

NETWORK ASPECTS OF THE FERMILAB CONTROL SYSTEM

H. R. Barton, Jr.

Fermi National Accelerator Laboratory*
Batavia, Illinois 60510

Abstract

The control system of the Fermi National Accelerator is a heavily computerized network of distributed processors. One part of the control system includes a multidrop network of eleven Lockheed MAC-16 processors, a Digital Equipment Corporation PDP-11 computer, a Xerox 530, and a Control Data 6600 system. These computers exchange information using serial hardware and dedicated cable buses. This paper attempts to explain the individual functions of the central processing units in this network, describes the message protocols for computer communications, and deduces design guidelines for future distributed processing control systems.

Introduction

Over the last several years, a number of publications¹⁻⁵ have described methods of functionally distributing a computer system over a number of central processing units which are executing independent calculations in parallel while overall synchronization of the system is maintained by the exchange of messages between the processors via a communications network. At the Fermi National Accelerator Laboratory, several dozen processing units have been incorporated into the control system over a five-year period.^{6,7} The recent addition of a PDP-11/50 computer to the control system has resulted in this investigation of the computer network now at the Laboratory.

Figure 1 shows the network connections existing between 14 of the computers in the control system. The processing nodes are linearly connected and generally speaking the flow of information in the system is directionally oriented from left to right in the figure. The processors to the left of Figure 1 perform data acquisition and real-time control while those to the right are primarily responsible for doing more complex arithmetic calculations on the data. The next several sections briefly describe some of the functions assigned to the processors in this network.

Accelerator Mac-16 Computers

The programs which execute in these eight computers control the operation of the ring of magnets in the main accelerator, acceleration at the radio frequency stations, and the extraction of beam from the accelerator. All the A/D and D/A converters throughout the accelerator are fed to these minicomputers via a serial CAMAC highway.

Accelerator XDS 530

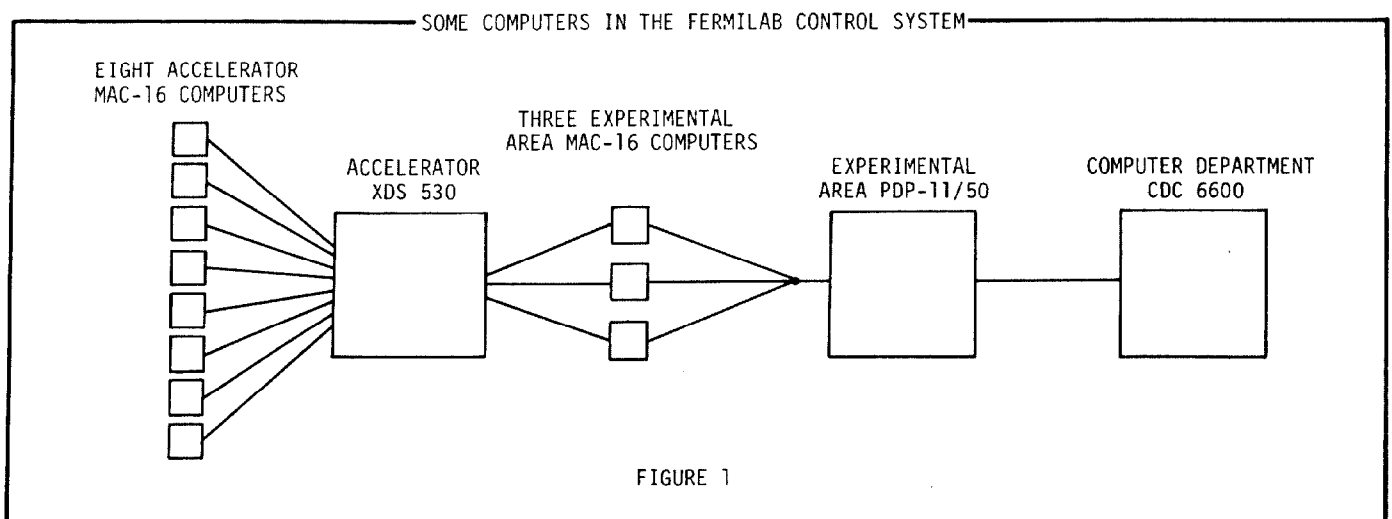
Although some adjustments of accelerator components are under closed-loop control by the MAC-16 minicomputers, the status of other components is monitored by interactive graphical displays in the accelerator control room. Using data transmitted from the MAC-16 computers, the XDS 530 produces these displays as required.

Experimental Area Mac-16 Computers

The elements in the beam lines are controlled by these three minicomputers. The A/D readings obtained via a serial CAMAC highway are displayed on CRT terminals (16 terminals on each computer) at each active experiment. Manual adjustments of the beam-line elements are accomplished by keyboard commands to these three computers. The computers have an extensive data base on fixed head disk to convert A/D readings to physical units and to check access rights to beam-line elements.

