IEEE Transactions on Nuclear Science, Vol. NS-28, No. 5, October 1981

VAX/LSI-11/CAMAC NUCLEAR DATA ACQUISITION SYSTEM UNDER DEVELOPMENT AT THE W.K. KELLOGG RADIATION LABORATORY, CALTECH

J.D. Melvin, M.H. Mendenhall, R. McKeown, and T.A. Tombrello
Kellogg Radiation Laboratory
California Institute of Technology 106-38
Pasadena, California 91125

Summary. A CAMAC data acquisition system is currently under development at the W.K. Kellogg Radiation Laboratory which utilizes LSI-11 crate controllers and a 250 Kbyte/sec 7-computer Q-bus DMA network. The central network computer is a VAX 11/750 with Unibus and Q-bus adaptors. Processing is distributed between the LSI-11s and the VAX as required to meet input-output and event processing speeds needed by the experiment. Use of the Q-bus on all machines allows CAMAC controllers and tape drives to be moved as needed between the LSI-11s and the VAX.

General Description

The Kellogg Laboratory operates three Van de Graaff accelerators. In the past, data acquisition has been carried out with hardwired single-parameter multi-channel analyzers and one obsolete computer driven single or multi-parameter analyzer, which is difficult to program. Current data acquisition needs require a computer based, easily programmed, multiparameter, laboratory data acquisition system.

Because of other computing needs of the laboratory in the areas of theoretical nuclear physics, text processing, and microprocessor development, as well as general programming, a data acquisition system has been designed which includes a central computer that can carry out these tasks as well as high speed real-time data analysis. A VAX computer was chosen for the central computer because of its 32 bit architecture and multiprocessing virtual memory operating system which can handle large numbers and sizes of programs. In addition, careful study of the VAX VMS operating system has shown it to be excellent for high speed real-time applications.

Since much time is spent by any data collection computer servicing interrupts, front-end processors are also incorporated. Various front-end processors were considered, including the Bi-Ra microprogrammable branch driver which has very high throughput. The decision was made to use a standard microcomputer for which many software development aids and peripherals are available, because of limitations on personnel available for development of custom programs and circuitry. The LSI-11 was chosen because its instructions are similar to the simpler VAX instructions and because an excellent real-time operating system, RT-11, is available for it. Also, the LSI-11 system allows operating two processors of differing cost and performance, the 11/2 and 11/23. CAMAC was chosen to provide a standard interface to the actual experiments.

The standardization of hardware and software allowed by the system components chosen has greatly facilitated development of the data acquisition system. This standardization also will allow the final system to be easily reconfigured for different experiments. Furthermore, by adding a Q-bus (LSI-11 bus) to the VAX Unibus though a commercially available converter (which is required for connection of the VAX-to-LSI-11 communication network), and by selecting a Q-bus controller for the tape drive on the VAX, we have provided a laboratory wide standard for peripheral-to-computer interfaces. The CAMAC crate controllers and tape drive can be moved between the VAX and LSI-11s for each experiment to meet requirements of input-output and event processing speed.

Various options were examined for communication between the front-end LSI-11s and the VAX computer. A newly available commercial product, which implements a general purpose, high speed, star network between seven processors, was selected. Drawing on the networking experience of one of the authors (Mendenhall), remote disk support for the RT-11 operating system on the satellite processors was implemented through the network. Subsequent software effort has shown that writing the front-end software using the tools of RT-11 rather than as a stand-alone program has greatly simplified the task of communicating with the hardware. High throughput is still achievable if large buffers are used in the LSI-11s and the VAX since time consuming system operations are restricted to the relatively rare occasions of buffer switching.

The data acquisition for a given experiment has a software configuration divisible into four parts: (1) data collection, (2) event recording on tape, (3) real-time event analysis (such as histogramming), and (4) experiment monitoring (such as displaying of histograms) and control. These parts can be placed partially on the satellite and partially on the VAX host, with the network providing a software transparent link in between. The flexibility for moving the hardware between the LSI-11s and the VAX is thus paralleled in the software.

