

## THE ON-LINE CONTROL, ANALYSIS AND DISPLAY SYSTEM FOR THE SLAC MULTI-GeV/c SPECTROMETERS\*

A. Boyarski

Stanford Linear Accelerator Center  
Stanford University, Stanford, California

### INTRODUCTION

The on-line experimental system I will describe here has been in operation just over two years now, and has resulted in publication of over ten papers to date in the field of high energy physics. All these experiments have used magnetic spectrometers with counters to measure the angle and momentum of particles scattered or produced from a target which is bombarded by high-energy electrons or photons. The on-line computer system consists of an SDS 9300 with 32 K of core, an extensive set of peripherals, software including an assembler, a real time FORTRAN IV compiler, a disk monitor and a multiplexed interface system able to communicate with over 300 devices each having up to 24 bits of information. The computer is dedicated to one experiment at a time, and the large number of communication channels allows permanent connection of devices from all spectrometers and simplifies the switch-over from one experiment to another. The major change between experiments is in the software programs.

The two-mile-long accelerator at the Stanford Linear Accelerator Center (SLAC) provides a high-energy beam of electrons which may be deflected into any of several areas located at the end of the accelerator. The accelerator produces 360 beam bursts or pulses per second, with a duration of about  $1.5 \mu\text{sec}$  per pulse. The beam can be deflected from one area to another by pulsed magnets, so that several areas may receive some of the beam pulses on a time-sequencing basis. End Station A, one of the larger experimental areas, contains three large spectrometers used in studying processes such as electron scattering off nuclei and photoproduction of unstable particles in gamma-nucleon collisions. The spectrometers measure the momenta and production angles of particles emerging from a target. Figure 1 shows the arrangement of the three spectrometers in End Station A. The spectrometers pivot about a common point. The largest is about 150 feet in length and has eleven separate magnets. Figure 2 is a photograph of the spectrometers.

The computer controls the magnet currents for the spectrometer in use, reads and monitors the status of various hardware devices, records onto tape both the status data and the event-by-event data of particles passing through the spectrometer, analyzes on-line as much of the data as time permits, and displays results during the data-taking interval. At the end of a data-taking run (usually from  $\sim 1$  to  $\sim 100$  minutes) results complete with cross sections can be printed out on the line printer. The ability to get the "final answer" immediately allows one to detect any error in the total system in a minimum of time. By plotting on-line data points with error bars, one can make decisions in an exploratory type of experiment, for example, as to the angle or energy at which subsequent data should be taken. In some instances the on-line results are good enough for publication, but more often the final corrections are better made by re-analyzing the data tape off-line. The off-line analysis handles all the events, whereas

---

\* Work supported by the U.S. Atomic Energy Commission.

in the on-line mode the fraction of events analyzed varies from  $\sim 50\%$  to  $100\%$ , depending on the rate of the incoming data. The off-line analysis is usually done on a central 360/91 computer since the 9300 is busy with the on-line operation.

The conditions at the spectrometer facility are such that normally only one of the three spectrometers is in operation at a time over intervals which last from a few days to a few weeks. Some parasiting may be done with another spectrometer, but usually beam conditions are sufficiently incompatible that good physics data cannot be taken in parasite mode. Hence only one experiment is conducted at a time, and the computer is dedicated to that experiment. No computer time-sharing is done. Program debugging is done during change-over between experiments (from  $\sim 3$  to  $\sim 10$  days) and during normal, scheduled accelerator shutdowns, which last a few days every month. This relatively small time that is available for debugging is sufficient because generally all the spectrometer experiments are somewhat similar as far as data acquisition is concerned, so that complete reprogramming is not necessary. The analysis programs are different, however, and so this is where most of the debugging effort is spent for new experiments. The use of FORTRAN IV facilitates writing and testing these analysis subroutines in a relatively short period of time. The same analysis subroutines are used off-line, which provides another means of checking out the analysis programs whenever a data tape is available. Data such as scalers, events, beam monitor data, etc., is recorded on-line with data type identification labels. The off-line routine reads the data tape, and from the data type identification the data are stored in appropriate labeled COMMON locations in memory (as it is on-line) so that the same analysis routines operate off-line as they do in the on-line mode. This provides a desirable similarity between the on-line and the off-line analysis with minimal differences in the overall programs of the two systems.

The electron (or photon) beam enters the end station and strikes a target. Scattered or produced particles enter the spectrometer, pass through a magnet system into a particle detector apparatus. In this case, four bend magnets, four quadrupoles, and three sextupoles focus the charged particles onto the P- and the  $\theta$ -focal planes in the vertical and horizontal directions, respectively. By placing scintillation counters (41 P counters and 32  $\theta$  counters), the momentum and production angle of a particle are measured. Figure 3 shows the arrangement of the particle detectors. Other hodoscopes (X and F) further determine the angles of the particle at the P and  $\theta$  foci, such angles being useful in determining if the particle might have scattered from magnet pole faces and also for limiting events to acceptable entrance angles into the gas differential counter C. A threshold gas Cerenkov counter G together with C is used for particle identification ( $\pi$ , K, protons). Signals from the Cerenkov counters feed into pulse-height analyzers, as does also the signal from a sandwiched lead-scintillator shower counter S used for electron identification. The latter counter provides a large pulse height for electrons or positrons, but usually small pulse heights for other charged particles. A range telescope R detects muons by their penetrating depth through the iron absorber. By using information from the C, G, S and R devices, the particle which is momentum and angle analyzed by the spectrometer may also be identified as an  $e^\pm$ ,  $\mu^\pm$ ,  $\pi^\pm$ ,  $K^\pm$ , etc. In addition to event-by-event data, the computer reads 40 scalers, and the integrated beam current given by voltages across charging capacitors. Other status data, such as the angle of the spectrometer, target type (hydrogen, no target, deuterium, etc.) position of slits within the spectrometer, pressures and temperatures of the gas in the Cerenkov counters, alignment of the spectrometer magnets with respect to a taut alignment wire, etc., are read and logged by the computer.

A rectangular-shaped counting house serves as the experimental control area for all the spectrometers. The spectrometer and beam-steering controls are located along one wall while the fast electronics associated with the hodoscope and particle identification counters occupy two other walls. The computer and peripherals take up the remaining end wall and part of the center of the counting house. Figure 4 shows the arrangement of the counting house. Cables are strung under the raised floor level. This arrangement is convenient to operate during experiments, where the experimenter has all the controls and displays surrounding him. It is not so convenient for the debugging

phases, since the number of people involved in hardware checking or modification tends to interfere with the tranquillity desirable for the programmer who is debugging on the computer.

### THE MULTIPLEXER INTERFACE

The SDS 9300 computer can input or output a 24-bit word of memory from a 24-line parallel input (PIN) or to a 24-line parallel output (POT) terminal. In order to send or receive data from many different devices, a multiplexer is necessary to route data to the desired device. We have built such a multiplexer at SLAC using SDS logic cards. It handles 24 lines in parallel (input or output) to match the computer word size, and is constructed in a modular form to allow for expansion and ease of installation. At present there are over 300 device terminals. A simplified schematic of the total multiplexer system is shown in Fig. 5. The various inputs and subsections will be described in the following.

In order to communicate with any desired device, the computer first executes an energize output mode or EOM instruction with an address of the device. This electrically selects the device data lines to the computer input/output lines. A subsequent instruction may then be executed to send/receive 24 bits of data in parallel to/from the device. On the 9300 it takes  $7 \mu\text{sec}$  to make one reading from a device and  $8.75 \mu\text{sec}$  to send out data to a device. If the device cannot respond at these speeds, it may insure proper communication by delaying (i. e., hanging up the computer execution of the PIN or POT) the completion signal that must normally be returned on another line of the multiplexer.

The multiplexer system is constructed of several modules, or multiplexer units (MUX), each of which can handle up to 31 devices at 24 data lines per device. We presently have 12 such multiplexer units, identified by MUX  $\alpha$ , where  $\alpha$  is the unit number ( $0 \leq \alpha \leq 11$ ). Figure 6 shows one of these units. These units are stationed at various locations around the counting room. The 31 possible terminals of each MUX are identified by  $\beta$  ( $1 \leq \beta \leq 31$ ). Hence any device has associated with it a MUX number  $\alpha$  and a device number  $\beta$ . When the 9300 issues an EOM instruction, the whole 24 bits of the instruction word (including its address part which contains the  $\alpha$  and  $\beta$ ) appear at the POT terminal of the 9300. The bits of a computer word are numbered 0 to 23 from left to right, according to the SDS convention, and the instruction code (the systems EOM) itself occupies bits 3 to 11. Decoding logic in the external device control (EDC) recognizes the EOM and picks out  $\alpha$  from bits 14 to 18, and  $\beta$  from bits 19 to 23 of the output POT lines. Also, a mode code, described below, is given by bits 0 to 2. As an example, the binary instruction in memory (and also on the output POT lines) to select MUX 1, device 5 with mode 1 would look like:

Bit No.	0	1	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	23			
	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>0</u>	<u>1</u>
	Mode			EOM (system)									$\alpha = 1$				$\beta = 5$						
	(MUX NO.)											(DEVICE NO.)											

This is symbolically written MPX  $\alpha, \beta, \text{MODE}$ . The mode bits indicate the kind of communication to be done as shown in Table 1.

The 1-2-4-8 BCD to BIN conversion is done by a hardware conversion device built into the multiplexer system. A complete conversion of 24 bits of BCD information into the binary equivalent is done in  $10 \mu\text{sec}$ . The SKS to follow mode simply sets a flip-flop to the state of the ready line from the selected device, so that a subsequent instruction may test the flip-flop state to determine if a device is ready.

Table 1  
Multiplexer I/O Mode

MODE BITS	DESCRIPTION OF I/O	
0 0 0	BIN-PIN	read binary into memory.
0 0 1	BIN-POT	send binary out of memory.
0 1 0	BCD-PIN	convert 1-2-4-8 BCD to binary, and read the binary into memory.
0 1 1	presently not used	(reserved for BCD-POT)
1 0 0	BIN-PIN-RESET	do BIN-PIN, then send reset pulse to device.
1 0 1	SKS to FOLLOW	select device and test for READY in next instruction (on SKS).
1 1 0	BCD-PIN-RESET	do BCD-PIN, then send reset pulse to device.
1 1 1	presently not used	

The two instructions, for example,

MPX  $\alpha, \beta, 0$   
PIN W

will read data from device  $\alpha, \beta$  into memory location W, and the instructions

MPX  $\alpha, \beta, 1$   
POT W

will send out data to device  $\alpha, \beta$  from memory location W.

The total interface consists of several multiplexer units. The outputs (or inputs) of these MUX units are eventually connected (via OR circuits) to the input or output registers of the external data control section. The EDC has an  $\alpha$  decoder which does the selecting of the proper MUX unit into the EDC. A schematic of a PIN MUX unit, which can handle 31 devices, is shown in Fig. 7. It takes two racks of SDS cards, consisting of diode cards and line drivers. If not all 31 terminals are used, some cards may be removed for reasons of economy. Burndy type M534 connectors are used at the terminals. Each terminal has the 24 data lines, a sign bit for BCD devices, an end of data transmission line (DATS), a line for sensing for device ready (EDSR), a reset line, and the 5 lines for the device code  $\beta$ . The  $\beta$  decoder is built within each MUX unit.

#### Manual Control Panel

It is very convenient during initial hardware checkout, or when problems occur during an experiment, to be able to manually select any device and observe the data from the device, or send out data to a device. A manual control panel, situated on top of the 9300 console as shown in Fig. 8, was constructed for this purpose. The device address  $\alpha, \beta$  and the MODE may be selected by push buttons (with indicators) on the address register and the device data will appear on the PIN register indicators for an input mode, or data may be inserted into the POT register buttons (with indicators) for sending out data. In addition, if a BCD-BINARY conversion is to be done, both the BCD and the BINARY appear on the panel. One master select button allows these registers to be operated either manually, or via the computer. If the latter is in use, the push buttons are locked out so that data transmission is protected against inadvertent button pushing. The manual control panel required construction of three registers to hold the

information for display purposes. The address register required 15 flip-flops while the PIN and POT registers each have 24 flip-flops.

Shown with the manual multiplexer control panel in Fig. 8 is the arrangement of push buttons (to the right of the console) used for manually producing an interrupt pulse to any of the 32 interrupt levels on the 9300. Each button also contains two indicator lights. One of these turns on if the interrupt level is armed (i. e., ready to respond to interrupt pulses) and the other comes on whenever an interrupt is waiting (i. e., an interrupt pulse has been received, but program execution has not been started or completed for that interrupt level). By manually depressing another control button, the active rather than the waiting states of interrupts are displayed (i. e., interrupt programs which have started execution). These indicators are especially useful during the debugging phases for programs in an interrupt environment. If, for example, a program halt occurs, the indicators make it possible to see the subroutines that were being executed on various interrupt levels.

### The Patch Panel

The interrupt inputs and sense line inputs to the 9300 appear at terminals of a patch panel. Pulse lines or logic levels of various devices are also connected to the patch panel. A user may then patch in any signal into any desired interrupt level or sense line according to the needs of the experiment. This makes it easy to change from one experimental program to another. A picture of the patch panel is shown in Fig. 9.

