UCRL- 91924
PREPRINT
CONF-841249--4

CONTROL OF A PULSE HEIGHT ANALYZER USING AN RSX WORKSTATION

Samuel Montelongo
David N. Hunt

1984 FALL DECUS U.S. SYMPOSIUM
ANAHEIM, CA.
December 8-14, 1984

December 1984

Lawrence Livermore National Laboratory

CONTROL OF A PULSE HEIGHT ANALYZER USING AN RSX WORKSTATION*

Samuel Montelongo and David N. Hunt
Lawrence Livermore National Laboratory
Livermore, California 94550

UCRL--91924

DE85 007977

ABSTRACT

The Nuclear Chemistry Division of Lawrence Livermore National Laboratory is in the midst of upgrading its radiation counting facilities to automate data acquisition and quality control. This upgrade requires control of a pulse height analyzer (PHA) from an interactive LSI-11/23 workstation running RSX-11M. The PHA is a micro-computer based multichannel analyzer system providing data acquisition, storage, display, manipulation and input/output from up to four independent acquisition interfaces. Control of the analyzer includes reading and writing energy spectra, issuing commands, and servicing device interrupts. The analyzer communicates to the host system over a 9600-baud serial line using the Digital Data Communications Link Level Protocol (DDCMP). We relieved the RSX workstation CPU from the DDCMP overhead by implementing a DEC compatible in-house designed DMA serial line board (the ISL-11) to communicate with the analyzer. An RSX I/O device driver was written to complete the path between the analyzer and the RSX system by providing the link between the communication board and an application task. The I/O driver is written to handle several ISL-11 cards all operating in parallel thus providing support for control of multiple analyzers from a single workstation. The RSX device driver, its design and use by application code controlling the analyzer, and its operating environment will be discussed.

INTRODUCTION

The Nuclear Chemistry Division of Lawrence Livermore National Laboratory is charged with support of Nuclear Test Program diagnostics. Support is also provided to on-going efforts in the Safeguards Program, the Environmental Program, Hazards Control Program, geological research, and basic research in nuclear science. The Nuclear Chemistry Counting Facility (NCCF) is a collection of over 100 experiments performing alpha, beta, x-ray counting and spectroscopy, and magnetic mass-spectroscopy in a variety of configurations using scaler/timers, multichannel scalers, single channel analyzers, and multichannel analyzers.

Currently the Nuclear Chemistry Division is in the midst of an engineering upgrade project to meet changing needs and to take advantage of new technologies. Efforts include automation of data acquisition and quality control, use of state-of-the-art technology, and replacement of unreliable and unmaintainable instrumentation.

In order to support the facility, a network of distributed control (interactive LSI-11/23 workstations running RSX-11M) with a centralized data base (VAX 11/750) for data reduction and storage has been developed and integrated into a flexible and reliable data acquisition network,[1] the NCCF-NET (see Figure 1).

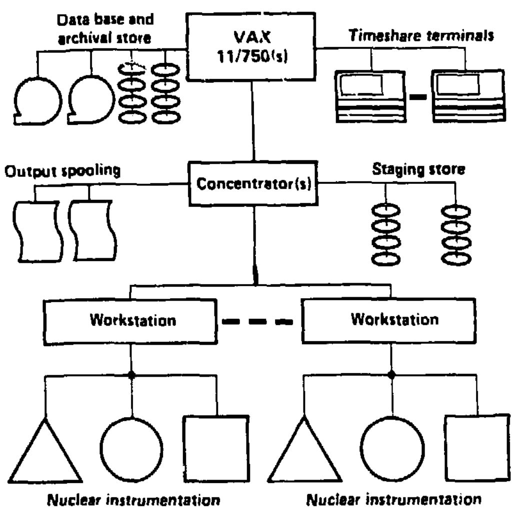

Figure 1. The LLNL Nuclear Chemistry Counting Facility.

*Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract number W-7405-ENG-48.

Through interaction with one of the workstations, users can control data acquisition and experiment scheduling on all attached nuclear instrumentation. Sample changers used to position samples in front of detectors are controlled by CAMAC hardware interfaced with the workstation. CAMAC design and implementation will not be discussed in this paper. Control of multiple pulse height analyzers (PHA) from a single workstation is acheived through the use of an in-house designed serial line communication card. The serial line card is interfaced with an I/O device driver to complete the path between the serial line card and application tasks implemented on the workstation.

