A Computer Utility for Interactive Instrument Control

by

Paul Day

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or 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.

For Presentation At:
Symposium on Computer Networking and Chemistry
Chicago, Ill. 8/24 - 29/ 1975

U of C-AUA-USAEC

ARGONNE NATIONAL LABORATORY, ARGONNE, ILLINOIS

MASTER

DISTRIBUTION OF THIS DOCUMENT-UNLIMITED

600

ACS SYMPOSIUM SERIES

Typeset and printed by Olin

SYMPOSIUM ON COMPUTER NETWORKING AND CHEMISTRY, DIVISION OF COMPUTERS IN CHEMISTRY, NATIONAL AMERICAN CHEMICAL SOCIETY MEETING IN CHICAGO, August 24-29, 1975

A COMPUTER UTILITY FOR INTERACTIVE INSTRUMENT CONTROL *

PAUL DAY

ARGONNE NATIONAL LABORATORY, ARGONNE, ILL. 60439

I. INTRODUCTION

With few exceptions, laboratory automation has proceeded by the interfacing of a mini-computer to each instrument that requires some type of real-time service; data acquisition, experiment control and/or the analysis of an on-going experiment. To those willing to invest considerable programming effort for each system or those whose needs may be satisfied by a commercially available package, this approach appears attractive. However, limited system flexibility is the price paid for such systems, which price is derived from the inherent limitations of mini-systems; 16 bit word size, fixed system resources, custom tailored programs (not modular), inherent difficulty of program development on a teletype, and unavailability of the computer for experiment control during code generation. The recent trend toward interconnecting several minis together into a network overcomes the limitations of fixed system size and provides the possibility of generating new code while real-time service is being provided. However, there is a class of instruments which require, and others which could greatly benefit by, real-time support which included the storage and analysis of large amounts of data (25K bytes/sec.) using such routines as fast fourier transformations, multi-point smoothing, least squares analysis and

* Work performed under the auspices of the U. S. Energy Research and Development Administration.

matrix manipulation. This type of support cannot generally be provided by a mini-computer network unless a computer with significantly more capability than a mini is connected to the network. This computer should provide sufficient storage capacity and computational ability (32 and 64 bit floating point arithmetic) to permit the real-time execution of 20 to 30K word FORTRAN programs, and it could further increase scientific productivity if all the final analysis for each experiment could be performed on the same system.

Rather than a network, our approach is more analogous to an octopus, in which all real-time support is provided by direct connection to a central computer. A careful study of the real-time requirements of our Chemistry Division in 1967 indicated that a central computer with suitable hardware features, properly controlled by an operating system, could provide all the advantages of a large computer without sacrificing the isolation advantage of the mini-computer in the laboratory (1). The operating system must utilize these hardware features in a manner which will provide each user with the level of real-time and other service required to run his experiment, while at the same time insuring a level of data and program integrity enjoyed by the isolated mini-computer user. To date it has not been necessary to enhance our central computer's service by the addition of a mini-computer between the central computer and the remote instrument. However, some of our remote interfaces do perform computer-like functions which interact with an instrument on a microsecond time scale.

At first glance, the isolated mini-system has three advantages over a larger system (including a mini-network): isolation insures no competition for resources; data and program integrity are as good as the stand-alone software package; since hardware failure rate is somewhat proportional to the number of components in a system, the mini will be out of service less often. However, our system has demonstrated that it is possible to meet the above service and integrity aspects. The only disadvantage of our central system is a hardware failure rate of about once in 20 days compared to minis which often run for many months without failure. However, our on-line users consider this insignificant compared to the following advantages of our central system:

1. The system is of sufficient size and speed to perform most data analysis in real-time or at

- the completion of an experiment. In the rare case that more computing capacity is required, 9-track magnetic tape is used for data transfer to a larger computer.
2. Dynamic sharing provides more resources (CPU, core, disk, magnetic tape) per experiment for a given capital investment.
3. Fast program generation and debugging with high speed card reader and line printer, along with the use of modular programming techniques makes program upgrading easy enough to keep up with the research scientist's changing needs. At least seven of our on-line experiments are considered state-of-the-art systems.
4. Time-shared execution of 25K word FORTRAN programs in real-time. Batch execution of up to 40K word programs for final analysis.
5. On-line keyboard displays and interactive graphics running at 9600 baud for working with data.
6. Disk storage (75 megabytes) for data (FORTRAN accessible), programs and program overlay storage.
7. Open-shop batch-processing permits on-line users to develop and debug their own FORTRAN programs which then have direct access to their data.
8. Only a small programming staff is required. A programming investment of 15 man-years has completed the operating system and applications programs for 21 on-line experiments. An additional 8 man-years was invested in the design and construction of the hardware interfaces.

All control and organizational aspects of the System were designed, coded and implemented by our Chemistry Division (2). The following Xerox software was interfaced to the system; loader (BCM), assembler (SYMBOL), compiler (FORTRAN IV-H) and FORTRAN run-time/arithmetic subroutine package. The central facility has been self-running (no computer operator) for over five years 24 hours/day, 7 days/week with the exception of 8 hours: 4 hours of preventive maintenance followed by 4 hours of System development each week. The system is continually evolving to support new peripherals (disks, interactive terminals, array processors, etc.) and to provide new services as the needs arise. Provisions have been built in for automatic recovery from power failures, to restart faulty peripherals and to cease using faulty memory when detected. System uptime

averages 157 hours per week with one system crash (down time averages about 40 minutes) about every 10 days (half are hardware related and the other system software). Automatic restart for users is available after weekly maintenance or a crash. Daily copying to magnetic tape of all program and data files stored on the disks and RAD (Rapid Access Device - fixed head per track disk) insures users against catastrophic failure. A partial loss of data collected over several hours has occurred three times in five years as a result of hardware failure.

