ORNL-4299
UC-32 – Mathematics and Computers

FLOTSAM – A TWO-PARAMETER ON-LINE DATA ACQUISITION PROGRAM FOR THE PDP-7

A. L. Marusak

OAK RIDGE NATIONAL LABORATORY
operated by
UNION CARBIDE CORPORATION
for the
U.S. ATOMIC ENERGY COMMISSION

Printed in the United States of America. Available from Clearinghouse for Federal
Scientific and Technical Information, National Bureau of Standards,
U.S. Department of Commerce, Springfield, Virginia 22151
Price: Printed Copy $3.00; Microfiche $0.65

ORNL-4299

Contract No. W-7405-eng-26

NEUTRON PHYSICS DIVISION

FLOTSAM - A TWO-PARAMETER ON-LINE DATA ACQUISITION PROGRAM FOR THE PDP-7

A. L. Marusak*

OCTOBER 1968

*Oak Ridge Graduate Fellow from The University of Tennessee under appointment from the Oak Ridge Associated Universities, Inc.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee
operated by
UNION CARBIDE CORPORATION
for the
U. S. ATOMIC ENERGY COMMISSION

CONTENTS

FLOTSAM — A TWO-PARAMETER ON-LINE DATA ACQUISITION PROGRAM FOR THE PDP-7

A. L. Marusak

ABSTRACT

FLOTSAM is a program for the PDP-7, designed to accept one- and/or two-parameter data from experiments. This report describes the particular two-parameter program used for measuring neutron cross sections by the time-of-flight technique from (d,n), (n,n'), and (p,n) reactions. Sufficient versatility is included so that conversion to one-parameter 4096-channel data acquisition or other configurations is, with some study of the program, easy and has been done. The data from ADC's are fed directly into the computer into one of two 64-word temporary buffers. While one buffer is being filled with data, events in the other are being analyzed and stored. A bank of six scalers can be read into the computer and stored. A continuous oscilloscope display of all or some of the data is provided, with three modes (linear, log, square root) available for the y-coordinate (number of counts in a channel). Light pen routines are provided to enable one to extract peak information from either old data stored on magnetic tape or from data as they are taken. Permanent storage of data is on IBM-compatible magnetic tape; the data can also be typed out or plotted on a CALCOMP plotter. This report includes a general description of the program, operating instructions, some flow charts, and the complete annotated listing of the program itself.

I. INTRODUCTION

The purpose of this report is threefold. First of all, a general description is provided to give one some idea of what the system can do. Intermixed in this description are some technical details which may be of interest. Secondly, a complete set of operating instructions is provided for those who merely wish to use the program. Finally, a complete anotated listing of the program itself is given along with some flow charts for those who wish to examine it in detail.

In its simplest form, a computer used for on-line data acquisition replaces a multichannel analyzer. In addition, FLOTSAM enables one to do extensive monitoring of an experiment, to extract peak information [area, centroid, and FWHM (full width at half-maximum)] as a first step in obtaining cross sections, and to store and retrieve data on magnetic tape quickly and easily. For monitoring purposes, a one-parameter 256-channel spectrum can be taken. Regions of interest of a particular spectrum can be typed out and summed if desired, or peak information can be extracted by means of a light pen. Spectra can also be plotted on a CALCOMP[1] plotter. The ADC's (analog-to-digital converter) are checked regularly against each other to make sure they are all counting. A bank of 6 scalers is read into the machine at intervals, and certain relations which should exist between different scalers are checked. An oscilloscope display exists which can display continuously regions of interest in the data in any of three y-coordinate (number of counts in a channel) modes (linear, log, or square root). For data storage and retrieval, routines are included to write on magnetic tape the data taken during a run along with pertinent monitoring information such as the monitor

[1] California Computer Products, Inc., Downey, California.

spectrum, the time of the run (all runs are timed), the six scalers, etc. Other routines exist to search on magnetic tape for a previous run and retrieve all or some of the data for that run.

II. STORAGE IN THE COMPUTER

Events are stored as two-parameter data, sorted according to pulse height and time, in up to eight groups (according to pulse height) of 512 channels (time) each. In addition a single parameter (time, energy, etc.) 256-channel spectrum for monitor purposes is stored. The two-parameter data are furthermore walk-corrected[2] and collapsed into a single group of 512 channels (henceforth called the Collapsed Spectrum). One additional region of 512 channels, called the Background, is provided. This region is used as temporary storage by some routines, and also is used in the peak stripping routines to extract the centroid, area, and FWHM of individual peaks. The rest of the $20{,}000_8$ word memory ($\sim 5400_8$ words) is used to store the actual program.

