IEEE Transactions on Nuclear Science, Vol. NS-26, No. 3, June 1979

THE CESR CONTROL SYSTEM*

Raphael Littauer, Selden E. Ball, Ray G. Helmke, Stuart B. Peck, and David H. Rice#

1. Abstract

Computer control of CESR, the Cornell (8 + 8)GeV e^+e^- colliding-beam facility, places special emphasis on response speed, operator convenience, and economy of programming effort. Individual subsystem computers (PDP-11/34@) run autonomously under standard operating systems. Each has its own high-speed data bus network through which it controls accelerator components and accepts operator commands. Message traffic between computers is restricted to links which give a central DECsystem-1070 direct memory access to each of the PDP-11s. The 1070 can control higher-level operations by using slave tasks running in the PDP-11s; it also makes its resources available for data storage and manipulation, editing of program source files (mostly FORTRAN), and preprocessing by the FLECS translator.[1]

2. System Configuration

The CESR control system takes advantage of the natural separation between the three major accelerator components--the e^+e^- linac, the synchrotron, and the 8-GeV storage ring CESR. Each component is served by its own PDP-11/34@ *unit computer*, running under the RSX-11M multi-tasking operating system. The three PDP-11s are identical except for the fact that only the CESR unit computer is equipped with a floating-point unit. Each computer has a full complement of 124k words of memory and two hard disks. The required tasks are present in main memory or are brought in from disk.

As indicated in Figure 1, each unit computer drives a high-speed distributed bus, the *X-bus*.[2] All accelerator components connect to their X-bus via interface cards placed in nearby crates. Except in the case of some special-purpose cards, interfacing signals have a standardized digital, analog, or relay-power format; this permits the components to be tested without the computer through the use of portable signal emulators.

The X-bus also serves the control panels through which most operator commands are accepted. Most of these panels are grouped in an area of the control room designated for each unit system. Here setup and maintenance procedures can be performed. It is also possible to enter commands at any crate in the X-bus system--a convenience for some local service procedures. There are several keyboard terminals for each unit computer.

In each unit control area there is a TV display of all adjustable variables within the system. Moving a cursor on this directory permits the operator to assign any of four control channels to any of the system variables. In addition to these four generalized controls we have several groupings on special-purpose panels which are preassigned to those frequently used variables requiring most convenient access.

The main control console has a directory which can be assigned to any of the unit computers and thus gives universal access to the whole system. We expect the console's instrumentation to be focused mainly on colliding-beam operations. However, there are small dedicated panels for the linac and synchrotron unit computers which make available to the operator preselected groups of often-used controls to be touched up conveniently.

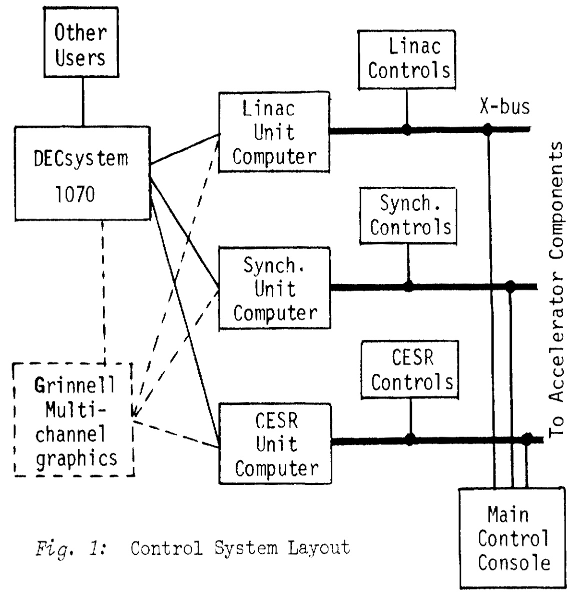

Fig. 1: Control System Layout

As described so far, the control system consists of three unit systems operating quite independently. This parallelism makes for simplicity of organization, but it does not provide for those control tasks which require coordinated action of two or more unit systems. This type of coordination may be achieved at two levels:

At the control-input level, it is possible simply to connect several X-bus interface inputs to the same knob or button, making the same information available to several unit computers. Alternatively, small amounts of information may be transferred between computers by connecting an X-bus output from one system to an X-bus input of another. For example, the CESR computer can tell the synchrotron computer what energy positrons it is set to accept. We have not exploited this hardware interconnection extensively yet, but it appears attractive in certain cases because of its simplicity.