II. System Services

Computer Configuration. The present computer configuration (Table I) has been incrementally expanded from a 24K word system, without disk, supporting 8 experiments to the present system supporting 21 experiments and six independent graphics/keyboard terminals.

Sigma 5 CPU	Central Processing Unit/ floating point arithmetic
Memory	56K 32 bit words plus parity
MIOP (2)	750K bytes/sec. bandwidth, 32 channels
CIOP	21 110 baud teletype, 2 1200 & 8 9600 baud terminals
Disk	251K bytes/sec. transfer, 25 megabytes x 3 spindles
RAD (fixed-head disk)	170 Kbyte/sec., 6 megabytes
9-track mag tape (2)	60 Kbytes/sec., 800 bpi
Card Reader	1500 cpm
Line Printer	650 lpm
Card Punch	300 cpm
Paper Tape	reader: 300 cps Punch: 120 cps
Digital Plotter	CALCOMP 565

TABLE I. Hardware Configuration

Data Transfer. All user and system data is handled by input/output hardware (MIOP) that interacts with all devices running concurrently; being capable of transferring data at an aggregate rate of 750K bytes/sec. Thus, a device seldom waits more than a few microseconds for transfer of a datum after it signals the MIOP that it is ready. Each user's data is moved directly between his assigned program core memory and a device; there is no system overhead required beyond the time required for MIOP transfer.

Do Not Type on This Line

Real-Time Response. A software priority is associated with each program controlling an interfaced instrument. These priorities are assigned on the basis of instrument requirements at system boot-in time. Upon receipt of a request for program execution (e.g. user's device signals "done"), the user's real-time program will commence execution after a time lapse that depends on its priority. The highest priority program will commence execution within 160 microseconds, whereas the lowest priority may not start execution until up to 200 milliseconds later, if one or more higher priority programs are currently "ready-to-run". A simplified real-time execution service cycle may be represented by the following:

1. Program requests a system service (e.g. start reading data from my device)
2. Program calls end-of-service (EOS) (program has nothing to do until data transfer is complete)
3. An MIOP interrupt for this device indicates transfer complete
4. Program processes data
5. Proceed to step "1"

Custom Designed Control Programs. Even though our engineers have established a basic interface design and our programmers have developed effective techniques for acquiring data and controlling instruments, each new system is custom designed at a level which will best serve the user's current and projected needs. A command language is established with which the user can easily communicate with his programs in a natural manner. Program prompts are provided (e.g. "how many scans") so the user does not have to memorize an unfamiliar mnemonic scheme. While our programming staff designs and writes the data acquisition and instrument control portions of the package, users familiar with FORTRAN may code and use their own FORTRAN programs for real-time processing and/or final analysis.

User Communication and Control. Human communication with the computer system is provided at each remote experimental site via a KSR33 teletype. At system boot-in time, input requests are initiated to all communications terminals. At any subsequent time, a user may log-on to the system; at which time sufficient core memory space is assigned for the core resident portion of his specific instrument control program and interact with his instrument through this program. Some of the remote sites also have interactive graphics terminals. In addition, several graphics terminals and keyboard

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Doc No. Type No. Date

displays are strategically located for shared use. A user may similarly log-on and interact with his graphics and/or computation programs from these terminals. The system provides the user with complete control of his instrument interface through the construction and execution of any I/O command sequence that can be executed by the MIOP and his device. This permits the buffering of data into a user's core area, with interrupt indications being passed to the user's control program as each buffer fills, while the MIOP may initiate further data transfer into another user-specified area within less than 10 microseconds after a buffer fills. System integrity is insured by checking for core bounds violation before an I/O operation is initiated on the user's assigned device.

Mass Storage Usage. The system was designed to maximize data throughput. Therefore, disk (and RAD) file structures were simply designed to provide random access in record sizes of the user's choice. When defining a new file, the user provides the name and size of the file to be defined. The system assigns a block of contiguous sectors and records the file name, location, size and creation date in the user's disk-resident file index. The file may be "opened" at any subsequent time, which then places the file's sector bounds in the requesting level's PDT (program description table). The file space within these bounds may then be used in any manner the user chooses; however, data transfer always commences at the beginning of the logical disk sector (LDS) specified (one sector = 256 words). Each request contains the LDS, the core address for data transfer and the number of words to transfer. The actual starting sector address is then derived by the addition of the LDS to the lower sector bound (core-resident after file is "opened"). After insuring that the user's core and file bounds will not be exceeded, the operation is initiated.

To facilitate the rapid updating of large histograms, only a portion of which can reside in core memory at one time, a write-read feature has been provided. This permits the writing back on the disk (or RAD) of a just-updated portion of a histogram and the reading into the same core area of the adjacent portion of the disk-stored histogram; all in one disk rotation.

Non-Resident Program Execution. Our real-time computational requirements vary over a wide range. Our pulsed NMR experiment requires scan averaging a 16K word histogram every 300 milliseconds, taking about 100 milliseconds per update. While this is the highest

usage required by this experiment, it is needed occasionally. Other experiments require the execution of a 25K word histogram transformation program (nuclear fission detectors) every minute, taking about 5 seconds. Other users require this type of execution every 10 minutes for times ranging from a few seconds to 30 seconds.

