Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

Accelerator & Fusion Research Division

To be presented at the 6th IEEE Conference on the Application of Accelerators in Research and Industry, Denton, TX, November 3-5, 1980; and to be published in the proceedings.

A HIGH PERFORMANCE/LOW COST ACCELERATOR CONTROL SYSTEM

S. Magyary, J. Glatz, H. Lancaster, F. Selph, M. Fahmie,
A. Ritchie, C. Timossi, C. Hinkson, and R. Benjegerdes

October 1980

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

S. Magyary, J. Glatz, H. Lancaster, F. Selph, M. Fahmie,
A. Ritchie, C. Timossi, J. Hinkson, R. Benjergedes

Lawrence Berkeley Laboratory

ABSTRACT

Implementation of a high performance computer control system tailored to the requirements of the SuperHILAC accelerator is described. This system uses a distributed (star-type) structure with fiber optic data links; multiple CPU's operate in parallel at each node. A large number (20) of the latest 16-bit microcomputer boards are used to get a significant processor bandwidth (exceeding that of many mini-computers) at a reasonable price. Because of the large CPU bandwidth, software costs and complexity are significantly reduced and programming can be less "real-time" critical. In addition all programming can be in a high level language. Dynamically assigned and labeled knobs together with touch-screens allow a flexible and efficient operator interface. An X-Y vector graphics system allows display and labeling of real-time signals as well as general plotting functions. Both the accelerator parameters and the graphics system can be driven from "BASIC" interactive programs in addition to the pre-canned user routines. This allows new applications to be developed quickly and efficiently by physicists, operators, etc. The system, by its very nature and design, is easily upgraded (via next generation of boards) and repaired (by swapping of boards) without a large hardware support group. This control system is now being tested on an existing beamline and is performing well. The techniques used in this system can be readily applied to industrial control systems.

INTRODUCTION

The availability of powerful computers at low cost, in the form of 16-bit microprocessors, has opened new possibilities for simplifying control systems and for reducing their cost. The methods described here, for controlling an accelerator, can also be applied to other large systems, for example to industrial plants. This is because the modular nature of the control system architecture, and the great flexibility for choosing the characteristics of the components allows one to design a system that fits the requirements of the plant to be controlled. Although a large number of processors are used, sophisticated network software is unnecessary, because each processor is programmed to run independently of the others, and the database (which is the link between them) occupies prescribed areas of memory. Further, only standard high level languages are used in the programming; therefore, it is possible, even preferable, for the programming to be done by the engineering and technical personnel who are familiar with the detailed operation of the plant.

The SuperHILAC (Fig. 1) is a linear accelerator designed to accelerate all ions to a maximum energy of 8.5 MeV/amu. The linac is pulsed at 36 Hz. In order to obtain the greatest possible utility from the machine, three independent injection systems are provided, and pulsed parameters (such as RF, switching magnets, etc.) are changed before each pulse. This allows any ion from any injector to be accelerated on any pulse. A computer control system provides a practical means for rapidly switching a

large number of parameters, and since 1975 a network of minicomputers has been employed for control of the linac and the experimental area. The injectors and beam diagnostics have not been placed on this system, in part because of the great additional computational load this would place upon the central computer.

The microprocessor control system described in this paper will be used as a major addition to the SuperHILAC control system. It will be used to control the injection lines, to collect data from beam probes, and to drive alphanumeric and graphical displays. Although this system has stand-alone control capability, an interface will be provided to the older part of the control system, so that a single database can be referenced for archival purposes.

Fig. 1. SuperHILAC Schematic Layout

DESIGN PHILOSOPHY

In planning a new control system, based upon 16-bit microprocessors, we decided to implement the following basic concepts:

a. "Design" the computer to fit the needs of the control system. In conventional control systems one buys a standard computer system and then tries to design around its shortcomings.

b. Trade off hardware for software in order to reduce the ever-increasing cost of programming. To accomplish this we added computer processing power in exchange for simplified software. Parallel processing offered a way of doing this.

c. De-couple the scientific programming requirements from those required to control the accelerator. This allowed us to tailor each system optimally for each requirement and also reduced the need for system programming or modification.

d. Higher level language was used even if it meant having to add more computer processing power to make up for its slower execution speed.

e. Design a modular system that had a reasonably long future as far as upgradeability was concerned. In addition, maintenance and repairability should be a sufficiently simple task so a special group is not required.

f. The control system has a reasonable way to increase its processing power as the needs increase. We didn't want to write large amounts of applications software only to find out that we ran out of

computing power and had no graceful way to increase it except for purchase of a new computer. (That would require starting all over again with the software development cycle.)

g. The computers used should have a sufficiently large directly addressable memory space so a large but simple database can be easily accommodated.

