knob, the current control value is fetched from the FEC and combined with the master data base information in the 11/45.

The operator at the midiconsole has a more restricted access. Firstly he does not have the interpreter available, only touch panels and knobs. Secondly he has only a subset copy of the data base for that process which does not include directories and he cannot access data base sets, only individual parameters.

Thirdly, although he always has free access to acquisition values, when the FEC isin normal operation with a link to MCR and the 11/45's, he must request (whether via the intercom or a BIDUL request is not yet decided) MCR clearance before attempting to access parameters for control. MCR gives this clearance which can concern an individaul parameter or s set, by releasing a VETO bit. When the central link is cut, this VETO bit should be automatically cleared, or ignored, allowing the midiconsole control access. However in both modes control access is restricted to a small number of variables at a time - one per shaft encoder.

When the midiconsole operator accesses a parameter, via the touch panel, the relevant data base information comes from the data base subset copy on the FEC disk plus the set of current control values in FEC core, thus making him independent of the central link. In addition to the VETO bit, there are BUSY bits to ensure that two operators at different consoles do not control the same parameter at the same time.

The actual operations performed at the midiconsole are not yet fully defined but will include facilities for hardware debugging, drift correction, etc. The functions needed will be provided via the touch panel options (e.g. timing options or pulse type) or function buttons (e.g. hook chosen parameter to knob 1).

8.4. There is also thought to be a need for local, perhaps mobile, consoles. Again these would be subsets of MCR consoles. They would be interfaced, via CAMAC, at a number of predefined access points - similar to the present mobile console in the Booster control room. They would enable engineers to perform problem diagnoses and to test hardware at any time, given that a proper procedure is followed in order to maintain the integrity of the overall system (see part 3. above on midiconsole use).

8.5. Program facilities on the operator consoles

This topic cyn be split into two sections :

i) how existing software is used (to control the accelerator),

ii) how additions are made to this software from programs developed in the program development computer.

The operator's tools

In designing tools for the operator the main philosophy has been that the operator needs to control the accelerator. In fact the operator does not wish to control the REAL accelerator which only knows about CAMAC or STAR addresses. He would rather control an IDEAL accelerator which can be controlled by specifying names, such as I-DV05, etc. This IDEAL accelerator would have characteristics which are not present in the REAL world. For example :

a) informing the user when his control value is out of range, out of tolerance

b) allowing increment, decrement and absolute setting of values,

c) enabling several discreet units to be controlled as if they were one item,

d) any many others.

All the tools provided enable the operator to control this IDEAL accelerator. In practice it is the Data Base which provides the IDEAL accelerator and converts all actions from the IDEAL world into the REAL world.

It is still inconvenient however to be faced with only this IDEAL accelerator. Some easy rules of protocaol have be to introduced before the accelerator can be used. For example, with a keyboard some form of CONTROL LANGUAGE (e.g. ISAAC) is needed. Alternatively a set of push-buttons and implicit functions need to be provided (e.g. a midi-console).

This protocol and the characteristics of the IDEAL accelerator is all that an operator needs to know to perform his job.

This situation roughly corresponds to ISAAC without any support routines or programs. Thus other facilities exist which offer more complex services (i.e. to plot beam losses, to set up bumps, etc.). Since the number of these facilities may increase with time a program library has been provided. This library is one way in which the facilities available to the operator may be increased.

The program library

The program library is an integral part of the facilities available to the MAXI (MCR) console user. Most of the components of this library will initially start as source text on the program development machine. After debugging they have to be incorporated into the program library. This is done upon request from the MAXI (MCR) console, resulting in the program or procedure being integrated into the program library. Transmission of the program would be via BIDUL.

The library is not intended to hold only routines but can serve as a depository for any of the many forms that utilities take. For example :

FUNCTIONS
ROUTINES
TASKS

The manpower required to implement fully this program library facility is estimated to be quite large ($\simeq$ 1 man year). In view of this the implementation may proceed in stages, expanding the available facilities as the manpower becomes available.

For example, an initial aim could be to produce two libraries one for ESAU and one for applications programs. The latter would use the standard DEC librarian facilities (requiring no work), while the ESAU library could be the prototype program library.