### Description of Devices

SDS logic operates at 0 and +8 volt-logic levels, with +8 being the true state. Inputs such as switch status or encoder positions simply need open or short the input lines to the PIN MUX to indicate true or false bits. Devices with flip-flop registers which are not within the 0 to 8 volt logic require level converters.

Most of the fast trigger and coincidence circuitry associated with the counter hodoscopes is made of commercially available plug-in units (Chronetics modules). This manufacturer also supplied the flip-flop buffer storage units (8 FF's per module) necessary to store the pulsed information from the hodoscope counters for subsequent reading by the computer and resetting after the read is completed. The fast circuitry itself was not disabled by the experimental stop-start control signal, but the event trigger interrupt signal to the computer and all scalers and pulse height analysers were gated by the control signal.

The forty scalers received by the computer are manufactured by Transistor Specialties Incorporated capable of 100 MHz operation. Each scaler has seven decades, but only the least significant six were read by the computer since six 1-2-4-8 BCD decades use the total 24 data lines available on the multiplexer system. Overflows from the 6<sup>th</sup> decade of each scaler set flags, and the setting of any flag causes an interrupt to the computer. The computer then reads and resets these flags and keeps track of overflows for all the scalers.

Eight pulse height analyzers, manufactured by RIDL, are used. Only the ADC portion are used. A considerable amount of interfacing was required for these units in order to read and reset them. Since most PHA's require microsecond long pulses, and the counters produce only ~ 10 nsec pulses, linear stretch circuits (Chronetics) are used, followed by amplifiers with a gain of 10.

Three digital voltmeters, manufactured by Dana, are accessible to the computer. Each has a ~ 200 element scanner to connect to 200 different voltage sources. In order for the computer to read a voltage from some device, a channel address is first issued via the POT multiplexer to the DVM, which sets up the DVM scanner and the read cycle for the DVM. Autorange is used normally. Since it takes .8 seconds or more to complete the reading, asynchronous programming can be used, and the program can

continue with other tasks until the interrupt from the DVM completion signal arrives. This interrupt program reads the voltage, together with scale range and returns the answer to the calling program with a flag parameter to indicate the completion of the reading. Items such as beam monitor data, gas pressures, slit positions and general voltage data are read via the DVM.

Approximately 25 nixie display units are situated around the control room. These are used for outputting diagnostic results from the computer, from simple things like a run number to the more complex such as the angle that Cerenkov light is emitted by a particle in one of the Cerenkov counters. This angle is calculated from the measured temperature and pressure of the gas in the Cerenkov counter and the known momentum and mass of the particle.

## THE ON-LINE PROGRAM SYSTEM

### Operating Characteristics

Before a new experiment is in operation, several weeks are spent writing or changing existing programs to meet the new needs of the experiment. A batch processing monitor with real time capabilities is used to compile, assemble and load binary subprograms, link edit and execute the job. During the debugging phase, it is convenient to store and update subprograms on a disk library. Each experiment has its own library name, and in the loading process, the experiment library is searched for subroutine first before proceeding to the general library which includes the Fortran subprograms. Routines useful for all experiments are put in the general library. In order to simulate experimental data, fake programs using random numbers are used. Also the computer can issue out programmed interrupts to simulate signals from the hardware. Analysis routines may also be debugged by reading data from a tape generated by a test run or a previous experiment. A few hours before the experiment starts, the final program is loaded, magnets are tested and set, and the computer is then dedicated to the experiment. Usually the core program is swapped out onto the disk, so that later it may be rapidly re-instated in memory if for example it is necessary to momentarily use the computer for another job.

The on-line system uses subprograms which may be written either in FORTRAN IV or the SDS META SYMBOL machine language. Those routines which are common to all experiments, such as data logging, are mostly written in the more efficient META SYMBOL form. Analysis routines or on-line routines which print messages on the teletype or line printer are much better written in FORTRAN. Of course one pays some overhead in more memory used by the Fortran subroutines, but the increased ease of coding is worth the loss of some memory. One of the nice features of SDS FORTRAN IV is the ability to write machine language statements intermixed with regular Fortran statements. This proves very useful for reading multiplexer data directly with Fortran coded programs and for doing bit manipulation logic. Since many different interrupt levels are used, both the monitor and the Fortran library routines must recognize re-entrant calling of subroutines from different interrupt levels. SDS supplies a real time Fortran IV system as well as a real time rapid access disk (RAD) monitor for the 9300. However, a SLAC RAD monitor system was written which uses considerably less core memory (4100 words rather than ~ 6500) and has an easy to use library editing system. In addition, the SDS formatting routines (used with READ or WRITE statements) were re-written with a reduction of about 2 K words of memory, but no loss in format capability.

The on-line program system consists of several major subsections, namely, data logging, experiment control such as starting and ending runs, magnet programs for setting the spectrometer momentum, the display executive system, equipment status checks, and the on-line analysis programs.

Extensive use of interrupts is made in the on-line program system. Table 2 gives a list of interrupts used during one of the experiments. Some interrupts, such as the event signal, must be honored immediately, while others, such as manually causing execution of a program which reads in new constants from a card reader, are used only because of the convenience of being able to randomly call a subroutine by pushing a button. The signals to the interrupts are routed via the patch panel.

Table 2  
List of Interrupts Used in the On-Line Program System  
(The highest priority level is 040; the lowest is 077. Not all 32 levels are used.)

Priority Level for Interrupt	Source of Interrupt	Description of Program Connected to the Interrupt
040		
041		
042		
043	event trigger	read in event data and store in memory.
044		
045		
046	scaler overflow	read in scaler overflow flags.
047		
050	program	log data buffer in memory onto tape (360 words).
051	DVM pulse	signal from end of DVM0 operation.
052	DVM pulse	signal from end of DVM1 operation.
053	DVM pulse	signal from end of DVM2 operation.
054	program	write a block of listing on the "microfilm" magnetic tape.
055	push button	read new bin efficiency data cards.
056	4 pps pulser	update a nixie display of accelerator beam settings.
057	push button	halt event sampling
060	50 pps pulser	magnet executive (adjusts power supplies at 50 steps/sec).
061	push button	print out latest Cerenkov angles for $\pi$ , K and proton.
062	60 pps pulser	set up data channel to display contents of a scope buffer on a scope (done 60/sec).
063	event trigger	analyze the data of latest event.
064	push button	read new data cuts from TY.
065	1 pps pulser	update magnet display (if display is selected) at 1 pps.
066	scaler preset	bump momentum by a small step (i.e., next minirun).
067	push button	abort display that is being updated.
070	push button or program	compute a new image into the scope buffer memory.
071	push button	read new scan limits from TY.
072	push button or program	start, reset, stop, etc., a run, according to the push button.
073	push button	read a setting for an individual magnet from TY.
074	program	completion of encoder movement on magnet supplies.
075	push button	read new momentum setting from TY.
076	program	completion of magnet adjustment.
077		

### Data Logging Program

Experimental data are logged on magnetic tape for future off-line analysis. As data are read from the hardware, they are stored into one of the two 360 word buffer areas in memory. Whenever new data can no longer fit into this array, the second array is used for data storage while the first array is written out onto tape. By using the buffers alternately for data storage, no data are lost in the logging process at rates up to 10 K words per second.

All types of data are logged by calling one subroutine ZLOG(I,N,DATA), where I is a data type number, N is the number of words, and DATA is the data array to be

logged. One logical block is written into the next available area in the memory buffer, with the first word containing both I and N followed by the N words of DATA. Since any interrupt level may call ZLOG, this program must be re-entrant. The number of logical data blocks in a full 360-word buffer is variable, depending on the sizes of the individual data blocks. Off-line, AZLOG, a complementary program to ZLOG, can be called to obtain the next logical data block from the tape, together with the values of I, N and DATA. At present, there are approximately 30 data types, some of which are listed below.

<u>Data</u>	<u>Number of Words</u>
Event (counters, bits, and pulse heights)	10
Scalers, with overflows and names	120
Constants, with names	151
Beam monitor data (voltages or scalers)	15
START, STOP, RESET, etc., messages	20 each
Magnet settings (voltages, encoders)	80
Comment from typewriter	18
Status data such as target number, slit settings, gas pressures in Cerenkov counters, etc.	--

Sufficient data are logged so that in the off-line analysis, final cross sections can be computed (aside from mistakes or final correction factors) from the data tape alone, without resort to the usual log book.

#### Experiment Control Program

A master RUN or STOP button controls the hardware gating logic which starts or stops event triggers and beam monitor operation. A computer interrupt is generated if any such control button is pushed, and if nothing else, a message together with the time of day is printed and logged to record which button was activated. A list of control buttons, and both their hardware and software effect within the program, are listed below. Each button sets a flip-flop flag which the computer can read and then reset. These buttons are shown in the bottom of Fig. 11.

<u>Control Button</u>	<u>Result</u>
RESET	Clear beam monitors, scalers and histograms.
RUN	Enable hardware, check if switch status O.K.
BEGIN RUN	Bump the run number, read target number, slit positions, angle, Cerenkov gas pressures, etc.; then do the RESET and RUN operations.
STOP	Disable hardware to stop run.
ENDRUN	Stop run if not already done, read values of scalers and beam monitors.
SETCONST	Enter a new value via typewriter into the constant table.
COMMENT	Read a comment from the typewriter.
DUMP	Print out specified histograms and final results.
NEW RUNNO	Enter a new run number from typewriter.
NEW TAPENO	Enter a new datalog tape number from typewriter.
ZEROSKLR	Reset the scalers, but only if ENDRUN has been done.
UNLOAD	Write EOF mark on tape and unload the tape; if ENDRUN was not done, do it first and then unload.



All these buttons are OR'd into one interrupt level. If another button is pushed while one is in operation, it is remembered and executed next in line.

From the beginning to the end of a run, the history of pushed buttons and various scaler and beam monitor data is printed as shown in Fig. 10a. At the end of a run various histograms may be printed out as well as a summary page (Fig. 10b), which shows all constants, cuts, scaler and event data and cross sections for various particle types.

### Magnet Control

There are fourteen power supplies available to operate magnets on the spectrometers. The most complicated spectrometer is the 20-BeV which requires eleven power supplies to operate it. Each power supply can be driven either manually or by a potentiometer remotely driven by a stepping encoder. The encoder can be stepped either up or down by a pulse issued by a POT multiplexer on an appropriate line. The position of the encoder can also be read by a PIN multiplexer. The actual current coming out of a power supply is measured by reading the voltage developed across a shunt which is placed in series with one of the output lines of the power supply.

In order to set the current to a desired value in one of the magnets, connected to a corresponding power supply, the corresponding voltage set point is stored in a memory location and certain bits are stored in a corresponding task flag word. An interrupt program, driven by a  $\sim 50$ -pps pulser, examines the task bits and performs the corresponding duties such as reading the shunt voltage, stepping the encoder or checking if the voltage has reached the set point voltage. If no task bits are on, the program exits very quickly. The task bits are turned off one by one each time a specified task is finished. If a DVM reading has been initiated, further readings are not attempted until the initiated one has been completed ( $\sim 0.8$  sec). When a new voltage set point is desired, the corresponding encoder set point is calculated (for small changes by extrapolating from the present voltage-encoder values or for large changes by using an approximately known encoder vs voltage relation) and the encoder is stepped at 50 pps until the encoder set point is reached. At this point the shunt voltage is read and compared with the desired value. If out of tolerance, new iterations of the encoder set point are made until the measured voltage meets the desired set point (within 0.1 mV or about 0.1% of the setting). The same program can step all power supplies together, although the voltage readings have to be done sequentially since there is only one DVM. It takes about 2 minutes to set all 11 supplies from zero to the maximum momentum setting.

The spectrometers have also been operated in a scan mode, where many short runs (miniruns) are taken with only a small change in momentum between miniruns. An interrupt generated every time a beam monitor increases by a specified amount, stops the run, computes the next momentum and recontinues the run as soon as the encoders have settled ( $\sim 0.5$  sec). The voltmeter then makes its time-consuming readings of the magnet shunts while the run is in progress. This mode of operation not only allows for a larger momentum bite than that given by a single setting ( $\sim 4\%$  in  $\Delta p/p$ ), but also smooths out bumps in the missing mass distribution due to non-uniform bin efficiencies in the momentum counters. Calculations of the beam flux together with counter deadtimes, momentum-dependent corrections and known counter efficiencies are accumulated in a large array for all the miniruns. This together with the histogram of counts from all miniruns gives a bin-by-bin cross section for the whole scan.

### The Display Executive Program

The analysis and display programs are apt to change from one experiment to another and even during one experiment. It is desirable to keep the programming simple in concept so that if a new scope display is to be included, nothing need be changed in the other displays already in use. Also, it must be easy to call any display by simply setting a switch (or possibly typing a name). Superimposing displays is also useful, and provisions for outputting data on command onto the line printer or other device should

be possible. The following hardware and software features were developed to satisfy the above needs.

The software display system builds an array or file of display programs in the following manner. Any routine which creates data to be displayed at any future time (e.g., an analysis routine which produces a histogram of counts, or a routine which makes an array of shunt voltages for magnets), makes a call to a filing program, thus:

CALL FILE (K, SUB).

