JOURNAL OF COMPUTATIONAL PHYSICS 1, 305–321 (1967)

The Use of an On-Line Digital Computer in Closed-Loop High-Energy Physics Experiments[1]

PAUL FRIEDL, CHARLES SEDERHOLM, AND WILLIAM DYE
IBM Corporation,
Scientific Center, Palo Alto, California 94304

AND

HALL CRANNELL AND M. R. YEARIAN
High-Energy Physics Laboratory,
Stanford University, Stanford, California 94305

ABSTRACT

Since April 1965, IBM and the W. W. Hansen High-Energy Physics Laboratory at Stanford University have been engaged in a joint study of use of on-line computers in high-energy physics experiments.

In the High-Energy Laboratory, electrons with energies up to 1.2 BeV, produced by the Stanford Mark III 310-ft linear accelerator, are allowed to bombard various target materials. Electrons, or secondary particles produced by the electrons, are detected with several different kinds of detectors. Information from the detectors is relayed to the data-acquisition area where it is processed and stored, while control information is fed back to nuclear instrumentation by the computer. This paper not only contains a description of the computer and instrument interfaces, but also the application-oriented programming approaches used to operate these devices in various experiments. Several different experiments using the data-acquisition computer system are described, and plans for future modifications and improvements are discussed.

I. INTRODUCTION

The pursuit of research in modern experimental nuclear physics has been accompanied by the development and use of very special equipment such as high-energy nuclear particle accelerators and high-performance nuclear detection and measurement instrumentation. The development of this modern apparatus has made it possible for the nuclear physicist to perform higly complex experiments

---

[1] This work was supported in part by the U.S. Office of Naval Research. Contract NONR 225 (67).

which produce large amounts of experimental data. Because progress in nuclear physics is highly dependent upon the rate at which the nuclear physicist is able to collect, store, retrieve, and analyze these data, the application of digital computers to these tasks has become a subject of increasing importance.

In order to investigate the application of digital computers in nuclear physics research, a joint study project was initiated between International Business Machines Corporation (IBM) and the High-Energy Physics Laboratory (HEPL) of Stanford University. The scope of the project includes the monitoring and control of the Mark III linear electron accelerator [1]; closed-loop control of experimental apparatus; the acquisition, monitoring, reduction, and display of data obtained in high-energy physics experiments; as well as the development of special problem-oriented programming languages to simplify the application of digital computers to these tasks.

Many control systems employing on-line computers in physics experiments are being developed [2]–[6]. The PHYLIS system [2] at Argonne and the NBS system [3] at the National Bureau of Standards are examples of software and hardware oriented control systems respectively. These control systems are designed to enable the physicist to conveniently control the execution of desired programs during the course of his experiment. The approach presented in this paper does not address the control problem *per se*, but rather the task of specifying the desired operations by means of a problem-oriented language.

The present paper describes a meta-linguistic programming approach which is oriented to solving the programming problems faced by a physicist in conducting on-line computer-assisted experiments. This approach is at present based on conventional macros, but will be extended to include other facilities such as "immediate assemble and execute" and "display-aided trace and debug" features (similar to those described in the XPOP system [7]).

II. FACILITIES

A. COMPUTER

The IBM Data Acquisition and Control System consists of the configuration of equipment shown in Fig. 1.

This solid-state system has a data path which is 18-data-bits wide plus parity, a core storage with 2-μsec cycle time, and a capacity of 49,152 words.

The processing unit has an extended arithmetic register (AC-MQ) and three index registers. Its instruction set includes fixed-point arithmetic, Boolean logic, compare and branch, interrupt control, and execute operations. Each instruction requires two words: one specifies the operation, arithemtic register, index register, and addressing mode (direct, normal, or indirect), while the other specifies the address. Instruction execution times range from 4 to 24 μsec with an average of 3–8 μsec.

The three data channels may operate simultaneously with the processor in an overlapped mode (core cycle stealing) to a maximum word rate of 500 Kc.