A detailed description of this library will be produced soon.

9. OPERATING SYSTEMS AND LANGUAGES

Real-time operating systems are complex programs which require extensive effort to understand their implementation, debug and maintain. In view of this it is naturally desirable to restrict their number to the minimum necessary.

The following three operating systems will initiall be supported :

i) RSX-11D - Main 11/45 computers (core $\simeq$ 25K)

ii) RSX-11M - Front End and Lab. computers with RK05 disk(core $\simeq$ 8K)

iii) MINIMON - Front End computers without disk (core $\simeq$ 4K)

The RSX operating systems are upwards compatible, i.e. RSX 11M is a subset of RSX 11D. At a later date MINIMON may be phased out if a version of RSX 11M without the need of a disk appears (details on RSX 11M are given in an appendix to this chapter).

Four languages will be installed on the system. They are listed below together with an indication of the conditions appropriate for their use.

1. FORTRAN IV

A compiler for mathematical calculations involving non-simple floating point operations and multi-dimensional arrays. Routines will be available enabling users to access process data, and manipulate control variables. No knowledge of the PDP machine architecture necessary. Not available for the FEC and Lab. computer.

2. PL 11

A compiler designed to replace assembly language which is useful for systems work and for applications programming when the main emphasis is on fixed point calculations or floating point operations of a simple nature, and where execution speed and efficiency of core utilisation are important. A knowledge of the PDP-11 computer architecture is necessary. The compiler may be used to produce code for any member of the PDP-11 family.

3. MACRO-11

The use of this assembler will be mainly restricted to cases where a direct interface with RSX is necessary.

4. ESAU*)

An interpreter providing interactive control of the accelerator by the operator. At a later date, depending on the need, a compiler version of ESAU may be produced. ESAU will also run on the Lab. computers and in conjunction with PL-11 should provide all the freedom and flexibility necessary for debugging CAMAC equipment. For an example of the use of ESAU see Ref. 3 below.

References

1. Preliminary specifications of PL-11, R. Russel, OM Note SW-29.
2. Extensions to PL-11, R. Russell, OM Note SW-39.
3. New interpretive language for the PS control: external specifications, MPS/CCI/Note 74-12.

---

*) ESAU resembles NODAL, defined in Lab. II. Differences are mainly due to the fact that we want to conserve the commands of the interpreter used successfully over two years on the MCR consoles already and also as the aims are not similar, neither are the hardware and software environments.

APPENDIX

The characteristics of RSX11-M may be summarized as follows :

A. A subset of RSX11-D. This means that :

i) the interface between program and operating system is the same as RSX-11D,

ii) object modules produced under RSX-11D may be used in the RSX-11M task builder,

iii) programs ultimately destined to run under RSX-11M may be run initially under RSX-11D in so far as the operating system is concerned,

iv) the operator commands are similar to those of RSX-11D.

B. Wide range of the PDP11 family may be used with RSX-11M. Thus the use of RSX-11M implies the freedom to upgrade the power of the front end computer if future needs dictate.

C. Supports a wide range of peripherals. Thus, with the exception of CAMAC, no need for in-house software on I/O drivers or insertion of file handling capability.

D. Supported by DEC. Thus reference documentation and implementation notes will be provided by DEC.

E. Will be released in November. The system is now running at DEC Maynard and will be released to 'experimental stations' in the next few weeks.

In the absence of RSX-11M we may have considered the ASEA operating system but in view of the above points we now feel that RSX-11M provides both more services for the present and more flexibility for the future. We therefore propose that RSX-11M should be the 'standard' for the FEC's.

10. PROGRAM DEVELOPMENT FACILITIES

Program development facilities are provided to enable users to produce or develop their programs on-line to the computer. This means that the user will normally prepare and develop his programs at an alpha-numeric display connected to the program development computer. In order to achieve this on-line environment, all of the users' source text, object modules and finished tasks will be entrusted to the computer file system. Editing, compiling, listing and testing services will be provided as tools to produce a comfortable working environment for program development.