The FILE program saves the display code number K and the corresponding display subroutine name SUB into a file table, whenever a call is made with values of K and SUB which have not already been filed. A call to the subroutine SUB would execute code which creates a display image into a scope buffer located in core memory (1000 words). An interrupt program, driven at 60 pps, then outputs the contents of the scope buffer to the oscilloscope (via a data channel) to produce the display. The subroutine SUB is called whenever the corresponding code number K is first selected on a display control panel, or if FILE is called again by the analysis routine while the display is already on the scope, or if any of the buttons on the display panel are changed. This allows the latest information to be updated, if new data arrives or a new display is desired.

A display control panel, shown in the top of Fig. 11 provides a means of selecting any display stored in the display file. The panel is divided into six rows, where each row contains an ON/OFF select switch, an 8-octal digit thumbwheel, plus three other select switches for specification to histogram displays, namely, manual or automatic scale factors, linear or log scales and presence or absence of axis.

An interrupt program, driven at 10 pps, examines the status of all 6 sets of buttons. If any select button is on, a display code number is read from the thumbwheels (left 3 digits) and if this number exists in the display file, the corresponding display subroutine is called; if not then a "MISSING PROGRAM" indicator is lit. The data from the rest of the thumbwheel digits, and status of the other buttons, are stored in a labeled common area, and are available as parameters to the display program. Up to 6 displays may be so selected and superimposed on the scope face.

A row of 10 buttons below the panel allows any display program selected by the above panel to be operated in a different way (e.g., clear a histogram, print display on-line printer, etc.) whenever a particular one of the buttons is pushed. The number of the button that is pushed (OPUSWTCH) is stored in a labelled common area of memory, and any display program can use this variable to decide what specific action is to be taken. An ordered convention has been established for the buttons as follows:

CLEAR; LINE PRINT; PLOT; PUNCH; CRIS-CROS  
(change line direction on isometric displays); ISOMETRIC  
ANGLES (change by teletype TY); manual SCALE FACTORS  
(enter on TY); CUTS (enter on TY) and SLICE (project 2-D  
display onto a 1-D display).

If no button is selected, a scope operation is performed.

General purpose display subroutines have been written to generate one and two dimensional histogram displays. The routine OPUS1 can display a one-dimensional histogram on the scope as shown in Fig. 12, or print out on the line printer as in Fig. 13, clear the histogram, change manual scale factors, etc., according to the value of the variable OPUSWTCH. OPUS1 also has the option of displaying error bars (proportional to  $\sqrt{\text{Number of events}}$ ) on the data points. A similar routine OPUS2 provides the same options for 2-dimensional histograms. The arguments for these routines are the histogram functions for the 2 (or 3) axis, number of points, a title, and an array which contains four general-purpose accumulation registers and scale factors. If the SLICE button is set, then OPUS2 will project the 2-D histogram into one or the other axis,

within slicing bin limits as specified by settings on the thumbwheel digits on the display panel. Isometric angles may be changed to examine the display from the best direction. Some examples of 2-D isometric scope displays are shown in Fig. 14 and a typical line printer output is shown in Fig. 15.

Finally, by calling another scope executive program, all display programs in the file may be called in turn irregardless of the thumbwheels. By setting OPUSWTCH = 1, this feature is useful for clearing all histograms and software scalers in memory at the beginning of a run, without requiring the knowledge of which analysis routines are actually loaded.

### Equipment Status Checks

In an automated on-line system, it becomes desirable to build in system checks wherever possible. These checks are especially important since the human being tends to over-look details the more automated the system becomes. Consistency in the final answers tends to become the criterion for working equipment, and flaws which produce only small effects may go unnoticed for some time.

At the beginning of each run then, the following items are read and checked automatically by the computer.

1. Check that spectrometer magnet power supplies are set to within tolerance.
2. Check special patterns on special test inputs to the multiplexer interface.
3. Check crucial switch positions, where available.
4. Check Cerenkov gas pressures.
5. Check if special clearing magnets, etc., are on.

Any malfunctions appear as a message on the teletype. Also, during a run, or at the end of a run, known relationships such as ratios between the integrated flux from various beam monitors, or internal consistency checks of the data may be printed out. Important check data such as distribution of counts in all hodoscopes and pulse height distributions are always available on a display, and are printed out at the end of a run.

During the initial phases of an experiment, special diagnostic routines are loaded. For example, the complete operation of all counters can be seen by a "one event" display as shown in Fig. 16. Counters which are struck by a particle are intensified, and pulse height channel numbers are shown.

### On-Line Analysis Program

The amount of on-line analysis that can be done depends on the amount of core memory that is available. Each analysis routine is a separate subroutine, so that changing the analysis system generally requires changing only certain subroutine calls in one master analysis program. A sample analysis routine which builds a histogram and allows it to be displayed is shown below. Each time SORTP is called, an event is added to the histogram HIST, while a call to ACTP causes a display to be generated or the histogram to be cleared depending on the value of OPUSWTCH.

```
SUBROUTINE SORTP
C  ENTER HERE TO ACCUMULATE P HISTOGRAM
COMMON/EVCODES/IEV(30)
C  IEV(1) CONTAINS THE BIN NUMBER OF P HODOSCOPE
```

```

EXTERNAL ACTP
INTEGER HIST(40), TITLE(2)
DATA TITLE/1, IHP/
REAL Q(8)

CALL FILE (1, ACTP)
Q(1) = Q(1) + 1.
I = IEV(1)
IF(I.LT.1.OR.I.GT.40) RETURN
Q(2) = Q(2) + 1.
HIST(I) = HIST(I) + 1
RETURN
C
C
SUBROUTINE ACTP
C THIS ENTRY POINT IS FOR DISPLAYING, CLEARING, ETC.

X(I) = I
Y(I) = HIST(I)/Q(2)*40.
C THESE STATEMENT FUNCTION ARE USED FOR THE X AND Y AXIS.
  NOTICE Y IS NORMALIZED TO TOTAL COUNTS IN THE HISTOGRAM.

CALL OPUS1(HIST, 40, X, Y, TITLE, Q)
C THIS RESULTS IN A DISPLAY WITH BOTH X AND Y AXIS,
C THE POINTS X(I), Y(I) FOR I = 1, 40, A
C TITLE 'P', AND THE VALUES OF Q(1), Q(2), Q(3), Q(4).
RETURN
END

```

With 32 K of memory on the 9300, the monitor, display executive, logging programs, control and magnet programs left room for the following analysis subprograms in a recent photoproduction experiment.

1. P (momentum) hodoscope distribution.
2. T (theta) hodoscope distribution.
3. X hodoscope distribution.
4. F hodoscope distribution.
5. PT (2-dimensional) P vs T hodoscope distribution.
6. SR (shower pulse height vs range hodoscope).
7. Eight correlated pulse height distributions from 3 PHA's connected to gas Cerenkov counters.
8. A 2-dimensional Cerenkov PHA spectrum.
9. A missing-mass display for one spectrometer setting.
10. A missing-mass display for a combination of many settings (100 bins  $\times$  3 particle types).
11. A list of accumulated counts for various particle types.
12. A list of constants with names for the run.
13. Status of magnets on the spectrometer.

Some overlaying was done for subprograms that do not need rapid interrupt servicing. In addition, at the end of a run, the on-line core load could be swapped with another analysis program stored on the disk to do further analysis such as least squares fits of models to the missing-mass data. It takes about 1 sec to swap core loads. At the end of the least squares fit, answers are printed out and the on-line program is swapped back into memory. Data communication between the two core loads is via a magnetic tape or disk, where certain summarized data are automatically stored at the end of every run. The logging tape (raw data) is never touched, however.

### Memory Map

The following table gives the size of major program sections as used in a recent photoproduction experiment. The first seven items are written in assembly language, while the remainder are mainly Fortran.

<u>Program Subsection</u>	<u>Words of Memory</u>
Resident RAD monitor	4100
Core swapping executive	300
Pushdown lists (variable)	≤ 900
Microfilm output routine	1200
Data logging	1100
Display executive	600
Display buffer	1000
1-D and 2-D display subroutines	3100
Magnets	1500
Control	1500
DVM	300
Beam monitors	700
Scalers	300
Plotter	500
Event analysis (not including histograms)	6300
Event histograms and weights	3000
Summary output (onto tape)	400
Fortran subroutines — formatting	2500
— other	1100
Miscellaneous	<u>3000</u>
Total	33400
Overlay total	3000
Overlay area	900
Maximum program size in memory	31300
Total external references	= 300 (25 are labelled common)
Time to load and link edit the full program	= 35 seconds

### CONCLUSION

An SDS 9300 with 32 K of core, 32 priority interrupt levels, card reader, card punch, line printer, teletype, three magnetic tapes, scope display, plotter and a 2 million byte disc is being used on-line with magnetic spectrometers. The computer is dedicated to one experiment at a time. A disc monitor and real time Fortran IV are used. Both overlaying of subroutines and core load swapping are done. Analysis routines are written such that they can be used on-line as well as off-line. One core load normally contains all the control programs, magnet routines, hardware reading routines as well as the analysis. About 30% of the memory is left for analysis programs. Core swapping is done for more data analysis. A large modular multiplexer system was built to interface with the experiment, and a display panel allows selection of any analysis routine that has been loaded to produce a display. The system is now in constant use.

## FIGURE CAPTIONS

- FIG. 1--Plan view of the experimental area at SLAC, showing the three magnetic spectrometers in end station A. The spectrometers pivot on rails about a common target point. A beam of electrons or photons strikes the target, and the angle and momentum of scattered or produced particles ( $e^\pm$ ,  $\mu^\pm$ ,  $\pi^\pm$ ,  $K^\pm$ , p, etc.) are detected by the spectrometers.
- FIG. 2--A photograph of the spectrometers. Large frames support the magnets, while the concrete huts at the end of the spectrometers house the detection equipment.
- FIG. 3--The particle detection equipment in the 20 BeV/c spectrometer for a typical experiment. The equipment consists of scintillation counter hodoscopes and trigger counters, gas Cerenkov counters, shower counters and range counters. All signals are strung about 400 feet to the counting house.
- FIG. 4--Photograph of the counting house taken during an experiment, showing the arrangement of the equipment and computer peripherals. The central processing unit, 3 magnetic tape units and disk are located in the area behind the scope.
- FIG. 5--Block diagram showing the interface to the SDS 9300.
- FIG. 6--Picture of one of the input multiplexer units or bins, showing the cables from devices plugged into the 31 possible jacks.
- FIG. 7--A schematic of an input multiplexer unit. SDS logic cards are used. The five address lines are decoded and 24 lines of data from one of the jacks (J1 to J31) is logically connected to the 24 data lines to the computer.
- FIG. 8--The computer console with the additional multiplexer manual control panel (above) and the 32 interrupt push buttons on the right.
- FIG. 9--Patch panel for routing external signals to the interrupts and sense lines. It may be easily changed from one experimental setup to the next.
- FIG. 10(a)--A typical log of a data-taking run. Messages are logged and printed for any buttons that are pushed, and scalers, beam monitors and constants are logged at the beginning and end of a run.
- FIG. 10(b)--A typical summary page written at the end of each run.
- FIG. 11(a)--The experimenter control panel showing the display control panel (top), the run control buttons (middle) and sense lines (bottom).
- FIG. 11(b)--A close-up of the control panel.

FIG. 12(a) and (b)--Examples of one dimensional histogram displays on the scope. Rounded scale factors (linear or logarithmic) are computed and error bars are optionally displayed. These displays are generated by the subroutine OPUS1.

FIG. 13-- Printout made by OPUS1 for a one-dimensional histogram.

FIG. 14-- Examples of two-dimensional isometric displays made by the subroutine OPUS2. (a) shows timing or delay curves for 20 counters in which the z axis gives counts in a counter number (x axis) versus delay in nanoseconds of the master trigger signal (y axis). (b) shows events (z axis) having a certain shower pulse height (y axis) and range (x axis), allowing identification of electrons, muons and mesons (or protons).

FIG. 15-- Printout made by OPUS2 for a two-dimensional histogram.

FIG. 16-- A diagnostic "one event" display which uses line intensification to show which counters are struck, displays the bin number of the struck counter, and also shows pulse height channel numbers for the event.