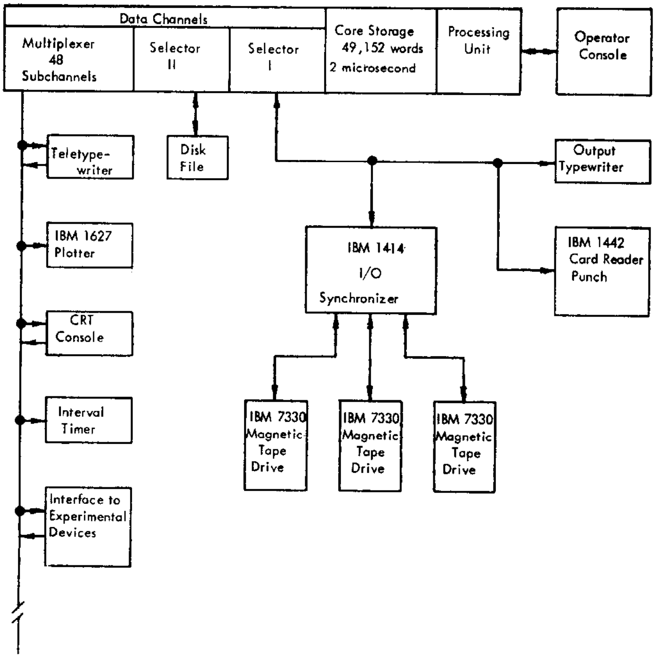

FIG. 1. Experimental computer system.

Conventional I/O devices are attached to the two selector channels. Three magnetic tape drives, a card reader/punch, and an output typewriter are on selector Channel I, and a disk file is on selector Channel II. The multiplexor channel has facilities for simultaneous operation of 48 subchannels, some of which have been implemented. These subchannels provide data paths for instrumentation and man-machine interaction (teletype, CRT display, etc.) as described in Section IV.

The programming system provides the framework for system operation — utility programs, a job monitor, a symbolic assembler, relocatable and absolute loaders, and a library of fixed- and floating-point mathematical subroutines.

B. ACCELERATOR

The main component of the Mark III accelerator is a radio-frequency waveguide operating in the $\frac{3}{2}\pi$ mode. The accelerator consists of 31 identical 10-ft sections. The power to each section is supplied by a high-power, four-cavity klystron. The electron energies range from 100 MeV to 1.2 BeV. The acceleration of electrons takes place in pulses of about 1-μsec duration. A maximum of 2×10^{11} electrons per pulse can be generated with a pulse rate of 60 Hz. In practice this means that, while particle detection and certain preliminary analysis must be accomplished in only 1 μsec, there is a relatively long time (16 000 μsec) between accelerator pulses in which to accomplish the data storage and analysis.

C. EXPERIMENTAL EQUIPMENT

The computer has been used, at least in part, in three different experiments. These experiments are described in more detail in Section III. In principle, however, all the experiments are similar. The beam of high-energy electrons is allowed to strike a target. The electrons scatter from the target material and in the process produce secondary particles. The experiments consist of detecting the electrons and/or the secondary particles. In addition, the momentum and angular distribution of the detected particle is often determined. Information concerning the number, kind, angle, charge, and momentum of the particles is then fed from the target area to the data-acquisition area. In addition to this information concerning the detected particles, other data such as the incident electron energy, the energy spread in the beam, and the number of incident electrons is relayed to the data acquisition area. The data acquisition area is located at some distance from the target area. Almost all equipment that requires adjustment is controlled from the data-acquisition area. In the past, most of the information from the target area was registered on scalers, recorded laboriously by hand, and necessary adjustments in the equipment were made by the experimenter. The following sections contain descriptions of interfaces and software which allow automatic recording and reduction of the data, as well as automatic adjustment of the necessary experimental parameters.

III. EXPERIMENTS

As mentioned in the previous section, the data acquisition system has been employed in three different experiments. The first use of the computer was to achieve very fast recording of data which had been stored in a two-parameter, multichannel analyzer. The 4096 words of information recorded in the analyzer were rapidly transferred to the core of the computer, releasing the analyzer for further recording. The information stored in the computer was then recorded on magnetic tape for further analysis.

The second experiment is a continuing effort in which electrons scattered from targets are analyzed in momentum and angle [8]. An array of 100 detectors placed in the image plane of a momentum spectrometer is used to obtain the desired information [9]. The data resulting from one accelerator pulse may be transferred directly to the computer or may be accumulated in an auxiliary core which is transferred on command to the computer. At present, additional experimental parameters, such as the incident electron energy and the scattered electron angle are entered into the computer via the teletypewriter. Work is now in progress to make the recording of these additional parameters automatic.

In these experiments a large amount of data is accumulated in a relatively short time. In the past it has taken several days or even weeks before the results of the measurements are known. At times apparatus failures have gone undetected until the data was finally analyzed. In order to reduce these difficulties, on-line analysis has been implemented.

During an experiment the data may be analyzed in any of several different ways by setting the appropriate switch at the console. Using pertinent data, the efficiency of the different detectors can be calculated [10]. The data from each of the detectors can be corrected for various experimental effects, and corrected results presented either as a typed output and/or as a plot. Another option allows presentation of a list of relevant calculated experimental parameters. The raw data is written on tape in a format which is compatible with the requirements for further analysis by existing IBM 7090 data-analysis programs. As an additional feature, any data already on tape can be recalled and reanalzyed or compared with current data. These features provide rapid monitoring of the progress of the experiment, thereby allowing man–machine interaction which helps maximize useful information during running time of the accelerator.

