CONF-811040--93

A CAMAC-BASED INTER-COMPUTER COMMUNICATIONS SYSTEM

D. E. Greenwood, R. D. Burris
Computer Sciences at
Oak Ridge National Laboratory*
Oak Ridge, Tennessee 37830

and

K. G. Young
Fusion Energy Division
Oak Ridge National Laboratory*
Oak Ridge, Tennessee 37830

Summary

We have used CAMAC hardware to provide communications between dissimilar computers for the ELMO Bumpy Torus (EBT) experiment. The software supports file and individual message transfers. The system has proven to be both reliable and fast, with transmission rates of about 36,000 baud.

Introduction

Primary computer support for the EBT experiment is provided by two PDP-11/34's, two PDP-8's and a Nuclear Data ND6600.[1] A MIK-11/23 will be added shortly, with additional systems (a VAX 780 and two PDP-11/24's) planned for the future. All of the 11-based systems run under RSX-11M v3.2. One of the 11/34's is used to monitor the status of various machine parameters while the experiment is in operation. The other 11/34 is the center node of a star shaped local network and maintains the official system clock and various identification files for all diagnostics. (This system is known as "EBC", while all other systems are collectively known as "outboard" systems.) All of the systems acquire data and perform some preliminary data analysis.

Most analysis work, and all permanent data storage, is done on a PDP-10 maintained by the Fusion Energy Division (FED) of ORNL. A PDP-11/45 acts as a communications front end to the PDP-10, connecting it to data acquisition computers at the various FED experiments (including EBT), to other ORNL computer systems and to the entire MFE network. Virtually all of EBT's information flow is toward the 11/45 and the 10. The communications software running on the computers used by the FED experiments was written locally and provides an efficient means of transferring files between computers.[2,3] The two EBT 11/34's are part of this local FED network.

At EBT, however, we encountered three problems with attempting to use the existing network. First, specific hardware was required for connecting any pair of computers. This limited our options for selecting new computer systems. Second, while the existing network software provided the ability to translate files from the format required by one computer to that required for another, each file could consist of only a single data type (i.e., single precision floating point, ASCII, etc.). The data for any diagnostic for EBT usually consists of multiple data types, all of which had to be stored in a single data file on the PDP-10. Finally, the existing network provided no message transfer capability. It is necessary for each of our computers to have access to the official date and time and other experiment coordinating information. Requests for information of this sort are known as "function requests".

We decided to develop a communications system based upon CAMAC hardware for several reasons. First, all of our computers except for the ND6600 were already or could be interfaced with CAMAC. Second, CAMAC modules which were to be used for data acquisition could also be used for communication. Third, faulty units could easily be replaced from backup supplies which we would maintain anyway. Fourth, additional computers could be easily added to a CAMAC communications system, provided that those computers could be interfaced to CAMAC. Finally, it was possible to use modules which would provide 24-bit parallel transmission with a hardware interlocking feature which meant that reliable transmissions with high transfer rates were at least theoretically attainable.

CAMAC Hardware

The primary set of modules we decided to use are quad input and output modules made by Joerger. Each module is a four-channel input or output register in a single width CAMAC module. Each channel provides 24 bits of parallel data transmission plus handshaking signals. The handshaking keeps the output register (QOR) from overwriting the input register (QIR) until the data has been read from the QIR by the computer. Connecting two computers requires only a standard ribbon cable with connecters at each end. We currently use cables up to about 20 meters in length. Longer cables could presumably be used but are not required for our purposes.

A disadvantage of these modules is that input data cannot be read using direct memory access (DMA). The requirement that each incoming word be read individually by the software resulted in considerable system overhead, causing slower data transfer rates than were desirable and some degradation of response time for the system as a whole. The maximum transmission rate from a PDP-8 to EBC using just QIR and QOR modules was about 4,000 baud.