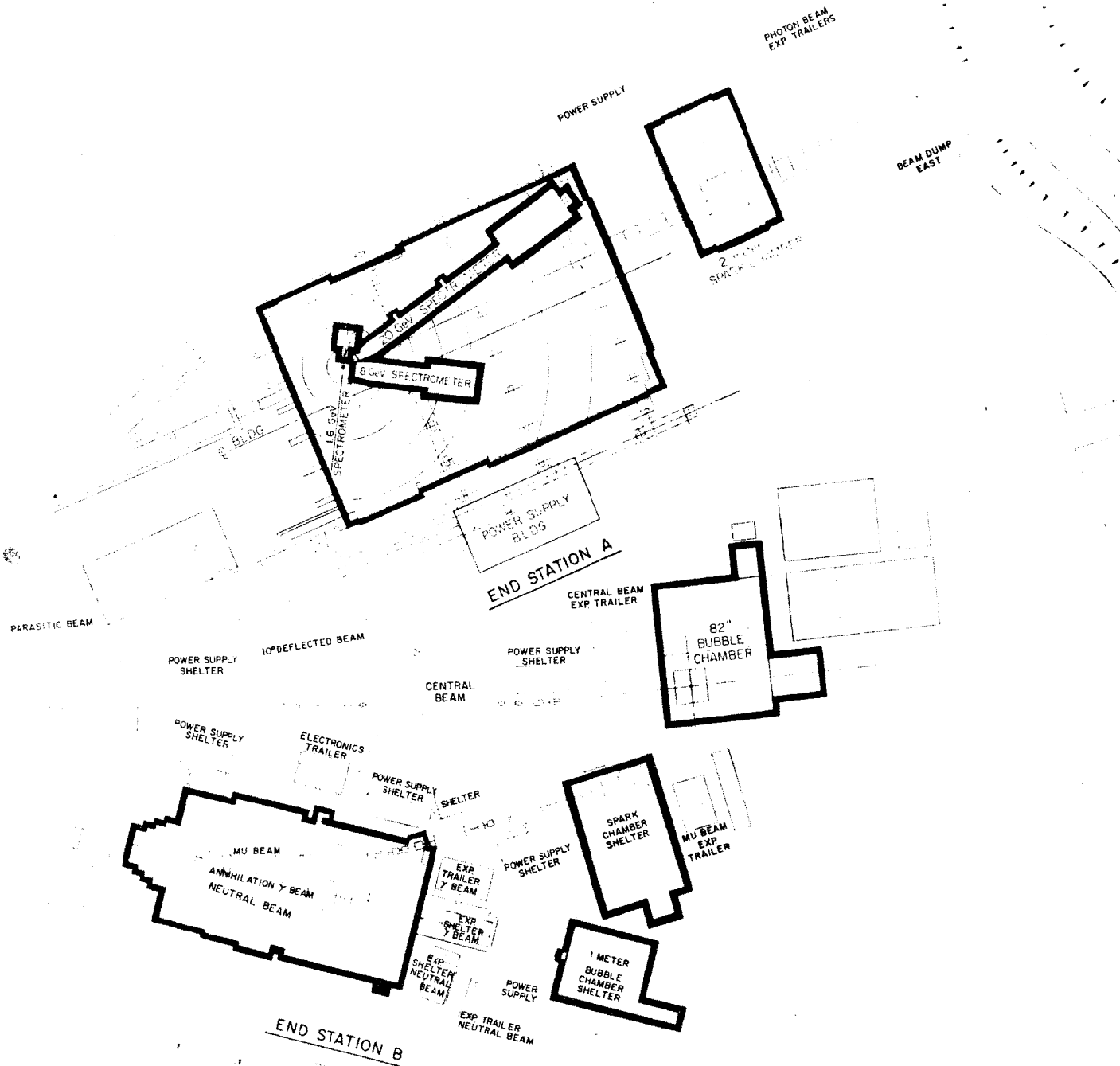


Fig. 1



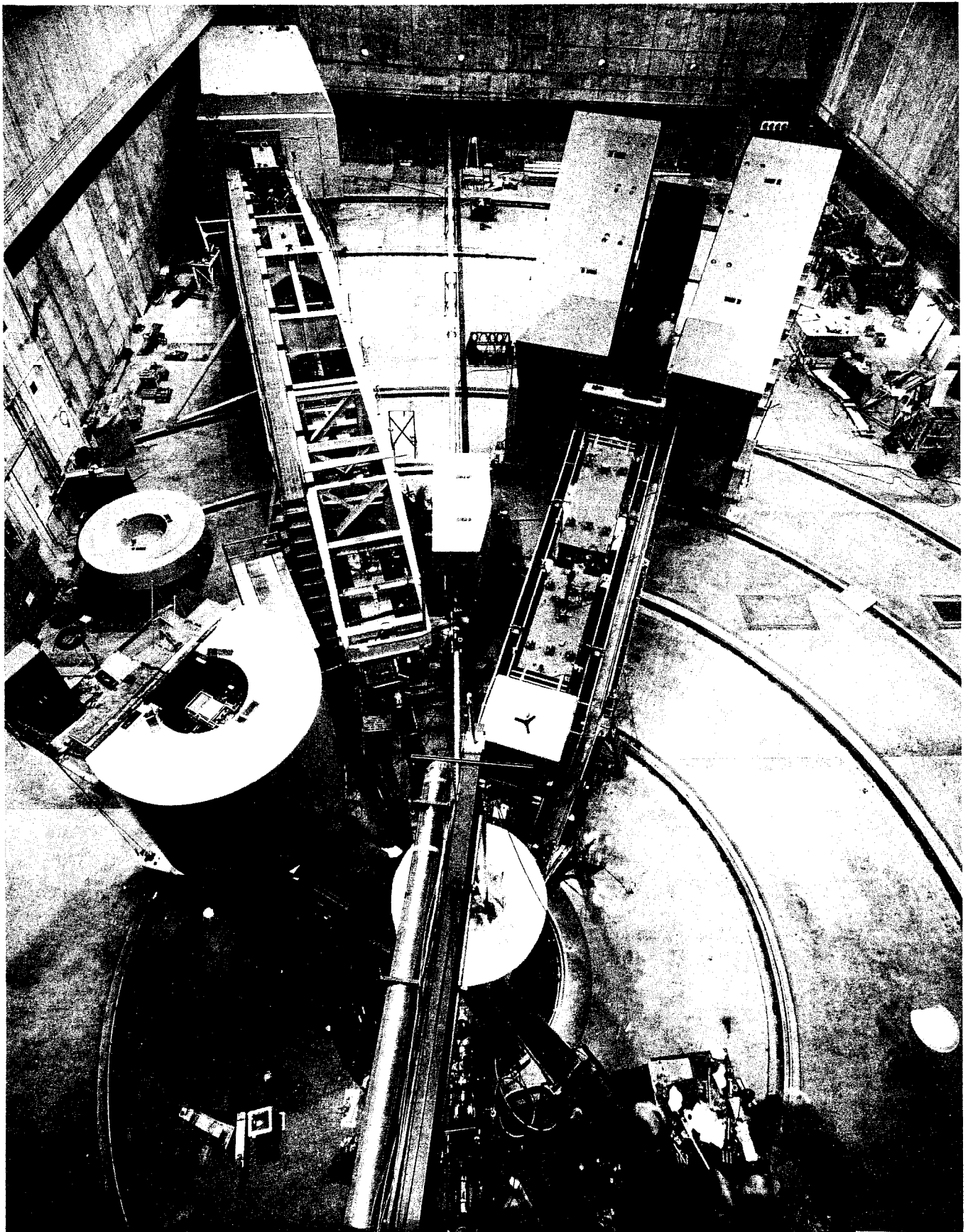


Fig. 2

1271A1

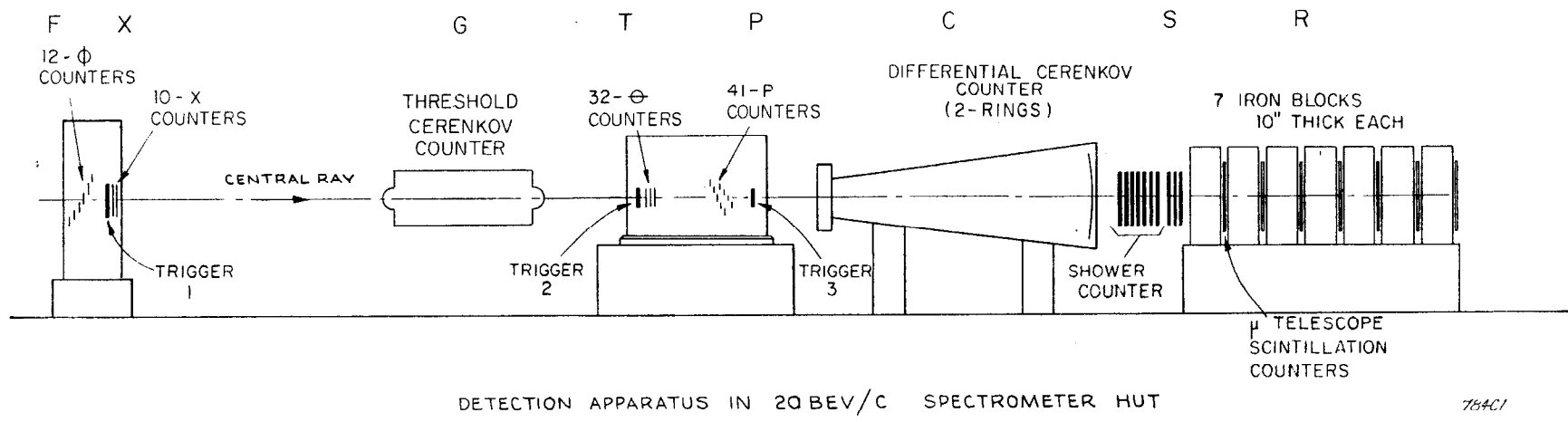
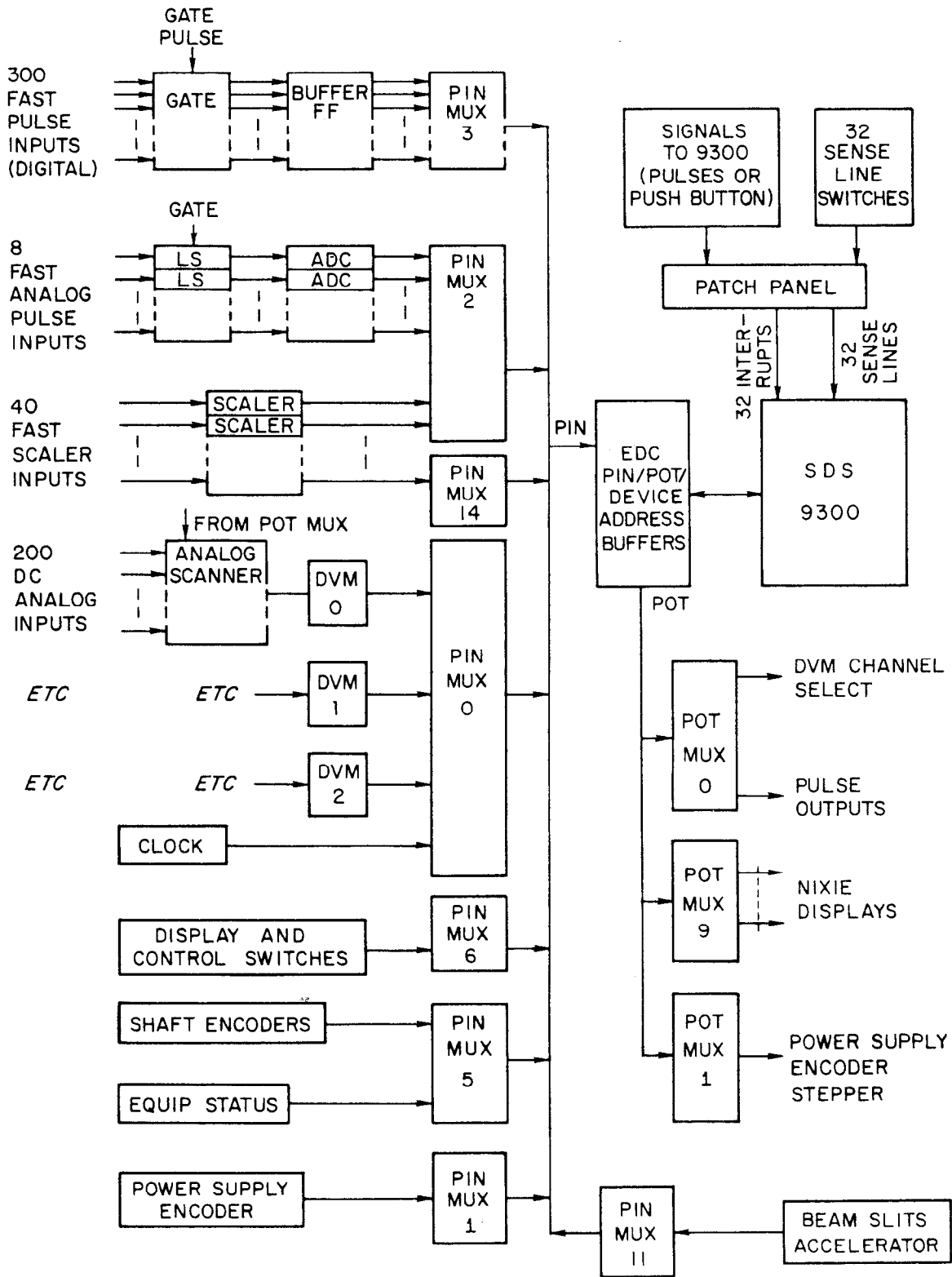


Fig. 3



Fig. 4

1271A2



752BI  
(Rev)

