THE NEW CONTROL SYSTEM OF THE SACLAY LINEAR ACCELERATOR

J.F. Gournay, G. Gourcy, F. Garreau, A. Giraud, J. Rouault
Service de Physique Nucléaire- Accélérateur Linéaire, CEN Saclay
91191 Gif-sur-Yvette, France

A new control system for the Saclay Linear Accelerator designed during the two past years is now in operation. The computer control architecture is based on 3 dedicated VME crates : one crate with a disk-based operating system runs the high level application programs and the database management facilities, another one manages the man-machine communications and the third one interfaces the system to the linac equipments. At the present time,communications between the VME micro-computers are done through 16 bit parallel links. The software is modular and organized in specific layers, the database is fully distributed. About 90% of the code is written in Fortran. The present status of the system is discussed and the hardware and software developments are described.

INTRODUCTION

The ALS, in operation since 1969 [1], was primarily manually controlled, a computer was introduced into the accelerator control system in 1974 for centralization of informations, automatic surveillance of the main parameters and control of the beam switchyard [2,3]. During the next 8 years this system was expanded with a second computer and with ponctual local processing power (8 bit micro-processors), but it became obvious in 1982 that the system had to be replaced very soon. The old technology computers were completely obsolete, their maintenance was difficult and expensive and the software developments very tedious. In this paper we describe the solutions that we have adopted for the new system.

HARDWARE SYSTEM

For the new control system it was decided to take advantage of the progresses in the micro-processors technology to ensure reliability, flexibility, versatility and a good cost effectiveness. A solution based on a standard and modern bus with powerful CPU and I/O boards available from many different manufacturers looked very attractive. Among the different possible choices, the VMEbus and the MC68000 microprocessors family seemed to be the best solution for the process computers.

Process computers

The functions of the system are distributed between 3 dedicated VME crates (fig.1) :

. the "HOST" crate with 1MByte of memory, a 20 MByte disk drive, a floppy disk for backup. This station is running under a disc-based operating system.

. the "LINA" crate with 512kByte of memory and all the interfaces to the equipements of the linac.

. the "OPER" crate with 1MByte of memory and all the interfaces to the man-machine communication devices.

The LINA and OPER stations have no disk and their software is downloaded from the HOST station. The three stations are close enough to be interconnected with parallel links, the protocol used enables 50 kBaud data transfer rates on these links. The stations are equiped with "first generation" VME CPU boards: 8MHz MC68000, small in-board memory size, no MMU, no floating-point coprocessor and a limited private bus reserved for I/O.

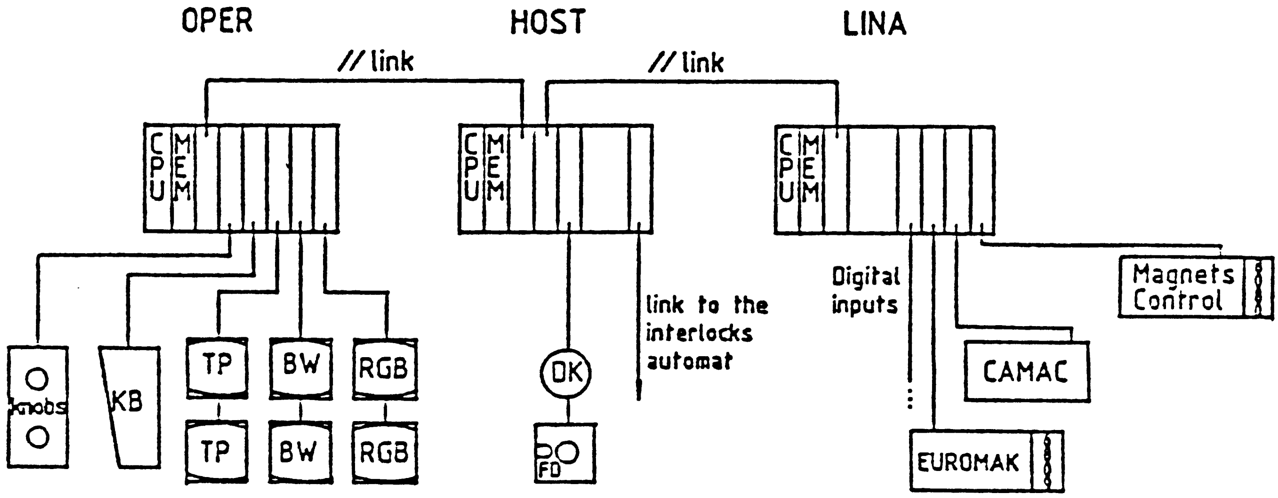

Fig.1 - Hardware Configuration

Interface to the equipments

All the equipments were previously interfaced through CAMAC modules. This solution suffers important disadvantages (expensive, relatively fragile, nuclear but no industrial standard) compared to its advantages (international recognised system, great variety of modules), so it was decided to progressively replace CAMAC by more modern standards.