RSX WORKSTATIONS

The RSX workstation is an interactive device through which scientists enter counting parameters that control experiment scheduling, data acquisition, and data analysis. Each workstation maintains its own schedule and count execution parameters.

The workstation (see Figure 2) consists of an LSI-11/23 processor with 22-bit memory address support running under the RSX-11M V4.1 operating system. The processor has 512 Kbytes of task memory to preclude foreseeable memory contention problems.

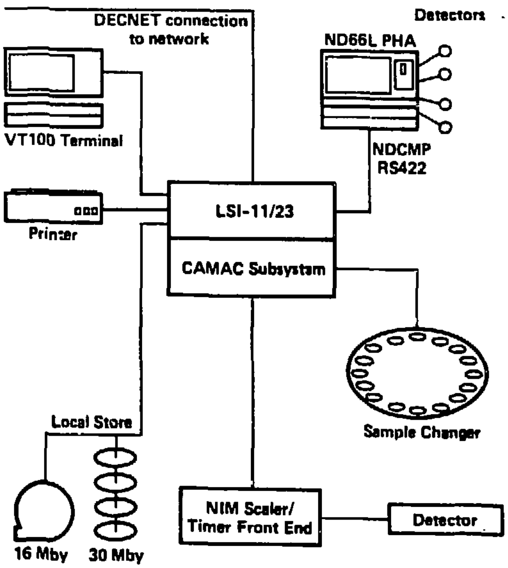

Figure 2. Block diagram of a typical workstation and its attached instrumentation.

There are four communications lines, 2 local terminal lines for user interaction, and 2 DECNET lines which provide both a link to the central node (VAX 11/750) and a path for remote user interaction. 30 Mbytes of mass storage for local data manipulation and storage are provided by a DSD 890 which also includes a cartridge tape drive for data transfer/backup. A small desk top printer provides user output and diagnostics. The terminal used is a standard DEC VT-100 with advanced video options.

Also included with the workstation is a CAMAC subsystem for manipulation of CAMAC modules attached to the workstation. The amount of instrumentation controlled by the workstation is limited by the amount of backplane space in the LSI-11 chassis. Currently we are configured for 3 PHAs and one CAMAC crate. The three PHAs allow control of up to 12 detectors (4 detectors/PHA). The CAMAC crate supplies not only interfacing to all instrumentation other than the PHAs, but also includes modules which provide for communication and control of remote CAMAC crates.

When executing a count, workstations move a sample in front of the detector, set up the PHA with a series of low level software commands and wait for count completion. Upon count completion, PHA data is dumped to the workstation and forwarded to the VAX by application code specifically designed for that purpose. It is the function of the workstation to collect data for use by the central computer's data base. The data is stored, then sent to the VAX for reduction, analysis, and report generation. Virtually every aspect of the pulse height analyzer is controlled from the workstation.

PULSE HEIGHT ANALYZER

The PHA chosen was the Nuclear Data ND66L. It is a microcomputer-based, multichannel analyzer system with the capability of being operated both as a stand-alone analyzer and a remote computer terminal.[5] The ND66L provides us with complete data acquisition, storage, display, manipulation, and input/output capability and flexibility. It supports acquisition from up to four independent analog to digital converters for simultaneous acquisition in three modes: Pulse height analysis, multichannel scaling (MCS), and sequential listing (LIST). Data is selectively acquired in separate memory groups ranging in size from 256 to 8192 channels to permit acquisition and storage of multiple spectra. In addition to acquisition and display of spectra, the analyzer can intensify, expand and overlay portions of the spectra, and provide quantitative values of interest.

Using the Remote Terminal and Communications Firmware provided as an option with the ND66L, the ND66L can be operated as a remote acquisition system under control of an external host computer.[5] The RSX workstation previously described serves as the host computer remotely controlling the ND66L. External control of the ND66L system requires generation of a program which uses commands in the same manner as an operator uses the keyboard on the ND66L. All user-initiated keyboard operations can also be requested by the workstation. In addition, spectra can be either dumped or loaded over this interface. This allows the RSX workstation to set up acquisition parameters, start and stop acquisition, perform data manipulation and send or receive data. ND66L Communications Firmware uses Digital Data Communications Message Protocol (DDCMP)[7] to facilitate message transmission between computer systems over a dedicated asynchronous serial line unit with data transfer rates up to 38.4 Kbaud. On top of DDCMP, Nuclear Data has added NDCMP,[5] an application specific transport layer which completes the necessary robust communications environment.