Fig. 5

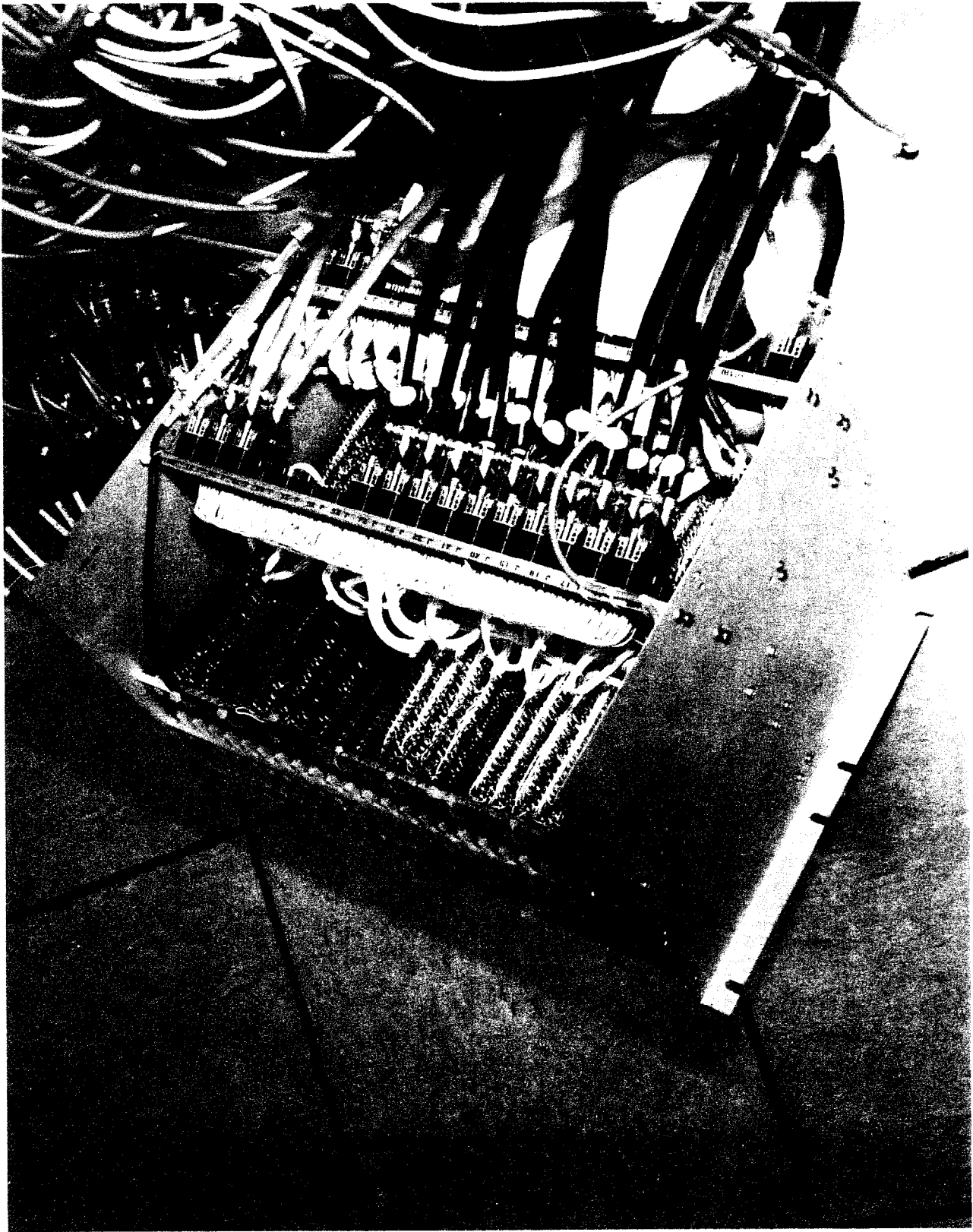


Fig. 6

1271A3

# MULTIPLEXER UNIT LOGIC DIAGRAM (FOR DEVICES INPUT TO COMPUTER)

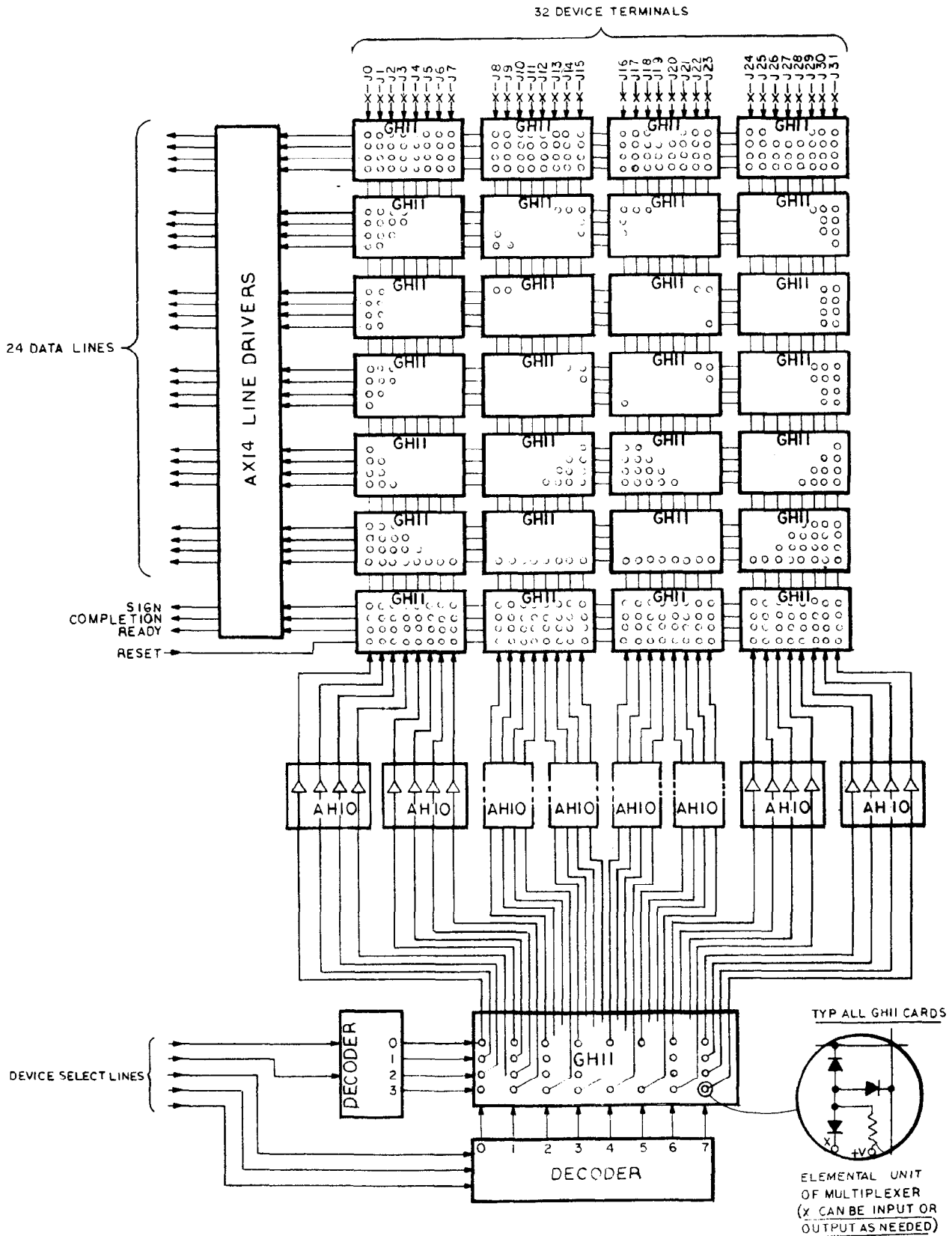


Fig. 7

1271A4

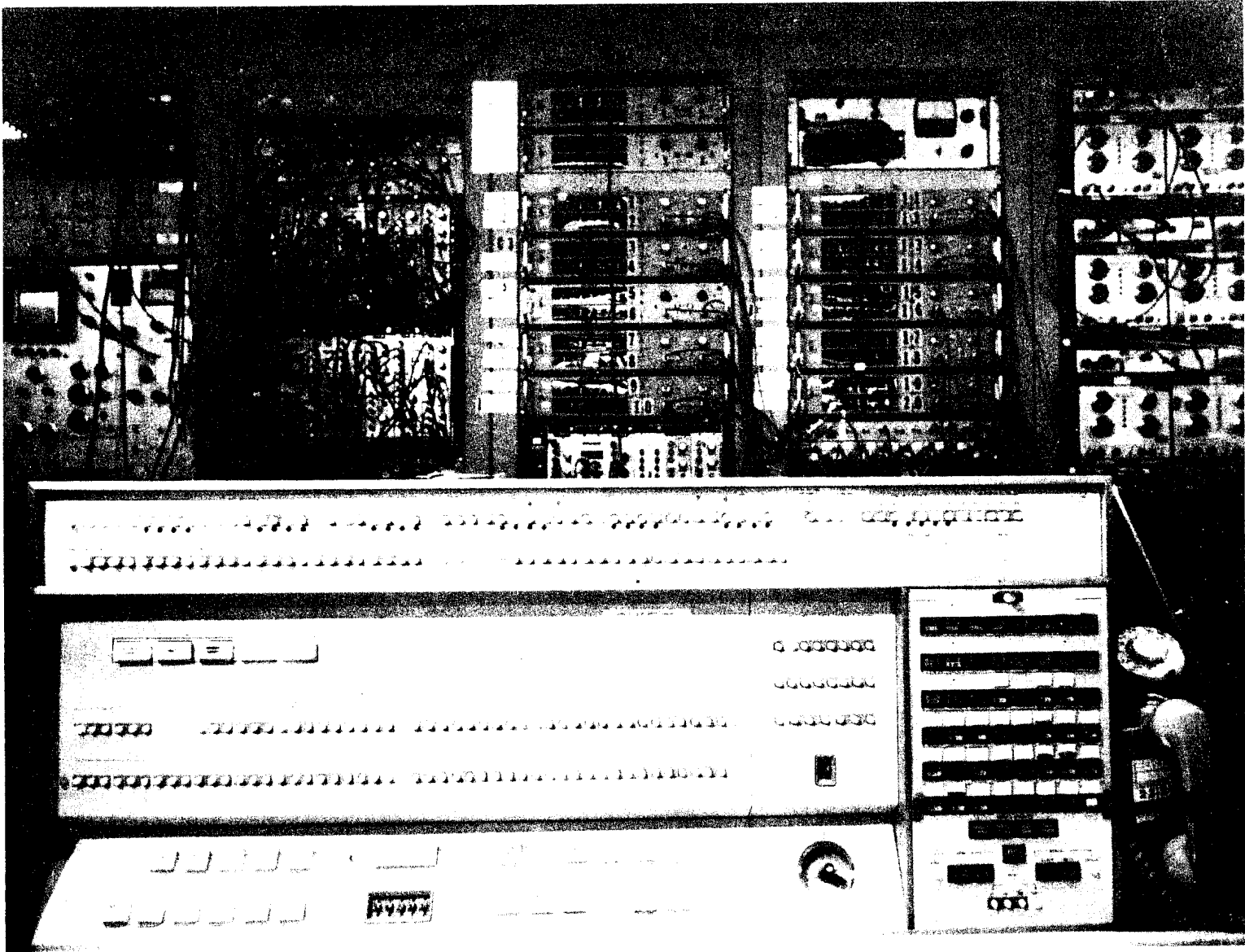
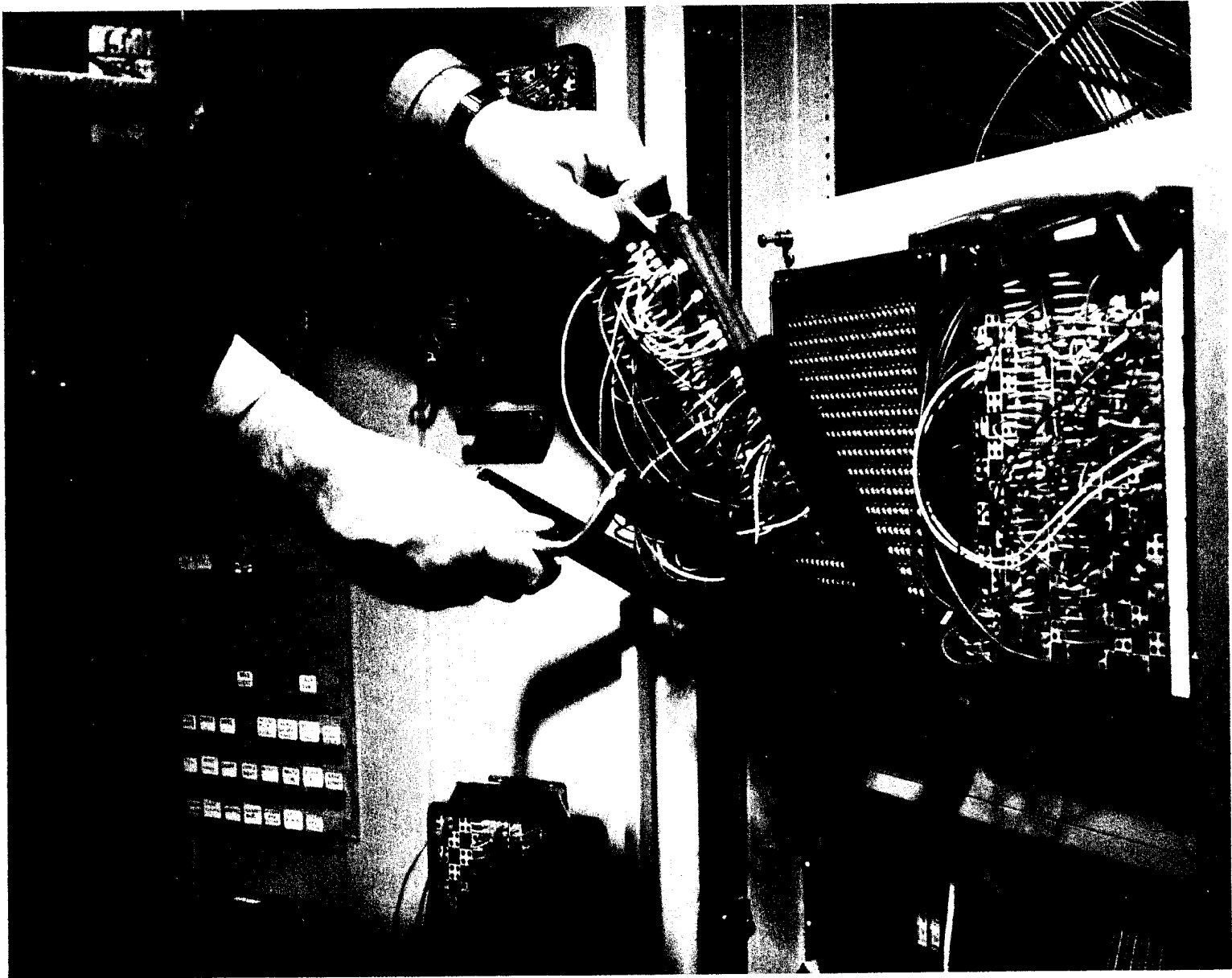


