COMPUTERS AND NUCLEAR PHYSICS

Recent developments in both hardware and software hold great promise for effective use of computer systems in nuclear-physics laboratories. Interaction between physicists and computers is, consequently, being simplified.

JOEL BIRNBAUM and MARTIN W. SACHS

TRANSFORMING RAW DATA from a nuclear-physics experiment into physical parameters and then to a publication follows a fairly universal course. For the past 20 years computers have performed more and more transformations. The first applications were primarily to the last stage, that is, transformation of reduced data into physically meaningful terms. As input and output devices connected to computers become more sophisticated, computers are put to work on more and more additional tasks. Today it is possible to have a computer perform most of the routine tasks in an experiment, perhaps even including the editing of a text for publication.

Especially in the last five years computer data-acquisition systems have been burgeoning in nuclear-physics laboratories.[1] There has also been a corresponding proliferation of papers proclaiming the brave new world of nuclear-physics experimentation. Unfortunately, however, aside from the obvious advantages that accrue from a machine with universal adaptability, the "revolution" of *physics* performed has been rather slow aborning. Although one can no longer argue about a computer system's increased flexibility, speed, capacity and convenience relative to its recent and sophisticated predecessor, the fixed-wire multiparameter analyzer, in most cases the computer is used in a manner that amounts to a replacement of one box of electronics with another functionally identical box that performs better. The conceptually exciting benefits of on-line systems, which have generated many proselytes, result from its use as an analytical instrument rather than as a means for rapidly ingesting large quantities of data and displaying them on a cathode-ray tube or a printer. To be used effectively a computer must simultaneously acquire, reduce and analyze data, and inform the experimenter of these results while his experiment is in progress.

The new generation of laboratory computers, especially the larger systems, have the necessary power for considerable simultaneous analysis:

Joel Birnbaum received a bachelor's degree in engineering in 1960 from Cornell University. In 1965 he took his doctorate in experimental nuclear physics at Yale. Since then he has been a staff member of IBM Watson Research Center, Yorktown Heights, N. Y., involved with research in computer-assisted laboratory experimentation.

Martin Sachs earned an AB from Harvard (1959) and an MS and PhD from Yale (1964). He was a postdoctoral fellow at the Weizmann Institute, Rehovoth, Israel, and is currently a research associate with the Wright Nuclear Structure Laboratory at Yale. He is responsible for the Yale portion of the Yale—IBM study of nuclear-data acquisition.

For this reason we will concentrate on these larger systems and emphasize the possibilities for enhancing an experimenter's analytical and judgmental powers rather than automating experiments. Although we deal with nuclear-physics applications, the general concepts are applicable to most areas of research. We summarize the current status of computer systems and, perhaps more importantly, underline difficulties. We attempt also to propose solutions to problems most amenable to eventual remedy so that physicists can use the computer's full potential for nuclear physics. This potential is not the ability to obtain more data (although this is undoubtedly important), nor is it to take better data (which is still more important) but rather to provide a basis for solving physical problems that cannot be handled in other ways.

It is not surprising that computers, with very few exceptions, have not yet been used in a revolutionary fashion. Innovation generally demands complete familiarity with the problem, and almost all computer systems have required much of the physicist's time for this familiarity. He must be conversant not only with the details of his equipment, but also, in most cases, with the program structure and programming system that guides his experiment. Most physicists are not interested in devoting time to achieve this expertise; so, if they have used a computer system at all, it has been provided by others or already exists in a conventional manner. To be fully effective, therefore, a system must be dynamic, interactive and amenable to reprogramming or restructuring with little or no effort.

What's new in computers

Before proceeding to the particulars of installing a computer in a nuclear-physics laboratory, it would be well to review briefly the many advances in the computer industry during the last five years. Improvements in circuit techniques, mass-production capabilities and input–output devices have now enabled one to purchase for a single laboratory a physically small and relatively inexpensive computer whose power frequently exceeds that which was formerly provided in a computing center for an entire university. Increase in power is not only attributable to the greater speed of microelectronics, but also because computers can now routinely overlap input–output operation with simultaneous arithmetic processing. Advances in reliability of computing systems have been equally important. Since most physics laboratories operate on an essentially full-time basis, and since experience has proved that an on-line computing facility will be involved in virtually all experiments, the computer's downtime becomes a vitally significant parameter. The current generation of laboratory computers does not significantly affect overall reliability of the total accelerator–experiment–data-handling complex.

Another difficulty that has arisen in many installations is that time-consuming use of a computer for data acquisition precludes program preparation and subsequent analysis of previously acquired data. New developments in programming techniques and in computer hardware allow computer sharing among several tasks that are effectively performed simultaneously.

Finally there have been advances in programming techniques, particularly in programming systems that relieve one of tedious and repetitive programming tasks. Most systems provide languages such as Fortran that permit programs to be written in a way that is completely independent of the computer's detailed operation. Some also support terminals that allow one to interact with an experiment in progress. In fact this interaction will produce significant changes in the performance of experimental physics.

In the next sections we present a brief summary of some of the hardware and programming-systems innovations and the ramifications for low-energy nuclear-physics experiment design. We make no attempt at a comprehensive review of existing systems (the reader is referred to reference 1 for recent summaries); rather, we discuss general principles with examples drawn principally from our experience at Yale University Wright Nuclear Structure Laboratory.