A third experiment, referred to as the closed-loop experiment, is a study of the development of high-energy electron-induced cascade showers in different materials [11]–[13]. A beam of high-energy electrons is allowed to impinge upon a large block of material. A small, energy-sensitive probe is inserted into the material and the energy deposited at the location of the probe is measured. By moving the probe radially (perpendicular to the direction of the beam) and longitudinally (along the direction of the beam), the distribution of energy deposited in the entire block is mapped.

A block diagram of the experimental apparatus is shown in Fig. 2. The probe gives a signal proportional to the energy deposited at a given point in the material. The monitor gives a signal proportional to the intensity of the beam. Variable-gain amplifiers are needed to analyze the large range of signals that are obtained from the probe and the monitor during the experiment. A motor and associated readout system drives the probe in and out of the material. The computer has been interfaced with each of these pieces of equipment and programmed to measure a radial energy deposition curve.

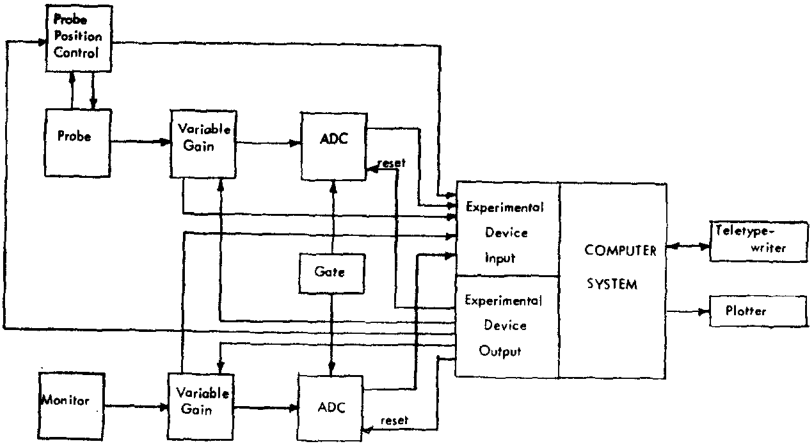

FIG. 2. Block diagram for closed-loop experiment.

The important experimental quantity is the ratio of the probe to monitor signals. After each beam pulse these signals are digitized and fed to the computer, the ratio of the signals is computed and appropriate action taken.

At the beginning of each measurement the computer positions the probe at the center of the material and steers the beam of electrons at right angles to the motion of the probe to center the beam. The probe is then withdrawn in steps. The step size is computed by considering the rate of change of the energy deposition. This logic is shown in Fig. 3. At each step, the ratio of probe to monitor signals is determined, and usually an average of 60 pulses is used to ensure good statistics. The raw data and calculated ratio are recorded on tape and the log of the ratio as a function of the probe position is plotted. When the signal from the probe or the monitor falls outside a predetermined range, the gain of the appropriate amplifier is changed. When it becomes necessary to change a parameter that is not under computer control (e.g., beam intenisty), an appropriate message is typed.

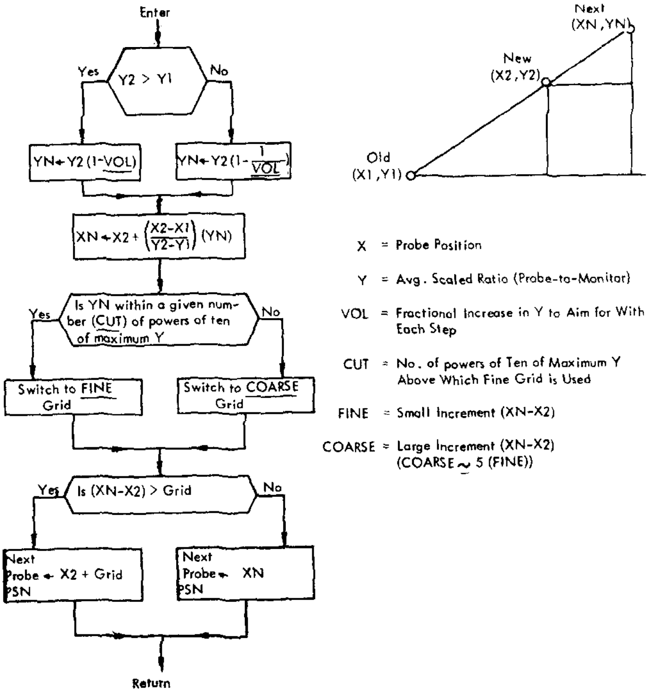

FIG. 3. Logical flow: Subroutine B—compute new probe position.