Fig. 8

1271A5



1271A6

Fig. 9



\*\*\*\* RUN 2675 \*\*\*\*

1035, 15 JAN 69. PAGE 0037

```
*BEGIN RUN                RUN 2675 *** TAPE 1624 *** TIME 0312 1654 05.6

*BEAM MONITORS
*SCALERS
  VET0 0 0          FLS1 0 0          0 0          0 0
  VACC 0 0          DEAD 0 0          0 0          0 0
  T2.3 0 0          S 0 0            0 0          0 0
  TR1 0 0           CIN 0 0           0 0          0 0
  TR2 0 0           COUT 0 0          0 0          0 0
  TR3 0 0           TKIL 0 0          0 0          0 0
  TPLE 0 0          TRIG 0 0          0 0          0 0
  ACC1 0 0          TRUE 0 0          0 0          0 0
  R1 0 0            2DB 0 0            0 0          0 0
  FLS2 0 0          CT 0 0            0 0          0 0

*CONSTANTS
  CARY 1 .0011 VBLTS 1.0010E-05 FDS .0000E 00 DC0UL .0000E 00 EQ 5.7896E 76 EQPP
  CARY 2 .0014 VBLTS 1.0030E-06 FDS .0000E 00 DC0UL .0000E 00 EQ 5.7896E 76 EQPP
  CARY 3 .0051 VBLTS 1.0050E-04 FDS .0000E 00 DC0UL .0000E 00 EQ 5.7896E 76 EQPP
  TOR01D 0 PULSES 1.0E 12 GAIN .0000E 00 DC0UL .0000E 00 EQ 5.7896E 76 EQPP
  QDVH 169 2 1.6900E-06 DC0UL 6.5724E 12 EQ

CLEARED P VS T
CLEARED X
CLEARED F
CLEARED P-T NORM
CLEARED CI VS CO
CLEARED CHERENKOV SPECTRA
CLEARED SH VS R
CLEARED MM2
CLEARED MM SCAN
*RESET                RUN 2675 *** TAPE 1624 *** TIME 0312 1654 25.6

*RUN                RUN 2675 *** TAPE 1624 *** TIME 0312 1654 26.7

*COMMENT                RUN 2675 *** TAPE 1624 *** TIME 0312 1654 27.9
  P1+ M2

*STOP                RUN 2675 *** TAPE 1624 *** TIME 0312 1700 09.9

*RUN                RUN 2675 *** TAPE 1624 *** TIME 0312 1713 36.3

*STOP                RUN 2675 *** TAPE 1624 *** TIME 0312 1716 55.6

*SCALERS
  VET0 0 645          FLS1 0 34          0 0          0 0
  VACC 0 2            DEAD 0 8483          0 0          0 0
  T2.3 0 9174          S 0 1920          0 0          0 0
  TR1 0 42243          CIN 0 30017          0 0          0 0
  TR2 0 14476          COUT 0 21874          0 0          0 0
  TR3 0 15799          TKIL 0 8084          0 0          0 0
  TPLE 0 8540          TRIG 0 8543          0 0          0 0
  ACC1 0 34            TRUE 0 8540          0 0          0 0
  R1 0 20238          2DB 0 8531          0 0          0 0
  FLS2 0 8540          CT 0 20250          0 0          0 0
```

\*\*\*\* RUN 2675 \*\*\*\*

EQ= 16.050 1.011 DEG M2 +VES Δ= .080 MM2= 1.11

1036, 15 JAN 69. PAGE 0038

```
*BEAM MONITORS
*CONSTANTS
  CARY 1 9.7665 VBLTS 1.0010E-05 FDS 9.7752E-05 DC0UL 1.6814E 14 EQ 8.6625E 08 EQPP
  CARY 2 2.0469 VBLTS 1.0030E-06 FDS 2.0516E-06 DC0UL 1.6739E 14 EQ 8.6239E 08 EQPP
  CARY 3 3.1567 VBLTS 1.0050E-04 FDS 3.1674E-04 DC0UL 1.6814E 14 EQ 8.6623E 08 EQPP
  TOR01D 194103 PULSES 1.0E 12 GAIN 7.4535E-04 DC0UL 6.1215E 15 EQ 3.1537E 10 EQPP
  QDVH 37611 2 3.7611E-04 DC0UL 1.4627E 15 EQ

**ENDRUN                RUN 2675 *** TAPE 1624 *** TIME 0312 1717 02.5
                          15.885 = .1681E 15
```

1271A7

Fig. 10(a)

PI+ H2

E = 16.05	P = 15.89	THET = 1.011	GT = 1.8469E-02	GP = 6.4903E-03
M1 = .9382	TNO = 1.000	NSLY = .9500	SSLY = .9494	VSLT = .2807
P2 = 15.89	FLTR = .0000	CP = 1.994	CT = 9.623	COLL = 4.001
PLSE = 180.0	TLEN = 26.99	DP = 1.000	STER = 4.4100E-07	RAT0 = .1540
MON0 = 3.000	AE = 1.000	AMU = 1.000	A1 = 1.406	A2 = 1.372
A3 = 1.460	CAL1 = 4.4230E 09	CAL2 = 2.0980E 11	CAL3 = 1.3650E 09	CALT = 2.1118E 10
EBKD = 6.0980E-02	ASR = 1.220	CAL6 = 1.0000E 10	ABS = .0000	DIPA = .3000
CTH1 = 62.13	CTH2 = 54.52	CTH3 = 21.40	GTH1 = 25.19	GTH2 = .0000
GTH3 = .0000	SHGN = 36.00	TDEN = 6.8850E-02	ATWT = 1.008	MRUN = .0000
CHPI = 1.015	CHK = 1.040	CHP = 1.080	AR = 1.098	AS = 1.111
ABSG = 1.024	ABST = 1.010	ABSD = 1.072	AM0D = 1.000	ABS = 1.109
SML1 = .0000	SML2 = .0000	PIH0 = 1.000		

EVCUTS

( 1- 10)	1	9	327	50	90	120	100	0	0	0
( 11- 20)	0	2	25	35	45	55	60	0	160	160
( 21- 30)	30	60	100	160	2	19	6	25	40	4
( 31- 40)	0	35	20	20	10	0	2	0	0	0
( 41- 50)	6	18	100	0	379	24	1	7	23	0
( 51- 60)	2	40	50	12	1	70	9999	160	33	0
( 61- 70)	0	0	-20	40	-80	80	2	0	0	10

BEAM MONITORING

CARY 1	9.7665 VOLTS	1.0010E-05	FDS	9.7752E-05	DC0UL	1.6814E 14 EQ	8.6625E 08 EQPP
CARY 2	2.0469 VOLTS	1.0030E-06	FDS	2.0516E-06	DC0UL	1.6739E 14 EQ	8.6239E 08 EQPP
CARY 3	3.1567 VOLTS	1.0050E-04	FDS	3.1674E-04	DC0UL	1.6814E 14 EQ	8.6623E 08 EQPP
TOR0ID	194103 PULSES	1.0E 12	GAIN	7.4535E-04	DC0UL	6.1215E 15 EQ	3.1537E 10 EQPP
QDVM	37611	2		3.7611E-04	DC0UL	1.4627E 15 EQ	

SCALERS

VET0	0	645	FLS1	0	34	0	0	0	0
VACC	0	2	DEAD	0	8483	0	0	0	0
T2,3	0	9174	S	0	1920	0	0	0	0
TR1	0	42243	CIN	0	30017	0	0	0	0
TR2	0	14476	CBUT	0	21874	0	0	0	0
TR3	0	15799	TKIL	0	8084	0	0	0	0
TPL0	0	8540	TRIG	0	8543	0	0	0	0
ACC1	0	34	TRUE	0	8540	0	0	0	0
R1	0	20238	2DB	0	8531	0	0	0	0
FLS2	0	8540	CT	0	20250	0	0	0	0

EVNTCNTS

READ	8076	0	.8603	.0039	FH0D	.0063	.0010	VANG	.0001	.0001	DBLF	.0109	.0012
LOGGED	8076	1	.1209	.0036	XH0D	.0287	.0019	XPIV	.0000	.0001	DBLX	.0316	.0019
LOST	0	2	.0161	.0014	TH0D	.0119	.0012	XB4	.0062	.0009	DBLT	.0140	.0013
SAMPLED	8076	3	.0022	.0005	PH0D	.0537	.0025	FLS1	.0000	.0001	DBLP	.0211	.0016
ANALYSED	6948	4	.0005	.0002	HANG	.0119	.0012	FLS2	.0000	.0001	FREE	.0386	.0021

YIELDS CH\*\*2/10\*\*30-0-STER-GEV/C

ERAW	592	.0739	.0030	MU	91	.0114	.0012	STRG	3607	.5426	.0090		
N I	0	.0000	.0017	N II	398	.4672	.0234	NIII	24	.0284	.0058		
I	305	.0793	.0045	II	3051	.5516	.0100	III	207	.0377	.0026	JUNK	44 .0002 .0010
ECOR	375	.0468	.0091	RL0	431	.1195	.0058	NMID	1323	.3668	.0101		

Fig. 10(b)

1271A8

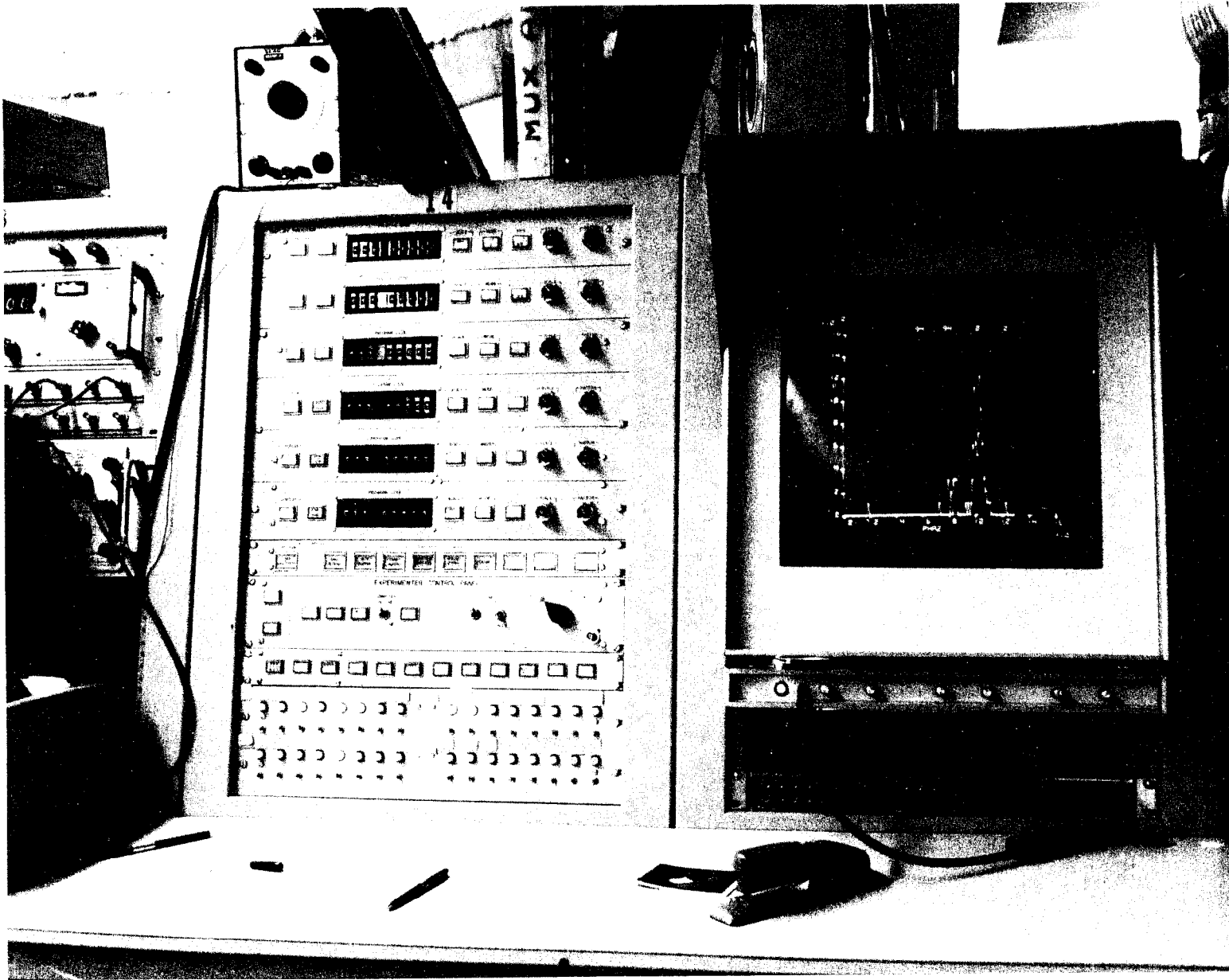


Fig. 11(a)

