COMITATO NAZIONALE PER L'ENERGIA NUCLEARE
Laboratori Nazionali di Frascati

LNF-73/10
29 Marzo 1973

M. Matera, S. Tazzari and G. Vignola: COMPUTER CONTROL SYSTEM USING THE HP BASIC LANGUAGE, FOR THE STORAGE RING ADONE. -

Servizio Documentazione
dei Laboratori Nazionali di Frascati del CNEN
Casella Postale 70 - Frascati (Roma)

Laboratori Nazionali di Frascati del CNEN
Servizio Documentazione

LNF-73/10
29 Marzo 1973

M. Matera, S. Tazzari and G. Vignola: COMPUTER CONTROL SYSTEM USING THE HP BASIC LANGUAGE, FOR THE STORAGE RING ADONE.-

ABSTRACT. -

The Adone storage ring control room has been implemented with a computer facility. The system has been in operation for over one year. This paper gives a short description of the control system hardware and deals mainly with some aspects of the software organization.

1. - INTRODUCTION. -

Many discussions on accelerator control by computer can be found in the literature[1], a great deal of attention being given to software problems. A very definite trend is emerging towards the use of interpreter-based software packages.

In this paper we discuss some aspects of the software organization for the Adone Control Room, after a brief description of the overall system. It has to be considered that the Adone control system was initially designed for manual operation and computerized control was implemented on a running machine, which turned out to be a task requiring a great deal of programming flexibility. It has also to be pointed out that the architecture of the system rests heavily on the hypothesis that neither very fast reaction times, nor real-time multiprogramming provisions are actually needed for our machine.

The software we have developed is based on the existing H. P. Basic Compiler, and one of its main assets is that its development required considerably less than one man-year.

2. - HARDWARE AND CONTROL FUNCTIONS. -

The computer around which the control system is built is a H. P. 2116B, having an 8K core memory (16-bit word) and a 1247K disc memory (2 moving-head discs, one of which removable).

Besides the usual peripherals (TTY, fast punched tape reader and fast tape-punch), the system is equipped with a CRT storage display and a keyboard, to ease communications with the operator.

Most control operations are effected by means of a modular data acquisition and transmission system[2], communicating with the computer via a "data bus". The bus runs along all of the control

room. The system modules are addressable in a way very similar to that of CAMAC.

There also is a system for digitizing very fast repetitive pulses, mostly used to analyze the circulating beam signals.

Most machine parameters under computer control (either single variables like a power supply current, or complex variables like a betatron wave number) are accessible to the operator through the usual manual devices (knobs, push-buttons and the like), but the action is always taken by the computer. This is a very useful feature, in that fine adjustments by the operator are rendered much easier (the interface is completely transparent) while maintaining computer supervision and on-line computing power.

3. - SOFTWARE. -

3.1. - General remarks. -

It was clear to us, right from the beginning, that a software package based on some kind of core resident compiler-interpreter, would in our case be the only reasonable choice, in view of the problems posed by interventions on a running machine.

The elementary control functions would be carried out by dedicated assembler-language programs (drivers) controlling the appropriate interface hardware, while complex control functions, calling and linkage of the drivers, and computations would be handled by the interpreter.

It is quite obvious, that such an organization, while providing easy on-line access to the programs, and hence maximum flexibility, may cause problems in connection with execution times. We will come back to this point later, but let us just say that this particular limi-

tation proved to be much less severe than anticipated.

3.2. - Organization. -

The system is based upon a suitably modified version of the HP BASIC Compiler.

The HP BASIC is a core resident program allocated as shown in Fig. 1. The storage area can be divided into three parts, two of which are occupied by the compiler proper, the third being reserved to user programs.

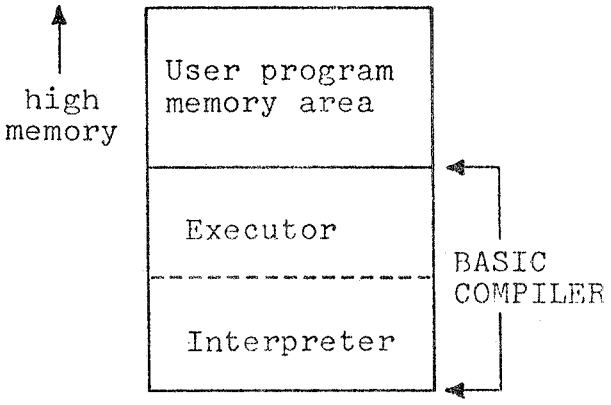

FIG. 1 - Memory map.

