presented at:
1981 Nuclear Science Symp.,
21-23 October 1981, San Francisco

BNL 30337

BNL--30337

DE82 005236

A DATA ACQUISITION AND EXPERIMENT CONTROL
SYSTEM FOR HIGH-DATA-RATE EXPERIMENTS AT THE
NATIONAL SYNCHROTRON LIGHT SOURCE*

J. L. Alberi and F. W. Stubblefield

Brookhaven National Laboratory
Upton, New York 11973

November, 1981

MASTER

*This research was supported by the U. S. Department of Energy:
Contract No. DE-AC02-76CH00016.

A DATA ACQUISITION AND EXPERIMENT CONTROL
SYSTEM FOR HIGH-DATA-RATE EXPERIMENTS AT THE
NATIONAL SYNCHROTRON LIGHT SOURCE*

J. L. Alberi and F. W. Stubblefield

Brookhaven National Laboratory
Upton, New York 11973

ABSTRACT

A data acquisition and experiment control system for experiments at the Biology Small-Angle X-ray Scattering Station at the National Synchrotron Light Source has been developed based on a multiprocessor, functionally distributed architecture. The system controls an x-ray monochromator and spectrometer and acquires data from any one of three position-sensitive x-ray detectors. The average data rate from the position-sensitive detector is $\sim 10^6$ events/sec. Data is stored in a one megaword histogramming memory. The experiments at this Station require that x-ray diffraction patterns be correlated with timed stimuli at the sample. Therefore, depending on which detector is in use, up to 10^3 time-correlated diffraction patterns may be held in the system memory simultaneously. The operation of the system is functionally distributed over four processors communicating via a multiport memory.

INTRODUCTION

Small-angle x-ray scattering experiments have been used to investigate relatively large order structures in biological systems.¹⁻³ The structure in these systems is measured by a diffraction pattern generated when the incident x-rays scatter from the sample. These experiments seek to correlate changes in the biological structure, measured by change in the associated diffraction pattern, with various external stimuli applied to the sample. Whether the stimulus is electrical, chemical or mechanical, the objective is to record a series of time-resolved diffraction patterns correlated with the applied stimulus. Auxiliary information, including monochromator and spectrometer settings, is also recorded. However, this information is constant over a series of time-resolved diffraction patterns. This paper describes the design and partial implementation of a data acquisition and experiment control system capable of satisfying the requirements of the time-resolved diffraction experiments outlined above. The system is being developed for the Biology Small-Angle X-ray Scattering Station at the National Synchrotron Light Source.

The necessity for a complex, noncommercial data acquisition system stems from the data flow and storage requirements set by the nature of the time-resolved experiments planned for the biology beamline. The detector of the x-ray diffraction patterns yields high data collection rates and requires large memory arrays to store the collected data. X-ray diffraction data are taken on position-sensitive detectors, three of which are planned for use with the data acquisition system. The first is a one-dimensional detector with a global position readout of 1024 channels capable of digitizing an event every microsecond. The second detector is also one-dimensional with an amplifier, discriminator, and scaler chain on each of ~ 100 anode elements. This detector has the capacity to accumulate data in each scaler during data acquisition at $\sim 10^6$ events/sec/channel. The third detector is a two-dimensional detector with approximately 500 channels in each

*This research was supported by the U. S. Department of Energy: Contract No. DE-AC02-76CH00016.

DISCLAIMER

This book was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

dimension, capable of digitizing an event every 200 nsec. These three detectors, which are now under development, are interchangeable in the experiment and are not used to collect data simultaneously. The number of time-resolved diffraction patterns, hereinafter called time slices, is inversely proportional to the amount of memory needed to store a single diffraction pattern and associated data, and directly proportional to the amount of data memory in the acquisition system. The performance objectives that the data collection system must meet to be useful to the experimenter are summarized below.

PERFORMANCE

The following are the design expectations of the data acquisition system.