1271A9

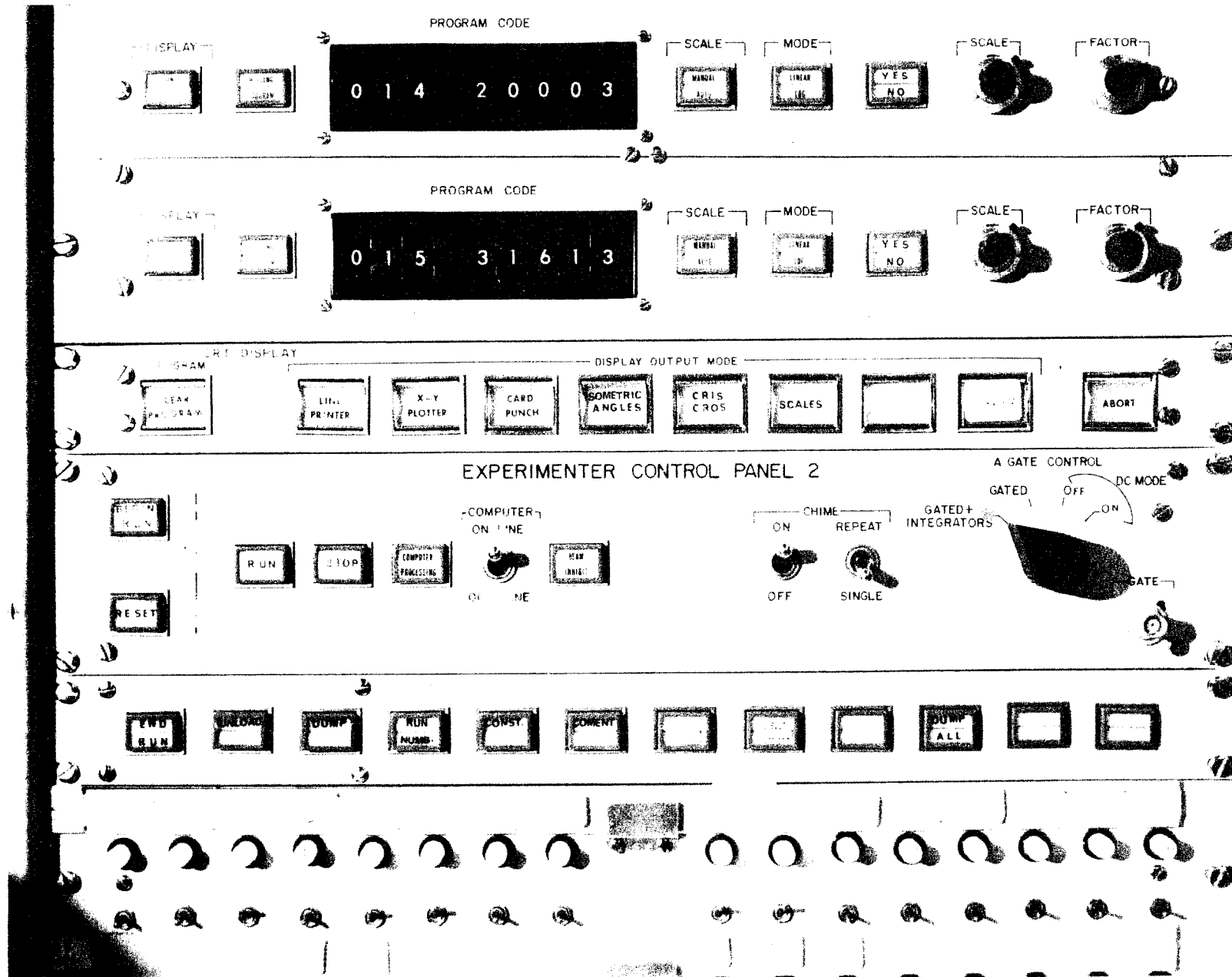


Fig. 11(b)

1271A10

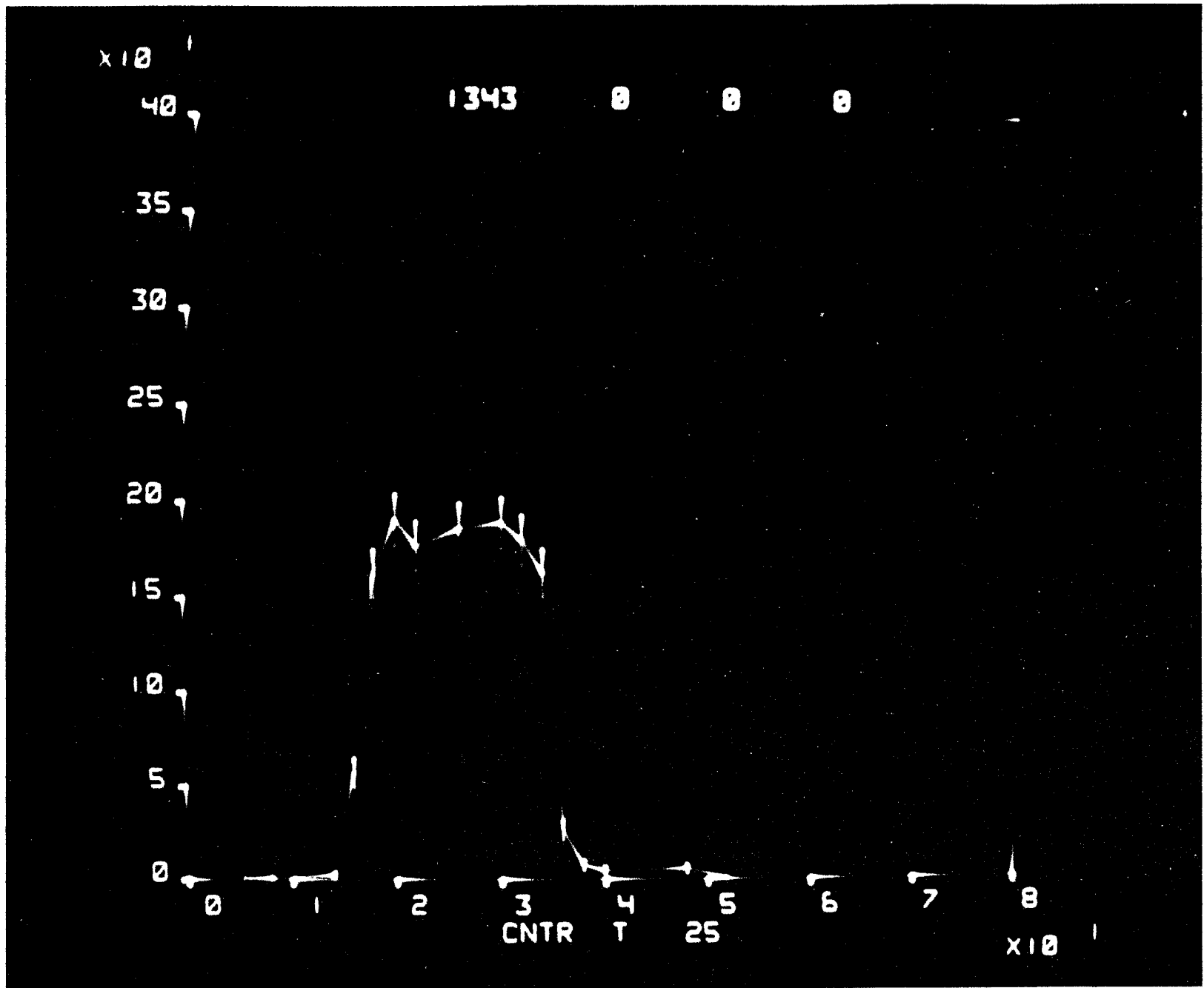
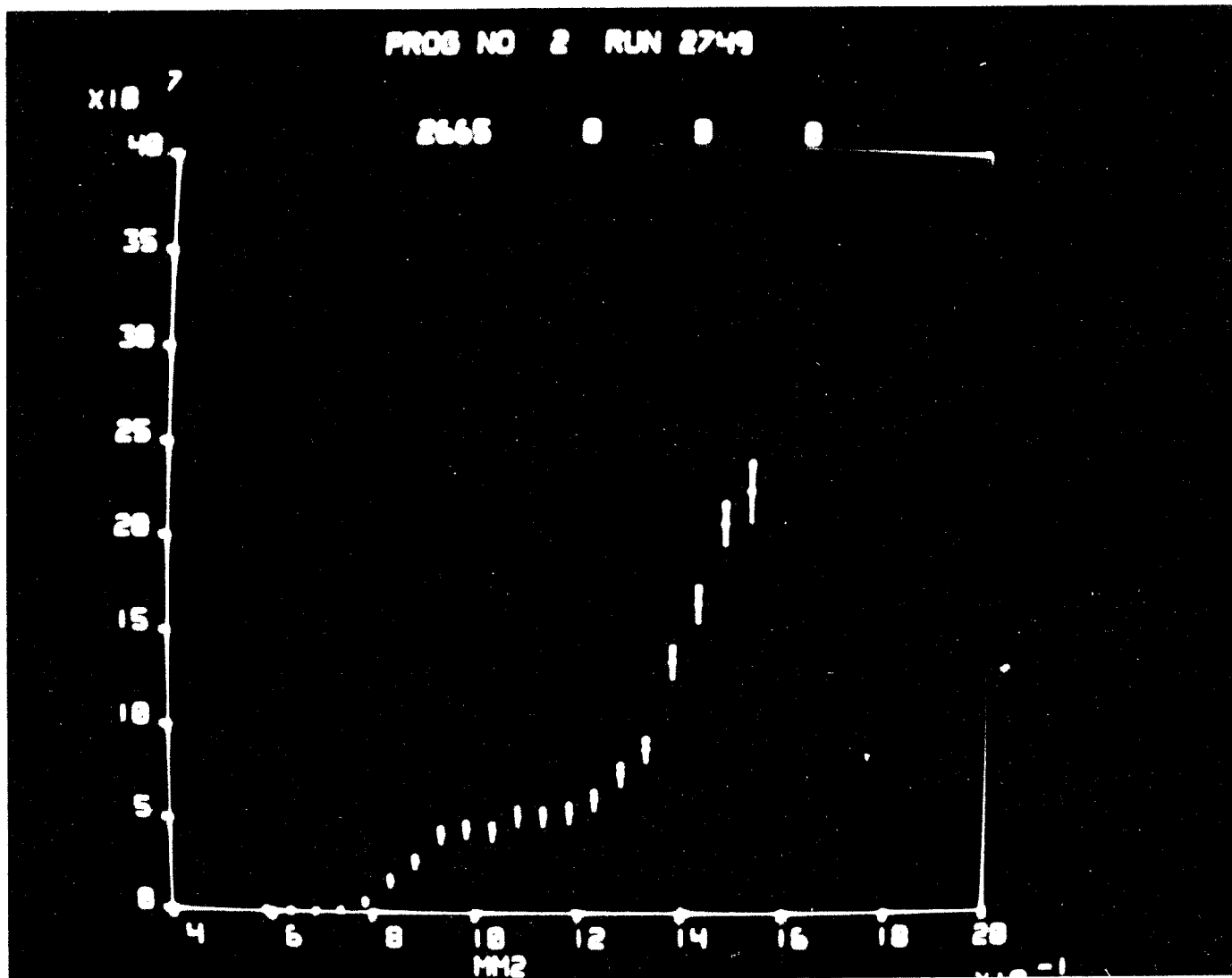