To satisfy this variety of demand without requiring an inordinate amount of core memory, the operating system provides for the time-shared execution of non-resident programs (not always in core) in the background core area. These programs are disk-resident core images of relatively large programs required infrequently and without severe time constraints. Two queues are provided: one with a 1 and the other with a 32 second execution time limit. These programs are usually FORTRAN which are user-written or from user-group libraries. This system function also makes it easy for the casual user, who is conversant with FORTRAN, to perform his own real-time processing.

Prior to execution, a non-resident program must be "opened", which consists of creating and storing the core image on a disk file and setting the relevant file information in the user's PDT. Multiple programs may be opened and/or open at one time. At execution request time, using the core resident size indication, the system writes a sufficient portion of the current lower priority task on the disk and then reads in the core image non-resident program. The total time required between the request and the start of non-resident execution is about 250 msec. for a 15K word program, if no other request is pending; otherwise it must wait its turn in the queue. Several options are available at execution time; core page size if larger than the code, automatic "opening" of user's current foreground files to the non-resident program, queue selection (1 or 32 sec. time limit) and a "save" option (write program back on disk at exit time). The "save" option allows step-wise execution. At entry time, a register contains the address of the requestor's calling function parameter table (which may contain an argument list of any length) facilitating argument transfer. Non-resident memory write locks are "ORed" with the requesting level's write locks, permitting modification of the foreground core.

Concurrent User Operations. A user's program has the option of overlapping different system services. This is effected through a "type" option in each call a program makes to the system, specifying the type of

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on This Line

action the system should take after initiating the request:

1. Initiate request and return immediately
2. Initiate request and return when request is completed
3. Return if or when the previous request of this kind is completed
4. Initiate request, call end-of-service and return when request finished
5. Initiate request, return immediately and return when request finished

Multiple "request finished" indicators are saved in the requestor's PDT, with user notification being provided on a first-in-first-out basis. Using combinations of the above "types", a user program may overlap a number of operations that do not have to be completed in a specific order. For example:

1. Start device reading into next data buffer
2. Move previous data to magnetic tape
3. Write data to disk
4. Execute a non-resident program which generates display data
5. Computation

Having the option of being notified about the completion of the various processes, the user program can proceed as required.

Graphics Facilities. Graphics facilities consist of a CALCOMP 565 digital plotter and eight Tektronix 4010 graphics units, three of which have 4610 hard copy units. Their respective FORTRAN callable driver sub-routines write calculated move-data to the user's disk plotter-file. After executing the FORTRAN program in the batch mode, a real-time CALCOMP utility program is initiated, which reads the move-data from the disk file and writes this data to the MIOP-coupled CALCOMP. Thus the slow moving CALCOMP plotting is performed while other batch jobs are running.

The Tektronix interactive graphics units are supported with driver routines that execute at a foreground level which in turn makes calls to the user's non-resident code, return to the foreground level reads the move data from the disk and transmits it to the Tektronix screen. Or, the foreground writes an inquiry on the screen and, after a subsequent user response, returns the response to the non-resident program via a new execute non-resident program request. Thus, non-resident core is available for other users during writing, reading or waiting for a user to respond to an inquiry. Provision is also made for the user's program to ascertain the coordinates of thumbwheel controlled

ACS SYMPOSIUM SERIES

Type Single Spaced Only

cursor crosshairs, providing the user further interaction with his program.

Time-Sharing. Previously compiled and stored (batch mode) FORTRAN IV-H programs can be executed in a time-shared mode from any remote keyboard terminal (numbering 31). The keyboard terminal is treated as the I/O device in place of the card reader and line printer. Actual code execution takes place in background core as a non-resident program between I/O statements. Programs executed in such a manner also have access to disk and RAD storage. In addition to the users' specialized programs a library of conversational programs is available. Routines for interactive graphics terminals include: peak resolving, data smoothing, spectral displaying. Interactive routines are also available for manuscript generation using a text-editing package.

Open-Shop Batch-Processing. A very important feature of the system is the support of an open-shop batch-processing service in which jobs are queued through the card reader. This type of queuing provides the casual user with immediate feedback for the rapid debugging of his programs. Batch is the only level in the system which has access to the "batch peripherals"; card reader, line printer, card punch and paper tape reader/punch. The batch level also has access to disk, RAD and magnetic tape. Although the on-line user has the option of performing extensive analysis of his experiment from his remote terminal, the batch level is often utilized where large amounts of output are involved and for the transferring of file data onto magnetic tape for archiving or transport to another computer. The batch level is also used extensively to generate and debug code for the control of on-line experiments.

The batch level uses all CPU cycles not used by on-going foreground processes while no higher priority use (e.g. non-resident execution, foreground loading, core image generation) is required of the background core area. Under normal daytime system loading, the foreground users require about 10 percent of the CPU cycles and the non-resident program execution about another 40 percent. Thus, it appears to the batch user that his program is executing on a computer with about half the speed of a Sigma 5 computer which was dedicated to batch-processing.

Long-Term Computation. CPU utilization seldom exceeds 30 percent in a 24 hour period even with heavy real-time and batch usage. The remaining CPU cycles

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on This Line

are made available for executing very long (hours to weeks) batch-type computations running at a priority level below batch processing. These jobs differ from normal batch in that they only have access to disk and RAD files, not the "batch peripherals". Once initiated (from the card reader), the job is rolled into background core anytime there is sufficient core space and higher priority usage permits. The daily saving of files onto magnetic tape also copies the current core image (now stored on disk) of the long term job along with its disk and RAD files. Automatic file (and long-term) restoration at system boot-in supports executions extending over long periods.