The actual form that the software support will take is uncertain at this time. There are two main alternatives :

i) All (or most) of the facilities are developed by CCI. This ensures that these facilities can be presented to the user in a consistent manner (i.e. commands for editing within the interpreter are similar to those for editing non-interpreter programs). However, this approach requires a considerable expenditure of manpower.

ii) We use the software supplied by DEC. At the moment, however, this is not very satisfactory but DEC are about to announce some improved products.

In view of ii) the final decision of which to adopt will be postponed until late 1974. For present facilities see MPS/CCI/Note 74-16, RSX multi-entry system, J. Gamble.

We can, however, state the type of facilities that will be provided :

1) a comprehensive text editor,
2) an interpreter,
3) compilers - FORTRAN, PL 11, MACRO-11 and eventually a compiler for interpreter programs may be written,
4) facilities for generating executable tasks from the modules produced by the compiler. This will almost certainly be supplied by DEC.

5) testing and debugging facilities. Very little effort has been put into this topic. In the long term sufficient facilities should be supplied to enable users to fully debug their programs before the program is allowed to communicate with the process.

All of the above will be available simultaneously from several consoles (VT05's). For users who do not wish to sit at consoles, the facilities will be available from a batch processing system.

Peripheral requirements

To support all of these various activities, a number of peripherals will be needed. The required peripherals have been listed under 4 a) already. It is sufficient to state that we shall need still to acquire disk packs for mass storage, cassettes and displays (VT05 or any other teletype compatible system).

The actuel number of VT05's needed is uncertain but a figure of 8 seems sufficient, supporting about twenty applications and system programmers.

The RP04 disk packs*) (80 Mbytes) will hold the file system for the program development machine.

Cassettes are expected to replace paper tape, since the 11E10 configurations for the FEC's and Lab. computers do not have paper tape. However, a few configurations exist (the message switcher 11/10) which only have paper tape and for these paper tape equipment is needed.

11. DATA BASE SYSTEM **)

11.1. This subject has been treated in some recent notes[1]. We therefore summarize briefly the main elements of the DBS. However, the user access, which has only been described so far to the Data Base Study Group and CCI/SSS[2], will be somewhat more detailed.

11.2. In the Data Base system we distinguish two major parts :

- the Data Base proper which contains a description on how to control every process variable and what its current working condition is,

*) or equivalent **) the centralized DBS has been replaced recently by a distributed data base (see MPS/CCI/Note 74-44)

- the Data Base management which provides the mechanism for fast access and use of this information (as required by the users[1,2]).

Each main process, i.e. Linac, Booster and PS, will have its own Data Base.

Each process variable is recorded by a name[3] and is described by around 40 words (e.g. acquisitions and control address, current control values, reference values, etc.).

We estimated about 1800 variables per process, which will require the Data Base to be stored on disk. However, the routines which provide the Data Base Management (e.g. to map the name of a variable into a disk address , to read its description, etc.) are permanently in core.

Upon a user's request, part of the Data Base he needs for his particular applications is called into core by an initialization procedure[1].

11.3. Two types of variables can be handled :

- simple ones : i.e. all individual hardware variables such as steering magnets, quadrupoles, etc. which require set point control,
- process variables which require special routines for acquisition and/or control e.g. Q values, mean radial position, etc. These routines are integrated within the Data Base System.

11.4. Functions

They are routines that act on data supplied to them by the user in the form of arguments in the call, e.g. : CALL BUMP (DIPOLE1, DIPOLE3), to crease a BUMP using two dipoles, where DIPOLE and DIPOLE 3 are defined, for example, interactively.

These function names will be stored in a separate table which will not be managed by the Data Base Management.

11.5. Use

One distinguishes between two types of use :

a) the off-line use (i.e. in the sense that the operator is not connected to the process) ; this implies all updating activities, in particular appending and deleting. An interactive program is being designed which allows a user to :

- define a process variable from scratch
- update the description of a process variable.

Currently this activity is performed on the teletype, but in future it will use an alphanumerical display and keyboard.

b) the on-line use. As a general rule, the user calls a process variable by its name ; in return he gets the specification he has requested.