Data channels

A distinguishing characteristic of most nuclear-physics data acquisition is that data occur randomly in time and that one frequently must require a sampling of instruments in time-coincident combinations. Fast data channels located in laboratory-size computers minimize the overhead associated with responding to these data. The data channel can be regarded as an input–output computer that executes its own program and performs input–output operations that are completely overlapped in time with activities of the central processing unit (CPU). With a channel for input, one can store data as they arrive without interrupting CPU activity and without imposing excessive "deadtime" on the measuring instruments themselves.

In general, measuring instruments are connected to the channel through a control unit, called an "interface," which has some provision for associating specific instruments or groups of instruments with particular types of events. Data from the instruments are transferred with any necessary identification words in an uninterrupted stream into a reserved list of words in the computer memory. When this buffer area is full, the CPU is informed of this fact by an interruption from the data channel and begins to process these data while a second buffer is being filled with more data.

This scheme allows much higher average and instantaneous counting rates than arrangements in which the CPU must respond to each event as it occurs. For most nuclear-physics experiments the time lag, in this scheme, between receipt of an event and processing is inconsequential. To handle cases in which response speed is important, control units must be able to notify the CPU as soon as the event has occurred so that it can be processed immediately instead of waiting for the buffer in which it is located to fill up.

Data channels simultaneously accumulate data, log it on magnetic tape, reduce it (or, when data reduction in real time is not needed, perform independent processing), and operate a display unit. The CPU spends most of its time reducing data; other operations are performed concurrently through the channels. In this manner, even without special add-1-to-memory circuitry, the Yale system, for example, can perform approximately 70 000 pulse-height analyses per second.

The control unit at Yale has been previously described.[2] Its operation can be briefly summarized as follows: Data-acquisition components [principally scalers, timers and analog-to-digital converters (ADC)] are grouped

PARTICLE IDENTIFICATION WITH AN ON-LINE COMPUTER

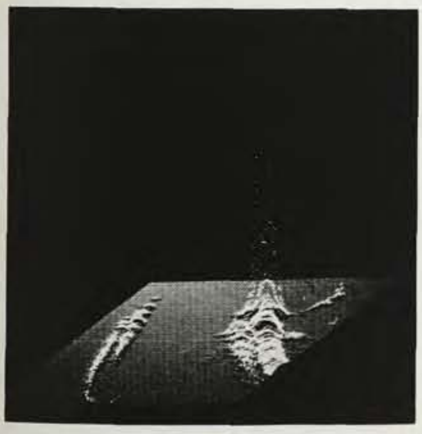
Identification of the reaction products of heavy-ion reactions at energies significantly above the Coulomb barrier is an important and challenging experimental problem. Essentially all reaction channels are open at these energies, and the investigation of a particular energy level of a given isotope will generally not be amenable to kinematic-coincidence techniques in themselves, particularly for reactions with low cross sections. It is necessary to measure some additional parameter to identify explicitly the particles.

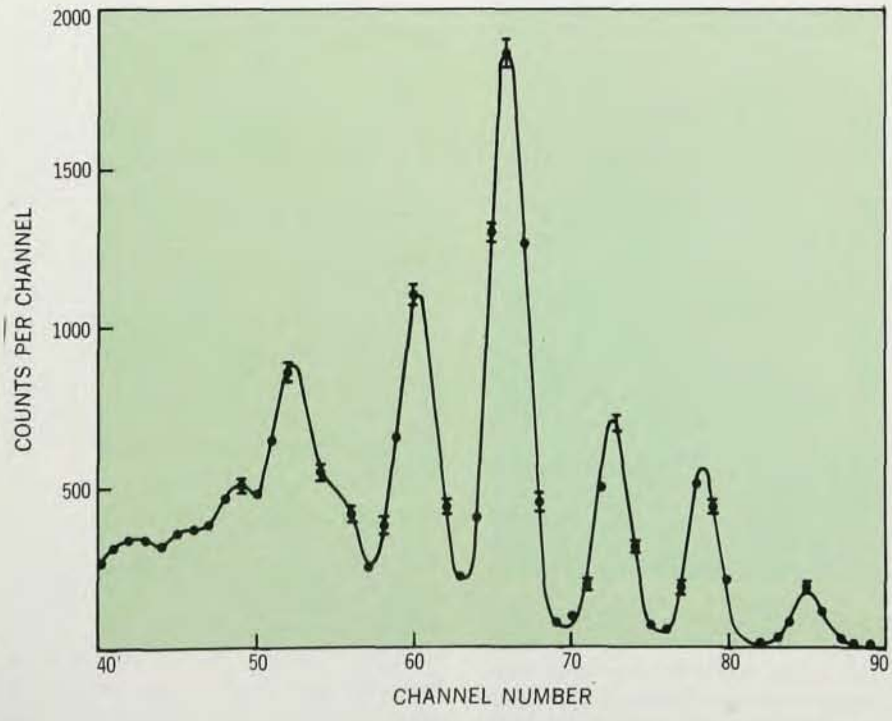
The most widely employed procedures of particle identification are based on the Bethe equation, which relates the rate of energy loss of a charged particle moving through nuclear matter to its mass, charge, energy and the properties of the absorber. If relativistic and slowly varying terms are ignored, this equation may be approximated for many experimentally meaningful situations by $dE/dx = (KMZ^2/E)$, where dE/dx is the rate of energy loss of particle, M and Z are the mass and charge of the particle, and K is a constant that combines various physical constants and parameters of the detector system. Thus the product of E and dE/dx provides a unique identification of the particle since no two known nuclides have the same value of MZ^2. The locus of a particular isotope in a plot of energy loss against energy approaches a rectangular hyperbola. The experimental technique uses a stack of detectors (known as a "particle telescope"), composed of one or more thin detectors that measure the energy loss, and a residual-energy detector thick enough to stop the particle.