Incremental Expansion. As more experiments are added to the system, one is concerned with two types of system expansion; hardware and software. The Xerox hardware architecture is such that essentially all hardware capability is modularly expandable except the CPU. For example memory can be expanded in increments of 8K words to a maximum of 128K; I/O channels can be added in groups of 8 up to 64 (2 MICPS) until another memory port is added which would then permit up to another 64 channels. The addition of a new interfaced instrument requires the physical connection of its associated device controller to the MIOP bus structure, a one hour job, during which computer power is shut down.

Software expansion need only be concerned with the generation of applications programs for each newly interfaced instrument. The operating system does not require any modification for a new experiment. At system boot-in time, all of the required PDT's are generated from level parameter cards, one per interface. Therefore, a new instrument requires the addition of only one card to the boot-deck. The applications programs for a new instrument are designed and written in close consultation with the users while the new interface is being designed and constructed. Total time from conception to operation runs about 3 to 6 months.

III. Current On-Line Experiments

The 21 experiments presently interfaced to the Sigma 5 require widely differing types and amounts of service. Data transmission rates vary from one byte per minute to 100,000 bytes per second over short intervals. Control requirements vary from none, to computing the angular coordinates of a goniometer from a crystallographic orientation matrix, to providing pulse sequencing for radio frequency generators. Mass storage requirements vary from 1000 to 1,000,000 words.

Real-time processing includes; scan averaging 16K channel spectra, performing mass-energy transformations on correlated pulse-height events, fast fourier transforms, least squares analysis, data smoothing and peak finding. Table II is a list of the currently supported programs with their associated instruments.

Quantity	Research Program	Instrument Type
5	fission and nuclear spectra	Packard 45 pulse height analyzer
2	routine chemical analysis	mass spectrometers (12 in. 60° sec)
1	pulsed radiolysis	Biomation 8100 transient recorder
1	stopped-flow kinetics	Biomation 802 transient recorder
1	precision heat capacity	Vidar 5204 data acquisition system
1	crossed molecular beams	time-of-flight detector
1	crystal structures	neutron diffractometer
1	biological system structures	Varian HR220 super-con pulsed NMR
1	very fast kinetics	Nuclear Chicago Multiscalar (400)
1	chlorophyll studies	Cary 14 spectrophotometer
1	matrix isolation	Cary 17H spectrophotometer
1	30 ft. grating spectrometer	plate measuring comparator
1	primary photosynthesis	Varian E-9 ESR spectrometer
1	plant pigment studies	Varian E-700 ENDOR spectrometer
1	environmental radioactivity	multi-detector (22) counting system
1	high-temperature chemistry	effusate spectrometer

TABLE II. Current Interfaced Instruments

Pulsed Nuclear Magnetic Resonance Spectroscopy.

This system consists of a Varian HR220 super-con NMR operating in pulsed mode for producing fourier transform ^1H and ^{13}C spectra (3,4). Automation is provided to control pulse sequencing with up to 10 interval-width pairs: pulse-widths of 0.5 to 511 microsecond in 0.5 microsecond steps and intervals of

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on This Line

Start typing on
this line

Start first page
of this line

M x 2^N (where M and N range over 0 to 31). Data is digitized with an ADC (8 to 13 bit selection) and transmitted directly to computer memory in groups of up to 16,384 channels with dwell times of as low as 16 μ sec without losing more than 5 percent of the scans. Data loss is hardware detected and software compensated; incomplete scans are discarded. For shorter dwell time a hardware/software interface feature has been provided which takes every N-th (N = 2, 4, 8, 16) data point from N successive scans in an advancing manner, requiring up to 16 times longer accumulation periods, but permitting acquisition of spectra with dwell times as short as 1 μ sec. Data is scan-averaged in real-time onto a disk-stored histogram. After accumulating the desired number of scans the data is base-line corrected, apodized, fourier-transformed (4K channels requires 3 sec.) and phase corrected. Spectra may be stored for future comparison or analysis. A "tau" mode is provided which will automatically collect and analyze spectra at up to 10 specified intervals. The spectrometer is primarily used for natural products chemical studies, especially in photo-synthetic pigments and in electron transport proteins.

Neutron Diffractometer. We have an extensive program investigating crystal and molecular structures in the 10-300°K temperature range on the following types of materials: hydrogen bonded such as hydrated protons; high conductivity inorganic complexes; bonding electron density studies on inorganic and biological compounds. Sigma 5 support for this effort includes automation of a four-circle neutron diffractometer (5,6), located one mile from the computer, which is controlled at any level of interaction selected by the experimentalist. This interaction can be either at the reactor or in the user's remotely located office (not at the reactor). User commands range over the following: move circle "X" to N-degrees, take a count, scan a diffraction peak in a specified region and store its coordinates, compute a preliminary orientation matrix, determine the locations of a set of reflections using the preliminary matrix, compute a least-squares adjusted orientation matrix using selected reflections, scan reflections (up to 10,000) in selected regions of reciprocal space. These operations may extend over periods from minutes to weeks (the program is automatically restarted at the appropriate point after system shut-down or crash) without human intervention or they may be suspended, discontinued or terminated at any time. With typical crystal sizes,

Do not type
below this line

Start typing on
this line

50 about 150 to 400 reflections are scanned per day.