One solution could be to interface the equipments directly to the VMEbus but this solution is rather expensive due to the complexity of the bus. So direct I/O interfaces on the VMEbus are reserved for the console equipments where fast response is primordial and for digital inputs where the VME module can directly interrupt on the CPU board. This last module was designed in our laboratory. For the console operation most modules were commercially available (high resolution colour graphic boards, touch panel and keyboard interfaces) except a shaft encoder interface that we have also designed. This module handles a pair of encoders and provides two convenient features: a software controllable brake for limiting the operator actions and 2 software controllable LEDs for warning the operator of the encoder(s) assignment.

For machine equipments interfacing we prefered to use a low cost 8-bit standard bus with industrial cards mechanically and electrically robust. The French EUROMAK bus was chosen : it uses single EUROcards, MC6809 micro-processors and 15 slots crates. A lot of industrial modules are available from the manufacturer. A first experience with EUROMAK is now acquired with the digitization of about 300 analog input channels splitted in 3 crates each containing multiplexers and ADC boards. The LINA station and the EUROMAK crates are temporarily linked with serial RS232C channels.

Future improvements

The major improvements planned in the next few months concern the hardware links between the VME crates and between the VME and EUROMAK crates. An ETHERNET network is now available in the VME standard and will replace our parallel links, the data transfer rates will be much higher and it will be easy to interconnect the VME crates to the micro-computer development system (MOTOROLA EXORmacs) for software down-loading.

It is planned also to interconnect our system to the nuclear physicist computers through the ETHERNET network, in order to make the linac parameters directly available to the physicists. Due to the bus structure of this network, it will be straightforward to add new VME crates if they are needed to increase the power of the system. The objection that this network is not very suitable for a control system since a maximum response time is not guaranteed [4] is not founded in our case: we have a small number of stations and the stations exchange only non time critical messages (updates of local and central data bases, operator commands, display lists).

The same type of network could be used to interconnect the VME and the EUROMAK crates, however as we planned in the future to increase the number of EUROMAK crates, an ETHERNET network will become prohibitively expensive. Following the choice of CERN for the LEP project [4] the aircraft MIL/STD-1553B network will be used for this purpose. The 1553B interfacing to the EUROMAK bus was done in our laboratory, two boards are used to manage the communications: the bus controller (BC) and the remote terminal unit (RT), one BC can control a network of up to 30 RTs. Following arrangements with the manufacturer the EUROMAK 1553B network is now commercially available.

To interconnect a VME master crate with EUROMAK slave crates equipped with 1553B RTs a VME compatible BC must be furnished. One such module for the military and aerospace market is available, but its price is incompatible with the type of applications aimed here. Therefore, taking advantage of the experience gained with the EUROMAK modules the VME BC module was also designed in our laboratory. As shown in figure 2, it is built around the SMC COM 1553B chip and is controlled by a MC6809 microprocessor which unloads the main processor from the 1553B hardware protocol (management of the 32 words 1553B messages, of the command and status words). Two dual-ported 16 Kbyte RAM and two semaphores are used to manage the exchanges between the local MC6809 and the main processor. These two areas can be used completely independantly: while the MC6809 performs a list of commands in one area, the host processor can prepare a new list or can read response to a previous command list in the second one. An interrupt is sent on the VME bus to signal the end of processing of a list by the local MC6809.