The control and acquisition program accesses the Data Base through a set of routines called functional routines. They are the interface between the application programmer and the Data Base system[2]. They provide the application programmer with ready to use routines which perform a well-defined activity on the process variables. They give him full control of the contents of the Data Base Management buffer and optimize the transaction within this buffer and between this buffer and the disk. A more comprehensive list was given in Ref. 2[*]. Example : suppose a user wishes to increase the current in a steering magnet by 1 Amp, then the recommended procedure that he will have to follow will : be

1) CALL INIT (I-DV05 CONTR,ERR)

This calls from disk into core everything he needs to CONtrol I-DV05. The ERRor code will warn him of, for example, illegal names, or if this variable is being controlled by someone else, etc. From now no-one else can control I-DV05. Note that initialization of a large SET may take some time (cf. section 11.7).

2) DELTA = 1.00 AMP

DELTA is the increment to be applied to the current control value of I-DV05 in AMP.

3) CALL INCRS (I-DV05, DELTA ERR)

This will INCReaSe the current control value of I-DV05 by 1.00 AMP

---

*) The note also describes the advantages and disadvantages of: One DMB buffer per user and one common DBMB for all users. In the note it was proposed to use the first solution.

whereas the ERRor code will tell him if this command was executed properly or if, for example, the resulting control value exceeds limits, etc.

4) CALL RELEASE (I-DV05, ERR)

This will perform the necessary housekeeping and make it free again for control by someone else.

This recommanded procedure ensures safe control and fast servicing within the program.

As some application programmers may find it awkward to follow the recommended procedure, default conditions will be introduced so as to make the Data Base more transparent. Example : suppose a user does not want to initialize a variable (e.g. I-DV05) before incrementing its control value, he could :

CALL INCRS (I-DV05, DELTA, ERR)

and if ERR indicates that I-DV05 has not been initialized, then

CALL INIT (I-DV05, CONT, ERR)

will be defaulted for him. Obviously there is not a guarantee that after the first time he has requested

CALL INCRS (I-DV05, DELTA, ERR)

he will be given control of I-DV05 ! Someone else may be using it and he will only be aware of it after his attempt to initialize. Also, he will have to access the disk for every variable he wants to control this way for the first time.

Under default conditions, use of the DB is simple but there are restrictions and it is expected to be less efficient.

The interpreter is considered as any ordinary applications program and will therefore have to map his commands in a sequence of routines (as described higher) depending on how an operator wants to access the process.

Communication between Data Bases in different computers is also handled by the Data Base Management through the communication monitor.

Remark : the suggested naming conventions for the Data Base have been circulated (see Ref. 3).

11.6. Subset of Data Base in FEC

The complete Data Base of each main process is housed on the disk of its main computer. The FEC (Front End Computer) contains part of the Data Base relevant to the sub-process driven by this computer. The "front-end" part of the Data Base is built and maintained (i.e. updated) from the main computer so that it is a duplicate of what is in the "main" Data Base. However, as the FEC performs some restricted operation, the current control values may be modified. The main computer will read the up-to-date current control values in the FEC when it needs them : therefore they may be kept in core, core space permitting, for fast access (if the FEC has no fast fixed head disk). To prevent conflicts, operation on the midiconsole and on the maxiconsole will be handled as follows :

Midiconsole operation in FEC

The number of the push-button which is pressed to select a process variable is mapped directly into the disk address where the description of this variable is stored. The midiconsole software checks for VETO bits and also whether a variable is free (i.e. not used by the main computer), and if so sets it busy for any other user. The process variable's specification essential for midiconsole operation is loaded into core. The current control value is updated when a new process variable is selected. (Note : the VETO bits check could be automatically bypassed if the communication between main computer and FEC is broken, provided such information is available in the FEC).

11.7. Operation from MCR consoles (often called maxiconsoles)

When the maxiconsole requests control of a process variable which may also be controlled from a midiconsole, the following procedure will be followed :

at INITialization

- check both in FEC and Main Computer if this variable is free for control,
- if so, set it busy for other users (in FEC and Main Computer),
- pick up the current control values which are in the FEC.

when CONTRolling

- control the variable.

when RELEASing

- after control, update the current control value both in the main Data Base and the FEC one,
- make this variable free again.