49 Provision is also made for real-time interactive

48 editing and graphic display of data.

47 Analysis of structures containing up to 50 atoms
46 in the unit cell are generally run in batch mode. The
45 analysis includes usage of: the Canterbury fourier
44 program for obtaining 3-dimensional fourier analysis
43 of intensity; ORFLS-3 least squares routine with iso-
42 tropic temperature factors; ORFFE-3 for obtaining
41 atom distances, angles and dihedral angles. These
40 programs may be run during the data acquisition phase
39 to assist in establishing the measurement strategy of
38 a particularly difficult crystal system.

37
36 Pulse Radiolysis. A broad range of pulse radioly-
35 sis experiments is conducted at the "LINAC" (electron
34 accelerator) located about 1200 feet from the computer.
33 They have developed a new method to record the time
32 dependence absorption or emission spectrum produced by
31 a single pulse of electrons. The time resolved spec-
30 trum is produced by an image converter camera with a
29 streak capability (7). The streak image on the phos-
28 phor, which is a two dimensional array of the absorp-
27 tion (or emission), is scanned by a TV camera and
26 stored on a video disk. Under computer control: the
25 video disk image is scanned one line at a time; the
24 scan is digitized and stored as 2000 points in a Bio-
23 mation transient recorder; the Biomation data is then
22 transmitted to a disk file on the computer. This
21 process is repeated until the data from about 100 scan
20 lines has been stored. The data from one scan line
19 represents the light intensity at one wavelength as a
18 function of time. The experimenter may view this data
17 and interact with his analysis programs via a Tektronix
16 graphic display unit, which has a 4610 hard copy unit.
15 After further processing, a FORTRAN program is used to
14 generate a 3-dimensional display of intensity versus
13 wavelength and time. Alternate programs are available
12 to give 2-dimensional plots of intensity versus wave-
11 length. The entire process of obtaining the data and
10 storing in the Sigma 5 takes a matter of minutes com-
9 pared to days by the conventional methods and makes
8 working with highly radioactive materials more feasible.

7 In measuring fast kinetics (50 picosecond risetime)
6 which are initiated by a pulse of electrons, signals are
5 often small and signal-to-noise ratios are limited by
4 shot noise, which means a great deal of signal averag-
3 ing may be necessary. The computer system allows the
2 saving of partial averages, so that long averaging
1 times may be used without fear of losing everything
caused by an experimental problem in the last five

Do not type
past this
lineStart first page
on this line

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on This Line

Start typing on
this line

minutes. Good partial averages are averaged and kinetic data are extracted using non-linear least-squares curve fitting; editing and display are provided on an interactive graphics terminal located at the experimental site. This scan averaging system is effected through an interfaced 400 channel Nuclear Chicago multi-scaling analyzer.

Start text page
on this line

ENDOR and ESR Spectroscopy. A significant portion of our efforts to characterize primary photosynthesis is being supported by two interfaced spin resonance spectrometers (8,9): a Varian E-700 ENDOR (electron nuclear double resonance) and a Varian E-9 ESR (electron spin resonance) buffered by a Nicolet 1074 CAT (signal averager). The uniqueness of these spectrometer systems derives from the large memory and computational speed (floating point arithmetic) of the Sigma 5 which allows the acquisition and manipulation of large quantities of three dimensional data (3-D) via two ADC's. A single 3-D experiment may be composed of up to 640,000 data points. For example in the ENDOR mode signal intensity and radio frequency are recorded as a function of magnetic field. In the ESR mode, the typical intensity versus field spectrum is recorded as a function of a third dimension, such as time for kinetic studies, or light intensity for photochemical studies. Furthermore, spectrometer operation is controlled by the computer; for example, in the ENDOR mode the computer synthesizes the radio frequency value, makes decisions about data validity, and terminates spectrometer operation if the radio frequency power is insufficient. All data calculations, including sophisticated spectrum simulations as well as FFT (fast fourier transform) analysis is performed in real-time by user-written FORTRAN programs. These unique facilities have greatly simplified and speeded up the unraveling of various fundamental aspects of primary photosynthesis in our laboratory.

Stopped-Flow Kinetics. Using the stopped-flow technique, kinetic studies are being made on systems with short-lived reactants and products at a quantitative level never before achieved (10,11). Current studies cover a variety of actinide redox reactions, including work with Np(VII), one of the most powerful oxidants known in acid solution.

The apparatus consists of a Durrum model D-110 stopped-flow spectrophotometer whose output is digitized and stored in a Biomation 802 transient recorder, which in turn is interfaced to the computer. One thousand data points spanning a time range of 0.5 msec. to

Do not type
below this
line

50 20 sec. are collected per experiment. Real-time storage
49 and analysis of the data allow output of the least
48 squares fit results at the remote site within 30 sec.
47 of the end of the experiment. The fast turn around
46 allows the experimenter to adjust conditions continual-
45 ly for optimum results.

44 Spectroscopic Plate Reading. The spectroscopic
43 group is using a 30 foot grating spectrometer to inter-
42 pret actinide spectra, each of which may have up to
41 100,000 spectral lines. An automatic comparator
40 measures the position of spectrum lines on an exposed
39 photographic plate by moving it continuously under a
38 scanning slit and photocell; the local blackening at
37 selected intervals is obtained by digital voltmeter
36 (DVM) readings of the photocell current. About 60,000
35 readings are taken on one scan of a 250 mm. plate,
34 requiring about 4 minutes. The digitized (DVM) data
33 is transmitted directly to the computer, stored and
32 analyzed in real-time using non-resident execution.
31 The smoothed readings and derivatives are scanned for
30 peaks, and the peak positions are converted to wave-
29 length by a polynomial formula derived in a separate
28 scan of known standard wavelengths. An important con-
27 venience is a disk file of standard wavelengths and
26 positions; only one pair of knowns is required as input
25 calibration, so the system requires much less effort
24 than manual measurement and at the same time gives
23 increased accuracy (1 in 10^7).