The key to our implementation was the use of the MULTIBUS² which makes the concept of parallel processing possible, therefore allowing addition of computer power when needed. The bus also permits a modular and building block approach to the control system. A wide variety of manufacturers build MULTIBUS compatible CPU, display, interface boards, thereby allowing improvements when newer and faster CPU boards, etc., become available. (Manufacturers have announced chips/boards 3 to 5 times as powerful as the ones we are now using.) In addition, repair and maintenance is reduced to swapping boards and having spare boards for critical components.

DESIGN REQUIREMENTS AND RESTRAINTS

The SuperHILAC accelerates ions from three injectors on a pulse-to-pulse basis, therefore requiring a new machine state every 27.7 msec. This places severe speed requirements on those parts of the control system that must change on a pulse-to-pulse basis. The number of devices to be controlled in the third injector control project are approximately 130 analog type devices and 300-400 bit control devices. In addition there are hundreds of beam monitoring and real-time analog scope signals that must be dealt with. Future addition of the other two injectors will greatly increase the number of signals.

Experience with the operation of the SuperHILAC has shown that a system response time of 100 msec or better to an operator command is required. This and graphic displays of monitoring devices imposes severe computational and CPU bandwidth constraints.

Parallel processing enabled us to meet these demands on the control system.

ARCHITECTURE

The system architecture is shown in Figure 2. It is of the star type for two reasons:

a. It lends itself to the physical layout of the accelerator since most of the equipment is clustered around the machine.

b. It offers the most potential for fast data transfer between the distributed parts of the system.

Each module in our system is an Intel SBC 660 card cage containing 8 slots with the MULTIBUS back-plane connecting the slots. Within a "module" the multiple computers "talk" to each other over the MULTIBUS via the dual ported memory available on each board. However, when executing code, or addressing on-board resources the MULTIBUS is not accessed; this allows parallel processing, since the MULTIBUS is only used for messages.

There are several distributed input-output micro modules (IOMM); these units interface to the accelerator equipment and are generally 100-200 feet from the operator stations. The IOMMs control the accelerator parameters in synchronism with the accelerator timing pulses. Data is collected from the various IOMMs in the collector micro module (CMM) and is available for use there by the display micro module (DMM). These units (CMM, DMM) need not be tied to the real-time interrupts of the accelerator, but instead serve as a collector and disseminator of information and as interface between the operator and the control system.

To link the IOMMs with the CMM, serial data transmission is used over fiber optic links running at 38 kbaud. These links have high noise immunity and provide for a significant link speed upgrade (up to 5 mbits/sec). Data access into the CMM from the DMM is provided by a parallel extension of the MULTIBUS data and address lines, this distance being several meters.

Fig. 2. System Architecture

DATABASE

Each IOMM has its own database (in EPROM and RAM) describing all the required quantities so it can function in a stand alone environment. During initialization the IOMMs send all their data to the CMM so it contains the sum of all the IOMM databases (in the CMM all the data is in RAM). Once this initialization is done only those parts of the database that change dynamically are updated in the CMM. This allows the data links to run efficiently, updating only the data that is of interest to the operator.

The database consists of three major parts and at any instant contains all the information needed to describe the state of the accelerator:

a. The common passive. This is that part of the database that every channel has (i.e., name, element type, points to the other two databases, etc.).

b. The particular passive. This is data unique to a channel (i.e., channel number, units, conversion constants, etc.).

c. The particular active. This data contains the dynamically changing part of the data (i.e., setpoints, monitored values, time stamp, etc.). Only this part must be in RAM in the IOMM.

INPUT-OUTPUT MICROCOMPUTER MODULE (IOMM)

Each IOMM is an 8- to 12-slot card cage designed to hold MULTIBUS compatible boards (Fig. 3). All the boards used are commercially available. The IOMM has a common set of external interrupts and accelerator mode data supplied by a high-speed serial fiber-optic link. This synchronizes the IOMMs with the SuperHILAC pulsing sequence. To satisfy our timing requirements and to simplify software, we require that all I/O programs be completed in one pulse (27.7 msec).