Preliminary runs with this equipment have shown that radial distribution can be determined in 5–10 min, which compares very favorably with the 30–60 min using older methods. An additional advantage is that the data is analyzed instantly, whereas before the job of analysis often took several months.

IV. I/O IMPLEMENTATION

The experiments described above required the interfacing of numerous external devices to the computer. Three types of data paths are available. Two selector channels communicate with all of those external devices which require high-speed transmission of blocks of data, as mentioned in Section II. A multiplexor channel handles data from several subchannels. Each of these subchannels is capable of addressing sequentially a block of core, one word at a time, even though successive words of data entering the multiplexor do not come from a single subchannel. Almost all of the attachments of experimental equipment to the computer are made through this multiplexor channel. This allows several different devices to be communicating with the computer in an overlapped mode, without intervention of the central processing unit (CPU). Finally, a command-and-sense bus offers a third method of communication with external devices. This data path is used when one bit of information needs to be passed between any device and the CPU.

The multiplexor channel presently has seven subchannels in active use, consisting of one 18-bit parallel digital input subchannel; two 18-bit parallel digital output subchannels; a plotter subchannel; one input and one output teletypewriter subchannel; and an interval-timer subchannel.

One digital output subchannel is interfaced to a CRT display console. The image on the CRT is regenerated from a buffer within main core. The system is capable of maintaining 3072 characters of text or 511 plotted points or a mix of the two. No more than 0.8% of the core time is used to maintain this display.

One digital output and one digital input subchannel are used for all experimental devices. At present, the only device on the digital output subchannel is a set-point controller which positions the probe for the closed-loop experiment. A single word is presented through the digital output subchannel to the external set-point controller which positions the probe. A sense instruction is used to determine when the probe has reached the required position.

A digital input subchannel is used to gather data from a large number of devices within the counting room. Two of these devices, the multichannel analyzer and the array of 100 detectors, transmit entire blocks of data and are directly addressed by the subchannel. The array transmits a block of the addresses of all of those detectors which detected radiation during the last beam pulse. An interrupt is sent to the computer when this variable-length block of data has been transferred from the array. Many other devices in the experimental area present only one word of digital data. Because the number of these devices exceeds the addressing capabilities of the subchannel, a digital multiplexor has been attached to this subchannel. Command instructions from the CPU control this multiplexor. The data presently multiplexed into this subchannel consists of (1) the present electron-shower-probe position, (2) two analog-signal-gain settings which are also used in the closed-loop experiment, (3) data from a pair of coincident analog-to-digital converters (ADC's), (4) a large number of digital readouts indicating various parameters of the experiment. With one selection of the digital input subchannel, all or any fraction of the experimental data and parameters can be read into the associated core buffer in any order.

The following command operations have been implemented: (1) multiplexing data as mentioned above, (2) resetting the ADC's, (3) controlling the gain of either of the two analog amplifiers used in the closed-loop experiment, (4) stepping of controls for beam steering. Three sense operations have been implemented: (1) ADC's conversion complete? (2) Probe positioning complete? (3) Digital input subchannel storage completed? The command operation is implemented as a single machine instruction. This means that the command is completed before the programming proceeds. The sense instruction is implemented as a skip-on-condition machine instruction which implies that the answer is returned before the program continues.

The address portion of both the command and sense instructions is partially decoded and a 40-bit data bus is presented. Over 100 separate external devices may be attached. Two additional output lines (command and interrogate) and one input line (true) are associated with these instructions. Each device, acting on these instructions, continuously monitors the data-bus address. A particular device is conditional whenever it detects any address associated with it. Upon receiving a command pulse, an operation is performed. If an interrogate pulse occurs, the addressed device returns a pulse on the true line if the state interrogated is true, and does nothing if the state is false. A pulse on the true line following an interrogate pulse causes the computer to skip the next instruction.

V. PROGRAMMING APPROACH

An assembly language was all that existed upon installation of the system. A programming language having the following characteristics was determined to be highly desirable:

(a) The user need have very little knowledge of machine language or of the inner workings of the computer.

(b) It must be simple to use, providing ease of coding, ease of debugging program logic, and ease of checking out instrumentation links.

(c) It should provide the ability to handle the major tasks involved with a closed-loop experiment: data acquisition, storage, movement, analysis, and display; control of peripheral computer devices and experimental instrumentation; entry of parametric and descriptive information and control of logical program flow during the course of the experiment.