22 Nuclear Particle Detection. A broad area of re-
21 search effort in our Chemistry Division is concerned
20 with eliciting the details of the fission process. In
19 studying the dynamics of a multi-nucleon fissioning
18 system, it is necessary to measure the changes in the
17 fragment mass and kinetic energy as the initial exci-
16 tation energy, angular momentum, and target mass and
15 spin are altered. The initial conditions are most
14 easily varied by using direct-interaction induced fis-
13 sion reactions such as (d,pf) which allow the simul-
12 taneous observation of excitation energies from below
11 to 10 MeV above the fission barrier. Characterizing
10 these reactions requires a five parameter event-by-
9 event correlation of the fission fragment energies as
8 well as the light particle type and energy. These
7 quantities are converted to mass and kinetic energy
6 distributions in real-time by user-written programs
5 (executing as non-resident programs), permitting suf-
4 ficient interaction between the experimentalist and a
3 real-time display of his transformed data; facilitating
2 optimum use of the experimentalist's and accelerator
1 time.

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on Dotted Line

The computer system supports five data acquisition areas located in the Chemistry building, Cyclotron building and at the Van De Graaff accelerator in the Physics building located about 2,000 feet distant. Experiments range from single pulse height spectra (using Packard 45 memories and model 160 ADC's), recording of event-time data (at resolutions of 40 μ sec.) to multi-particle pulse height spectra on up to six ADC's. Singles, correlated and transformed spectra are generated in real-time and automatically displayed at a selected interval. An option is also provided for storing event data pulse-heights (one or more ADC's) on magnetic tape for subsequent analysis in batch mode or on to the disk for real-time display, editing and analysis. Event rates of up to 10,000 ADC-events/sec. may be processed.

UV, Visible and Infrared Spectroscopy. Two interfaced Cary spectrophotometers (14 and 17H) support the Division's chlorophyll and surface studies groups. Digital data from the Cary 14 is sent directly to the computer for subsequent deconvolution and plotting on the Calcomp plotter. On the Cary 17H, the scan speed, chart speed and intensity range are under computer control (12,13). Direct digitization of the photomultiplier output has enhanced the absorbance accuracy by at least a factor of two. Automatic absorbance scale ranging is also provided during a spectral sweep such that the useful absorbance range has been increased from 0.3 to 0.4. In addition, the problem of data fidelity depending on scan speed has been eliminated. Repetitive scans with data averaging provides a ready means of improving signal-to-noise ratios. The digitized data are available for analysis by programs which include base line correction, Gaussian-Lorentzian deconvolution, peak area measurement, and plotting of the original or program-modified data.

Heat Capacity Measurements. This system is used for the precise measurement of the heat capacities of various chemical compounds between 0.1°K and 350°K, a platinum resistance thermometer being used between 4°K and 350°K and a germanium resistance thermometer between 0.1°K and 25°K (14,15). Thermometer resistances are measured with a 6-digit potentiometer whose dial positions are automatically transmitted to the computer on command. A fifteenth order polynomial (for germanium) which converts voltage to temperature is solved by a non-resident program, returning the time, temperature and temperature drift. After drifting becomes constant, a heating cycle is entered in which readings

Do Not Type on This Line

of heater power and time are read automatically by an interfaced Vidar 5204D-DAS and transmitted to the computer. Following heating, the thermometer current and resistances are again followed, until drifts become constant. The system makes the recording of the data much easier and more reliable and in addition it is much easier for the experimenter to decide when equilibrium has been attained after the heating period, so that another heating can be initiated.

IV. New Directions

Our central computer system has supported present-day instrumentation quite well. However, as the art of automation advances, experimentalists' learn to better utilize their new sophisticated tools and new instrument capabilities evolve, it is clear that new and more difficult data acquisition problems will continue to arise. One of the more demanding problems is the scan-averaging of spectra acquired at rates approaching a million channels per second. Another is more sophisticated real-time control of experiments requiring logic decisions (computation) on a micro-second time scale. One might try to solve these problems by adding a mini-computer between these new instruments and the central system. However, one is then faced with the problem of programming these individual computers (probably of different manufacture) and designing an interface to the central system. In addition, a mini-computer's response time is faster than a central computer only by virtue of the fact that it takes 4 nanoseconds per foot (2 way transmission) for data transmission and communication.

We are intending to provide higher speed capabilities in the future by using microprocessor support at the experimental site. Xerox has a System Control Unit which has a wide variety of configurations and options including 8 registers, interrupts, stacks, clocks, read-only and writable memory (16 bit, 350 and 700 nanosecond) in increments of 1K words, and optional floating point. We will load these computers with code from the central computer for each phase of its data acquisition effort, using them more as a sophisticated input/output device than as a computer. This will keep the programming logically simple and thus efficient. In addition, this type of usage will not require much memory, thus keeping the cost down below that which would be required by using a mini-computer.