The two sections of the compiler proper control writing and execution of a Basic program and namely:

- The interpreter section translates the Basic language program, statement-by-statement, into an operational code. It also checks the statement sintax on-line (issuing suitable error codes), and stores the statement in the program area.
- The executor section executes the program statement-by-statement. Only one Basic program at a time can be executed.

The HP BASIC includes a CALL statement used to interface absolute assembler language subroutines (drivers) to the program. Any number (64 max) of these can therefore be appended to the standard system, to create specialized configurations. The drivers are usually stored in the program area, thus reducing the memory available to the Basic user program, while both the number of the drivers and the length of the user program increase proportionally to the

amount of control functions to be performed. A large storage capacity and suitable programming techniques (segmentation) are therefore needed. Recalling that BASIC executes a program statement-by-statement, and that a CALL instruction occupies one statement only, a "virtual" memory expansion seems quite natural.

Our system was therefore implemented with a disc memory unit. The disc is also very useful for storing the fairly large amount of data necessary to a keep a log on machine operation.

Moreover, in view of the fact that the two sections of the compiler are completely indipendent and operate in sequence, BASIC was modified so that the driver corresponding to a given CALL statement (automatically recalled from disc, unless present) overlaps the Interpreter section of BASIC. The interpreter is automatically recalled in memory at the end of the execution phase.

Segmentation and linkage of programs written in BASIC, is effected through a BASIC-callable driver capable of transferring the programs to and from disc (the programs are transferred after having being translated).

The standard BASIC has been further modified to improve its Interpretative Capabilities. In particular a variable can be designated by any number of alphanumeric characters in between two " ' " characters.

As an example of the system capabilities assume it is desidered to set the voltage of power supply 'BUMP' to X.

The driver executing the elementary function of setting a power supply is 5. The Basic program is then as follows:

```
i     LET 'BUMP' = X
i+j   CALL (5, 'BUMP')
```

3.3.- Application.-

We have found it is useful to distinguish two separate modes of operation of the accelerator:

- normal operation(when running experiments and the machine is controlled by the operators)
- accelerator testing (the accelerator is then usually run by the physicists on the accelerator staff);

Under normal operation conditions the computer runs on a standard program subdivided into suitable subsections, each performing operations of a variable degree of complexity. Each subsection can be called from a push-button selection panel, to allow for some flexibility in the sequence of operations. Dialogue with the operator is through the CRT display and a keyboard.

Under accelerator testing conditions, while the usual facilities remain available, the physicists can write their own programs in Basic, and run them right-off.

4.- CONCLUDING REMARKS.-

Our system is rather small but we think, quite interesting for its use of an existing BASIC program, its expansion capabilities due to the disc-based organization, and the relatively small programming effort required.

The advantages of a system based an a core-resident compiler-interpreter did in our opinion outweigh its drawbacks by far.

As far as the execution time is concerned, and given our limited requirements, two observation can be made:

i) if the execution time is referred to the operator, the speed is mostly more than adequate even when running BASIC programs

and making extensive use of the disc.

ii) When a program or an operating procedure are sufficiently consolidated to make further program modifications unlikely, the whole batch can be translated into assembler and reduced, at BASIC language level, to a single statement.

As a last observation, we found it of invaluable help, in reducing programming times to have a core-resident library, belonging to the BASIC Compiler, and immediately available to the programmer of Assembler drivers.

ACKNOWLEDGEMENTS. -

We gladly acknowledge the important contribution Dr. F. Soso gave to the early stages of system design.

All of the hardware was designed, built and put into operation by the Adone Electronics Laboratory (LEA).

REFERENCES. -

(1) - See for instance: Proceedings of the 8th International Conference on High-Energy Accelerators, CERN (1971).

(2) - LNF Report, to be published.