We solved this problem by using LeCroy 8801/16 dual port memory modules. These single width modules each have 16K memory of 16-bit words which can be accessed for reading or writing via either the CAMAC dataway or an external port. This module can be read DMA, although without modification it cannot be written using DMA. We connect one channel of a QOR module to the external port of the 8801. Data is output from the QOR and stored temporarily in the memory of the 8801. The connecting cable is still a ribbon cable, except that the clock line must be coax. In addition, the clocking pulse is inverted by a Joerger TTL Fanout module before the pulse reaches the 8801. (The data words received by the 8801 must also be inverted or complemented by the software.) The modules just described permit transfers in one direction only. A second 8801 must be used for transfers in the other direction. (The TTL Fanout module has two channels and so the same module may be used for transfers in both directions.) As with regular QOR to QIR cables, we have used cables for an 8801 of up to 20 meters. Data is transmitted at about 36,000 baud using this hardware,

MGU

even though the data is only read using DMA, -- not written.

It should be noted that the path through an 8801 supplements, not replaces, the QOR to QIR path. This is due to the fact that the sending computer cannot conveniently use the 8801 to notify the receiving system that a message is available.

Software

Several criteria had to be met when designing the communications software. First, we did not want to make any system software changes. Second, the software had to permit user routines to obtain experiment status information at any time, without creating and/or sending files. Third, the communications software was to do whatever translation was necessary to convert data from one system's format to another's. Fourth, it was necessary to transfer files from the CAMAC network to the FED network for transmission to the PDP-10. Fifth, the user interface was to be as clean and simple as possible. In fact, for the primary use of the network, which is to prepare and send files to the EBT database on the 10 and to acquire experiment coordination information, the user should not have to be aware of the communications aspects of the process at all. Finally, it was desirable that user routines should be able to run on either 11/34 (or other 11-based processor) or either PDP-8 without modification to the source code, recompiling or even, if possible, reloading.

On EBC, communication is handled by a separate task for each outboard node. These tasks are clones of a single task image file, each installed under a unique name. A command string passed to the task when it is initially run specifies the node name and type of computer with which it communicates and the device names and channels of the CAMAC modules to be used. (Device names are converted to crate and station numbers by the CAMAC software driver.)

The communications tasks are normally in a stopped state, so that they require virtually no system resources (in particular, CPU time and memory). A separate task, called SNUPER, scans all QIR channels for input from outboard systems, unstopping the appropriate communications task when input is received. (Essentially, SNUPER simulates the LAM interrupts which our CAMAC software driver does not support.) If a file is to be transferred, the command and the file are written to disk. A separate task, called XRT, reads the command and translates the file, if necessary. The translated file is then routed to the destination node via PIP (if EBC is the destination node) or via the FED network software. The only responsibility of the communications tasks, then, is to acquire the file from the outboard computer and pass the file to XRT.

Function requests are checked for legal function codes and correct argument lengths. The request is passed to a task called HDR which processes the request and returns an answer. The answer is then transmitted to the outboard system.

Both HDR and XRT may be accessed directly by data acquisition tasks running on EBC. Subroutines have been provided to minimize the effort required of the applications programmer to create files for translating and transmission and to make function requests.

Tasks named XRT and HDR also run on all PDP-11's other than EBC. All XRT tasks are the same regardless of the system on which they are running. For outboard 11's, HDR is actually a communications task which transmits the function request to EBC for actual processing. The result is that application tasks may run on any PDP-11 without modification due to communications reasons.

For PDP-8's a subroutine call is made for all communications requirements. Since the PDP-8 is a single process machine, all communications software must be loaded with the user routines. Several subroutines and programs have been written to make access to communications as easy as possible. In particular, all message formatting, whether for files or function requests, is internal to the communications subroutines.

Protocol

General

The communications protocol is based loosely on DDCMP.[4] Messages may contain control information only or control information plus data. Unlike DDCMP, our protocol places control information both before and after data. In addition, control information is always three bytes (24 bits) long. Data information is always passed as 16-bit words, with the high order eight bits of the 24-bit parallel path for QOR-QIR transmission passed as zeros. The general format of a message is a 24-bit (3-byte) header, followed by any number of data words, a check word and a 3-byte end-of-message trailer. Either the header or trailer may be transmitted by itself.