The E and dE/dx signals may be combined with an analog pulse multiplier to calculate the function $(dE/dx)(E + hdE/dx + V_0)$, with h and V_0 being adjustable parameters. V_0 and h are adjusted until the output is independent of E and depends only on MZ^2. By setting voltage gates on the regions of interest in the multiplier spectrum, one can route data on individual nuclear species to separate regions of multichannel analyzers. In this fashion one obtains directly a conventional energy spectrum by requiring a time coincidence between the E-detector pulse and the identifying signal from the multiplier.

A multiparameter analyzer simplifies the experimental procedure further: The dE/dx and E signals are presented to analog-to-digital converters, and the appropriate cell in a magnetic core memory is updated when a time coincidence is detected. In this manner the isotopic hyperbolas are recorded directly, are viewed during the course of the experiment, and are available on magnetic tape for subsequent off-line analysis by computer. The two photographs illustrate computer-produced displays (isometric and contour) of data from an experiment in which a N^{15} gas target was bombarded with a 114-MeV B^{11} beam. The graph is an entirely computer-produced energy spectrum for Be^{10}. [From J. E. Poth, J. C. Overly, D. Bromley, Phys. Rev. **164** 1295 (1967).]

The replacement of the multiparameter-analyzer–off-line-computer combination by analog-to-digital converters coupled to an on-line computer yields still further advantages. Then the creation of the energy spectra of individual isotopes can be carried out with any degree of sophistication desired, while the experiment is still in progress. Relations other than the Bethe equation may be readily substituted when appropriate (for example, algorithms based on time of flight or range energy have proved useful in certain situations). The many advantages of pulse-height analysis that accrue with an on-line computer are also applicable. For example, only regions of interest in the spectra need be considered, and these can be discontinuous and have different resolutions. Megachannel analysis, using associative memory techniques (wherein a memory cell is assigned only to those channels containing meaningful information), can be employed when high resolution is required. Digital stabilization, graphical manipulation for theoretical comparison and other procedures discussed above may be freely invoked. Furthermore *all* data can be saved on magnetic tape or disk if desired; so the selection of the reaction products to be analyzed is not irrevocable, as it is with many hard-wired schemes.





at the beginning of the experiment by setting pins in a diode-matrix pinboard that associates arbitary combinations of these components with an externally provided event signal. For example, the event signal might be the output of a coincidence detector, an overflow signal from a scaler or an electronic signal from a program command. When an event signal is presented to the interface and when all components have completed their measurements, data from the components are transferred in sequence to the computer and are preceded by a word that identifies the event. Individual components are modular and totally interchangeable logically, electronically and mechanically. An experimenter may, therefore, arrange them in any desired order. A subroutine that processes an event treats it simply as a list of 16-bit data words in a specified order. This subroutine is not concerned with the components' physical locations and needs not even be concerned with their nature (for example, the same program might handle either pulse-height analysis from an ADC or time-of-flight analysis with a scaler–timer). Other functions are provided, such as rejection of unwanted coincidences between events and optional generation of an external interruption to notify the computer when a specific event has occurred. If desired, a hierarchy of priorities can be associated with the events.

If, as is frequently the case, one desires to log data in its raw form, no additional programming is necessary. The buffer may be written unchanged onto the appropriate output unit by system routines. In the Yale system one can use the same Fortran program in real time with or without data logging, or process data previously logged on tape. One selects the particular mode of operation by a single control card at execution time.

Because one can have automatic classification of data into unique events and because format is independent of devices that produce it, there are many advantages. Coding (in Fortran) to process each event can be written independently of other events, thus allowing versatility without having to modify existing event routines. In addition, data logged on magnetic tape can include additional events not processed in a program by assigning them to events having no routines in the program. These data would later be analyzed by replaying the event tape. In fact several experiments can be run simultaneously by assigning to each a unique group of events.

An interface such as Yale's quickly becomes quite complicated; an alternative approach is to replace this interface with a small computer that is connected by a data channel to a larger computer. This approach is quite advantageous: Such an interface is entirely flexible, can be changed under program control, and can otherwise perform tasks that leave the large computer free for data reduction and analysis. On the other hand, since hardware functions are being replaced by programs, this approach may not sustain as high a data rate as a purely electronic sorting and identifying scheme (for the same expenditure of funds).