Experimental Area PDP-11/50

A complex protocol allows a task in the PDP-11 to obtain current device readings from the MAC-16. These data can be supplied to the location of an experiment in graphical form on a CRT display unit. The data can be stored on the PDP-11 disk to give a history of device readings over a longer period.



Data files can be transmitted periodically from the PDP-11 to the CDC 6600 in the Laboratory's Computer Department using a specially designed CAMAC communication link.⁸ The purpose of this is to create an archival file of the data for later, extensive analysis.

Standard Link Hardware

Except as previously noted for the connection to the CDC 6600, the same hardware is used for all link connections in the network. As in several other network applications,⁸⁻¹⁰ an attempt has been made to take advantage of existing CAMAC hardware. Each link connection terminates in three single width CAMAC modules: a transmitter module, a receiver module and a control module.¹¹ Each processor node in the network is specially interfaced to a CAMAC crate which contains a set of these CAMAC link modules. The communication line between processor nodes consists of two coaxial cables (RG-213) which give the hardware full duplex capability.

The hardware at the transmitting node constructs frames¹² in two formats: 6-bit frames for control information or 24-bit frames containing 16 bits of transmitted data. The receiving hardware extracts 16 bits of transmitted data from each frame as it arrives and returns an acknowledge to the sending node. The acknowledge is used as a signal to begin transferring the next frame. The transmission rate is 100K 16-bit data words per second.

Each different type of computer in this network requires a specially designed interface to the link hardware. The interface acts as a crate controller for the CAMAC crate containing the link modules and allows the processor or the memory of the processor to accept the arriving data. In the MAC-16 computer, the interface connects to the computer's Multiplexed Data Channel. Once the processor is interrupted by the first frame of an arriving message, the data channel transfers words of data to computer memory without requiring program intervention. This is accomplished by interleaving channel operation between computer instructions. At these data rates, this offers only minor interference with the CPU program running in the MAC-16.

The interface to the PDP-11 incorporates the DR11-C general-purpose interface between the CAMAC crate and the computer's UNIBUS. The DR11-C is a programmed data transfer interface which requires that the software in an interrupt service routine be executed for each word of data transmitted over the link. The result is that the data transfer rates possible at the PDP-11 end of the link are one order of magnitude slower than at the MAC-16 end. Although this delays message transmission, it also overloads the CPU because the processor spends unnecessary time repeatedly executing the interrupt service routine.

The problem of the interface represents a challenge to the design engineer. He must determine whether more than one link to a node will be in operation simultaneously. Will unsolicited arrivals of messages be allowed? The data presented to the processor may be collected by direct memory access or by programmed data transfer. When the interface for the PDP-11 was designed, it was decided that all link operations (input as well as output) would be initiated by the PDP-11 operating system and that concurrent transmissions to several links would not be permitted.

As is traditional in systems design, the PDP-11 link was implemented as a structure of layers. Each layer consists of distinct hardware or software modules with well-defined functions. At least four layers can be identified.¹³ The dialogue level is a complete set of user service routines which may be called from FORTRAN or assembly programs in order to perform user-requested operations on the network. This level of software support makes it possible for users to do meaningful communications between processor nodes without knowing the details of the link protocols.

At the logical link level, a series of messages is created which will be transmitted over the link. This level is responsible for synchronizing program execution in the computers which form the network. The physical link level includes software which creates I/O requests in the proper sequence so that programs will resume execution when message transmission is complete. The hardware level is the actual CAMAC modules which make up the link hardware as well as the interface which connects to the DR11-C on the PDP-11 UNIBUS.

Redundant Links

Besides the links shown in Figure 1, two additional links exist between each of the Experimental Area MAC-16 computers and the PDP-11/50. The first of these allows the MAC-16 to act as a terminal concentrator retransmitting command lines from keyboards in the experimental areas to the operating system of the PDP-11/50. In this mode of operation, any keyboard connected to a MAC-16 computer functions as though it were connected directly to the PDP-11. To accomplish this, one crate on each of the three serial CAMAC highways from the experimental area MAC-16 computers is located close to the PDP-11/50. A buffer module placed in this crate is connected to a DL11 communications interface on the UNIBUS.

The second additional link allows the PDP-11 to transfer 256-word blocks to any module on one of the serial CAMAC highways which are attached to the experimental area MAC-16 computers.¹⁴ This capability is used by the PDP-11 to transfer a graphical display to any experiment. The UNIBUS is not directly connected to any experimental area serial CAMAC highway. To remedy this, a parallel CAMAC branch driver was obtained for the PDP-11 and made it possible for the computer to address a parallel CAMAC system. A buffer module (with RS-232 port) in the parallel CAMAC crate is connected by twisted pair to a similar module in the nearby serial CAMAC crate. This scheme allows data to travel from the PDP-11 as far as a module on one of the serial highways. At this point, a message is sent over the standard link (Figure 1) to one of the experimental area MAC-16 computers to notify it that data has been deposited. The MAC-16 can then issue a serial CAMAC command to block transfer the data around the serial highway. Since the amount of data needed to make up a normal graph exceeds the memory of the buffer modules, usually several block transfers are required.