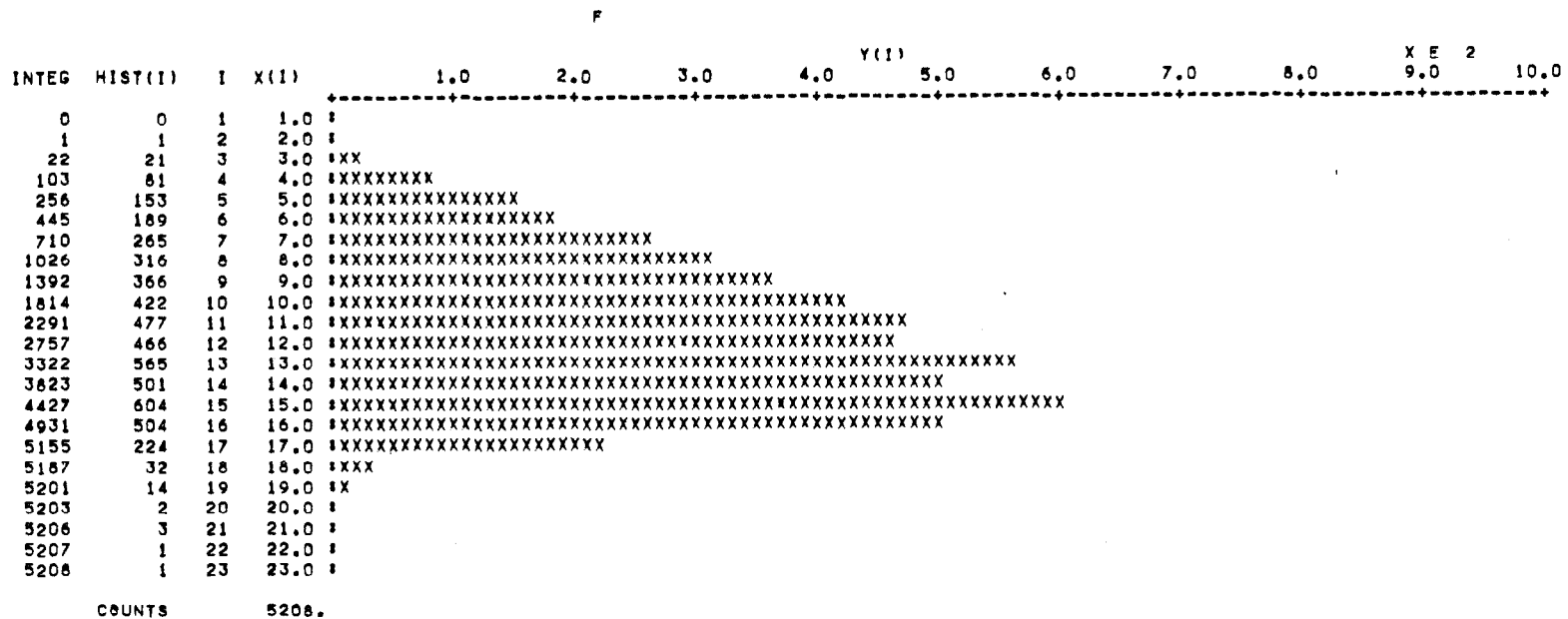
Fig. 12(a)

1271A11



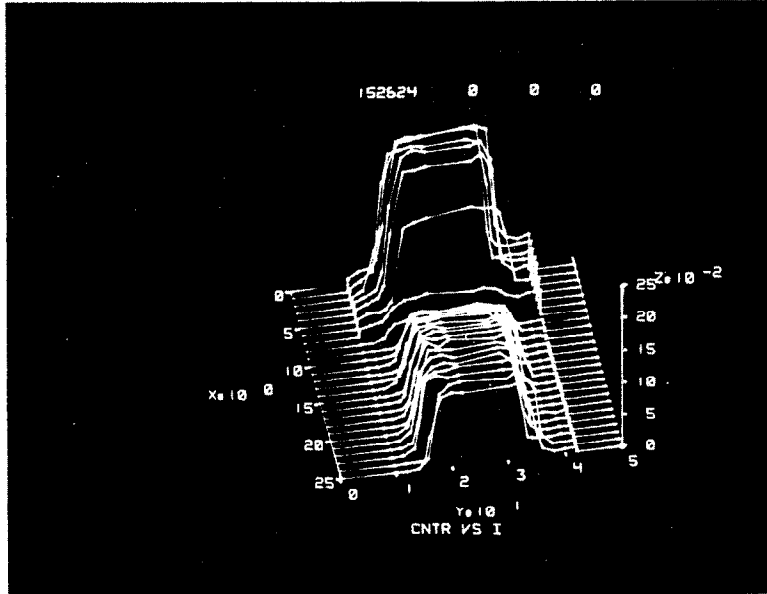
1271A12

Fig. 12(b)



1271A13

Fig. 13



20 COUNTERS

1271A14

Fig. 14(a)



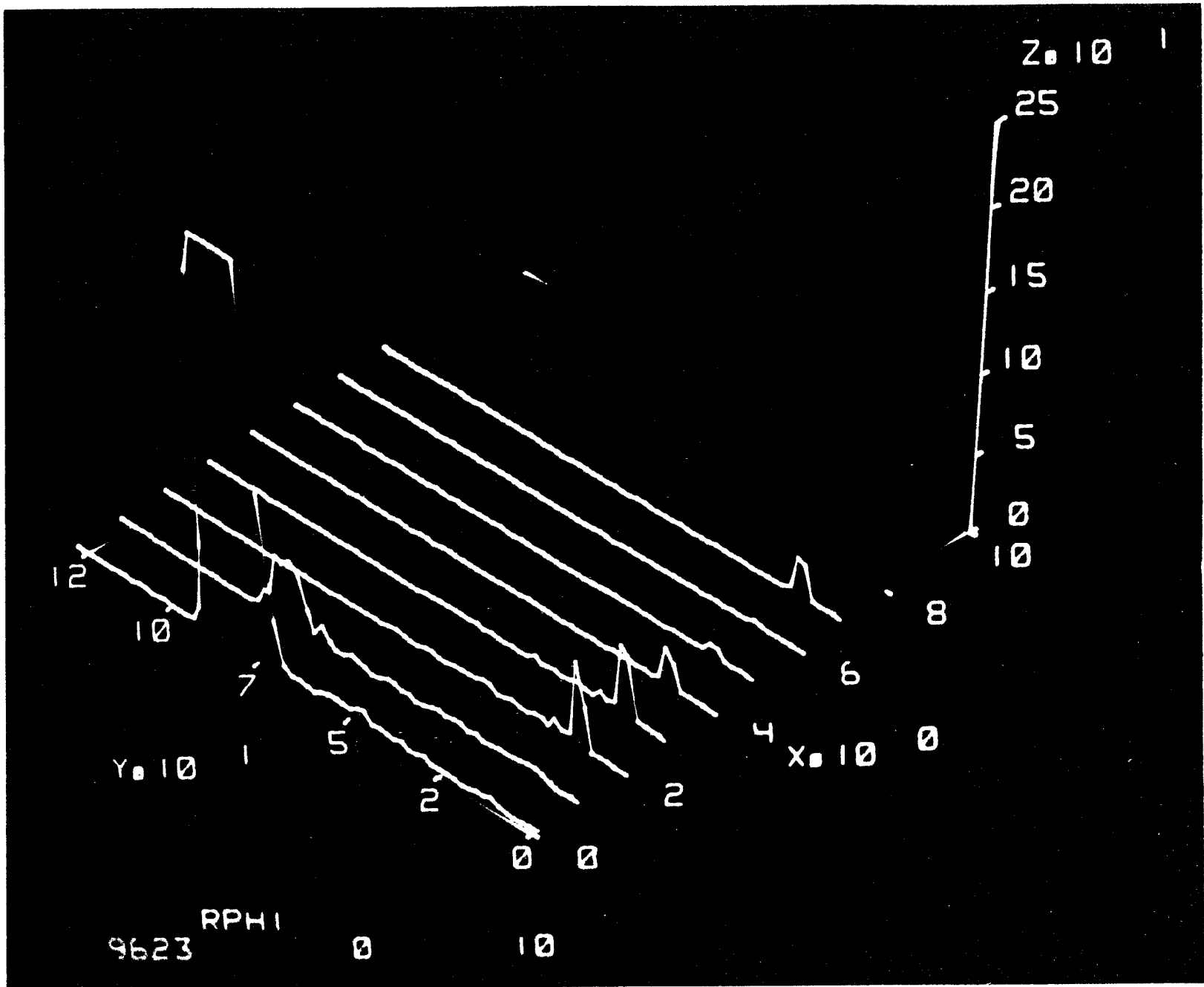


Fig. 14(b)

1271A15

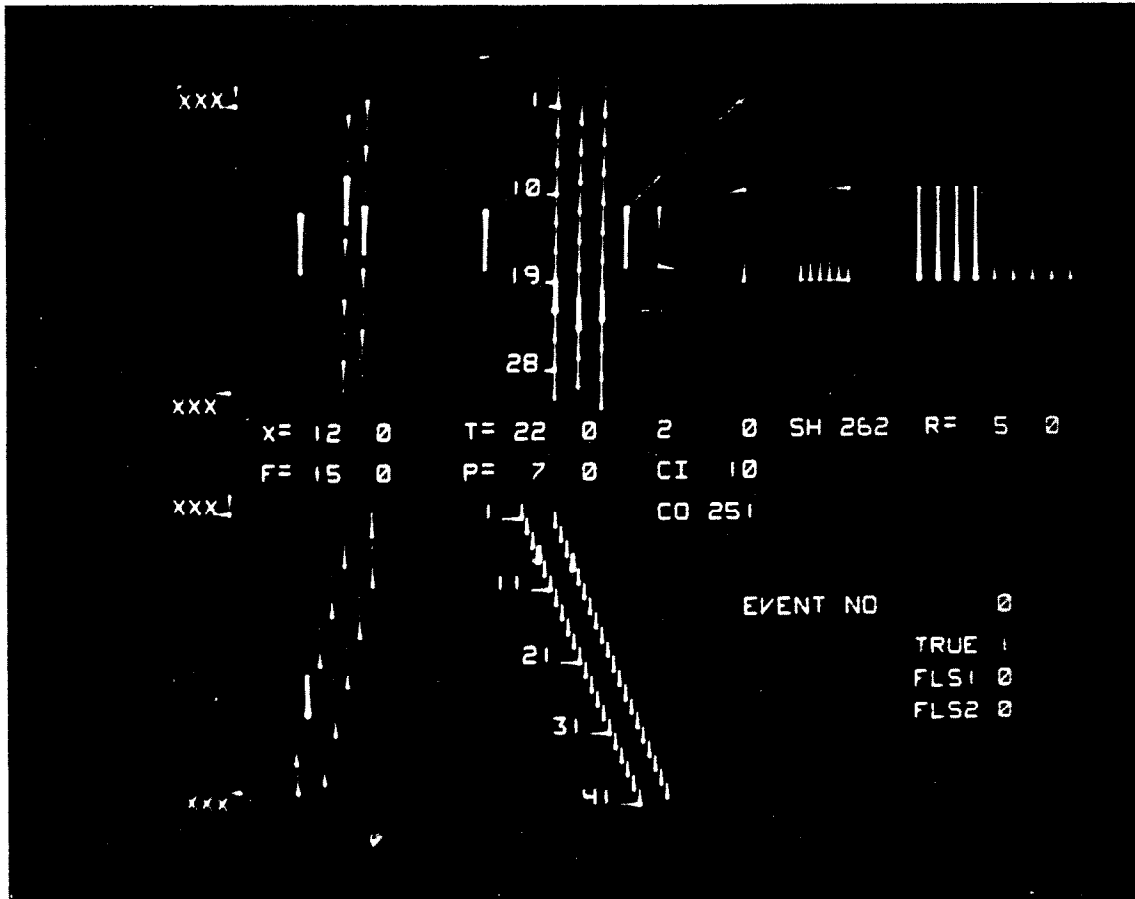
SH VS R

J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
XVY	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	10	67	5	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	12	23	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	14	9	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	16	13	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	18	3	2	4	46	67	38	13	5	1	11											
11	20	9	2	16	240	298	184	82	15	5	41											
12	22	2	4	24	160	184	87	39	19	14	22											
13	24	6	2	15	59	77	31	11	4	1	5											
14	26	3	0	6	35	42	17	7	2	1	8											
15	28	2	0	4	28	22	12	4	0	1	0											
16	30	3	1	3	18	11	7	2	0	2	2											
17	32	3	1	6	13	6	3	3	1	0	2											
18	34	2	0	7	8	11	7	2	1	0	0											
19	36	0	0	4	2	4	3	2	0	1	0											
20	38	2	1	1	6	2	2	1	0	1	0											
21	40	19	7	23	90	66	24	9	3	1	0											
22	75	19	4	27	69	37	15	8	0	1	0											
23	110	16	12	36	67	36	10	4	0	1	0											
24	145	11	7	35	72	50	15	1	1	0	0											
25	180	12	12	49	70	29	13	3	0	0	0											
26	215	13	10	51	94	28	9	0	0	0	0											
27	250	19	24	67	72	31	3	2	0	0	0											
28	285	17	35	59	64	16	3	0	0	1	0											
29	320	22	31	67	40	18	9	0	0	0	0											
30	355	24	20	43	31	7	3	0	0	0	0											
31	390	19	18	30	25	9	0	0	0	0	0											
32	425	13	26	24	14	3	0	0	0	0	0											
33	460	6	12	16	11	1	2	0	0	0	0											
34	495	10	6	8	2	0	0	0	0	0	0											
35	530	8	10	4	4	0	0	0	0	0	0											
36	565	38	22	6	0	0	0	0	0	0	0											
37	600	91	36	3	0	1	1	0	0	0	0											
38	635	51	18	3	0	0	0	0	0	0	0											
39	670	8	4	0	0	0	0	0	0	0	0											
40	705	0	0	0	0	0	0	0	0	0	0											
41	740	0	0	0	0	0	0	0	0	0	0											
42	775	0	0	0	0	0	0	0	0	0	0											
43	810	0	0	0	0	0	0	0	0	0	0											
44	845	0	0	0	1	0	0	0	0	0	0											

COUNTS 4828. . . . .  
 RLCOR= 1.1403 .0062 RSEFF= .8683 .0052 RS PERCENTILES .85 312 5 .90 357 6 .95 440 7

Fig. 15

1271A16



1271A17

Fig. 16