SPECIFICATION	ANALYZER GATE SCALER
ACTION	CREATE ENTER EVENT MODE EVENT n EVENT END CLEAR PHA NPHA SCALE DGATE IGATE

DATA-ACQUISITION STATEMENTS in the Yale System. The three specification statements define memory-core storage images of the multichannel analyzers, gates (single-channel analyzers) and scalers. —FIG. 1

Display hardware

The cathode-ray-tube display has become indispensable in experimental nuclear physics. It not only allows data presentation in an unlimited variety of forms, but also can serve as an input medium with a light pen for manipulation and selection of data.

There are several types of display and several ways of connecting them to a computer. All that we have mentioned about data acquisition and data channels is relevant as well for connection of displays. The simplest form of display involves a point-plotting oscilloscope driven by digital-to-analog converters and connected to the computer by a data channel. This setup is the basis of the display unit in the Yale system. With this organization, the CPU is required only when structuring coördinate buffers in the computer's main memory; subsequent regeneration is completely under channel control. Its advantage is relative simplicity and high speed with which one can present data to the terminal. Its disadvantage is expensiveness in terms of computer memory for storing data for the display. This situation is particularly unsatisfactory if one requires multiple independent displays. A more satisfactory solution provides external storage of data for the display unit so, once loaded with display coördinates, the display is self-sustaining. Possible schemes include magnetic drums or disks to hold information being displayed, storage oscilloscopes and auxiliary computers for display control and information storage. Magnetic drums and disks have the advantage of large data volume and regeneration rate at the expense of a complex control-unit design; the storage oscilloscope has the virtue of economy; an auxiliary computer provides the greatest flexibility.

At Yale, several special features in the display unit augment interaction between physicist and computer. These features include relocation hardware in the display control unit so an image on the screen can be moved by simply changing a coördinate specifying its origin rather than by recalculating each set of coördinates that describes all points in the display. There is also character-generation hardware that enables rapid production of alphanumeric information in arbitrary font and variable size.

Control and monitoring

Many computers now have as optional features control units that send and receive both digital and analog signals, sense the position of external contacts, read external voltage levels, and so forth; hence one can easily use them for apparatus control and monitoring.

COMMUNICATIONS TERMINAL showing the display CRT, light pen and function keyboard. The oscilloscope at right is connected with large CRT and is used for photographic purposes. —FIG. 2

One general scheme of operation employs a digital computer to initiate and then to monitor continuously a process while in progress. A second scheme involves sending the desired final value of parameters associated with each piece of apparatus to an external unit that will, without computer control, monitor the operation and notify the computer when the process ends. The advantage of the first scheme is that it requires construction of little or no special-purpose hardware. Its disadvantage is that it consumes more computer time than the second procedure, which requires the computer's intervention only at the beginning and end of a process. One can structure programs associated with apparatus control in a general way so a small family of quite broad functions can serve the needs of an entire laboratory.

At Yale, to free the CPU as much as possible we have built a simple interface box containing input and output registers that can be addressed by the CPU and that are easily connected to a wide variety of digitizers. Each unit, which has extensive device-multiplexing capabilities, also contains a digital comparator so the CPU need only provide the ultimate value for a given parameter. After this step external hardware monitors that parameter, stops associated variation and signals the computer when the destination has been reached. The CPU, of course, can read the actual value of a parameter at any time; it can, therefore, confirm operation completion.

An apparatus-control-and-monitoring facility provides a major improvement in data quality by temporarily interrupting counting when specified parameters exceed predetermined limits. Such interaction between control and data-acquisition functions is particularly easy to implement when the same computer is doing both jobs. For example, in an experiment that is dependent on careful control of beam optics, the computer monitors beam centering by measuring various slit currents and makes necessary corrections (either automatically or by notifying the accelerator operator of deviations), meanwhile stopping counting until the deviation is corrected. Similarly in fast-coincidence work, a computer could monitor the instantaneous beam current for unwanted fluctuation, again stopping counting and taking corrective action when this fluctuation exceeds a specified limit.

Data-acquisition language

With programs that analyze and display data in higher level languages such as Fortran, a physicist has much greater freedom to experiment with innovative techniques and to reduce drastically the amount of time that he must spend learning to program the computer. To write real-time programs in this way requires a far more complex operating system than is needed if real-time programming is done in machine-dependent assembly language; however, program preparation and modification are simpler and more efficient.

Special application-oriented statements for data acquisition have been incorporated into Fortran at Yale. These statements are shown in figure 1. Each of these statement types includes parameters (not shown) specifying vital statistics such as number of channels in an analyzer, overflow value and statement to which control is to go on overflow of a scaler, etc. CREATE sets up data areas and control information for analyzers, gates and scalers. ENTER EVENT MODE turns control over to the real-time supervisor and initiates input from the nuclear-data interface. EVENT *n* and EVENT END define the start and finish of a block of Fortran coding that processes event *n* whenever it is found in the buffer. CLEAR clears an analyzer, gate or scaler. PHA and NPHA perform pulse-height analysis (+1 and −1, respectively) with the input datum as a channel number. SCALE adds an input datum to the software scaler. DGATE performs single-channel analysis on a datum, and IGATE integrates a region of a spectrum.

These statements may be freely used in normal Fortran programs. A precompiler first scans the program and translates these special statements into Fortran coding. The resulting program is then compiled into machine language in the usual way. Data-acquisition language is implemented by a precompiler rather than by modifying the Fortran compiler since the precompiler is easier to alter and expand than the compiler. In fact the precomplier itself is written in Fortran.