Two kinds of graphics display are provided: Tektronix 4010 storage scope terminals with interaction via cursors, and DMA (Direct Memory Access, i.e., without computer intervention) vector point plotting interfaces on the Q-bus. The latter are fully programmable, and are capable of supporting live, interactive display. In addition, graphics hard copy is provided by a Versatec printer-plotter and several Decwriters with a plotting modification.

Initial development of the system will be completed during the summer of 1981. The system will provide the primary data acquisition capability for a new multi-purpose high current, high stability 3 MV tandem accelerator to be installed in the laboratory during the summer of 1981. It will also support counter development for a neutrino detection experiment at LAMPF. As resources permit, a 200 foot extension of the network to reach the existing 6MV tandem accelerator will be implemented using high frequency serial fiber optics transmission.

Details of the Hardware Configuration

Figure 1 diagrams the hardware layout of the VAX and of one satellite LSI-11. The VAX is a standard 11/750 configuration from DEC with two RK07 disk drives. To the Unibus we have added the Versatec V80-211 interface and graphics printer, and to the RK07 controller a Unibus extension cable to an Able Model 10067 Quniverter Unibus-to-Q-bus converter which

0018-9499/81/1000-3739$00.75©1981 IEEE

plugs into an LSI-11 extension chassis. In the Q-bus is a Western Peripherals Model TC-151 tape drive controller, software compatible with the DEC TM-11 controller, connected to a Perkin Elmer Model 1275, 800-1600 bpi, 125 ips tape drive.

Also on the Q-bus is the Peritek HEX-11 network system consisting of a host DMA interface and a network processor which attaches via 40 line ribbon