A software package was shipped with the ND66L which provides for communication between an RT-11 system and the PHA via a standard serial line card. This package implemented both the DDCMP protocol and the NDCMP transport layer. The initial design of the Nuclear Chemistry Upgrade Project fit well with the delivered communication package as the design called for an RT-11 system to act as a data and control channel between the workstation and the PHA. Later design changes required direct communication between the ND66L and a workstation running under the RSX-11M operating system. Since no software package existed which provided for communication between the PHA and workstation we were forced to look for a viable solution. An in-house designed serial line card, the Intelligent Serial Line (ISL-11) for LSI-11 processors proved to be the answer to our problem. Communication between the workstation and PHAs take place via the ISL-11.

ISL-11 COMMUNICATION CARD

The Intelligent Serial Line card (ISL-11) provides Q-bus compatibility and DMA capabilities to the workstation with a standard interface to the PHA (see Figure 3). The ISL-11 operates at 9600-baud using the DDCMP protocol thus relieving the workstation CPU from the DDCMP overhead. Messages passing through the ISL-11 are transferred to/from the LSI-11 memory via the ISL DMA interface.

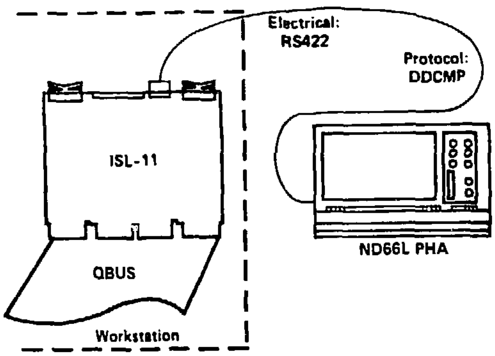

Figure 3. The ISL-11 interfaces the ND66L to the RSX workstation.

DDCMPK,[2] the version of DDCMP implemented on the ISL-11, performs the functions necessary to complete these transfers. The handshaking used to synchronize DDCMPK and the ISL driver is specified by the ISL driver documentation. The description of DDCMPK and the handshaking involved is beyond the scope of this paper. The in-house designed ISL-11 card is a general purpose communication card. While originally developed for a communications multiplexor and programmed for the SCULL (Serial Communication Utility for Long Links) Protocol it was adaptable for our job and DDCMP.[3] Our project design team designed and implemented the DDCMP protocol on the card in firmware to ensure reliable and compatible communications between the workstation and the PHA.[2]

The ISL-11 communication card (see Figure 4) is a programmable serial line card with an on-board microprocessor. There are three major components of the card: the microprocessor including program and data memory; a differential (RS-422) serial line interface; and a DMA (Direct Memory Access) interface to the LSI-11. Address and vector assignments are strappable and the board supports both 18-bit addressing and four-level interrupt arbitration. There are four independent interrupt bits, each with its own interrupt vector and strappable to QBUS interrupt levels. Included on the card is a diagnostic connector for analysis of the on-board microprocessor operation. This card is packaged as a standard Digital quad-height, four-layer, printed circuit board and is compatible with LSI-11/2s and LSI-11/23s. The board contains 87 integrated circuits and is fully socketed.

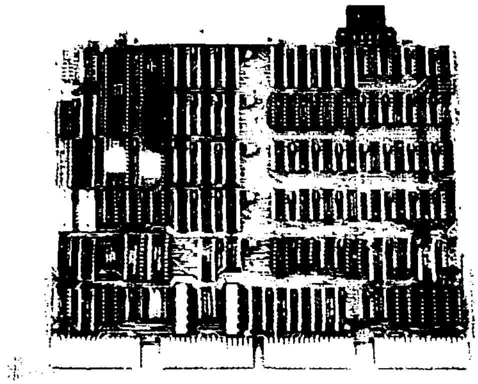

Figure 4. ISL-11 printed circuit board.