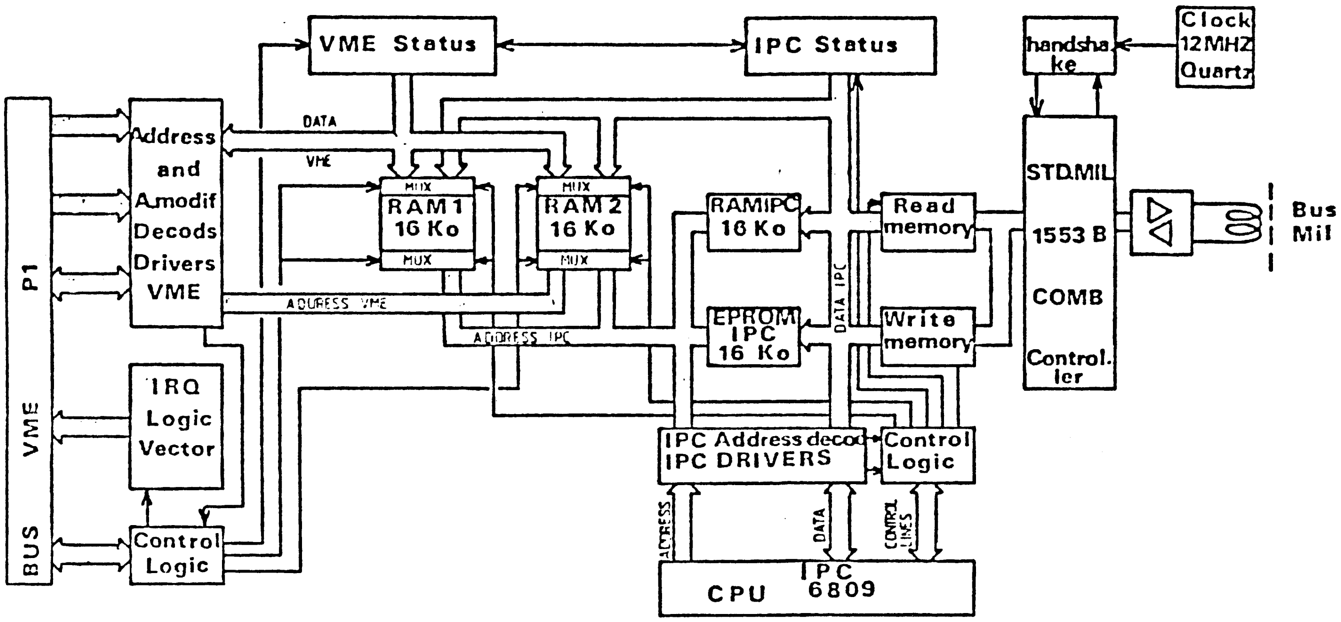

Fig.2 - MIL STD-1553B Bus Controller Block Diagram

SOFTWARE SYSTEM

A proportion of 80% of the old system software was written in three different assembly languages and therefore must be completely rewritten. An high level language was needed to rewrite rapidly all the software developed during 8 years and FORTRAN was chosen because it is widely used and understood in our Laboratory. Furthermore many of its disadvantages against more structured languages have vanished with the 77 versions.

The ease to write and maintain the programs with a high level language is paid by their least efficiency : the response times are increased by a factor from 4 to 6 regarding our previous system. Improvements are expected with the new generation of VME boards and also with the availability of a better FORTRAN compiler.

The software developments are done with an EXORmacs micro-computer from MOTOROLA. It supports easily 4 to 6 simultaneous users with 1 Mbyte of memory.

3 layers can be distinguished in the control system software:

Operating system

The EXORmacs and HOST station run the same VERSAdos disk-based O.S. The LINA and OPER stations use the RMS68k real-time kernel. VERSAdos is built around this kernel. This feature is very convenient : the programs are almost completely debugged on the development system before their final integration on the target systems.

Service software

This level concerns all the general purpose services which are furnished to the high level application software.

Intertask Communication : A message system has been developped which handles in the same way local messages in one station as well as messages exchanged between two different stations. The receiver of a message is addressed with a symbolic name :

name = STAT, SSTA, TASK

where STAT is the name of the VME station
SSTA will be used to address a substation connected to the 1553B network
TASK is the name of the receiving task in a VME station or an EUROMAK substation.

The management of the transfers can be modified by attributes specified by the receiving task:

- the receiving task can be blocked until a message is sent to it (default option) .
- the receiving task can test if there is a pending message for it.
- the receiving task can receive only messages of selected sending tasks,and in this case, discard or delay the other messages.

Data base management system : A distributed data base contains all the informations concerning the accelerator parameters. The data base management system was strongly influenced by the system used at SLAC for the SLC control system [5,6]. The accesses to the data base are done through standard functions with symbolic parameters. The data base is structured in subsets belonging to each station. The HOST station maintains a copy of the whole data base. Each subset is structured in blocks, the first block contains all the pointers necessary to access the data contained in the other blocks. In the four other blocks, the data are structured regarding their types : stable parameters, HOST write only parameters, HOST read only parameters, HOST only parameters. A Fortran-like format with or without a repeat specification is associated with each piece of information : integer 16 or 32 bits, real, character string.