1. The derandomized data-rate capacity is $\geq 10^6$ events stored/sec.
2. The dead time for data entry into the derandomizing buffer is negligible compared to the digitization time of the position-sensitive detector readout.
3. The data memory capacity is 10^6 16-bit words with provision for expansion included. The user may select the number of time slices.
4. The input port to the data acquisition system accepts data from any one of three position-sensitive detectors described above.
5. The user programs the number and length of time slices, and the number and time sequence of stimuli to the sample during a run. (A "run" consists of a user-selectable number of passes through a specified sequence of time slices and sample stimuli.)
6. The system controls automatically the incident beam monochromator and the x-ray spectrometer on which the detector and sample are mounted.
7. The measurement sequence is user-programmable in FORTRAN. Therefore, program development is done under a commercially available operating system. All data collection functions are accessible to the user via FORTRAN-callable subroutines. The user-written program specifying the measurement sequence runs under the operating system with full access to the file management provided by the operating system.

Details of the architecture and implementation of the system are presented below.

ARCHITECTURE

General Concepts

In the present system a commercial operating system and a minicomputer with a more or less standard set of peripherals form the control element for the data acquisition and experiment control process. The minicomputer used is a Digital Equipment Corporation (DEC) LSI-11/23 and the operating system is DEC RSX-11M. There are several advantages inherent in this choice:

- (a) The LSI-11/23 is rapidly becoming a de facto standard minicomputer for medium-size data-acquisition systems acquiring data from scientific experiments. Thus it is reasonable to expect that LSI-11 bus-compatible peripherals and memories will continue to be developed and will become even more readily available.
- (b) The LSI-11/23 has a fully implemented method of memory management and protection compatible with the RSX-11M operating system. Thus a data acquisition system employing this minicomputer can be protected against errors caused by user-generated experiment control code.
- (c) Because the RSX-11M operating system is the primary commercial operating system for several popular minicomputers, it is likely a large effort will be made to maintain and upgrade it in the next few years.

Balancing these advantages are a set of disadvantages that preclude use of this processor and operating system as the sole control element in the data acquisition system:

- (a) The relatively large and complex operating system cannot respond rapidly enough to real-time events to be useful in generating time-slice control data.
- (b) The LSI-11/23 has a maximum memory size, using the current version of RSX-11M, of 128K words, but the anticipated data array size is 1M words.
- (c) The direct-memory-access time is much greater than the 0.5 μ sec necessary to insure a data acquisition rate of 10^6 events/sec.
- (d) Use of a single computer (CPU) in a data acquisition system implies that only one instruction sequence (program) can be executed by the CPU at any one time during the data acquisition process. Also, this one program can extend over only one logical address space. (The LSI-11/23 is limited, during an instruction cycle, to a logical address space of 64K bytes.)

Larger computer systems, such as the new lines of 32 bit minicomputers, are perhaps fast enough and contain enough memory, but they are expensive and have more computational ability than is necessary for a data acquisition system.

To overcome the disadvantages described above without undertaking extensive modifications of hardware and software in the LSI-11/23, three LSI-11/2 processors with

necessary peripherals, termed subfunction processors, are provided to carry out real-time data collection subfunctions for the data acquisition system. The following subfunctions have been isolated in this manner:

- (1) time-slice and sample-stimulus generation
- (2) real-time data display
- (3) input/output operations management.

Each of these functions is confined to an individual processor with a minimal set of support components. Additional components and peripheral devices are added as required to perform the specific function assigned. The functions assigned to the three subfunction processors are discussed in more detail below.

Along with the subfunction processors is a histogramming memory with a fast read/increment/write processor, all on a separate data bus segment called the MBUS. This subsystem is capable of storing detector data at 10^6 events/sec in noninterleaved operation and 2×10^6 events/sec when the addresses are interleaved by a factor of two.