Because of the above requirements, it was clear that some form of high-level language such as FORTRAN or ALGOL should be developed along with an appropriate set of special-purpose subroutines or procedures.[2] The high-level language would satisfy requirement (a) and the first two parts of (b). The ability to assemble a library of user-defined subroutines answers the task requirements listed in (c). It should be particularly noted that checkout of experimental interface linkages is facilitated by the use of machine-language subroutines. During debugging operations they provide the closest possible man-equipment-program interaction.

A considerable effort would have been needed to develop a FORTRAN or ALGOL compiler. Therefore, it was decided to attack the programming-language problem by giving the symbolic assembly program the added power of system and user-defined macros. (A macro is a single line of coding, closely resembling symbolic coding, which results in the generation of more than one computer instruction.) A set of macros was developed which answers nearly all of the previously stated requirements: (a) little knowledge of the interior of the computer (e.g., macros do not require knowledge of an accumulator or index register); (b) ease of use (in actual practice, the macros are as convenient to use as FORTRAN with the exception of coding a long string of mathematical operations); (c) ability to handle major tasks (the macros may be used to call closed subroutines and, in addition, to provide an inline or open subroutine capability which is efficient for short routines).

The set of macros was specifically designed to perform the data acquisition and control tasks needed for the closed-loop experiment discussed in Section III. A set of systems-oriented macros was written for basic arithmetic, logic, data movement, and operation of the computer I/O equipment such as tapes, plotter and teletype. Then, the set was expanded to include specific application macros (e.g., reading ADC's plotting coordinate axes, moving an experimental probe). In many cases this second set utilized macros defined in the first set. This set of special-purpose macros is being continually expanded to support additional activities.

Space limitations make it impractical to include within this paper all those macros presently in the system. It is hoped that the following presentation of examples of the major types will be sufficient to indicate the salient features of the method.

I/O OPERATIONS–CHANNEL A

MRD	CRD$, 27, INPUT	Read a card in BCD mode and transfer 27 words into core starting at symbolic address "INPUT."

---

[2] A programming approach based on the FORTRAN compiler has been presented by Kane [14], [15].

MWB	A3$, 350, OUTBUF	Write a 350-word binary-tape record on tape drive 3 starting from symbolic core address "OUTBUF."
BEOT	ENDT, WR	Branch to location "ENDT" if the macro at location "WR" caused an end-of-tape indication to occur.
Etc.		

CONVERSION

FLOAT	XVAR, FVAR	Convert the integer in location "XVAR" to a single precision floating-point number located at "FVAR."
FXIN	BUF, 3, VAR	Convert a BCD integer appearing anywhere within a 3-word field starting at location "BUF" to a binary integer located at "VAR."
Etc.		

MATHEMATICAL OPERATIONS

XADD	A, B, C	Add two integers located at "A" and "B" and put the result in location "C."
FDV	A, B, C	Perform a floating-point divide of the quantity located at "A" by that located at "B" and place the result in location "C."
FSIN	THETA, SINT	Compute the sine of the floating point variable located at "THETA" and place floating point result in location "SINT."
Etc.		

DATA MANIPULATION

INC	PO, 1	Increment the integer at "PO" by 1.
MOVEX	V, (3), W, K, 2	With two vectors *V*, *W*, move a block of two words beginning with the third element of *V* into vector *W* beginning with the *K*th element of *W*.
ZERO	A, N	Set the "N" words starting at location "A" to zero.
Etc.		

PROGRAM SWITCH STATEMENTS

FLAG	EXIT, 'S'	Set ('S') a program flag called "EXIT."

TFL	EXIT, OUT	Transfer to location "OUT" if flag called "EXIT" is set.
IF	VI, (16), A, B, C	Transfer to location "A" if the integer at "V1" is less than 16, to location "B" if equal to 16, and to location "C" if greater than 16.
GOTO	A, B, C, D, POINT	Transfer to locations A, B, C, or D if the integer located at "POINT" is 0, 1, 2, or 3, respectively.
REPEAT	LOC, (5), I, K	Repeat all statements (starting at location "LOC") 5 times, each time incrementing the integer at "I" by the integer at "K".

SPECIAL-PURPOSE STATEMENTS

TBLU	TABLE, LN, A	Look up the *N*th floating-point number in a table starting at location "TABLE" and place it in location "A." ("LN" = location of *N*).
MAXM	6, V, POINT	Find the maximum value in a 6-element vector of integers starting at location "V" and store in location "POINT" an integer which points to it.

TELETYPE I/O

TMSK	AMSG, 'A', 'C', 'X', T1, T2, T3, T4	Type the message starting in location "AMSG" and wait for a response. If the response is the depression of teletype key 'A', transfer to location "T1"; if key 'C', to location "T2"; if 'X', to "T3"; and if any other key is depressed, to "T4."