At a higher level, the unit computers are linked to the laboratory's DECsystem-1070 processor, which is also used for on-line physics work and by other users. The 1070 has direct memory access to each PDP-11 via a bus-window interface. Coordinated control programs can be run from the 1070, using slave tasks in the PDP-11s. The computational power of the 1070 and its extensive peripheral facilities can thus be called into play. Easy two-way file transfer between computers also permits us to use the 1070 as a file repository, both for system data and for program sources. These latter are normally created and edited in the 1070, using its FLECS preprocessor.[1]

*Work supported in part by National Science Foundation

#Wilson Synchrotron Laboratory, Cornell University, Ithaca, NY 14853

@Computer designations are trademarks of Digital Equipment Corporation

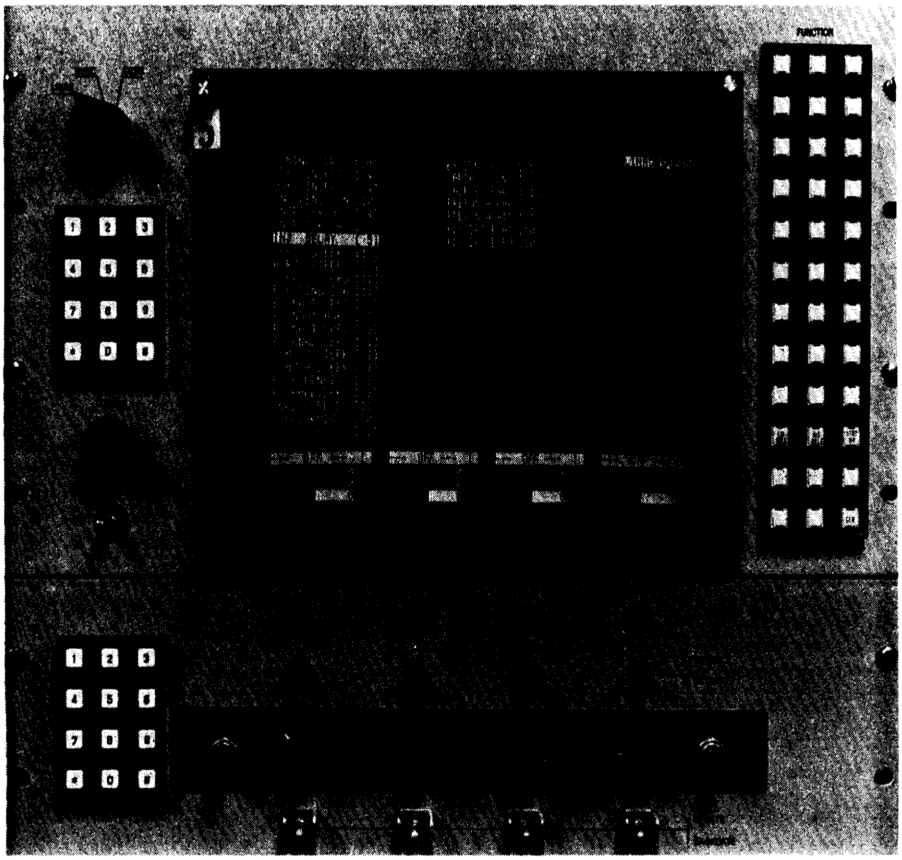

Figure 2:

Main Console Directory Area

3. Data Handling

Each X-bus operates in a polling mode under control of the currently active tasks in its unit computer. Although the X-bus design includes hardware interrupt facilities, we have so far not used them; this simplifies the control programs. Task priorities are handled by the task scheduler and can be assigned explicitly. The RSX-11M multi-task operating system has proved to be well adapted to the needs of a control system.

In the polling mode, response to operator commands from the control panels, and routine supervision of machine variables, depend on the execution of supervisory tasks at appropriate time intervals. Because of our emphasis on rapid response this background activity consumes an appreciable fraction of CPU time.

All data, including the current X-bus addresses of all machine components, are stored in a flexibly organized data base.* Machine variables can be addressed through easily recognizable eight-character mnemonics. Organization of this data base, together with implementation of the associated subroutines (which also control X-bus operation), represents the main software effort for development of the basic control system facilities.

With the exception of some low-level routines, systems programs are written in FORTRAN/FLECS, a convenient high-level language which speeds program development and yields flexible, well documented source files. The resulting machine code, after compilation, is however neither compact nor particularly fast in execution.

Users familiar with FORTRAN can pick up FLECS in a trice, and can also learn the data-base subroutine calls quite quickly. Almost anyone can thus write application programs for the control system. The chief obstacle to rapid program development lies in the need for compilation and task building in the PDP-11s. These are relatively slow processes and require the user to be familiar with the PDP-11 executive system. However, the fact that we have three independent computers helps productivity; also, we have a fourth PDP-11 set aside for program development and hardware testing.

If task *execution* speed ever becomes a serious limitation, the use of compiled code of course makes it possible to rewrite sections of it in more efficient (though less flexible) languages.

Data display in graphical and alphanumeric form is handled by a multichannel TV system.# This commercial system was adapted for use with several computers by the addition of a multichannel data buffer at its input. Each of our computers has access to the graphics system and can direct its output to any designated TV monitor. We presently use only black-and-white monitors; extension to color is possible.

Figure 2 shows an example of a TV monitor used to display the directory of variables available for control. This is the directory at the main console, assigned to the linac unit computer. Status information (command, analog value, and previous command) is included for each of the four control channels.

*We acknowledge the collaboration, during early phases of this work, of E. Knobil, present address: Systems Engineering Laboratories, Fort Lauderdale, Fl.33313.

#Model GMR-27, Grinnell Systems Corporation, Santa Clara, Ca.95050.