Mutiprogramming and time sharing

A principal problem in computer utilization in any application is optimization of resources (time, memory and external equipment) associated with the computer. With on-line data acquisition this problem often amounts to a simple dilemma: A computer must be available for full-time, on-line data acquisition and control and yet still be ready for program preparation and data analysis. Furthermore one frequently cannot write a single, straightforward program that both collects data and performs the desired immediate analysis without having to stop one task to do the other. Although in some situations one can substitute two smaller identical computers for a single large one, in many cases this is undesirable because the greater power of a large computer is often needed.

This problem is solvable because although a computer may be busy full time with data acquisition (on a time scale of seconds or minutes), it is frequently idle on a scale of milliseconds or even seconds as it is waiting for a new event to occur or for an input or output operation to be completed. One can design a programming system that uses these idle periods, together with any available equipment and memory space, for the execution of other programs that were written independently of the data-acquisition program and of each other. Such programs can concurrently analyze data being acquired, or they can be involved with assembly, compilation and testing of new programs as well as analysis of previously acquired data.

Various types of systems are in laboratories. Depending on the nature of the algorithm that divides the time and other resources among the various programs, they are called "multiprogramming," "time-sharing" or "time-slicing" systems. For such a system to be successful a computer must not spend too much time deciding which program is to run next and performing transfer of control from program to program. With recent computers, special hardware as well as new programming techniques do this job quite rapidly.

An interactive system

A computer system that has the facilities described above is necessarily an exceedingly complex structure; it is no surprise that the average physicist is unwilling to divert his energies from performing experiments to understanding such a system. A system enabling a physicist to think in terms of his experiment has been implemented at Yale. It is based on an intermediate-size computer with which a user interacts via a terminal comprising a CRT display, light pen, typewriter and function keyboard. The communications terminal is pictured in figure 2. The keyboard consists of an array of buttons that produce an external interruption in the computer and a code that uniquely identifies the particular button depressed. The programming system allows one to connect any Fortran or assembly-language program to any of these buttons by a simple control card. When a button is pressed, the system responds by locating and executing the associated program. The physicist views the system as an assemblage of functions that can be invoked at execution time in any sequence and for any reason. He must, therefore, only learn the operation's characteristics as the result of a key depression; he need not be concerned with the nature of the program or the system supporting it.

The graphic terminal is also important for deciding strategic alternatives during an experiment. With a light pen, which singles out information, the terminal is an effective medium for a physicist to exert his judgment and experience.

Nuclear-physics experiments are often exploratory, and, therefore, one would like data acquisition, display and control to be readily changeable at execution time without reprogramming (or advance provision of alternate decision paths). For this reason all data-acquisition statements (in the extended Fortran previously mentioned) are programmed as two-part instructions. An *action* statement performs a particular function (for example, pulse-height analysis or scaling); its associated *specification* statement determines the operation's parameters (for example, number of channels in pulse-height analysis and overflow limit for the scaler). Since specification statements are accessible during execution from the function keyboard, CRT and typewriter, a parameter change does not involve the user with the revision mechanism.

Data arrays in the Yale system are self-defining in that each is associated with additional information, called an "attribute table," that specifies parameters used in its creation. This means that the user can refer to any data (or variable, for that matter) in an entirely symbolic way; so-called "global-symbol linkages" give automatic information about referenced data. For instance, if a spectrum named "E1E2" were acquired, the user could produce a contour display by the statement, CALL CONTOR (E1E2) or the corresponding function-keyboard operation.

The display mentioned above is pro-

```
ANALYZER: ANG1 (1024); ANG2 (1024)
LOGICAL SWITCH/.TRUE./
INTEGER TYPER/4/, SCALER (9),
  DATAWD
CREATE: ANG1; ANG2
DO 5 I = 1, 9
5 SCALER (I) = 0
WRITE (TYPER, 9)
9 FORMAT ('HOW MANY MCAS
  AND SCALERS? (211)')
READ (TYPER, 10) NANALS, NSCLRS
10 FORMAT (2 I 1)
ENTER EVENT MODE

EVENT 9: START COUNTING, CLEAR
  SCLRS BY READING THEM
SWITCH = .FALSE.
EVENT END

EVENT 10: READ SCALERS, STOP
  COUNTING, PRINT OUT
SWITCH = .TRUE.
DO 20 I = 1, NSCLRS
20 SCALER (I) = DATAWD (I)
CALL DISPLY (NANALS)
CALL OUTPUT (NANALS, NSCLRS,
  SCALER)
EVENT END

EVENT 1
IF (SWITCH) GO TO 102
PHA: ANG1 (1)
102 EVENT END

EVENT 2
IF (SWITCH) GO TO 202
202 EVENT END

EVENT 16: DISPLAY UPDATE
  (EXT. TIMER)
CALL DISPLY (NANALS)
EVENT END
END
```

COMPUTER PROGRAM for data acquisition at Yale. Events 9 and 10 read a scaler bank; pulse-height analysis occurs between them. 1 and 2 are pulse-height-analysis events, each corresponding to a separate spectrum. The display can be connected to event 16. —Fig. 3

duced according to a standard set of specifications. Once created, a display can be manipulated, combined with others, stored for future reference and deleted with a powerful set of display-manipulation programs that operate on the attribute table attached (automatically by the system) to each list of display coördinates. Operations are independent of the type of display and are *not* included in coding user-written display programs.