Etc.

PLOTTER OUTPUT

AXES	15., 10., 1.	Plot an x axis of 15 inches, and a y axis of 10 inches with "tic" marks every 1 inch.
PLOTPT	XX, YY. .05	Plot a cross (+) of radius .05 inches at x, y coordinates located at "XX" and "YY", repectively.

Etc.

INSTRUMENTATION CONTROL

ADCR	PROBE, MONITOR	Read the dual analog/digital converter and place the pair of binary integers in locations "PROBE" and "MONITOR."
PPOS	NEW	Position the probe to a value stored in location "NEW."

Etc.

Figure 4 shows a portion of the listing for the data reading program which deals with part of the subroutine A. This portion is enclosed by the dotted line in Fig. 5. The only non-macro statement required in this section is the transfer statement (TRA) for which the basic symbolic assembly-language name is sufficient.

```
        IF      X,TOT,Z1,C,C
Z1      ADCR    PROBE,MONIT
        INC     X,1
        FLAG    BADATA,'R'
        IF      PROBE,(1024),Y3,Z2,Z2
Y3      IF      PROBE,(0),Z2,Z2,Z3
Z3      IF      PROBE,(128),Z4,Z5,Z5
Z2      INC     PO,1
        TRA     SETBDF
Z4      INC     PU,1
SETBDF  FLAG    BADATA, 'S'
Z5      IF      MONIT, (1024), Y7,Z6,Z6
Y7      IF      MONIT,(0),Z6,Z6,Z7
Z6      INC     MO,1
        TRA     STBDF
Z7      IF      MONIT, (128),Z8,A,A
Z8      INC     MU,1
STBDF   FLAG    BADATA, 'S'
A       TFL     BADATA,B
        INC     GPT,1
        FLOAT   PROBE, FLPRO
        FLOAT   MONIT,FLMON
        FAD     FLTOTP,FLPRO,FLTOTP
        FAD     FLTOTM,FLMON,FLTOTM
        FDV     FLPRO,FLMON,WORD1
        FAD     TOTRAT,WORD1,TOTRAT
```

FIG. 4. Data-read program.

Figure 6 shows a complete logical flow for the closed-loop experiment. The ease with which logical choice boxes shown there are implemented is worthy of note. The TMSK macro was designed to allow decisions to be made by the experimenter at program execution time. This is a good example of a man–machine

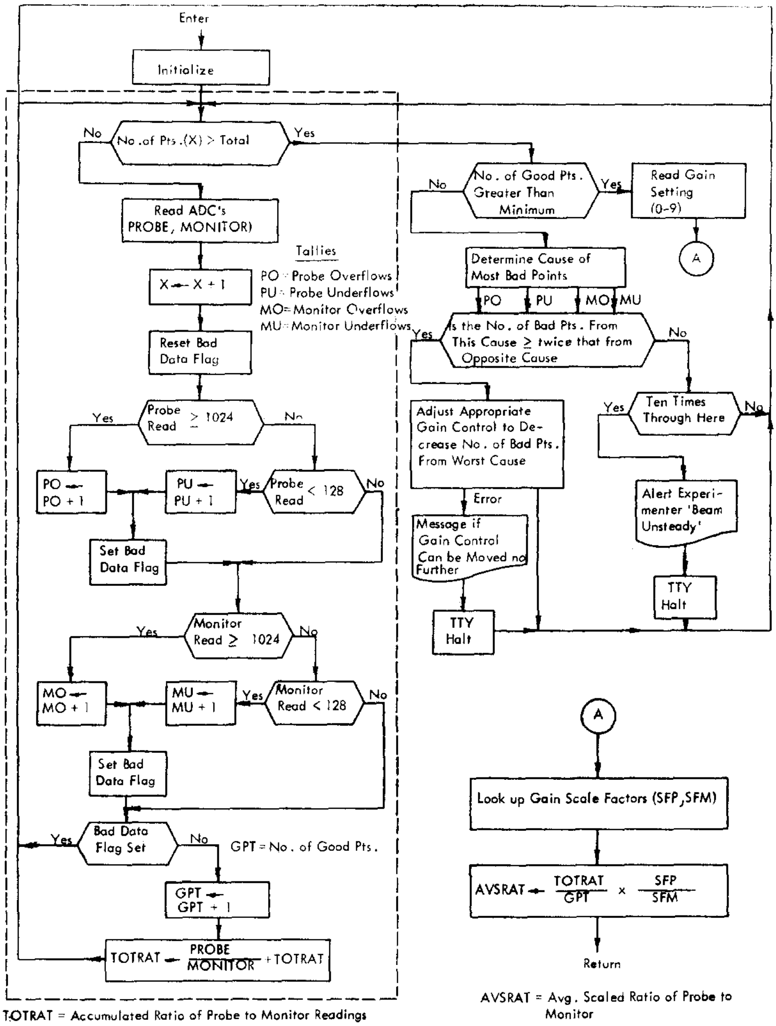

FIG. 5. Logical flow: Subroutine A—Data Reader.

macro which is as easy to program as it is convenient to execute (execution being simply the depression of an appropriate key followed by a carriage return on the teletypewriter).