The microprocessor controlling the board (see Figure 5) is the Intel 8039, a single-chip, eight bit processor with 128 bytes of RAM and 27 input/output lines. An interval timer is included on the chip and can be used for programmable time-outs in conjunction with the 9.216 MHz processor clock. The processor has separate program and data space with 6K words of EPROM available for the program and 1K words of RAM for data storage.

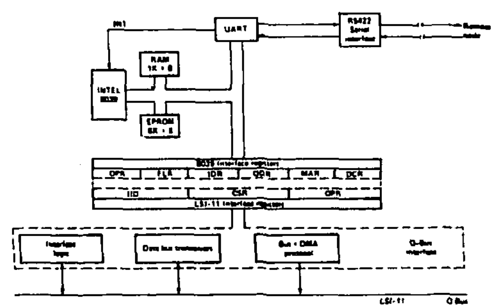

Figure 5. Block diagram of the Intelligent Serial Line (ISL-11).

Peripheral chips to the processor are mapped into the processor's data space.

The serial line interface is based around a UART (universal asynchronous receiver/transmitter) built by Western Digital, the WD8250 Asynchronous Communication Element. All characteristics such as baud rate, word length and parity detection are programmable via the processor. Using a 3.072 MHz clock rate (1/3 of the processor rate), the UART is capable of all standard baud rates up to 56K baud. The UART drives an EIA RS422 compatible serial line. This interface includes line protection circuitry and will drive lines up to 2 miles in length.

The LSI-11 interface is comprised of two parts: two memory-mapped I/O registers and a DMA interface. The registers include a control-status register, used for initialization and communication handshaking, and a parameter register used for parameter loading upon initialization. The DMA interface is capable of word or byte transfers, to or from the LSI-11 memory, without the aid of the processor. This interface is capable of single-cycle mode DMAs only.

The major role of the ISL-11 is to supply a reliable I/O communication interface between the PHA and the workstation, thus relieving the workstation CPU from the overhead involved with DDCMP. This also insulates the upper level software from the details of DDCMP on this device. Although an RSX I/O device driver software package existed for the ISL-11 device, it proved to have problems when transferring very large amounts of data from the PHA to LSI memory. This problem was intolerable, therefore, the ISL driver was rewritten to strictly adhere to RSX-11M programming conventions in order to ensure the integrity of the RSX Executive.

ISL I/O DEVICE DRIVER

Workstation application code sends messages to the PHA through the ISL-11 via RSX QIO system calls to the ISL device driver. The ISL I/O device driver services the needs of the ISL-11 communication card within the RSX workstation. The ISL driver receives and services I/O interrupts from the ISL-11, initiates I/O operations on the ISL when requested to do so by the RSX Executive, cancels in-progress I/O operations, and performs initialization of the device and the device database upon loading and power-up.

Only QIO calls designed to initiate a send or receive transmission on the ISL-11 are queued by the ISL driver. All other QIO calls are processed immediately. QIO calls requesting a receive function contain buffers to be filled by the ISL card. This QIO call may specify to the ISL driver an AST address or an event flag specific to the PHA in question. When the receive buffer has been filled, the ISL card causes an interrupt to happen, the driver then retrieves the data and causes the specified AST or event flag to occur. In our code, as soon as the owning task recognizes the AST, it posts another buffer to the ISL driver while still in the AST state. This ensures there is always a receive buffer posted to each ISL. Queuing buffers during the AST state has proven to be a reliable and effective means for synchronizing communications between a PHA and application code controlling that PHA.

The ISL device driver requires 2.6K bytes minimum of memory.[4] It is written to run on an LSI-11/23 processor running V3.2 or higher of RSX-11M. The language used is DEC MACRO-11 assembly language. The ISL driver is the I/O driver specifically written for the LLNL designed ISL-11 asynchronous communications card for LSI-11 processors. It is a user-written loadable driver thus making it easier to incorporate into a system. There are two main sections to the driver: the driver code which controls interaction of the ISL-11 with the CPU, and the driver database which contains data structures defining the configuration of the ISL-11s to the RSX Executive.