Having operated our accelerators for many years with conventional "knob-oriented" controls, we are reluctant to give up the convenience and immediacy such controls can offer. As a result we have developed inexpensive "knob" devices which are scanned so rapidly as to simulate instantaneous response.[3] Where these knobs are proliferated on special-purpose panels the associated display elements are simple LED modules driven by a single coaxial cable.

A data-base feature we have found useful during program development is the inclusion of a set of internal variables called *Program Input*. A control channel may be assigned to one of these variables, and another program can read the commanded value from the data base. This permits adjustment of many program parameters during execution, without the need for intermediate program recompilation.

4. Examples of Applications Programs

The strengths and weaknesses of a control system are best seen in the context of some specific examples of applications. Though we have not yet gone very far in program development, some basic features of our system have become apparent.

1. The operator-command program, which scans input knobs, lever switches, buttons, and keypads and sends instructions to the appropriate variables and tasks, runs just fast enough to permit effectively instantaneous response (provided the hardware is able to follow). Our usual scan rate is about ten times per second, and some care in program design is needed to avoid the system's becoming overloaded at this rate.

2. Status information can be updated at a similarly high rate wherever this is desirable. The multichannel graphics system operates so fast that quite elaborate displays can be generated on a current basis.

3. The control system cannot be expected to respond on a real-time basis as far as most accelerator time scales (rather than operator perception) are concerned. For example, to slew the magnet lattice from one operating point to another requires closely synchronized changes in each of 98 quadrupole power supplies and associated steering corrections.[4] This type of command is carried out in two steps: first the desired slew rate for each power supply is sent to an input latch which controls a binary rate multiplier; then a hardware clock slews all supplies together, each via its own rate multiplier to achieve the desired slew rate. It is easy for the CESR PDP-11 to handle this task, given a table of desired magnet changes. Although on-line computation and display of the current machine optics has been done on the PDP-11, *optimization* of parameters is not practicable, given the number of elements and the PDP-11's limited address space. Thus these computations are performed on the DECsystem-1070 instead, which permits us to use quite elaborate optimization programs. The CESR unit computer is relegated to the more basic task of implementing a specified table of magnet settings.

4. Some real-time response is available directly from the PDP-11s, however, provided the time scale is not too short. For example, we have swept the linac energy analyzing magnet ballistically by timed commands from its PDP-11. The associated task is run at high priority to ensure regular recurrence. In this manner it was easy to generate a "live" display of the linac's energy spectrum.

5. Beam-position information from the CESR beam detectors is also processed directly in the PDP-11, the response speed being limited by the coaxial relays in the detector cable network. This task will be transferred to the 1070 because it needs to be associated with an elaborate orbit-correction program which requires great computational power.

6. We have found it possible to calibrate the electrical centers of our beam detectors with respect to the magnetic axis of the adjacent quadrupole by finding that position of the *injected* beam where switching the quadrupole on and off produces zero steering effect (all CESR quadrupoles are powered independently). This calibration procedure will be directed by the 1070, using the beam-position program of the PDP-11 as a slave.

7. Evidently there are many injection and beam-manipulation tasks which require simultaneous control of parameters in different unit systems. Such tasks will normally be implemented in the 1070; however, hardware interconnections between X-buses of two or more computers may occasionally prove useful in transferring small amounts of information.

5. Summary

Program development effort has been minimized by use of standard operating systems in the unit computers, which run autonomously except for interaction with the DECsystem-1070. Though this organization yields only modest "network" power, the advantages of simplicity and immediacy of access to each subsystem are important. Similarly, the use of a standard high-level program language reduces the initial software investment, though at some sacrifice in speed of execution. However, in this respect the system can be improved progressively.

Total cost in manpower and dollars is difficult to define precisely because of the inevitable overlaps with boundary areas. Our best estimate is less than ten man-years of professional effort for system design and commissioning, including software development and hardware design for the X-bus, general-purpose interfaces, and video display system. The total cost has been below $1 million.

Beyond achieving these modest cost figures, we consider a distinguishing feature of this control system to be its rapid response and conventional control layout in areas where this was desired.

Major shortcomings, to the extent these have had a chance to declare themselves, are: the heavy CPU time requirements for maintaining a rapid scan rate; the severely limited logical address space of the PDP-11s; possible maintenance problems for three computers, each equipped with disks; and the slowness of compilation and task-building by the PDP-11s. Remedies envisaged include more extensive use of peripheral microprocessors, as in the knob scanner already implemented[3]; some task execution directly in the DECsystem-1070; and more systematic use of the extra PDP-11 available for development and testing purposes.

[1] Terry Beyer, Computing Center, University of Oregon, Eugene, Oregon 97403

[2] R.Helmke, D.Rice, and S.Ball, Interface Hardware for CESR Control System, paper E-14, this conference.

[3] D.Reaves, F.Dain, and R.Helmke, Applications of Distributed Microprocessors in the CESR Control System, paper D-45, this conference.

[4] D.Hartill and D.Rice, The CESR Magnet Power Supply System, paper J-22, this conference.