Future Control Networks

Although extensive interconnections have been installed between the computers of this control system, each computer was intended to perform its assigned function autonomously of the others.⁶ For example, no scheme for managing a distributed data base was implemented. The links were installed to facilitate the exchange of modest amounts of data. Now that the processors in the system have become

fully loaded, recent attempts have been made to share the computing load between several computers and to synchronize the computations.

In the past, it has been possible to assign control functions to separate processors without significant overlapping of responsibilities in either computation or data base. This makes it possible to distribute the system over a set of non-overlapping processing nodes. Synchronization of the processing at the separate nodes is no problem if the processors have no need to share resources (memory, peripherals, input data or calculated results).

Future physics experiments at the Laboratory will be done by colliding beams within the main ring tunnel. This appears to couple more closely the functions of particle acceleration, beam transport and physics research. In view of this, I anticipate that computers in the control system will be required to cooperate more closely than they do today. In addition, the technological advances in accelerator design such as super-conducting magnets may increase the computing load on the processors and this may require more load sharing to accomplish real-time control. As an example, several computers may have to exchange results to diagnose a malfunction requiring the acceleration cycle to be terminated prematurely. Such fault tracing may have to be done in real time in order to prevent periods of accelerator downtime.

Any improvements in the control system will have to be obtained using computers which are not significantly different than the ones already installed. Faster processors or memories are not likely to be produced in the next few years. Although new software systems will be available, these primarily offer new options and convenience, not any improvement in real-time performance. If the control system is carefully distributed over a number of coupled processors which are synchronized in parallel computations, performance improvements could be obtained with currently available hardware.

References

*Operated by the Universities Research Association, Inc., under contract with the United States Energy Research and Development Administration.

1. D. G. Dimmler, "Functional Distribution - An Architecture for Multi-User Computer Networks in Instrumentation," IEEE Transactions on Nuclear Science, Vol. NS-21, February 1974, 838-850.
2. F. W. Stubblefield and D. G. Dimmler, "Transaction Processing in the Common Node of a Distributed Function Laboratory Computer System," IEEE Transactions on Nuclear Science, Vol. NS-22, February 1975, 473-479.
3. J. D. Schoeffler and C. W. Rose, "Distributed Computer Intelligence for Data Acquisition and Control," IEEE Transactions on Nuclear Science, Vol. NS-23, February 1976, 38-54.
4. B. Zacharov, "Distributed Computer Systems in the Laboratory Environment," IEEE Transactions on Nuclear Science, Vol. NS-23, February 1976, 436-441.
5. J. V. R. L'Archevêque and G. Yan, "On the Selection of Architectures for Distributed Computer Systems in Real-Time Applications," IEEE Transactions on Nuclear Science, Vol. NS-24, February 1977, 454-459.
6. R. E. Daniels, R. W. Goodwin and M. R. Storm, "The NAL Computer Control System," IEEE Transactions on Nuclear Science, Vol. NS-20, June 1973, 505-509.
7. M. E. Johnson, E. J. Barsotti, H. R. Barton, Jr., J. Bobbitt, M. Haldeman, T. Lahey, R. L. Loveless, D. Purvis and J. Simanton, "Overview of a Multi-User High Energy Physics Control and Data Acquisition System," IEEE Transactions on Nuclear Science, Vol. NS-24, February 1977, 356-361.
8. A. E. Brenner, T. F. Droege and R. G. Martin, "BISON-NET, A High Speed Block Multiplexer Channel for Interconnecting Computers via CAMAC," IEEE Transactions on Nuclear Science, Vol. NS-22, February 1975, 493-498.
9. B. Zacharov, "CAMAC for Data Links and Computer Communications," IEEE Transactions on Nuclear Science, Vol. NS-21, February 1974, 898-902.
10. J. M. Potter, D. R. Machen, F. J. Naivar, E. P. Elkins and D. D. Simmonds, "CAMAC Based Computer-Computer Communications via Microprocessor Data Links," IEEE Transactions on Nuclear Science, Vol. NS-23, February 1976, 448-451.
11. S. R. Smith, R. W. Goodwin and M. R. Storm, "Intercomputer Communications in Real Time Control Systems," IEEE Transactions on Nuclear Science, Vol. NS-20, June 1973, 536-540.
12. IBM Synchronous Data Link Control General Information, GA27-3093-0, IBM Systems Development Division, Research Triangle Park, North Carolina, March 1974.
13. Product Description of DECNET, Digital Equipment Corporation, Maynard, Massachusetts.
14. E. J. Barsotti, H. C. Lau and J. R. Simanton, "Operational Aspects of a Serial CAMAC Control System," IEEE Transactions on Nuclear Science, Vol. NS-21, February 1974, 881-885.