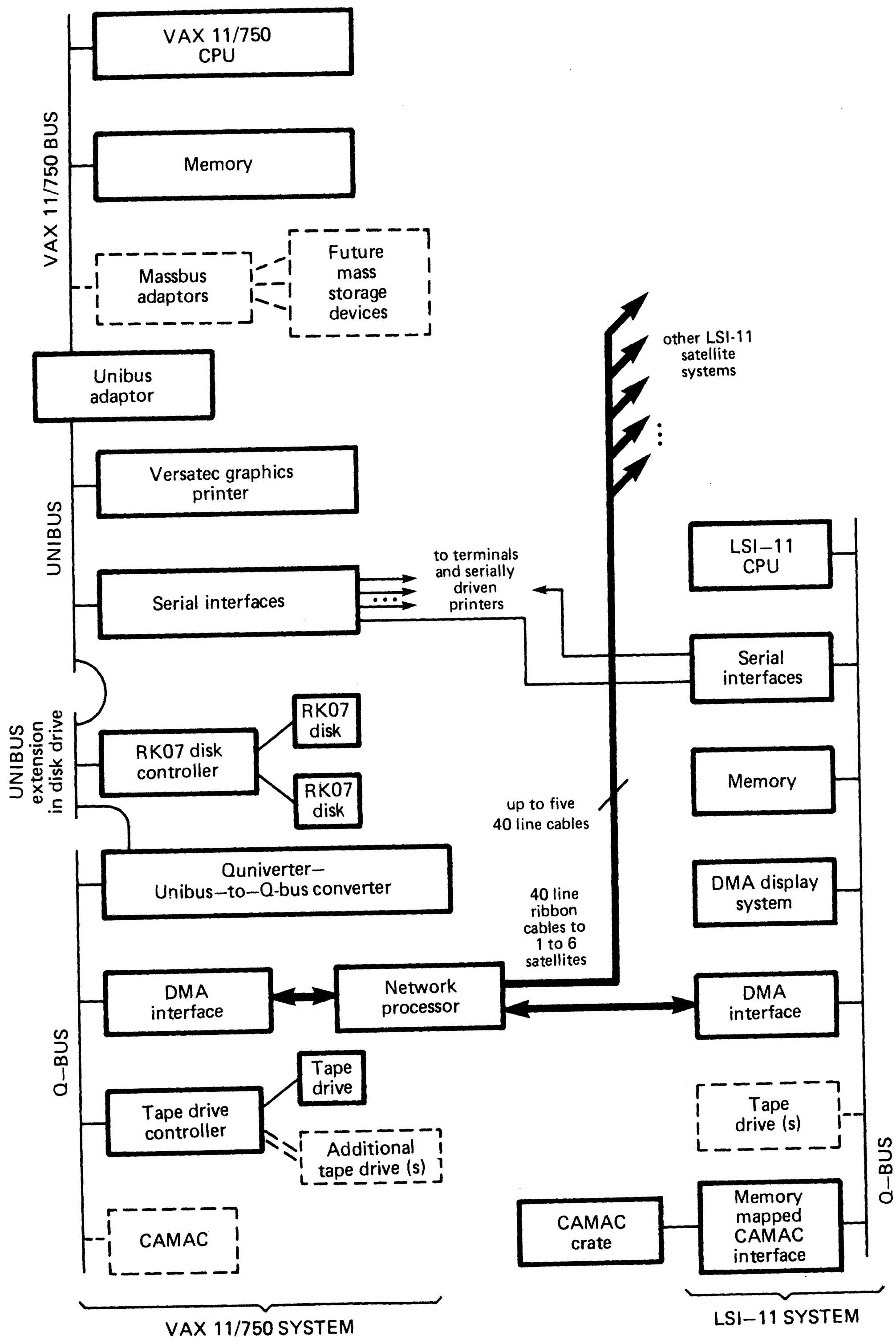

FIGURE 1 Hardware Layout of the Kellogg Data Acquisition System

cables to one to six satellite DMA interfaces. The network supports satellite-to-host interrupt with flags to indicate which satellite interrupts, satellite selection by the host, and communication with the selected satellite including (1) interrupting of the satellite, (2) setting and clearing of 6 satellite and 3 host flag bits, (3) sending four 16-bit mail box registers in either direction, and (4) DMA transfer of 1 to 32767 16-bit words in either direction. Furthermore, the satellites have ROMs which allow bootstrapping of programs sent from the host.

The satellites have serial ports which can be connected to the host (in which case the host can activate the bootstrapping ROM of the satellites under program control) or to local terminals and printers. They have memory mapped CAMAC interfaces, Bi-Ra Model 1311-2, which allow the LSI-11 to read or write 16 or 24 bits to the CAMAC dataway in 1 or 2 memory references, respectively, and to respond to 32 LAMs with vectored interrupts.

Currently we are using LSI-11/2 processors with 60KB RAM. 11/23 processors with up to 2MB RAM will be used if speed or peripheral buffer capacity require. Consideration also has been given to implementing a 68000 processor on the LSI-11 Q-bus to gain another factor of 2 to 3 in speed over the 11/23, should speed become a problem. However, this possiblity would preclude use of the RT-11 operating system. Since the VAX has a Q-bus, high rate experiments will more likely be accommodated by moving the CAMAC interface to the VAX. By using the unused BR7 Unibus interrupt priority level, the VAX 11/750 can be configured to vector directly to an interrupt routine in under 10 microseconds, and properly written software can do this in a manner that does not interfere with the integrity of the VMS operating system. (Software implementation of fast interrupt response on the VAX is discussed in the next section.)

A low cost live display is provided by the Data Translation Model DT2771 DMA interface which transfers 12 bit X and Y values, stored sequentially in an array, to digital-to-analog converters along with a delayed intensify pulse. The outputs of this interface directly drive any X-Y display scope. The most common mode of operation for this display involves generation of display arrays by the VAX from data arrays created by histogramming in the VAX, and transmission of the generated arrays to the satellite computers for display. Transmission of a 4K display over the network requires only 64 msec. Thus, real-time and interactive display can be handled easily in this manner. Besides the display board, Tektronix 4010 storage scope terminals are available for graphical data inspection.

For hard copy graphical output, the Versatec and Decwriters are used. The Decwriters are DEC LA36 terminals with the controller boards replaced by Selanar Graphics II boards which provide higher print speed, buffered interfaces, multiple character sets, and point and vector plotting.

Details of the Software System

Motivated by the lack of personnel to undertake a major programming effort, it was decided to make maximum use of tools provided by the VMS operating system on the VAX and the RT-11 operating system on the satellite LSI-11s to facilitate implementation of the multiple tasks required in data acquisition. This approach has facilitated software development, and has generated software which is modular and easy to configure for different experiments. Furthermore, if large buffers are utilized, the overhead involved in going through an operating system (which comes into play during buffer switching) can be minimized.