As an integral part of the RSX Executive, the ISL driver possesses its own context, allows and disallows interrupts, and synchronizes access to its shared data base with that of other RSX Executive processes.[6] This last detail, synchronizing access to a shared data base, proved to be one major problem with the initial version of the ISL driver. Design of the I/O data structures (i.e. DCB, UCB, SCB) for the ISL driver allows for control of multiple devices from the ISL device driver. By synchronizing access to the ISL driver data base, the ISL driver can handle several ISL-11 devices all operating in parallel thus allowing control of multiple PHAs from a single workstation (see Figure 6).

The ISL driver includes operating parameters which must be tailored to the specific environment. Hardware parameters define the device address and interrupt vectors. Software parameters define the UART characteristics. The driver code contains parameters defining the UART characteristics and the device addresses of the first ISL-11 device in the computer. The UART parameters define the characteristics for all ISL devices using the driver code. Parity, stop bits, bits/character and speed must be mutually agreed upon by the communicating parties. The device address register set is only for the first ISL device.

The ISL driver and ISL-11 card are specifically designed for 18-bit RSX systems but they can be used in 22-bit systems if they are loaded within the lowest 256 Kbyte of memory (18-bit addressable memory). Any application code interfacing to the ISL driver must also reside within the lowest 256 Kbyte of memory. Multiple ISL-11 devices on a single workstation are currently constrained to identical UART settings. The ISL-11 has four interrupt vectors: two for send transmissions and two for receive transmissions. As RSX limits the number of interrupts per device to two, a single ISL must be treated as two units. Receive transactions occur on even numbered units, and send transactions occur on odd numbered units.

ISL device driver capabilities include support for multiple ISL-11 devices, support of access to a single ISL-11 device by multiple tasks, transmission rates up to 19.2K baud, and a flexible user interface. This user interface allows loading of the driver when needed, automatic initialization of the driver and device, capture of the device by a single user task, multiple requests from a single task to be outstanding, and cancellation of outstanding requests. In addition, the user interface provides success/failure information on transmission, and monitoring information on transmit/receive completion.

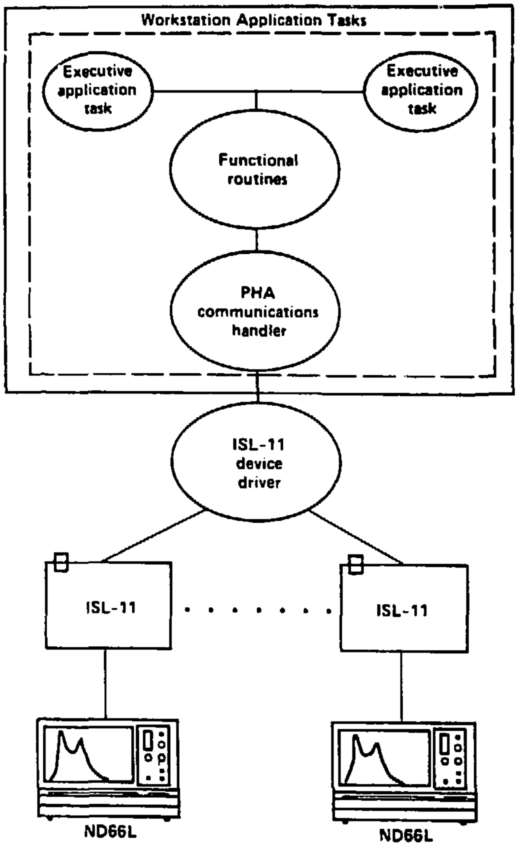

Figure 6. A PHA Communications Handler (PHACH) interfaces the ISL device driver to workstation application code.

The driver interface contains the four usual entry points required by a driver: I/O initiator, device time-out, cancel I/O, and power failure.

I/O initiation entry occurs in response to a direct QIO call to the driver from an application task. All phases of the communication process between the analyzer and the workstation can be moderated through this mechanism. QIO functions defined for the ISL driver provide for device initialization and termination, transmission initialization and cancellation, and statistics monitoring. The functions supported by the ISL driver are discussed below.