When any computer is restarted or reloaded, it clears all communications CAMAC modules at its end and issues INIT (see Table 1). Upon receipt of INIT, the other computer aborts any pending transfers, reinitializes its own CAMAC modules and issues INACK. The link is then initialized for transmission. EBC specifies the transmission route (i.e., QOR-QIR only or with one or more 8801's) as part of its INIT or INACK message.

TABLE 1. Protocol Formats

INIT	005	000	RTE
INACK	007	000	RTE
DATA	201	WC1	WC2
EOM	004	EOF	MSG
OK	006	000	LMR
NOK	025	000	LMR
FUNC	130	WCF	FNC
ANS	101	WCF	FNC
NOF	125	ARG	FNC
FARG	106	WCF	FNC

RTE specifies the route (e.g., QOR-QIR only)
WC1,WC2 are high and low order word counts for data messages
EOF is 377 for EOM(EOF), 000 otherwise
MSG is the no. of the current data message, mod 400
LMR is the no. of the last correctly received data message
FNC is the function code
WCF is the length (in words) of the argument or answer
ARG is 377 for NOF(ARG), 000 otherwise

Function Requests

Function requests are always initiated at outboard nodes. A FUNC message is sent to EBC, specifying the numeric code of the desired function and the length of the argument, if any. If the function code and argument are both legal, EBC will return ANS, the answer (if any) and EOM. Both the argument and answer are ASCII, with a maximum of 25 characters. Any necessary translation (such as to or from the PDP-8's six-bit ASCII) is done by the communications routines at EBC. If EBC detects an error in the function request, one of three responses is returned. NOF indicates that the function code is illegal. NOF(ARG) specifies that the function code is legal but the argument is not. FARG indicates that the function code is legal but the argument is of the wrong length. These messages make outboard software appear to verify function requests, when in reality the software on EBC does all verification.

File Transfers

File transfers are performed through a series of data messages. Each data message is preceded by DATA and followed by a check word and EOM. The first data message contains source and destination node and file name specifications and any switches specifying what translation is to be done on the file. All data from DATA messages after the first are placed in a file, until EOM(EOF) is received. The file is then passed to XRT for translating and routing.

One or more function requests may be made during file transmission. Only one file, however, may be transmitted at a time between any pair of computers.

Timing and Error Control

When transmitting any information via a QOR module to a QIR module, the sending computer waits for 12 seconds for the receiving computer to read the current information. If the 12 seconds elapse, the sender terminates the link and issues a diagnostic message to the console. After receiving EOM, other than EOM(EOF), and sending OK or NOK, a computer waits 36 seconds for the next message. If none is received, the link is terminated and a diagnostic message issued.

The software uses word counts, message numbers and check words to verify data and function messages. Message numbers are used only for data messages, are initialized to zero after each INIT-INACK sequence and are incremented for each data message, modulo 400 (octal). The check word is initialized to zero before each data or function message. Then, as each word of the message is read, the check word is rotated left and the message word exclusive-ored with the check word. Control information is not included in the check word. When an 8801 is used, an additional pair of data words are transmitted, one before the message and one after the check word. These words contain alternating bits of 1's and 0's and insure that all 16 data lines may be set or cleared when desired and that the correct number of words have been received at the 8801.

---

*Operated by Union Carbide Corporation under contract W-7405-eng-26 with the U. S. Department of Energy.

References

[1]R. D. Burris, D. E. Greenwood, J. S. Stanton and K. A. Geoffroy, "EBT Data Acquisition and Analysis System," ORNL/TM-7464 (1980).

[2]C. O. Kemper, "An Automatic Data Acquisition and Communications Computer Network for Fusion Experiments," Ninth Symposium on Engineering Problems of Fusion Research, October 26-29, 1981, Chicago, IL.

[3]R. D. Burris, C. E. Hammons, C. O. Kemper, "Net - A Powerful File Transfer Facility," Proceedings of the Digital Equipment Computer User's Society, Vol. 5, No. 2 (1978).

[4]J. E. McNamara, Technical Aspects of Data Communication, Digital Press, (1977).