This procedure will be transparent to the user (it will be handled by the Data Base Management).

11.8. System resources needed

Disk capacity :

128 K to store the description of about 1800 process variables.

Core requirement :

- In main computer

i) 9 K for the Data Base Management routines and some tables

ii) 24 K of (at most) of Data Base Management Buffer(s): however, it is intended to define empirically the optimum size of this buffer starting from e.g. a 10 K buffer. To give an idea of the storage capacity of such a buffer : a 10 K buffer could house around :-

- 400 acquisition (only variables), or
- 200 standard unpulsed control variables (including acquisition), or
- 120 pulsed (up to 5 different pulses) control variables (including acquisition).

- In FEC : 3 K to store all current control values of the 1800 process variables (if no fixed head disks) and some tables to prevent conflicts (see previous paragraph).

11.9. Error and recovery

In general the DB software will include error checks (e.g. against illegal names, etc.), and critical data may be transmitted twice to safeguard against transmission errors. In particular :

- If the main computer goes down, we rely on the back-up scheme.
- If the FEC goes down and the DB was modified in the main computer, one must provide a program to update the one in the FEC. However, this program is part of a more general recovery procedure.
- If the FEC disk goes wrong, only the midiconsole cannot be used, but as the last control values are still in core, the main computer can still control the process.
- If a program using process variables crashes, there will be an automatic clean up of the Data Bank Management buffer(s).

References

1) Standard Data Base for the CERN PS control, A. Daneels, MPS/CCI/Note 74-5, 30 January 1974.

2) Data Base Study Group, Notes on a discussion on Wednesday, 3 April 1974, A. Daneels, 6 May 1974.

3) DB Working Group, Naming conventions for the new CPS Data Base, 14.6.74, MPS/CCI/Note 74-34, A. Daneels, P. Mead.

12. TRANSITION FROM IBM 1800

One of the essential assumptions in building up our new computer system is to guarantee at every stage of the project a smooth computer-aided operation of the Linac, Booster and PS. In this chapter a way is indicated how this could be achieved with the computer configuration proposed. The relevant facts in respect to operations are (see Fig. on page 35).

- The IBM 1800 is not connected to BIDUL. Therefore no communications are possible between the IBM 1800 and main computers or front end computers (FEC's).
- The STAR control and acquisition system is shared between the IBM and a FEC thus making STAR available to the new computer system.
- The present consoles are connected via STAR to the computers, full console software is implemented in the IBM, PDS1-console software in the on-line PDP 11/45 as well.
- There will be a prototype of the new consoles installed in the present MCR. This prototype is connected via a PDP 11/10 to BIDUL.

Transition of software (application programs) from the IBM to the new computer system could be performed as follows. Computer-aided operation will continue to be done exclusively from the present consoles through all stages of transition until the new MCR and the completed computer system are available. To make the advanced computer power of the PDP11/45's (floating point hardware!) available for operation as soon as possible, it is suggested to implement PDS1 console software (the IBM resident part) in the on-line PDP 11/45 (see Fig.) This would allow to select new application programs in the 45 from the old consoles and display their results there. The new console, however, is used for three purposes :

- testing new application programs,
- testing transferred aplication programs,
- training operational staff.

The situation is clearly this : even if transferred application programs have been tested in the PDP 11/45 (program development computer) probably during MD time, the copies of these programs in the IBM stay operational and thus will continue to be called via the old consoles. New applications for the PDP 11/45 cover new devices which are typically connected via CAMAC to a FEC. They are first written to be tested on the new console, then made available to operation by adapting them to the PDS1 consoles. This is exactly the price one has to pay to have a clear cut between operation and system development*) The PDS1-console servicing software in the PDP 11/45 will try to simulate the new consoles, so that adaptation will not be a burden to the programmer. It is not recommended to operate new applications from the new consoles, being itself under development (hardware and software!) and thus not always available for operation. New application programs represent only a minor part of all applications we have to implement in the new computer system. Old applications stay unchanged anyway in the IBM. Conflicts in this situation can only arise if new applications ask for control variables from different data bases.