An estimate of programming ease when using the macros is that the entire main program (700 machine instructions) was coded in about four hours, and contained only two logic errors as first conceived. It was found convenient to write the ma-

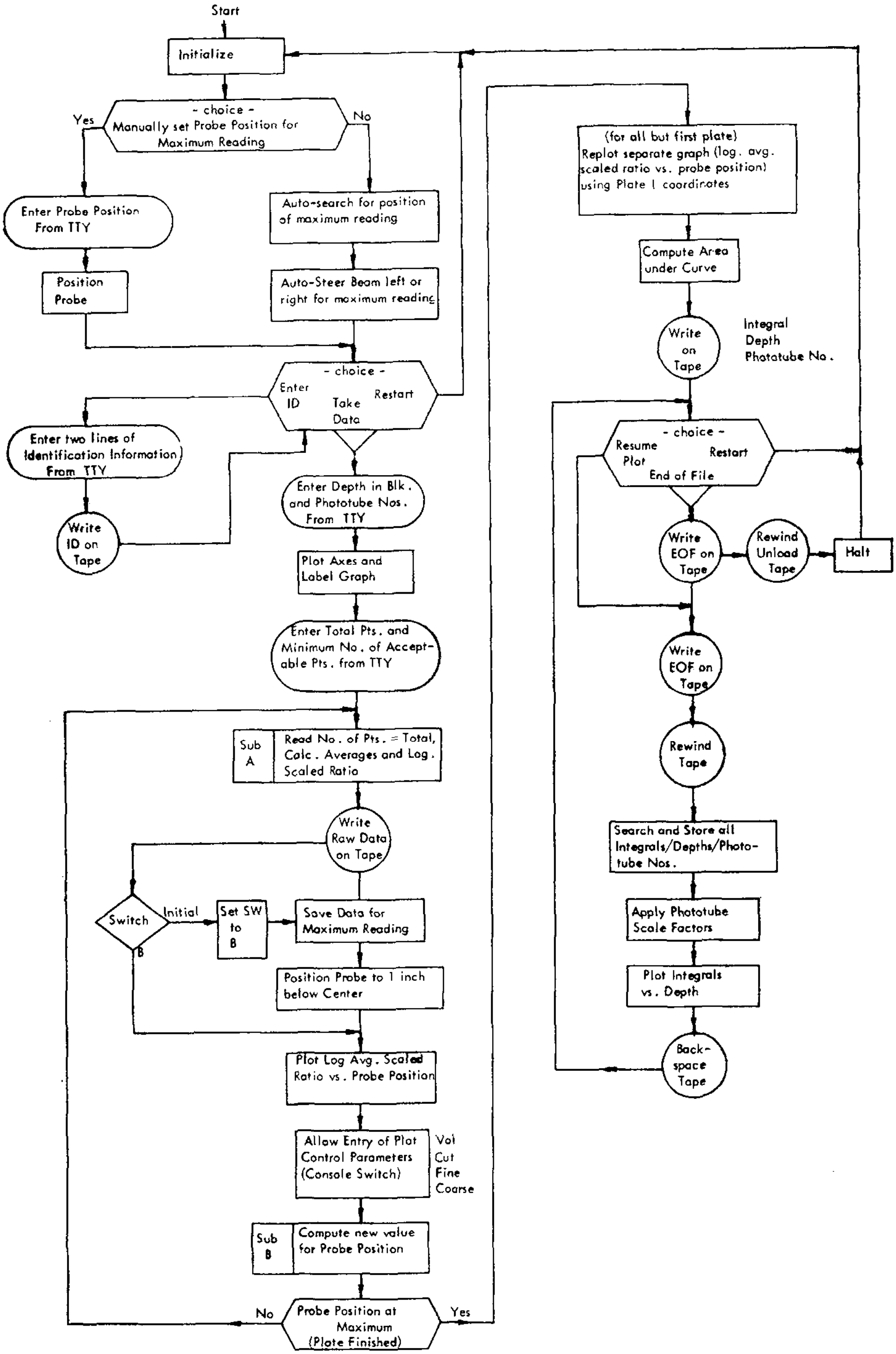

FIG. 6. Overall logical flow: closed-loop experiment.

cro program directly from a word description of the problem. Also, logical debugging of macro programs has been found to be no more difficult than with FORTRAN programs.