Interprocessor Communication

Communication among the four LSI-11 family processors is carried out through a commonly accessed memory array called the multiport memory subsystem.^{4,5} Figure 1 shows a block diagram of several processors interconnected via the multiport memory. Each processor, either an LSI-11 (QBUS) or PDP-11 (UNIBUS), has an access port through which it reads from or writes to the multiport memory. The access ports and memory share a bus, called the MBUS, with 26 bits of address space that is an extension of the DEC UNIBUS. Standard UNIBUS memory with some extra address decoding is used in the subsystem. The port connects on the processor side to the standard processor bus (QBUS or UNIBUS). Each port, which is constructed on one DEC quad printed circuit module, has a 4K word window in processor physical address space. Access to the multiport memory address space from processor address space is routed through the window by a mapping register contained in the processor I/O space. In this way all 26 bits of multiport memory space are available to the local processor. Contention for access to the multiport memory is resolved by a round-robin scheduling module labeled SCANNER in Fig. 1. The scanner asserts a PORT SELECT signal to each access port in turn for a duration of 100 nsec if no memory request is pending. If a request is pending, the port asserts the signal, SCAN STOP, to prevent the scanner from advancing. The scanner is released after the memory cycle and continues on to process memory requests from other access ports. Memory cycle time-outs are processed by the scanner module in such a way that no time-out in any processor can cause another processor to time-out.

A block diagram of the overall system architecture is shown in Fig. 2. The MBUS is shown at the left interconnecting four processors, the histogramming memory and fast read/increment/write processor, and interprocessor communication memory. Each of the group of four processors, with access to the multiport memory through an access port, is serviced by the multiport-memory scanner module as described above. The four processors are at different functional levels depending on their role in the system operation. A functional level is defined by the master-slave relationship of one processor to another. Each processor is defined as either a master, slave, or unconnected with respect to any other processor. These functional relationships are implemented by protocols for processor operation on master and

Fig. 1. General interconnection method of a multiprocessor system by the multiport memory subsystem.

slave parameter blocks and data buffers set up in the interprocessor communication portion of the multiport memory.

Experiment Control Processor

The highest level processor, called the experiment control processor, is shown on the bottom of Fig. 2. The user writes experiment control code in FORTRAN for this processor. Of the four minicomputer central processing units in the system, it is the one that is used most like a standard computer system. It contains a standard array of peripherals on an LSI-11 bus, including (1) a 9-track magnetic tape drive for data storage and system backup, (2) a 10 M-byte disk which serves as the RSX-11M system device, (3) a dual floppy disk used for transferring programs developed off-line to the system disk, and (4) a spectrometer motor controller. This last device, which has been described previously, can control up to 15 stepping motor-encoder pairs.⁶ Requests for real-time data collection services are implemented via FORTRAN-callable subroutines that place the parameters of the requests in parameter blocks in the interprocessor communication portion of the multiport memory.

Time Slice Processor

The time slice processor, which directly controls the duration and sequencing of time slices within the system, is a slave of the experiment control processor. It conditions the data flow from the position-sensitive detector by adding time-slice correlated offsets to the data to be put in the histogramming memory. The time-slice processor also controls generation of external

stimuli to the sample. The stimuli include pulses of 1 μ sec duration used to drive auxiliary apparatus and excitation voltages that may be supplied directly to the sample. There are 8 channels of each of these stimuli. Data taken from the experiment include the position-sensitive detector information as well as 8 channels of independently gated scalars. The scalars may be correlated individually with any combination of time slices or may be used to count overall incident beam flux. Time-integrated voltage measurements are taken by connecting the scalars to voltage-to-frequency converters. Details of the implementation of the time-slice interface are presented below.

Data Display Processor

The formatting and graphical display of multi-dimensional data is a task sufficiently complex to justify the assignment of an individual processor, called the data display processor, to this function. The display function operates in two modes: (1) the display may be controlled by the experiment control processor to which it is assigned as a slave processor, or (2) the display operates in stand-alone mode on data in the histogramming memory by responding to a set of operator-initiated keyboard display commands.