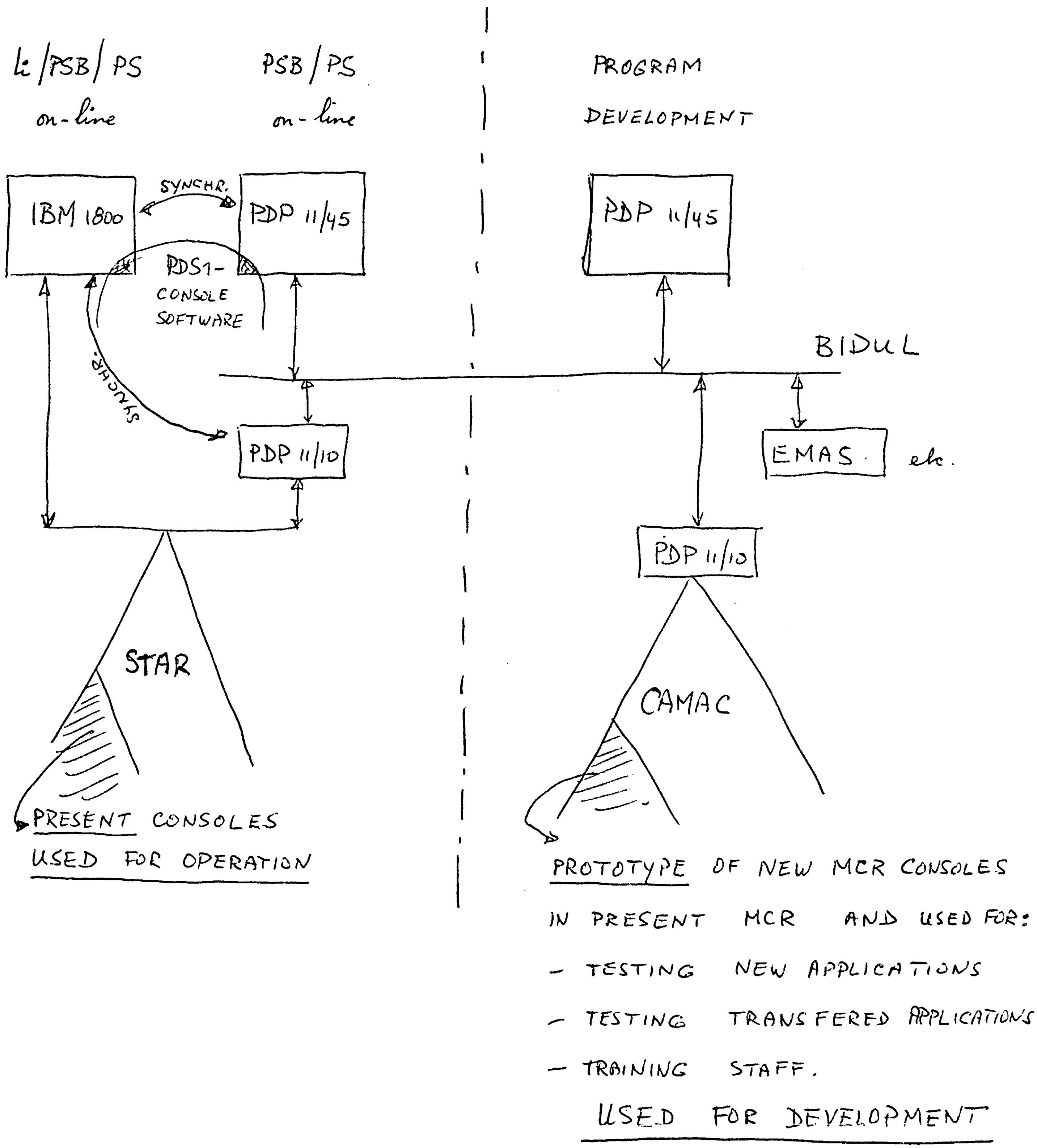
The IBM 1800 has still to play an important role in the overall system. Up to the time when essential parts of the new computer system are ready, all urgent new applications have to use STAR transmission and be implemented in the IBM. Only when the new computer system is completed and the new MCR operational will most of the burden fall from the goold old IBM. Then its destiny is connected to that of the old Linac.