V. Summary

The successful operation of our system over the past five years is a clear demonstration that a central computer, without remote mini-computers, can provide very sophisticated real-time support without suffering the disadvantages of unreliability often experienced in large systems. Users never experience the consequences of another user's errant program, only their own. The level of system integrity is such that we test a completely new instrument control program during normal system loading, without having to be concerned about interfering with any other user's interests. Evidence of the system's acceptance as users have been added over the years, is that we continue to have 6-8 new users wanting to be interfaced as funds become available. The total capital investment for the entire system is about \$50K per experiment, which compares favorably to the cost of sophisticated mini-systems. But, one has much more resource and capability to support each experiment: providing full experiment control in a flexibly interactive manner; fully interactive graphics capability with on-going and completed experiments; computational ability for complete final analysis of experiments.

APPENDIX

Memory Bus Structure. The 960 nanosecond core memory (32 bits plus parity) is modular; each 16K words has its own read-write and address recognition capability and may have up to six ports. The CPU and each processor (MIOP's and CIOP) is connected to its own "port bus" made by joining one port from each memory bank. Simultaneous access is achieved when requests are in different banks and cycle stealing takes place when more than one request coexists for a single memory bank.

Memory Protection. A 2-bit lock and key feature prevents a user's program from modifying unassigned core, while permitting the execution of reentrant code. The "key" resides in the current PSD (program status doubleword), the page (512 words) locks are set in fast memory in the CPU via a single instruction by the scheduler each time a program level commences execution. Implementation of the protection feature will trap a core-modify instruction before execution and yet does not increase instruction execution time.

Start typing on
this lineStart first page
on this line

50 Data Flow. Each MIOP (multiplexor input/output
49 processor) may handle an aggregate data flow of 375K
48 memory accesses per second (one or four bytes) on up to
47 32 concurrently active data channels (typically one
46 per experiment). Only a single instruction must be
45 executed by the CPU to initiate a data transfer between
44 any core memory area and a specified device; all sub-
43 sequent data movement and communication with the remote
42 device is handled by the MIOP. At the termination of
41 data transfer (up to 65,536 bytes), the MIOP generates
40 a single interrupt. Upon recognition of the interrupt,
39 the CPU ascertains the channel number and the channel-
38 end conditions by executing a single instruction.
37 Thus, high speed data transfer to individually selected
36 core areas (specified by each user's program) is com-
35 pletely handled by the MIOP, freeing the CPU to per-
34 form useful computations concurrently with data
33 transfers.
32

31 Terminal Communications. The CIOP (communications
30 input/output processor) handles up to 128 lines of
29 mixed speeds from 110 to 9600 baud; currently driving
28 21 teletypes, 8 interactive graphic units and 2 key-
27 board displays. This processor works with the CPU in
26 a manner similar to that used with the MIOP. Thus
25 message transfers take place to any core memory area
24 while the CPU is performing computations.
23

22 Experiment Interfaces. Each of the remotely lo-
21 cated on-line experiments is interfaced to a remote
20 controller. The simplest controllers consist of a one
19 byte buffer and a few logic lines. The more complex
18 ones have up to sixteen byte buffers, computer initia-
17 ted timing circuitry to control instrument functions to
16 a precision of 0.1 microsecond and numerous logic
15 lines. For transmission rates up to 100,000 bytes per
14 second at distances up to 1500 feet, a remote control-
13 ler is connected to a device controller via a bundle of
12 co-axial cable (8 bit wide data path). For transmis-
11 sion rates of up to 4K bytes/second and at distances of
10 up to one mile, four twisted pair are used. The device
9 controllers were designed by Argonne engineers and are
8 connected to the MIOP bus structure in the computer
7 main frame, weekly data transmission validity tests on
6 all interfaces running concurrently for about an hour
5 never detect a transmission fault. About every eight
4 months one of the interfaces fails solidly because of
3 component failure, which is readily fixed within a few
2 hours.
1

Do not type
on this line

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Printed Type on this page

Start of page 50
this page

Core Memory Assignment. With the exception of system areas, core memory is assigned dynamically. As users log-on from their terminals, sufficient core is assigned for the resident portion of their data acquisition program, which includes their own data buffer areas. The programs are structured into user-controlled overlays which are typically written onto their disk file and rolled into their core area as they require. Using this overlay technique, the average core residency required per program runs between 2 and 5 pages (page = 512 words). To perform real-time computations using FORTRAN, the foreground code may call for execution of a "non-resident" program. Foreground core area is assigned from the top of core down in the first available block of contiguous core pages. Background core (batch and system tasks) is assigned from lower core upward. This permits the maximum background core area to extend on up to 42,600 words when no real-time tasks are loaded. While background core is variable in size, the user is always guaranteed an area of at least 15,900 words. Real-time job lengths vary from minutes to weeks. On a typical day, about 60% of the instruments are on-line at any one time, requiring about 14,400 words of core. This leaves about 28,200 words of background core.

Start of page 51
of this page

Reentrant Code. The Xerox write-lock feature, which permits the execution of code anywhere in memory yet forbids the modification of memory without the proper "key", facilitates the utilization of reentrant code to perform many user functions. Reentrant code is defined as code that does not modify itself. About 800 words of reentrant code are in permanent residence for performing such functions as two-way translation of keyboard I/O and binary-coded-decimal conversion for some of the interfaced multi-channel analyzers. In addition, there are 128 words of EBCDIC-ASCII conversion tables available.