The global facility also permits programs independent of reference-data storage; that is, it is immaterial whether the data is in the core memory, a disk or elsewhere. For example, one can define multichannel analyzers whose data areas in core memory contain either 4, 2 or 1 "bytes" (a group of eight bits) per channel (count capacities $2^{31}-1$, $2^{15}-1$, and $2^{7}-1$, respectively). In the latter two cases the system provides a built-in overflow-counting mechanism wherein two bytes of overflows per channel are stored on a disk file. All references to data in an analyzer are independent of type and always retrieve the true number of counts. The user can emphasize speed (4 bytes per channel in core memory) or core economy by changing only his analyzer-specification statement.

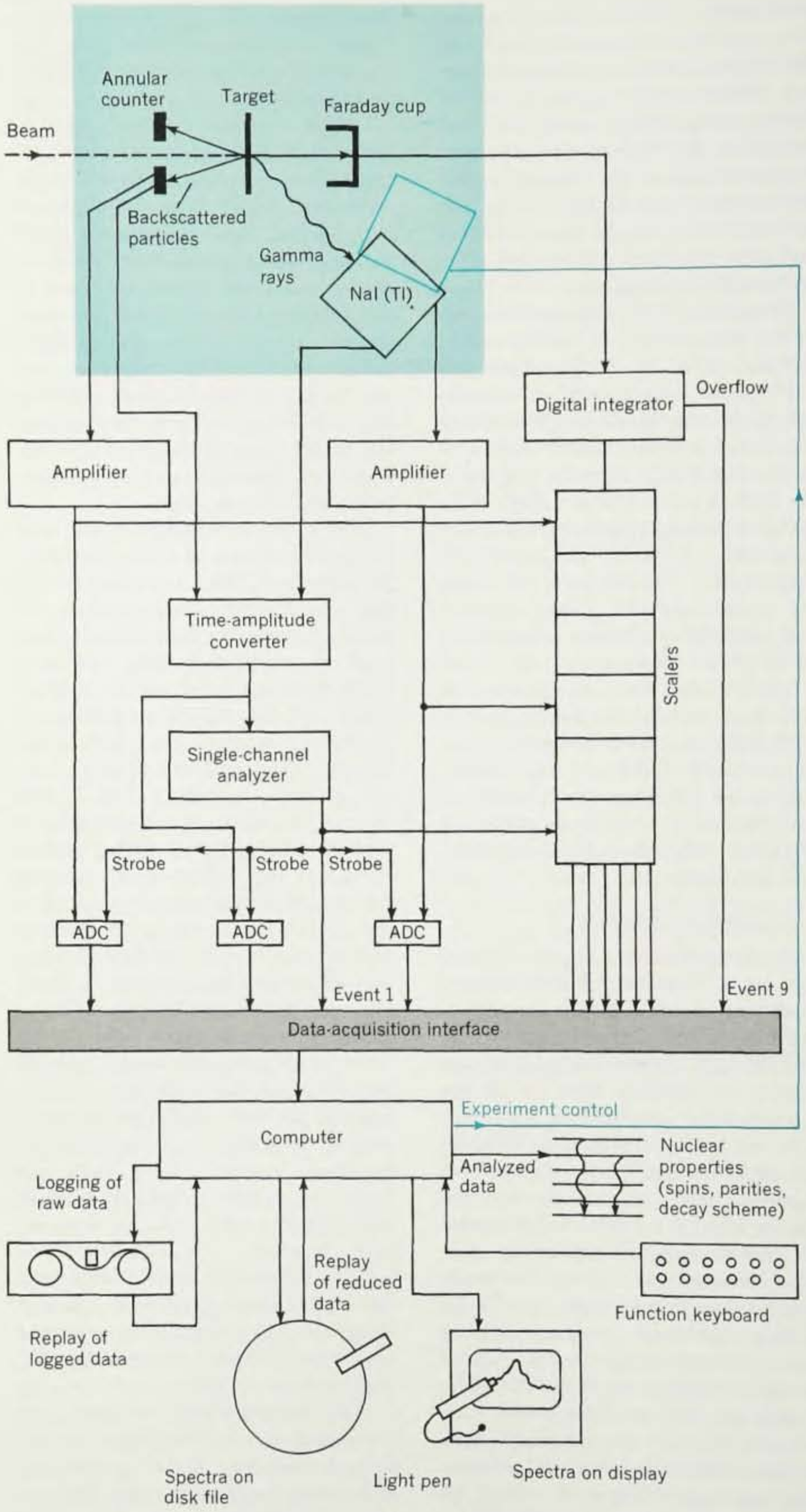
Yale system in operation

Several experimenters employ the system as a sophisticated multichannel analyzer, recording spectra from up to eight detectors, integrating peaks with the light-pen facility and putting out spectra on magnetic tape for off-line input to other data-analysis programs. The basic data-acquisition program for such an experiment is shown in figure 3. This program assumes that events 1 and 2 are pulse-height analysis events, each corresponding to a separate spectrum. It can be easily expanded to record more spectra by adding additional analyzers, EVENT and PHA statements. A scaler bank is read by both events 9 and 10. The scalers are read into the memory by event 9, but they are not used. This method effectively clears them since they are reset when read. Until an event 9 occurs, no pulse-height analysis takes place. From event 9 to event 10, pulse-height analysis occurs. Values read from scalers by event 10 are counts accumulated since the last event 9. One can update a display periodically by connecting an oscillator to event 16. The function keyboard selects the analyzers being displayed and performs the usual display-control functions.