A discussion of implementation of each of four major parts of a data acquisition software system follows. Then we discuss the use of the network to share these parts in different ways between the VAX and LSI-11 computers, depending upon the experiment. Finally, use of the network to support RT-11 on the satellite computers is described.

1. Data collection.

The primary data collection involves responding to LAMs generated by modules in the CAMAC crates attached to the experimental detectors, gates, registers, counters, etc. with user-programmed interrupt routines. In the simplest cases, these interrupt routines might increment a memory location, read a time or ADC value through the CAMAC interface and store the value obtained in a buffer, and/or output some control values to reset or restart modules in the CAMAC crate. In the most complicated cases, the interrupt routines might read a number of ADCs in the CAMAC crate to determine the path of one or more particles though arrays of detectors, evaluate with kinematic calculations whether the event is of interest, and, in the case of an interesting event, read various CAMAC modules and store the data in a buffer. In either case, an interrupt routine which fills a buffer must end by checking to see if it has finished filling a buffer, and if it has, switch to the next waiting buffer.

Depending on the complexity of the interrupt routine, it would prove best to connect data collection to a satellite LSI-11 or directly to the VAX. The VAX in all cases would service the interrupt faster. However, since both computers have a similar hardware-limited response time to interrupts (3 to 10 microseconds), those experiments with a large number of interrupts should be implemented on the LSI-11 when possible in order to free the VAX for data analysis and for carrying on other computation. Thus, implementing simple interrupt routines which fill buffers on the LSI-11, and complex data analysis of these buffers on the VAX, is the preferred mode of operation.

Careful study of the VMS and RT-11 operating systems shows that both systems support very fast interrupt response when the interrupt handling routines are part of an input-output driver. (A driver is a system program which communicates with hardware and presents a standardized interface to user programs). On both the VAX and the LSI-11, hardware causes the computer to jump (or "vector") to a particular routine in memory when an interrupt occurs. On the VAX 11/750 and on the LSI-11, this vector's address is provided by the interrupting hardware interface. The vector can be caused to point to a particular input-output driver's interrupt routine. Thus, the routine may start executing immediately after hardware vectoring (provided that the interrupt routine handles saving and restoring of any registers which it wants to utilize).

(We note that the widely used VAX 11/780 does not vector directly to an interrupt routine. However, there is a way to circumvent this situation for a single interface. The VAX 11/780 vectors to an address associated with the VAX interrupt priority level (referred to as "IPL"), of which there are 32 (=0,1,...,31). The VAX 11/780 supports several Unibus adaptors, each of which supports four interrupt priority levels, called bus request (or "BR") levels

4, 5, 6, and 7. Thus, on the VAX, the IPL associated vector may not correspond to a unique interrupting interface. Normally, software is invoked which reads VAX hardware registers to determine which Unibus (or other) adaptor caused the interrupt, reads the interrupting Unibus adaptor's hardware registers to determine the address provided by the interrupting interface, and based on these values, jumps to the appropriate interrupt service routine. This situation is simplified if the VAX has only one Unibus. Also, BR7 (the highest bus request level on the Unibus) is the only priority associated with IPL 23 (the highest device-associated interrupt priority level on the VAX). Futhermore, DEC makes no Unibus interfaces which operate at BR7. Therefore, if the CAMAC interface is configured to interrupt at BR7, a CAMAC driver's startup routine can write the starting address of its interrupt routine directly into the System Control Block (the data structure which contains the hardware interrupt routine addresses for each IPL level) of the VAX, at the location corresponding to IPL 23. The address provided by the Unibus interface is simply ignored, or is read within the interrupt routine to determine which LAM caused the interrupt. By this mechanism, interrupt vectoring to an interrupt service routine is accomplished on the 780 in a single hardware step just as on the LSI-11 and VAX 11/750.)