Two functions must be performed in each IOMM; one is to handle communication to and from the CMM; the

² MULTIBUS is a trademark of Intel Corp.

other is to service the analog and I/O boards that connect to the accelerator devices. To simplify programming and to increase performance we use one computer board for each of these functions.

Fig. 3. IOMM Configuration

A portable console-computer can be connected to the IOMM in place of the CMM for local control and debugging of the IOMM and all devices connected to it. After check-out the IOMM can be re-connected to the CMM without disturbing the rest of the system.

Fig. 4. CMM Configuration

THE COLLECTOR MICRO MODULE (CMM)

The CMM (Fig. 4) simultaneously collects database information from all the IOMMs (via fiber optic links) and stores it in a block of memory reserved for the database. Here again we use two computer boards to separate functions and to speed up throughput. One computer services the USARTS and their associated buffers, while the other stores the data to and from the database. With this approach a large number of IOMMs could be handled, without degrading performance, by addition of computer and USART boards. A MULTIBUS extension board permits direct

access to the database from the DMM (operator station computer).

Fig. 5. DMM Configuration

THE DISPLAY MICROCOMPUTER MODULE (DMM)

The DMM (Fig. 5) is the heart of the control system and it contains three computer boards.

CPU1, the console computer, evaluates messages sent from the actual control console. The control console (Fig. 6) consists of up to 16 knobs (each

Fig. 6. Operator Control and Display Station with 8 Knobs

with its own 12-character dynamically alterable alphanumeric display), as many as four TV-based touchscreens, and a number of 16-key numeric key pads. The knobs were built in-house (see Fig. 7) and come in groups of four. They can be dynamically assigned to control any analog parameter via the database. CPU1 interprets messages from the console and (if required) notifies CPU2 of any action it must perform.

CPU2, the operating computer, receives the messages from CPU1 (via its dual-ported memory) and puts up the appropriate display information. A memory-mapped alphanumeric display board enables display of characters with various attributes such as reverse video, blinking, cursor position, etc.

CPU3, the graphics computer, controls (via a fast parallel interface) a graphics translator. Real-time

Fig. 7. Knob Panel

analog scope signals can be mixed in with the graphic/alphanumeric data (Fig. 8), thereby giving

Fig. 8. Analog Scope Signals with Labels and Graphs

the system the capability of a programmable and labelable oscilloscope. This makes the handling of analog signals much more versatile and automatic, and thereby relieves the operator from a burdensome and difficult task. The touch-screen (Fig. 9), showing a prototype page for controlling the analog scope, can be used for any pushbutton control function; it has one character resolution. Displays (alphanumeric and

Fig. 9. Touch-Screen with Prototype Control Buttons for Analog Scope

graphic/analog) are refreshed on the 10 per second rate. Future requirements can be easily accommodated by simply adding computer boards.

AUXILIARY COMPUTING CAPABILITY

Often there is a need for background programming

with access to control parameters. Examples are auto-tuning, modeling and on-line testing of new devices. To accomplish this a separate link is provided so any stand-alone computer (shown as a development system in Fig. 5) can control accelerator parameters. This approach de-couples the scientific type processing required for this type of calculation from the hardware type function of the control system. This link is currently used to allow Basic (Fig. 10) programs executing in a development system to control accelerator parameters. Since the development system has a disk-based operating system and supports Basic, Pascal, Fortran and PL/M, we have a wide variety of languages to choose from, plus use of the disk and other peripherals. This approach

Fig. 10. Development System with Basic Program for Ramping a Power Supply

neatly simplifies the computer programming task. We no longer have to modify operating systems to accommodate the control system, nor do we have to make a specialized control system look like a general mini-computer. This allows us to use the best approach required by each process without having to compromise the performance of either system.

PROGRESS

At this time we have completed and tested the hardware and software for the IOMMs, the communications link, and a substantial portion of the DMM. This system is currently running on an existing beam line and performing well (Fig. 6). The combination of parallel processing, along with the dynamically assignable and labelable knobs, and touch-screens makes the system extremely fast and flexible. The graphic computer software is being written now and will be available when the third injector begins operation (early 1981).