Function	Purpose
IO.INL	Initializes the ISL-11 device and the ISL driver. The currently active I/O transmissions are aborted. Queued but not yet active I/O requests are not cancelled. Monitoring statistics are zeroed. On completion, if there were no queued requests, the device is operationally in an idle mode awaiting a transmission request. Otherwise, the driver commences processing of the queued requests. This call must be made on both channels whenever used. The receiver channel must be the first call, the sender channel must be the second call.
IO.TRM	Terminates all activity on the ISL-11 device. The currently active I/O transmissions are aborted. Queued but not yet active I/O requests are not cancelled, but neither will they be processed until function IO.INL is called.
IO.KIL	Cancels outstanding I/O transmissions. Both queued and active requests will be deleted.
IO.ATT	Attaches the ISL device to a single task. If the device is already attached, the request is queued by the monitor until the device becomes available.
IO.DET	Detaches the ISL device from a task.
IO.WLB	Sends a transmission over the ISL-11 link.
IO.RLB	Posts a buffer to be filled upon completion of a receive transmission from the ISL-11.
IO.SEC	Returns monitoring statistics to the application program.

Entry to the Cancel I/O code occurs in response to a request to kill the active transmission. Two techniques gain entry: the issuance by an application task of a QIO call with function IO.KIL or an MCR ABORT command. The current transmission is aborted if the requesting task owns the current I/O transmission. Queued but not yet active I/O transmissions are also aborted. Cancel I/O entry will occur only if an active I/O is in progress. The QIO call with function IO.KIL is used by an application task to cancel outstanding transmission requests for a particular unit. Since the ISL device is logically divided into two units, a call to each unit is required to totally eliminate all outstanding requests for the ISL channel.

Device time-out entry occurs in response to a time-out on a send transmission from the workstation to the analyzer. When a send operation is initiated, a time-out count is established. If an IO.WLB function fails to execute within the specified time interval, the RSX Executive notes the time lapse and calls the driver at this entry point. I/O processing is completed for the faulty I/O packet transmission and a status code is returned in the status buffer.

Entry to the power failure code occurs automatically when the driver is loaded with the MCR LOAD command or when a power failure occurs. All devices defined in the driver database are initialized. The driver internal state is initialized and statistical information is zeroed. The device is operationally in an idle mode awaiting a transmission request. This entry point can also be entered via the QIO call IO.INL specifically designed to initialize the ISL-11 and the I/O driver.

The ISL device interrupt routines are entered by a hardware interrupt from the analyzer or when a communication sequence between the ND66L and the workstation (previously initiated by the driver) completes its I/O operation.

The workstation interacts with the ND66L using application code designed to interface directly to the ISL driver.

APPLICATION CODE

Our application code consists of intercommunicating tasks which interact with scientists, maintain counting schedules, execute counts, and gather and forward data to a VAX for archiving and data reduction (see Figure 6). Interaction of these tasks is driven by an Executive Application Task (EAT), an application task implemented by the NCCF design team. An EAT is written to suit the needs of a specific user of the NCCF-Net. The EAT controls every aspect of the workstation. It includes code to define device-specific parameters, set up experiments, dump spectra, and process software interrupts. It also controls quality assurance testing of data, data archiving, data storage, and system diagnostics. Multiple EATs are envisioned to exist on a workstation, thus allowing control of the NCCF-Net by several users according to the design of the EAT. All code which users may wish to modify at some future time is written in FORTRAN 77 to facilitate easier understanding/modification by the user. FORTRAN 77 is the language most widely understood by users of the NCCF-Net.

The EAT interacts with the ISL driver through the PHA Communications Handler task (PHACH). The EAT incorporates functional routines which handle specific chores for a PHA. Functional routines issue device-specific commands to the PHACH which in turn formats ND66L commands to be issued to the PHA. The PHACH controls all interaction between the ISL driver and the workstation application code. It buffers all I/O to and from the driver. This includes commands, data, spectra read into or out of the analyzer, and PHA hardware interrupts fielded by the ISL driver. Hardware interrupts are returned as software interrupts to the EAT.

The PHACH interfaces to the ISL driver through the use of RSX QIO system calls. The QIO directive forces all I/O requests from a user task to go through the RSX System Executive. The RSX Executive works to prevent tasks from destructively interfering with each other and with the RSX Executive itself. QIO calls interact with the driver according to design specifications of the driver. The PHACH acts as a traffic cop between the ISL driver and application code wishing to communicate to a PHA. The PHACH funnels commands into the ISL driver and then acts as a fan-out when returning data from a PHA to a specific application task.