Unlike a normal Fortran program, these statements are not executed sequentially. The ENTER EVENT MODE statement, among other things, transmits starting locations of each event routine to the system's buffer-scanning routine and then turns control over to this routine. At that point the data channel starts reading the nuclear-data interface, and the buffer-scan routine begins scanning buffers, identifying each event and transferring control to the appropriate event routine. The EVENT END statement returns control to the buffer scanner.

The system is used more elaborately in particle–gamma angular-correlation experiments. These experiments, figure 4, involve measurements of gamma-ray spectra from various reactions in coincidence with back-scattered beam particles. Both sodium-iodide and lithium-drifted-germanium detectors measure the gamma-ray spectra. The real-time data-acquisition program records a 128 × 128 channel, two-dimensional spectrum of particle energy plotted against gamma energy in the sodium-iodide counter for true coincidence events, as well as the particle spectrum in coincidence with all gamma rays (sodium iodide), the sodium-iodide spectrum in coincidence with all particles, the lithium-drifted-germanium spectrum (at a different angle) in coincidence with all particles, and the pulse-height spectrum of the time-amplitude converter used as the particle–sodium-iodide coincidence circuit. Raw data are logged on magnetic tape, in 1024 or 4096 channel resolution, for later replay. In addition the particle–sodium-iodide accidental coincidences and the particle–lithium-drifted germanium true and accidental coincidences are logged (without real-time processing) for later analysis.

With the same program these various events can be reanalyzed in any desired resolution in the system's REPLAY mode by changing only the analyzer specifications and the EVENT statement for the coincidence event (to specify the desired one of the four coincidence events). Effective time resolution may be varied during replays by setting conditions on the co-



PARTICLE–GAMMA CORRELATION EXPERIMENT. Colored panel at top includes laboratory equipment: beam, target and detectors. Central region is the data-acquisition system. The six scalers are representative of a larger number. Below the data-acquisition interface is the computer, programmed as in figure 3, and its associated storage, printout and display software. Colored line from computer is a feedback link that controls the motor driving the crystal detector in angle. —FIG. 4

incidence interval for events to be processed.

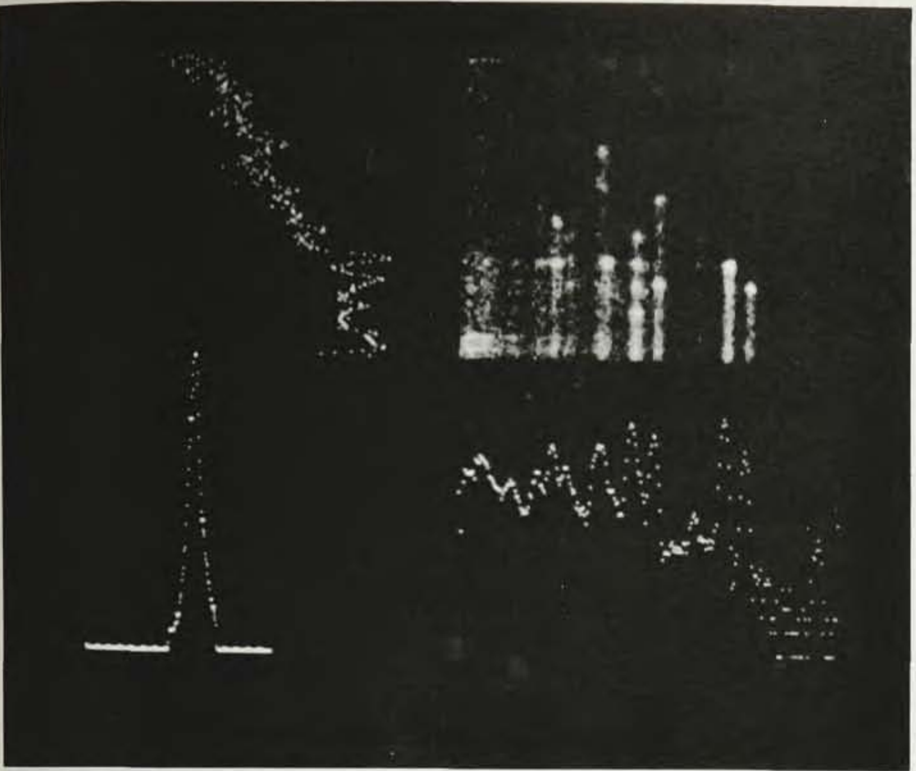
Figure 5 is an example of the type of display in this experiment. This particular picture is for Mg^{26} (α, pγ) Al^{29}. At the upper right is the coincidence spectrum, plotted as a contour map, with particle energy to the right, gamma energy plotted upward, and counts represented by intensity. At the lower right is the particle spectrum in coincidence with all gamma rays, and at the upper left is the gamma spectrum in coincidence with all particles. The latter two are actually at higher resolution than the contour display since they were obtained by separate pulse-height analysis statements rather than by summing in the coincidence spectrum. These two total-coincidence spectra, lined up with the corresponding loci in the contour display, are of considerable assistance in identifying the various groups. At the lower left is the spectrum of the time-amplitude converter. A keyboard function allows display of any one-dimensional spectra in this corner of the screen without disturbing the rest of the display.