The transfer of all IBM application programs constitues a heavy load to the Booster and PS programmers. Taking into account the number of programs involved and the effort of creating them, it is guessed that about 10 man/years are needed to redesign and rewrite them using the new system features such as data base, communication software, existence of FEC's, new consoles, floating point hardware, etc.

Problems are posed by operational aspects of urgent new applications like "program generation" via a FEC etc. Deadlines of single projects are difficult to match with the availability of the overall system. Instead of looking for special solutions where changes in configuration are involved, one should try to have a prototype of the computer system working as soon as possible. The concept of operating exclusively from the old consoles and making applications implemented in the PDP 11's available to operation via a copy of the PDS1 console software will help here. No waiting for new consoles is implied and the

*) We are not sure if we can afford to pay such a price, suggestions made on PDS-1 connections and presence of STAR-FEC are currently under review (1.9.74 ; JHBM).

Use of new computer system during transition from IBM 1800
(simplified)



effort of putting the console software into a PDP 11/45 is of the same order (about 6 months), as is gained by not implementing the communication monitor in the IBM which is now disconnected from BIDUL. This software can be made by the Exploitation Section which will anyway be involved in transition problems.

13. SYSTEM PERFORMANCE

In general such a study attempts to rate how well the control system (hardware and software) performs against the operator's requirements at peak times (MD and setting-up). Ideally this requires a performance study of every individual subsystem which is difficult, if not impossible, to achieve at this stage of the project. Some items of the overall control system have passed the first iteration of design, others not.

At the most we can state why in general we expect the new control system to perform better than the previous one. This can be illustrated by summarizing the figures presented in earlier work in this field[1,2].

13.1.The overall configuration[1,2]

The old control system using the IBM 1800 was able to successfully service four users : Linac, PSB, PS injection, PS ejection, not including some permanent tasks and program development.

The core size of the IBM 1800 is 48 K only, forcing extensive use of the disk and resulting in long response time. The constraints complicate application programming very considerably also.

The main changes introduced by the new system are :

1) replacement of 1 machine of 48 K by 2 on-line machines of 96K,

2) truncation of off-line work (program development) from on-line work,

3) withdrawal of some permanent tasks to FEC's.

The on-line machines

They will service Linac, PSB and PS. The final distribution of the load over the two machines will be decided later. However, as already explained (4 a)), to begin with : one machine for the new Linac, the other for the PSB and PS. We can assume that the PSB/PS computer will have most of the load and that they may have to house 6 to 7 application programs working in parallel[1,2].

In "IBM" terms this would require for each program on average[1,2]:

- core 4 K
- STARA 120 data/cycle
- STARC 30 data/cycle
- overlays 2 (related to core size)

Disc access

- many (from 0 to 100) at initialization time, i.e. when the operator requests a program
- normally 0/cycle during normal running of the program

Display

- from 0 (some very particular programs only) to 2500 (extreme case), on average around 500 data/cycle.

It is not quite clear how the core in the main computers will be organized. If we consider that :

i) the PDP 11/45's are about 5 times faster than the IBM,

ii) fixed head disks are about 25 times faster than the ones on the IBM configuration,

iii) permanent tasks such as watchdog functions, midiconsoles, general display (?), multiple pulse setting, etc., requiring real-time control are housed in the FEC,

iv) program development will no longer be interfering with the on-line tasks.

on can expect the new system to be able to cope with requirements.

The off-line machine

This machine will be dedicated to program development. Hence :

i) no interference with the on-line tasks (especially use of disks),

ii) all programmers have access at any time to this machine,

iii) the less reliable peripheral devices (line printer, etc.) are removed from the on-line machine,

iv) effective back-up by providing a bus extension to which program development devices are connected and which can be switched to either machine.