Two different types of display device are incorporated into the display process in the data acquisition system. The first is commercially available and consists of a DEC VT-100 video terminal modified for raster-scan graphics with a Selanar Graphics-100 add-on unit. The display terminal connects to the display processor with a 9600 baud, RS232-C serial data line. The command language of the modified graphics terminal

Fig. 2. Overall architecture of the data acquisition and experiment control system. Three types of bus segments are used to implement the system:

- (1) the LSI-11 bus,
- (2) the multiport memory bus,
- (3) the histogramming memory bus.

emulates that of a Tektronix 4010, and therefore the Tektronix Plot-10 graphics software can be used. Thus the display software has been implemented so that programs executing at the experiment control processor invoke subroutine calls that have the standard Plot-10 format. These calls are converted into function requests directed to the display processor. Software executing in the display processor sends the usual Plot-10 compatible ASCII character sequences to the terminal. Because a processor is dedicated to the display function, the experiment control processor also has available subroutine-callable display functions of a much higher level. For instance, a single function request leads to the complete generation of a standard display of position-sensitive detector data.

The second display device associated with the display processor is now under development at BNL. This device employs a high-speed, electrostatic deflection CRT (Hewlett-Packard 1304A) in conjunction with a vector/raster generator that reads data directly from the detector data memory. X-, Y-, and Z-axis data are combined using analog techniques so that rotatable

isometric and density plot representations may be easily achieved.

Input/Output Operations Manager Processor

The lowest level processor in the functional hierarchy, called the input/output operations manager processor, controls a set of peripheral devices and supplies programs to all other processors in the system except the experiment control processor. The most important of these peripherals is a medium capacity disk drive, a Data Systems Design 880. This disk utilizes Winchester technology and emulates a DEC RL01 with an extended capacity of 7.8 Mbytes. The input/output operations manager processor maintains copies of each subfunction processor's operating system on this disk, as well as a number of main program-overlay program sets written in FORTRAN for use by the subfunction processors.

At present, x-ray diffraction pattern data are stored on peripheral devices (disk or magnetic tape) connected to the experiment control processor. However,

a special purpose file management subsystem will be written for execution on the input/output operations manager processor. This file management will have a file structure that facilitates storage and recall of large data arrays and experiment parameters.

Histogramming Memory

Data are accumulated in the histogramming memory from the x-ray position-sensitive detector after being conditioned by the time-slice interface. The histogramming memory consists of three subunits, as shown in Fig. 2, connected to a separate bus segment called the HBUS. This bus is electrically and physically similar to the MBUS and has an identical 26-bit physical address space. The histogramming memory behaves like ordinary memory on the MBUS when accessed from any of the system processors. It is, however, a two-ported memory with data flowing from the position-sensitive detector into one port (the read/increment/write processor) and access requests from the acquisition system flowing through the other port (the HBUS-to-MBUS access port). Only requests for access to physical memory locations that lie within the histogramming memory address space are passed through the MBUS-to-HBUS access port. Therefore, interprocessor communication on the MBUS does not load down the HBUS. The read/increment/write processor has the ability to add from 1 to $2^{24}-1$ to any location in histogramming memory in 1 μ sec. The actual value added by the processor depends on the mode in which data is collected from the position-sensitive detector. The word size in memory is programmable to be either 16 or 32 bits.

Scheduling between the MBUS-to-HBUS access port and the read/increment/write processor is accomplished in a round-robin fashion similar to that described for the multiport memory. Priorities are not assigned to the ports because the HBUS bandwidth is properly allocated to assure no loss of data from the experiment. Derandomization of detector data in the read/increment/write processor insures that HBUS latency does not result in data loss.

IMPLEMENTATION

This section describes selected portions of hardware and software implementation in the data acquisition system. The time-slice interface, read/increment/write processor, interprocessor communication protocol, software implementation, and subfunction processor implementation are discussed.