The maximum data length of any I/O buffer transmitted between the ISL driver and the PHACH is 512 bytes. This adheres to the prescribed format of serial line communications from the ND66L. When processing I/O from the ND66L, the ISL driver strips off the DDCMP protocol and returns to the PHACH only the ND66L message overlayed by the NDCMP transport protocol. The PHACH uses the NDCMP protocol for message typing and data control. The PHACH then proceeds to parse the ND66L message and formulate a response to be returned either to upper level application tasks or the ND66L depending upon the message type. A storage buffer for messages received from the ND66L is always posted to the ISL-11 card by the PHACH. All ISL-11 I/O received by the ISL driver is checked for validity and processed according to its data type.

The PHACH is written to process only one functional command at a time. Functional commands come from upper level application tasks. These commands are designed to implement ND66L functions available to users of the ND66L. Since only one PHA command is processed at any one time by the PHACH, the PHACH is designed to queue all commands received while a command is being processed.

All tasks which must communicate with other tasks use the RSX directives RCST$ and SDAT$ to send and receive messages. These directives allow tasks to take advantage of the I/O queuing mechanism supported by RSX. This queuing mechanism allows multiple messages to be outstanding to a particular task thus preventing loss of messages. This task-to-task communication allows our application software to process task messages as they are pulled from the queue for a particular task. This queuing mechanism also allows application software to queue a message and continue with the task at hand. All tasks, when not processing a message, are in a dormant state ready to become active upon reception of a task communication.

Interrupt support is made available between the ISL driver and the PHACH through the use of asynchronous system trap (AST) handling provided by the RSX operating system. Since AST support is only available using the macro interfaces provided by DEC, we designed and implemented FORTRAN callable routines which made use of the macro AST support routines. This method is currently being implemented with various application tasks including the PHACH which relies heavily on AST support. This support includes time-outs on command processing, time-outs on message transmission between tasks and I/O drivers, unsolicited software interrupts from the ISL driver, and message synchronization between the ISL driver and the PHACH.

The PHACH incorporates functional commands to establish a logical software link cross-reference table between the workstation and the PHA devices. This is essentially a table linking ISL-11 devices to Executive Application Tasks. This is a very important feature of the PHACH which allows an EAT to control a specific acquisition interface on the ND66L. The table also serves as a cross-reference table for the PHACH in returning ND66L service request messages from a specific acquisition interface on the analyzer to the appropriate EAT.

CONCLUSION

A prototype workstation attached to an ND66L pulse height analyzer has been implemented and has been shown to meet all criteria of the Nuclear Chemistry Upgrade Project. The ISL-11 communication card and its associated device driver have been operating on the workstation in a most reliable manner. The design and implementation of the ISL-11 driver has met the needs of the Nuclear Chemistry Counting Facility. The design of the workstation, its control of attached instrumentation, its application software, the ISL-11 and its device driver, and the implementation of the workstation with the NCCF-Net are proving to be a system which is reliable, maintainable, and usable.

REFERENCES

1. D. W. O'Brien, S. L. Robinson, and D. N. Hunt, "The LLNL Nuclear Chemistry Automated Counting Facilities: Design and Implementation", IEEE Trans. Nucl. Sci., NS-28, No. 1 (1981).

2. D. N. Hunt, G. C. Kessler, "The DDCMP Protocol on a DMA Serial Line Board", Proceedings of the Digital Equipment User's Society, Anaheim, CA, December 1982.

3. R. M. Bryant, "User's Guide to the Intelligent Serial Line Card (ISL-11) for LSI-11 Systems", Lawrence Livermore National Laboratory, LEA 79-1438-76D.

4. R. Allsman, S. Montelongo, "User's Manual for the ISL-11 Device Driver for RSX-11M/M+ Systems", Lawrence Livermore National Laboratory, Internal Report, 1984.

5. Nuclear Data, "Analyzer/Remote Terminal Operator's Instruction Manual", 1981.

6. Digital Equipment Corp., "Guide to Writing an I/O Driver", Order No. AA-2600-TC, 1981.

7. Digital Equipment Corp., "DDCMP Specification V4.0", Order No. AA-D599A-TC, 1978.

DISCLAIMER

DISCLAIMER