Thus, both VMS and RT-11 allow immediate servicing of interrupts by device driver routines. There remains the problem of how to install interrupt routines into drivers without going through the major project of regenerating the driver for each new experiment. This can be done by passing the address of the interrupt routine and of pointers (such as the addresses at which the driver is to store information about current and next queued buffers) to the device driver in a special function call. This presents no problem in RT-11, but VMS will have to re-map such an interrupt routine into system space so that, on interrupt, the area of memory will be immediately available without having to (a) schedule the process which originally contained the interrupt routine, (b) switch context into that process, and then (c) execute the routine. Various tools, such as real-time system page table entries and the "connect to interrupt" driver, are provided by DEC on the VAX for problems such as this one.

There is no system protection against incorrectly written interrupt service routines installed by special function calls. They will operate at high priority with all privileges (on the VAX) and can do anything, including bringing down the system. However, the routines can be developed and checked out within a process--thereby affording full protection--and installation can be protected by having the driver respond to interrupt routine installation commands only from processes with special privileges. On the LSI-11, such protection is not available. However, since each satellite is generally assigned to a single experiment, bringing down the satellite operating system is not a serious problem.

The data acquistion device driver currently under development will support 32 different interrupt vectors, one for each CAMAC LAM, and eight independent "units" or channels. Each channel can queue buffers to fill and can release filled buffers independently of the other channels. In addition, the driver looks ahead when a new buffer is queued for filling: If it is the only buffer on the queue, filling is initiated and the number of interrupts lost while no buffer was available to fill is written into the buffer. If the buffer is queued to one currently being filled, the new buffer's beginning and end are stored in special locations so that, within 10 to 20 microseconds of completion of filling the previous buffer, filling of the new buffer can begin. In addition, the RT-11 form of the driver implements a trick so that completion routines (routines activated on release by the driver of a filled buffer) know the address of the buffer just filled. These routines can then pass the buffer to the data output driver (either for tape output or for data analysis and tape output).

2. Event recording on tape.

As soon as a buffer is filled with data, it is usually copied to magnetic tape, in case the subsequent real-time analysis contains errors or in case the data is coming in too fast for all of it to be analyzed in real-time. This storing of data is carried out by a simple program which passes every buffer read by the input data acquisition driver to a magnetic tape output driver. The magnetic tape system can be attached either to the VAX Unibus through the Q-bus extension and Quniverter, or to the LSI-11 on its Q-bus. RT-11 has software to support the TM-11 magnetic tape interface. VMS supports only TS-11 Unibus tape drives, but Western Peripherals is supplying a VMS driver to support their tape controller.

3. Real-time event analysis.

In order to monitor the progress of the experiment and to expedite calculating final results, as much (if not all) of the data analysis as possible is carried out during data collection. Such analysis requires passing of filled data buffers to an analysis program, during or after output to magnetic tape. This analysis program carries out the more complex selection of events of interest as discussed under "data collection" above, as well as histogramming of the data into one or more single or multidimensional arrays. Such analysis is implemented as a process under VMS or as a job under RT-11 with full context and protection. In order to make the software modular, a data analysis pseudo-driver will be used to activate and deactivate the analysis program: When data analysis is turned on, the program which passes filled data buffers to the tape output driver is told to pass them instead to the data analysis pseudo-driver. This driver is previously called by the data analysis program, which, after the call, suspends its own execution pending response from the driver. The driver issues the response in the form of the address and length of the buffer available for analysis. On reactivation by this driver response, the data analysis program sends the buffer to the tape driver for output. It also issues another call to the data analysis pseudo-driver in case it needs to be instucted to release the buffer for filling with new data before the analysis is complete, or in case a new buffer is ready for analysis. Then it carries out the analysis of the data in the buffer. On completion, it suspends its own execution pending response from the data analysis pseudo-driver.

4. Experiment monitoring and control.

Separately from the high speed data collection and analysis carried out by the parts of the data acquisition software system described above, a background program allows the experimenter to start and stop the experiment, reconfigure computer controlled parts of the experimental setup, and generate listings, logs, or displays of experimental parameters, data, and data histograms for experiment monitoring. This program must have access to the CAMAC