Time-Slice Interface

The time-slice interface, as shown in Fig. 3, conditions data from the position-sensitive detector and stimulates the sample synchronously with the individual time slices. The time-slice interface is a list driven device; the list is a series of 16-word entries contained in the local memory of the time-slice processor. Data are transferred to the interface by a direct-memory-access interface (DEC DRV-11B) where they are stored in a series of buffer registers until a match between the time in the event clock and the specified event time occurs. At this time, the data in the buffer registers are transferred to the output registers, and

Fig. 3. Block diagram of the time-slice interface showing the various outputs used for control of the experiment front end.

the function implied by these registers is executed. As soon as the data are transferred to the output registers, the buffer registers are reloaded from the next entry in the event list.

The data output from the interface consists of 16 function lines, a 16-bit map register, eight 1 μ sec pulse lines, 8 scaler gate lines, and 8 individual excitation voltage lines. The function lines are dedicated to controlling the front end of the experiment. They are currently used to start and stop the position-sensitive detector and scalars. Eleven of these lines are undedicated so that more complicated experiments may be carried out in the future. The map register is used to offset data in the histogramming memory to correlate with individual time slices. Data from the detector are conditioned by an adder that adds 16 bits from the map register to the high order 16 bits of the position data. Because the read/increment/write processor can accommodate 24 bits of position data, data may be offset in histogramming memory on 256 word boundaries. The pulse outputs are used along with the excitation voltages to stimulate directly the biological sample or to drive auxiliary apparatus. The scaler gates are used to control counting in auxiliary scalars at arbitrary times during a pass through the time slices.

Read/Increment/Write Processor

The read/increment/write processor, which is connected to the HBUS, forms the second major component that handles data from the position-sensitive x-ray detector on its way to storage in data memory. Depending on detector type, this specially built processor can add from 1 to $(2^{24}-1)$ to the contents of a 16- or 32-bit word in histogramming memory. The usual device of this type, found in most pulse-height analysis systems, can increment the contents of data memory only by unity. However, as mentioned in the Introduction, one of the detectors used in this experiment accumulates data in ~ 100 scalars before storing the data in a burst at the end of a time slice. The other detectors process data one x-ray event at a time and hence require the usual increment by unity.

A block diagram of the read/increment/write processor is shown in Fig. 4. Data entry is event-driven and begins upon detection of an event by the position-sensitive detector. Digitized position and count data enter the device and are latched. The data are then derandomized by 10 MHz FIFO buffers so that the full bandwidth of the histogramming memory can be utilized. Up to this point both position and count data are treated equally, but here the position data is used as an address in data memory, and the count data is used to update the contents of that address. The position data may be shifted left one bit to implement 32-bit memory increments. The count data must be multiplexed from 24 to 16 bits because the HBUS data word size is only 16 bits. Sixteen- or thirty-two-bit word size is programmable by the time-slice interface via control lines. The remainder of the processor consists of an adder, data latch, and address and data bus transceivers. The memory cycle controller is implemented as a microsequencer that has instructions for the memory update cycle stored in a fast bipolar PROM. The microsequencer is capable of rudimentary logical tests that conditionally cause execution of different portions of the update cycle. Currently implemented are three microsequences where: (1) single 16-bit memory words are incremented by 16-bit count data with overflow detection, (2) two 16-bit memory words (32 bits) are unconditionally incremented by 24-bit data in two memory cycles, and (3) two 16-bit memory words (32 bits) are conditionally incremented by 16-bit data. In the latter case the condition is that the high-order 16 bits are incremented

only if there is an overflow from the low-order 16 bits. In the first case, overflow from the 16-bit data word is flagged by setting all bits in that word to one. Microsequence selection is programmable via control lines from the time-slice interface. The single-word memory-update time is 1 μ sec using data memory with 425 nsec cycle time, 300 nsec read access time, and 200 nsec write access time.

Fig. 4. Block diagram of the read/increment/write processor connected in the histogramming memory via the HBUS.

Interprocessor Communication Protocol

Computer systems that consist of multiple processors must have available a protocol to formalize interprocessor communication. Communication protocols may have several levels with the number of levels and complexity of each level usually depending on the technique used to effect the communication. In the present system the potential complexity of interprocessor communication is reduced because of the short interprocessor distances involved. The remaining complexity has been removed from the system software by employing very sophisticated communication hardware, i.e., the multipoint memory subsystem. Interprocessor communication blocks reside in the shared memory module; processors at each end of the communication "link" have continuous access to each word of the communication blocks by means of ordinary memory word access.