13.2. Data Base

The data base system implies a load both on core and disk access.

i) Core requirement to house part of the data base :

- 1 single process variable needs about 40 words
- 1 set of 100 variables : about 5 K.

ii) Disk access

- If we assume we have fixed head disks with an average access time of 20 msec (DEC type) and that we are the only user of the disk, then :

Access to 1 single process variable	: 50 msec on average
Access to a set of 20 process variables	: around 500 msec
Access to a set of 100 process variables	: about 2 secs

However, this time delay occurs normally only at initialization time.

References

1. The PS performance requirements for the computer control system : data and commentary, B. Carpenter, J.H.B. Madsen, Ch. Serre, MPS/CCI/Note 74-8, 27th February 1974.

2. Attempt to estimate the PSB computer load, A. Daneels, MPS/CCI/Note 74-9.

14. THE ANALOGUE FUNCTION GENERATOR(FG)

14.1. Process terminals are not described in this report. However, the FG takes a particular place in the control system due to its real-time aspects. The present spare capacity on the VARIAN-based FG is not sufficient to furnish the large number of functions required for high intensity operation and for intensity modulation.

A new FG has been developed, which has been described in detail[1]. We give a short summary of its characteristics only.

14.2. Each FG contains a digital memory which has the information required to elaborate the analog function. The information - supplied by the main computer - is updated only if the operator has initialized a modification (on the maxiconsole). Thus during steady state operation the FG works independently of the computer. The time base of the FG can be an internal or external pulse train (C or B train). The start of the function is made by a local trigger pulse which can be conditioned with a program line. As the memory of the FG can be charged with several functions the program line conditioned start allows the use of one FG in complex cycle situations (intensity modulation, etc.).

14.3. The program for the FG is rather complicated and therefore handled in the main computer. This imposes limitations : no access from local consoles if central computer facilities not available. Anyway the new FG will be used in 74/75 via the PDS1's and IBM 1800, no local access.

Reference

1) E. Assêo, CERN/MPS/CO 72-5, R. Debordes, MPS/CO/Note 73-8 and 18 ; MPS/CCI/Note 74-20.

15. USER EDUCATION

The new computer control system together with the new control room will obviously present a completely new environment to operator and programmer alike, although all possible will be done to make the new system as easy to program and operate as possible.

At the appropriate time documentation will be published and courses will be organized, on all aspects of the system which present a new interface to the user. For example, courses will be run on PL-11, operation of the new consoles, ESAU interpreter, the process interface routines for Fortran users, etc. Operators will be expected to choose the course(s) appropriate to their needs and interests. Certainly one will not need to be an "expert" on all facets of system operation and programming in order to make use of it.

With regard to program development, users will have to know certain aspects of the RSX system. Suitable documentation and "in-house" courses will be arranged to cover this, and there should be no need for attendance at formal DEC courses. However, it will be left to the individual groups to decide on their own policy in this respect.

* * * * *

Acknowledgements

This report has benefited from the skill and effort of our secretary, Mrs. J. Eddison, and more recently, Miss. A. Seagroatt. We thank them both very much.

We thank all the future users of the new system for their contributions to our work. Without their appreciated interactions we cannot construct a general system for the MPS.

M. Bouthéon was the convenor of a working group on the users' aspects of man - computer interfaces (participants : W. Busse, A. Krusche, K. Schindl and P. Têtu). He made the proposal for the new MCR consoles and is presently working with us on the elaboration of this project. His interest and that of his colleagues in the Operations Group on our project has been of great help to us.

The working group on the Data Base (convenor : A. Daneels, participants : H. van der Beken, B. Carpenter, P. Heymans, H. Kugler, C. Serre and U. Tallgran) allowed us to include users' ideas from the start of its design.

The work done by the Linac Group on the controls for the new Linac (U. Tallgren) has generated numerous exchanges of ideas between them and us.

Besides the authors of this report, many people worked or are still working actively on the project. In the first place we mention H. van der Beken and B. Carpenter, who contributed to a very large extent to it. G. Daems has a major part in the design and construction of BIDUL, R. Debordes in that of the new function generator.

Distribution

MAC
CUC
PSS

G. Baribaud	F. Perriollat
S. Battisti	H. Reige
G. Daems	F. Rohner
R. Debordes	A. van der Schueren
J. Gruber	C. Serre
P. Heymans	P. Têtu