Program Description Tables. In providing a multi-programmed environment for a large number (currently 41) of concurrently running programs (user and system), it was found that a program description table (PDT) associated with each priority level was an efficient method of keeping track of system service requests and individual program status. Besides simplifying priority queuing and level termination at end-of-job, a PDT structure also simplifies the addition of a new program level to the system. PDT generation is controlled at system boot-in by one level-parameter card per terminal.

Do Not Turn on This Line

The PDT for each priority level is 80 words long plus two words for each I/O command pair needed by the level (ranging from 1 to 16) to control its associated device. The table content also includes; user ID number, level status, program status double words (PSD), memory write-lock image, time remaining, core bounds, disk memory file bounds and register values. These tables are stored in a write-protected area of core memory.

Scheduling. The system structure/operation and usage strategy have been developed concurrently. Thus a system has evolved which maximizes the system work per unit time in such a manner as to provide the required level of real-time interaction between the user, his keyboard and his instrument. However, the specific strategy used for any particular application is not dictated by the system as long as programs make requests which conform in time, core memory, device operation and file bounds. A hardware trap is set to disallow the execution of privileged instructions by user code.

The basic scheduling algorithm runs each task to completion, contingent upon its priority. The scheduler determines what priority level will execute next, through the processing of all I/O interrupts from the MIOP, and the maintenance of software priority level status using the PDT's. Once the identity and status of the level associated with an interrupt is determined, the scheduler compares its priority with that of the level interrupted. If it is of lower priority, the interrupted level is resumed. Otherwise, the execution dependent portion (PSD, execution time and registers) of its current state vector is moved to its PDT. Referencing the new (interrupt signalled) level's PDT, its execution is initiated by setting the memory write-protect locks, execution time, registers and the PSD.

Upon receiving an end-of-service (EOS) request from a level, the scheduler records this in its PDT. Then it scans down the priority chain through the PDT's until it finds another level that is "ready-to-run". State vectors are then adjusted as above to initiate execution.

A necessary feature in a multi-programmed system is the prevention of processing lockout by some higher priority level looping endlessly. A maximum service cycle time (execution: foreground=1 sec., non-resident=1 or 32 sec., batch and long term=no limit) and a time remaining value are stored in each PDT. Any time a level is executing, the time-remaining value is decremented by a CPU clock (2000 Hz). But each time a level

Do Not Type on This Line

calls EOS, the maximum service value is moved into the time-remaining PDT location. By breaking processing into logical service cycles, a program can readily perform its function forever, but will be terminated by time-out if it malfunctions time-wise.

ABSTRACT

A careful study of our laboratory automation needs in 1967 led us to the conclusion that a central computer could support all of the real-time needs of a diverse collection of research instruments. A suitable hardware configuration would require an operating system to provide effective protection, fast real-time response and efficient data transfer. An SDS Sigma 5 satisfied all our hardware criteria, however it was necessary to write our own operating system; services include program generation, experiment control real-time analysis, interactive graphics and final analysis.

Our system is providing real-time support for 21 concurrently running experiments, including an automated neutron diffractometer, a pulsed NMR spectrometer and multi-particle detection systems. It guarantees the protection of each user's interests and dynamically assigns core memory, disk space and 9-track magnetic tape usage. Multiplexor hardware capability allows the transfer of data between a user's device and assigned core area at rates of 100,000 bytes/sec. Real-time histogram generation for a user can proceed at rates of 50,000 points/sec. The facility has been self-running (no computer operator) for five years with a mean time between failures of 10 days and an uptime of 157 hours/week.

Literature Cited

1. Day, P. and Krejci, H., Proc. AFIPS FJCC (1968) 33 1187-1196
2. Day, P. and Hines, J., Operating Systems Review (1973) 7 (4) 28-37
3. Scheer, H. and Katz, J. J., Proc. Natl. Acad. Sci. USA (1974) 71 (5) 1626-1629
4. Crespi, H. L., Kostka, A. G. and Smith, U., Biochem. Biophys. Res. Comm. (1974) 61 (4) 1407-1414
5. Peterson, S. W., Willett, R. D. and Huston, J. L., J. Chem. Phys. (1973) 59 (1) 453-459
6. Williams, J. M., Petersen, J. L., Gerdes, H. M., and Peterson, S. W., Phys. Rev. Lett. (1974) 33 (18) 1079-1081

ACS SYMPOSIUM SERIES

Type Single Spaced Only

Do Not Type on This Line

7. Gordon, S., Schmidt, K. H. and Martin, J. E., Rev. Sci. Instrum. (1974) 45 (4) 552-558
8. Norris, J. R., Uphaus, R. A. and Katz, J. J., Chem. Phys. Lett. (1975) 31 157-161
9. Norris, J. R., Bruyan, M. E. and Katz, J. J., J. Am. Chem. Soc. (1973) 95 1680-1682
10. Watkins, K. O., Sullivan, J. C. and Deutsch, E., Inorg. Chem. (1974) 13 1712
11. Weschler, C. J., Sullivan, J. C. and Deutsch, E., Inorg. Chem. (1974) 13 2360
12. Gruen, D. M., Gaudioso, S. L., McBeth, R. L., and Lerner, J. L., J. Chem. Phys. (1974) 60 (1) 89-99
13. Green, D. W. and Gruen, D. M., J. Chem. Phys. (1974) 60 (5) 1797-1801
14. Osborne, D. W., Schreiner, F., Flotow, H. E. and Malm, J. G., J. Chem. Phys. (1972) 57 (8) 3401-3408
15. Osborne, D. W., Flotow, H. E., Fried, S. M., and Malm, J. G., J. Chem. Phys. (1974) 61 1463-1468