At the system software level, the basic unit of the communication protocol is the transaction. A transaction is a function request directed from a master processor to a slave processor. The master-slave relationships among the processors in the system are established at system generation time; these relationships define the functional hierarchy of the processors. At the highest level of hierarchy is the experiment control processor, which is always the master processor during any transactions in which it is involved. The experiment control processor uses transactions to request functions of any other processor in the system. The input/output operations manager processor, on the other hand, is the lowest level processor in the system. It always acts as transaction slave and services function requests made by all other system processors.

Transaction processing involves three types of memory blocks located in the multiport memory module. A master mode processor requests a function of a lower level processor by loading parameters of the function to a request transaction parameter block in shared memory. Transaction processing begins when the master mode processor sets the function request flag in this same request block.

Lower-level processors, which act as transaction slave mode processors, examine, in round-robin fashion, transaction request blocks for all potential transaction master processors. When a pending function request is located, the slave processor extracts the function parameters from the multiport memory and performs the requested function. Parameters that represent the output from the function are loaded to an acknowledge transaction parameter block in multiport memory, and a function completion flag is set within this block. The transaction master processor uses the completion flag as a signal that the requested function has been performed. The master processor extracts the parameters returned by the function from the transaction acknowledge block in multiport memory. The transaction request and acknowledge parameter blocks are each 64(10) words in length.

A third block, the transaction datablock, may be involved in a transaction. The datablock is the largest of the transaction blocks, having a maximum length of 8K words, and is also contained in the multiport memory. The datablock contains non-control information associated with the transaction. If the datablock information is to be transferred from the master to the slave, the master processor must load the datablock to multiport memory before the function request flag is set. If the datablock information is to be transferred in the opposite direction, the slave processor must place the requested data into multiport memory before setting the function completion flag.

A typical function request is one in which a master processor requests a FORTRAN overlay program for execution. The memory limit parameters of the overlay (start address, transfer address, etc.) are returned in the acknowledge transaction parameter block; the overlay program itself is returned in the transaction datablock.

Master and slave mode processor routines which allocate the transaction blocks appropriate to a particular master-slave pair and set the physical addresses of the blocks in map registers in the multiport-memory access port are part of the operating systems in the lower-level processors. The routines are initiated by invoking software traps and take the form of trap service routines in the processor synchronous system trap (SST) module.

Any slave processor that responds to transaction requests from several masters is termed a shared-service processor. A shared-service processor must resolve a potential conflict when transactions from several masters compete for the resources necessary to complete the transaction. Principle among such resources are parameter blocks and data buffers. The strategy used in this system is to avoid such competition by preallocating in interprocessor communication memory in the multiport memory subsystem individual interprocessor communication blocks and data buffers for any transaction that might cause such competition. Because the multiport memory has virtually unlimited address space and because of the present low cost of memory, this solution is deemed to be the most cost effective. Other types of resources that may be claimed for processing a transaction include slave processor peripheral devices such as disk and magnetic tape subsystems. Round-robin scheduling allows the slave processor to claim all resources required for a transaction before transaction processing begins and to release them before assertion of the completion flag.

Thus the ability to claim a complete set of slave resources is assured at the beginning of any transaction. This simple set of rules serves quite adequately in a data acquisition system, where data collection procedures are inherently sequential.

Software

As noted above, the data acquisition system uses as much commercially available software as possible, especially for the processor with which the users have the most interaction, the experiment control processor. The experiment control processor executes the DEC RSX-11M operating system. This choice has the advantage that program development can be carried out away from the data acquisition system on a separate system. The experimenter for which this acquisition system is being constructed has a DEC PDP-11/45 which executes RSX-11M and is available for program development.