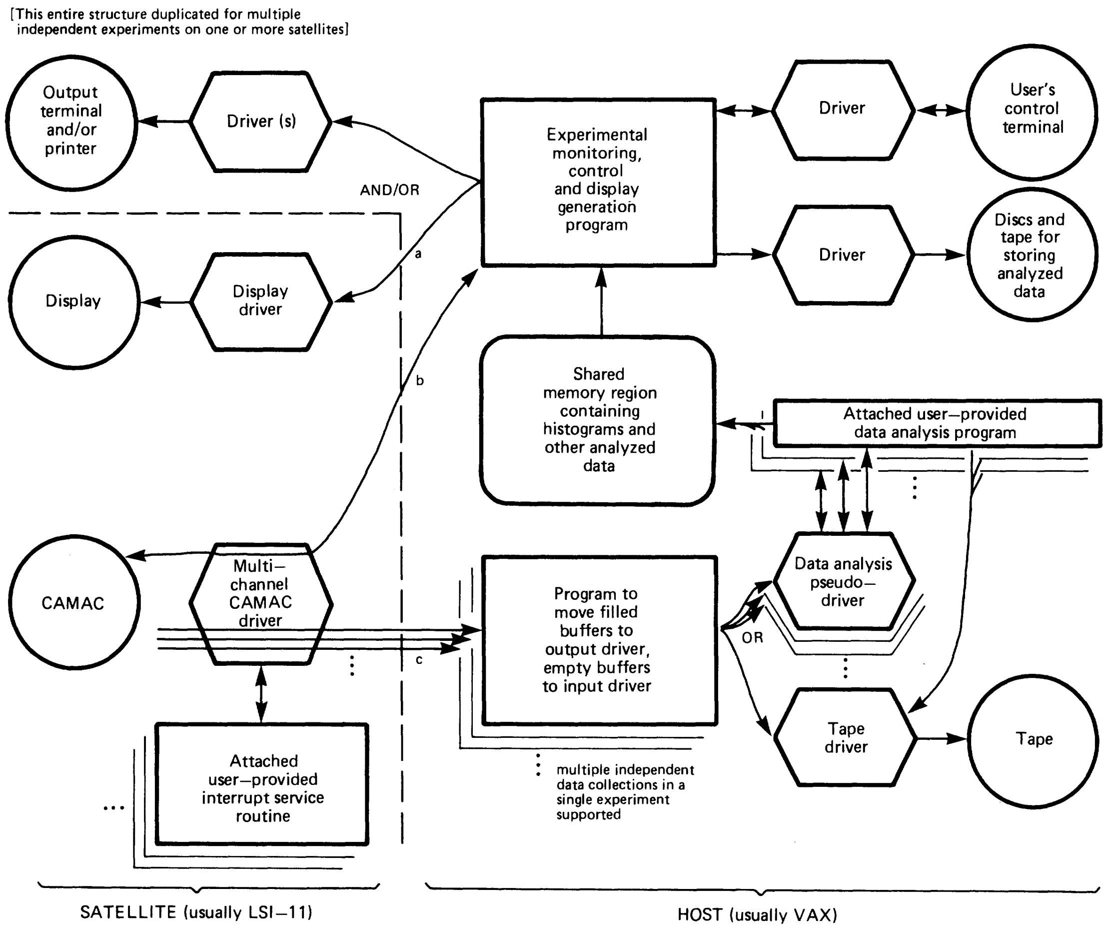

FIGURE 2a Software Structure of the Kellogg Data Acquisition System

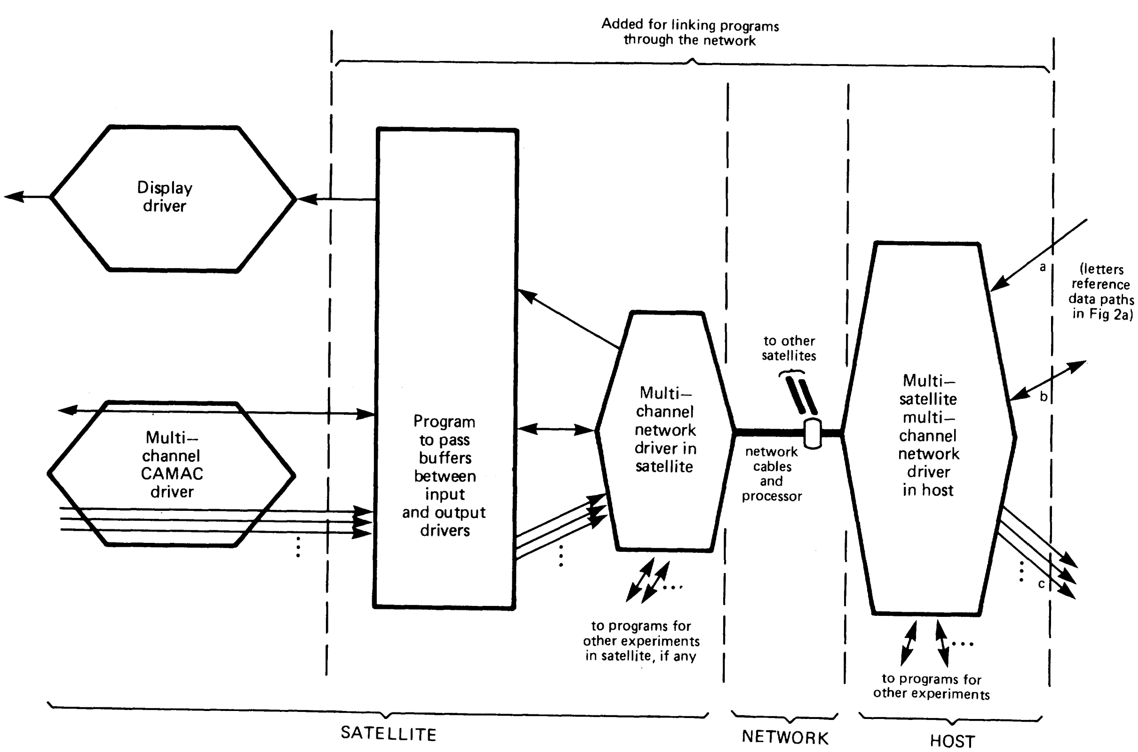

FIGURE 2b Software Structure of the Kellogg Data Acquisition System: Division of tasks between host and satellite by use of the Network

controller for simple FNA transactions (which, for the sake of protection and to avoid memory mapping problems, will be implemented on the VAX as special function calls to the data collection handler). Also, it must have access to tables, histograms, and other data generated by data analysis programs. Such access is direct on the LSI-11 and is easily implemented on the virtual memory VAX under VMS by mapping shared regions of memory into both the data analysis and experiment monitoring and control processes.

5. Network utilization for data acquisition.

The above four parts of a data acquisition system can be implemented entirely on an LSI-11 or on the VAX, or can be shared between both computers (the normal mode of operation). Key to this sharing is the network. From the software point of view, the network is implemented as a standard device driver on the satellites, with one exception. The driver has up to eight independent channels of communication, each of which can queue input - output transactions independently. Thus, data can pass regularly over one channel, while a control channel waits many minutes for the next command. These channels are software constructs, implemented under RT-11 as "units" of the "network device"; the hardware network provides a single connection between the satellite and host. All block transfers over the network are limited in length so that other channels on a given satellite and the drivers on other satellites can communicate between the block transfers.

Figure 2a illustrates the parts of the software data acquisition system described above, and Figure 2b illustrates the modular way in which the network can be inserted so as to divide these parts between the satellite and host. As an example of the division of tasks using the network, consider the following: Initially, a data acquisition program is brought up on an LSI-11. An interrupt routine is installed in the data acquisition driver, a command is issued to the CAMAC crate to start acquisition, and then the data acquisition program passes buffers which are filled by the CAMAC driver to the tape output driver. Subsequently, it becomes desirable to acquire the data on an LSI-11 and store the data on a tape drive connected to the VAX computer. The same software is run except that the output channel is reassigned from the tape driver on the LSI-11 to a channel of the network driver. This channel, at the VAX end, is assigned to the VAX tape output driver. All these assignments are made by simple commands at the terminal, both under RT-11 and VMS. They require no program changes because key parts of the data acquisition system have been implemented as drivers which have a standardized interface to user programs. For real-time data analysis, output on the VAX can be reassigned from the tape driver to the data analysis pseudo-driver. For off-line data analysis, input on the LSI-11 from CAMAC or on the VAX from the network can be reassigned to the tape driver, which is reading buffers from an event tape previously written during acquisition.