Another keyboard routine enables the examination of planes through the two-dimensional spectrum and generates the gamma spectrum in coincidence with any particle group—or the particle spectrum in coincidence with any gamma peak—with a light pen for selecting regions of interest. These spectra may be printed out or integrated with a light pen. Thus one can generate correlations for as many gamma lines as desired while the experiment is in progress.

With multiprogramming one can carry simultaneous analysis of data considerably further since the analysis is totally independent of incoming-data reduction.

The goniometer has been designed for remote digital control, and all of its degrees of freedom are under computer control.

Another experiment, which studies compound-nucleus formation in radiative-capture reactions, uses a computer to perform control and stabilization functions as well as data acquisition and analysis. Basically, gallium-phosphide light-emitting diodes, whose output has been matched to the spectral response of a 10 × 12-in. sodium-iodide crystal, provide precisely calibrated light pulses for a photomulti-



QUADRANT DISPLAY for particle–gamma angular-correlation experiment. Portions of display can be manipulated independently. See text for identification. —FIG. 5

plier while the experiment is in progress. The light-pulse spectrum is accumulated separately from gamma data (by utilizing a separate event input and event routine) and compared with the initial spectrum by using mathematical fitting procedures to detect both gain shifts and broadening. Utilizing the shift of reference peaks the computer regulates feedback of a correcting voltage to the photomultiplier. Reference-peak width is a measure of pileup in the counting system and optimizes counting rate. Automatic logging of accelerator and experiment parameters are other computer functions. Since more than 1000 four-minute runs are performed during an experiment, the advantages of computer-based information storage and retrieval are quite significant.

Future developments

Computer-based experimentation can take several directions. The most obvious replace or augment existing pieces of laboratory apparatus. For example, a time-amplitude converter used as a fast-coincidence circuit enables one to modify effectively the coincidence-resolving time when reanalyzing data by setting conditions on the coincidence interval for acceptable events. Similarly a computer, instead of analog circuitry, can directly calculate particle-identification information from measured parameters (E and dE/dX or time of flight). Besides saving cost of equipment, a computer allows more sophisticated functional relationships than are practical with hardware identifiers. For pulse-height stabilization a computer determines the presence of gain shifts, without special hardware, and then applies correction either by sending signals to external hardware or by remapping the spectrum according to the new gain. Remapping the spectrum avoids potential resolution losses associated with variable-gain elements. In either case, a computer need not follow a linear correction procedure but can perform its corrections according to any previously determined response function for the entire system.

One can also expect computer-based experimentation to grow in the direction of new techniques providing higher degrees of precision than would otherwise be possible. For instance, Cyril Broude and his coworkers at the Chalk River Laboratories[3] use a position-sensitive semiconductor detector to determine not only the energy of detected particles, but also their scattering angle. With this information a computer calculates corrections on an event-by-event basis for energy variation with angle. Broude has found that near 180 deg, where this variation is substantial, he can achieve significantly better energy resolution.

The digital computer also makes feasible experiments that would not otherwise be possible because of the enormous amount of data to be sorted in real time. An example of such an experiment is one in which hodoscope arrays of semiconductor or scintillation counters are used. Associative-memory techniques allow real-time sorting of multiparameter data in which the number of required channels is far greater than available memory space but in which large regions of the multiparameter space contain no counts. In this scheme memory space is occupied only by analyzer channels containing nonzero information. Each cell contains both a descriptor (channel number) and its associated number of counts. Dynamic storage allocation with the associative memory algorithm enables one to use storage for other purposes until it is needed to enlarge the analyzer, an important consideration in a multiuser environment.

These are only a few examples of the ways in which one can use computers to produce a significant improvement in the quality of data and in the information conveyed to the experimenter. As new developments in both programming and hardware become more and more common, they may indeed revolutionize the physics done with them.

* * *

This work was supported in part by the US Atomic Energy Commission.

References

1. *Proceedings of the Conference on the Utilization of Multiparameter Analyzers in Nuclear Physics*, L. Lidofsky, Ed. Columbia U., CU(PNPL)-227 (1962); *Proceedings of the Conference on Automatic Acquisition and Reduction of Nuclear Data,* K. H. Beckurts *et al.*, Eds., Gesellschaft für Kernforschung m.b.H. Karlsruhe (1964); *IEEE Trans. Nucl. Sci.* **NS-15**, 93 (1968).
2. H. L. Gelernter *et al.*, Nucl. Inst. Meth. **54,** 77 (1967); IBM Tech. Note TN 21.575-18; J. Birnbaum, H. L. Gelernter, in *IEEE Trans. Nucl. Sci.* **NS-15,** 109 (1968).
3. C. Broude, R. W. Ollerhead, Nucl. Inst. Meth. **41,** 135 (1968). □