The user's data-acquisition programs can take full advantage of the peripheral device control and file managements provided by RSX-11M. It is expected that the file management will be used to maintain extensive experiment parameter files on the acquisition system disk. X-ray diffraction pattern data are recorded on magnetic tape. Programs are transferred from the program development system to the data acquisition system via floppy diskette.

A FORTRAN program executing on the experiment control processor requests functions of lower-level processors in the functional hierarchy by executing FORTRAN subroutine calls. Subroutine code sets up in multiport memory the transaction parameter blocks required to control the function request. The FORTRAN subroutines do not directly access registers in the multiport-memory access port but invoke instead a set of software interrupt routines. The software interrupts are referred to as synchronous system traps (SST's) in RSX-11M. The interrupt service routines are maintained in an RSX-11M memory resident library. The user links his program to this library of standard subroutines and synchronous system trap definitions during the process of building his program.

Subfunction Processors

The time-slice processor, the data display processor and the input/output operations manager processor, denoted collectively as subfunction processors, all execute software that is structurally similar. The operating system is a slightly modified version of RSX-11S, the DEC memory resident version of RSX-11M. This

operating system was chosen because it supports elementary input/output operations to a large number of peripheral device types. The method (the "QIO" executive directive) of initiating input/output operations and the device-driver structure used by RSX-11S have been preserved intact.

Figure 5 shows the utilization of local memory for the subfunction processors. The operating system occupies approximately 6K words of processor physical address space. Functions requested of these processors are performed by code written in FORTRAN. The programs take the form of main program-overlay program sets, i.e., a FORTRAN main program remains resident in one part of processor memory and FORTRAN overlay programs are loaded into another part. The FORTRAN main program is typically 6K words long and the overlays typically 8K words long. A main program-overlay program set may contain as many as 63 overlays. Also resident in the memory of these processors is a library of synchronous system trap (SST) service routines. These routines are independent of the particular FORTRAN program executing in the processor. The routines manage system parameters, such as the number of the processor currently requesting a function, as well as managing higher level functions such as loading and initiating execution of FORTRAN main programs. The SST routines also control access to regions in the multiport memory subsystem.

Fig. 5. Function processor logical/physical address space memory map.

All processors at functional hierarchy levels above the lowest obtain their main program-overlay sets from the input/output operations manager processor, the lowest level processor, which cannot request programs and overlays from itself. This processor obtains programs from appropriate areas in the multiport memory. On system initialization, bootstrap programs in the processors execute a wait loop of a predetermined length during which the input/output operations manager processor loads its operating system, main program and first overlay from a read-only portion of multiport memory. The bootstrap programs then request the input/output operations processor to supply them with the operating system image and standard main program appropriate to their processors.

STATUS

As mentioned in the Introduction, the acquisition system has been partially implemented. This section briefly describes the current status of the data acquisition system. The hardware for all processors including the histogramming memory has been integrated and tested. The time-slice interface and the vector display interface are under development. Software for interprocessor communication, the main program-overlay set manager subsystem and the initial data display has been developed and tested. Current software development includes code for the time-slice processor functions and fast vector display. Application code for the experiment control processor is being developed by the scientific group which will carry out the experiments.

References

1. H. E. Huxley, R. M. Simmons, A. R. Faruqi, M. Kress, J. Bordas, M.H.J. Koch, Proceedings National Academy Sciences, Vol. 78, 2297-2301 (1981).
2. H. E. Huxley, A. R. Faruqi, M.H.J. Koch, and J. R. Milch, Nature **284**, 140 (1980).
3. E-M. Mandelkow, A. Harmsen, E. Mandelkow, and J. Bordas, Nature **287**, 595 (1980).
4. D. G. Dimmler and W. H. Hardy, II, IEEE Trans. Nuc. Sci. **NS-24**, No. 1, (1977) 469.
5. J. L. Alberi, "Multiport Memory Subsystem II", BNL 29056 (1978).
6. J. L. Alberi and F. W. Stubblefield, IEEE Trans. Nuc. Sci. **NS-28**, No. 1, (1981) 351.