The several channels of the network allow transmission of data from several acquistion channels on each satellite to the host, simultaneously with transmission of display arrays from the host to the satellites for DMA point plotting on scopes, and also simultaneously with transmission of control and status information between programs on the host and satellites. In addition, two channels in each satellite are reserved to implement a remote version of the RT-11 operating system. This remote operating system environment is described in the next section.

6. Use of the network to support RT-11 on the satellites.

When it was decided that the satellite software would be developed under an operating system rather than as a stand-alone program, one cost problem was clear. Operating systems and program development generally require disks. Because of the high cost of disks (even flexible disk systems matched the entire LSI-11/CAMAC hardware in expense), the network was configured so that the satellites can reference sections of the host disks as if they are local satellite disks. This network disk system was implemented by (1) providing the satellites with an RT-11 system device driver for their network interfaces, having a software bootstrap compatible with the ROM bootstraps on the network DMA boards, and by (2) running a program on the host which passes input-output requests coming through the network to various host devices and disk files, assigned by satellite number and satellite network driver unit number.

To save space, RT-11 was regenerated so that all shared routines (editors, assemblers, compilers, linkers, utilities, and libraries) are taken by monitor commands from a shared unit different from each satellite's system/user unit. The satellites' system/user units are mapped to individual files on the host disks, and all satellites are given read-only access to another file which contains the shared RT-11 programs.

Two of the satellite LSI-11s have 8 inch double-sided double-density flexible disk systems. An obvious byproduct of the network system is the ability to copy files at high speed between the VAX RK07 disks and the flexible disks. It is anticipated that this will help alleviate the current problem of limited VAX disk space.

Summary and Present Status of the System

In this paper we have described the hardware and software configuration of the new data acquisition system under development in the W.K. Kellogg Radiation Laboratory. The system is built of commercial components, drawing heavily on available software tools. The structure of the resulting system allows flexible hardware and software configuration to accommodate foreseeable experiments. Furthermore, up to six independent data collection stations can be supported. Each station consists of a small number of components: a CAMAC crate, a terminal, and an LSI-11 computer, including processor, memory, CAMAC crate controller, display board, and network interface. Software will accommodate any new station automatically simply by the station making itself known on the network. Furthermore, the VAX will support a large number of off-line data analysis users. The VAX processing speed appears adequate to simultaneously handle anticipated real-time data analysis for both tandem accelerators in the laboratory.

Siting and computer hardware problems delayed bringing the VAX 11/750 on-line until late May, 1981. Development of the network software, probably the most complex part of the system, is nearly complete on three LSI-11s and should be easily transferable to the VAX. The structure of the CAMAC data collection driver has been laid out. Specific interrupt routines, data analysis routines, and experiment monitoring and control programs will be developed for each experiment as required, with standard single and dual parameter acquisition programs being made available for general usage by the end of the summer, 1981.

The network throughput with full RT-11 device driver protocol has been measured to exceed 200KB per second for transferring 10KB buffers. CAMAC throughputs for single parameter data acquistion using an LSI-11/2 front-end processor are calculated to exceed 15KHz, and by using buffered ADCs such as the LeCroy Model 3512, to approach 100KHz. Use of an 11/23 will double or triple these throughputs.

Acknowledgements

The efforts of the Kellogg staff which have made possible rapid initial development of this data acquisition system are gratefully acknowledged. In particular, we wish to thank Alan Rice, Fernand Meulemans, and Al Massey.

Design of the described system at the Kellogg Laboratory would not have been possible without fruitful contacts over the past three years with Peter Parker at Yale University, Richard Kouzes at Princeton University, and David Balamuth at the University of Pennsylvania. In addition to visits to these institutions, one of us (J. Melvin) enjoyed productive visits to the University of Washington, Bell Laboratories in Murray Hill, Lawrence Livermore Laboratory, Stanford Linear Accelerator Center, Los Alamos Scientific Laboratory Van de Graaff, University of Indiana Cyclotron Facility, and Nuclear Data Corporation in Chicago. Hardware and software ideas were exchanged, and in several cases we were provided with tapes of useful software systems.

This project has been primarily supported by National Science Foundation Grant PHY79-23638.