Man-Machine communication facilities : A graphics support has been provided to display informations on the console screens. The output commands (draw a set of vectors, display a string of characters, fill an area) are passed to the graphics software and are stored in device independent graphic segments. Another set of commands is used to manipulate the segments (initialization, deletion, drawing, erasing, windowing) and to send them to specific device drivers.

The operator commands are communicated to the system through the use of touch panels and their operation are defined by symbolic files compiled by a specific program. This compiler allows the specifications of the location of a button, its label, and the actions to be taken when it is touched : e.g. the command:

```
COMMAND:05,07,     ,RUN SURVEILLANCE,SURV,RUN-SURV
```

generates a button at location 5,7 with the label RUN SURVEILLANCE and when it is touched, the message "RUN-SURV" will be sent to the task SURV. The command:

```
MENU   :05,08,   M,INDEX            ,IDLE,MENUINDX
```

generates a button at location 5,8 with the label "INDEX" in a medium size and, when it is pushed, the menu called "MENUINDX" will replace the current one.

Error handling facility

Based on the existing VERSAdos Error Messages Handler, a facility is provided to edit the warning or error messages in a standard way on the terminal attached to the HOST station. The messages appear always with the same format : date, originating task , message (with or without variable fields) - For example :

```
05/07  9h30m LINA-....-LACQ CAMAC ERROR IN CSSA  C=3  N=10  A=0  F=0
05/07 10h15m HOST-....-DBEX Protect code error (+4)
```

The most recent 200 messages are also logged on the disk for later analysis

Application software

The LINA station runs all the acquisition and control software under the control of interactive programs in the HOST station. It does also autonomous cyclic acquisition and surveillance. The interactive tasks running in the HOST station are disk resident except two tasks which display, on two dedicated screens, permanent informations about the accelerator status.

All the basic tasks available on the old system are now implemented.

CONCLUSION

According to the schedule, the system described here is operational since September 1985. Less than one month after the system start-up, it is obvious that some unavoidable youth problems are still present (minors bugs in the network software and in some application tasks, poor quality of our VME power supplies). Also as stated before the time responses must be decreased to improve the operator comfort. Nevertheless, with more than one year of experience in using VME microcomputers we can draw some conclusions:

. The choice of the VME standard, 2 years ago, was relevant : the number of vendors is extremely large and the module available are more and more performant, but the software furnished (compilers, utilities, system generation tools) is not always flawless or sufficient.

. The number of hardware failures is drastically decreased compared to our old technology system.

. The use of FORTRAN and of the software services mentioned above have been very successful to write efficiently the appplication programs.

. The hardware and software distributed architecture gives the system a great flexibility. This point will be particularly valuable for the extension of the control system to the ALSII project [7].

REFERENCES

[1] H.LEBOUTET et al., "First operation of the high duty cycle Saclay electron linac (ALS)", Proc. of the 1969 Particle Accelerator Conference Washington, Trans. Nucl., NS16, n°3 (1969) 299.

[2] G. BIANCHI et al., "A flexible dialogue with the computer in the control room of the Saclay's linac", Proc. of the 1976 Proton Linear Accelerator Conference, ChalkRiver (1976) 369.

[3] J.F. GOURNAY, "Logiciel du système de controle de l'Accélérateur Linéaire de Saclay" déc. 1979, Internal report 79-428.

[4] M.C. CROWLEY-MILLING, "Distributed Digital control of accelerators" Proc. of the 1983 Computing in Accelerator Design and Operation Conference, Berlin, Springer-Verlag (1984) 278.

[5] R.E. MELEN, "Design and performance of the Stanford Linear collider control system", Proc of the 1984 Nuclear Science Symposium, Orlando, Trans. Nucl. Sci., NS-32, n°1 (1985) 230.

[6] M. BREIDENBACH, N. PHINNEY, Private communications.

[7] B. AUNE et al., "ALS II : The Saclay proposal for a 2 GeV, CW electron facility" Proc. of 1983 Particle Accelerator Conference, Santa Fe, Trans Nuc. Sci. NS-30, n°4 (1983) 3296.