VI. FUTURE APPLICATIONS

The plans for this system fall into two general categories: (a) implementation of additional experiments; (b) development of programming systems providing easier man–machine interaction, both in program writing and in real-time control of experimental runs.

Various experimental parameters, controls, and detectors are being interfaced to the computer on a continuing basis. Likewise, additional application programs are being written for the system. A spark chamber is presently being constructed employing direct readout into the computer. Programs are being developed to identify events in real time. In the near future, experiments involving more complicated manipulation of experimental parameters and more sophisticated correction of data will be run using the array of 100 detectors. In particular, it is hoped to make pulse-by-pulse corrections to the data from the array based upon the intensity of each individual pulse.

Development of the CRT display as an operations console for the entire experiment is planned. By the installation of a large number of sense switches, it is expected that experiments can be totally controlled from that console.

Additional programs are being written for data manipulation and display at the console. This will not only allow the presentation of data now being gathered, but will also allow the recall of previously taken data.

The macro assembler and resident system are being modified to accommodate multitasking. In this mode of operation, numerous tasks can be operating simultaneously on a priority basis. Each task execution is overlapped with I/O channel operations of other tasks. These tasks are coordinated by appropriate macros.

In addition, programs are being written which will make the CRT display console a useful tool in the development and debugging of programs. The CRT will be capable of displaying a listing of object programs which have been generated. These programs may be modified from the display console, with immediate updating. Programs so generated may be assembled and executed from the display console. Current display or alteration of variables in programs being executed will be possible. Programs will be executable in a trace mode; i.e., execution will take place at full speed down to a point in the object program which has been indicated by the light pen. At this point, execution will pause, allowing the user to display or alter variables. Such capabilities will be of great help in the debugging of new programs, and in the development of new experimental interfaces.

ACKNOWLEDGMENTS

We wish to thank many persons for their assistance in this work. Support for the programming systems has been given by H. Goettsche, S. Graham, K. Marks, and R. Ryneveld. Much of the interfacing hardware has been developed by S. Berezin, L. Birkwood, C. Crannell, L. Hodges, K. Klotz, E. McGuire, D. Smith, L. Suelzle, and W. Wadensweiler.

REFERENCES

1. M. CHODOROW, E. L. GINZTON, W. W. HANSEN, R. L. KYL, R. B. NEAL, AND W. K. H. PANOFSKY. *Rev. Sci Inst.* **26**. 134 (1955).
2. W. F. MILLER, M. A. FISHERKELLER, AND K. HILLSTROM, in "Proceedings of EANDC Conference on Automatic Acquisition and Reduction of Nuclear Data" (July 13–16, 1964, at Karlsruhe), p. 475. Gesellschaft für Kenforschung, Karlsruhe (1964).
3. H. L. DAVIS, *Nucleonics* **22**, 57 (1964).
4. J. V. Kane, *Phys. Today* **19**, 59 (1966).
5. G. W. TAUTFEST, *Phys. Today*, **19**, 69 (1966).
6. S. J. LINDENBAUM, *Phys. Today* **18**, 19 (1965).
7. M. I. HALPERN, "Proceedings–Fall Joint Computer Conference." Spartan Books, Washington, D.C. (1964).
8. R. HOFSTADTER, *Ann. Rev. Nucl. Sc.* **7**, 231 (1957).
9. L. R. SUELZLE and M. R. YEARIAN, *Nucleon Structure* (R. Hofstadter and L. I. Schiff, Eds.) (Proceedings of the International Conference at Stanford University, June 24–27, 1963). Stanford University Press, Stanford, California (1964).
10. H. L. CRANNELL and L. R. SUELZLE, *Nucl. Int. Methods* **44**, 139 (1966).
11. A. D. KANTZ and R. HOFSTADTER, *Nucleonics* **12**, 36 (1954).
12. C. CRANNELL, *Bull. Am. Phys. Soc., Ser. II* **11**, 374 (1966).
13. C. D. BUCHANAN, H. COLLARD, C. CRANNELL, R. FROSCH, T. A. GRIFFY, R. HOFSTADTER, E. B. HUGHES, G. K. NÖLDEKE, R. J. OAKES, K. J. VAN OOSTRUM, R. E. RAND, L. SUELZLE, M. R. YEARIAN, B. C. CLARK AND R. HERMAN "Procedings of the International Conference on High-Energy Physics, Hamburg."
14. J. V. KANE, *Nucl. Instr. Methods* **31**, 83 (1964).
15. J. V. KANE, *in* Ref. [8] p. 500.