III. TWO-PARAMETER STORAGE

The data are stored in two parameters primarily to treat the problem of "walk".[2] The two-parameter storage also allows one to set arbitrary biases on the detector. Since most background consists of low energy neutrons and gamma rays, most of it falls into the lower groups corresponding to low pulse heights. One can arbitrarily discriminate against this background by not using low pulse-height groups for high energy neutron peaks, losing a small percentage of the peak area, while restoring the group if necessary for low energy peaks. "Wrap-around"

---

[2] W. E. Kinney, "Neutron Elastic and Inelastic Scattering from ^{56}Fe from 4.60 to 7.55 MeV," Ph.D. dissertation, ORNL-TM-2052, January 1968.

(because timing for the experiments is from a pulsed beam, neutrons with flight times differing by a time equal to the frequency^{-1} of the pulsed beam will fall in the same channel, the lower energy neutrons being "wrapped around"), low energy peaks can be eliminated in the same manner. Since the program allows one to collapse various sequences of groups "on-the-fly", one can quickly deduce background running conditions, peak shapes, etc.

IV. DATA MANIPULATION AND MONITORING

The program has built into it the capability of manipulating data stored in one particular region, the Background. By means of the light pen, the Background can be set equal to a Foreground (e.g., Monitor, one of the eight groups, or the Collapsed Spectrum. Any of these may later be referred to as Foreground.). Then the Background can be altered with the light pen in such a way that the difference between Foreground and Background is a peak of interest. A routine is provided to calculate the area, centroid, and FWHM of this peak.

Simple checks on some of the electronics is done by the computer. In our case, pertinent data is fed into a bank of six scalers as data are being taken. Normally definite relations hold between the values of certain scalers (in our case, under proper running conditions, scaler 3 > scaler 4, scaler 6 > scaler 5, and scaler 4 > scaler 6). At the option of the user, these scalers can be read into the computer periodically and checked. If the proper relations do not hold, the run is stopped and the teletype bell is rung to warn the operator. Similarly, at the discretion of the user, the two sets of ADC's (TOF and Monitor)

can be checked against each other to make sure both are counting. If, for instance, 576 consecutive events are TOF events, the computer will assume the Monitor ADC is not working, and the operator is warned as above.

The bit pattern of data fed in is set by patch-panels. By changing boards, one is able to feed in one parameter gamma-ray spectra to study pulse-height amplifier calibration. With another patch panel, one can feed in data from the γ-n pulse-shape discrimination circuit to set windows for γ-n discrimination.

V. INTERRUPT HANDLING

FLOTSAM is a program which operates completely on the PIC (Program Interrupt Control). There are no waiting loops for job-done or equipment-ready flags from input/output equipment. There is one and only one main program, the display routine. This routine is interrupted as needed to service peripheral equipment, and then control is returned to it. Since control is returned with the PIC enabled between all interrupts (e.g., when the teletype is serviced, control is returned between the typing of individual characters), other input/output devices can be initialized while one is being serviced.

The magnetic tape input/output control and the ADC interface transfer blocks of data by means of a high-priority interrupt, the data break. To initialize one of these devices, one first sets up through the program two registers, a memory address counter (MAC) and a word counter (WC). (The magnetic tape units and the ADC interface have different pairs of MAC's and WC's.) The MAC contains the starting address of the block of

data to be transferred, and the WC contains the negative of the number of words to be transferred. When one of these devices has been enabled and is ready to transfer a word of data, the computer "pauses" (that is, the flow of the program is stopped, or broken, to enable a transfer of 1 datum of information) for one machine cycle, 1.75 μsec, during which time the datum is transferred to the address in the MAC, and the MAC and the WC are both incremented by one. Then the flow of the program continues normally. Data breaks can occur whether or not the PIC is enabled. When the WC of either device overflows (reaches a value of 0), a regular interrupt flag is set. These flags, along with those from the clock, keyboard, teletype, paper tape reader, paper tape punch, light pen, CALCOMP plotter, or console, cause a break in the sequence of the program. When one of these occurs and the PIC is enabled, the program stores the address of the next instruction in the display loop to be executed, stores all pertinent registers, disables the PIC, and jumps to a routine to decide which interrupt occurred. That interrupt is serviced, pertinent registers are restored, the PIC is enabled, and the display loop is continued at the proper address. It is not unusual to have a situation in which new data are coming in from the ADC's, the clock is enabled and counting, old data are being read in from magnetic tape, the teletype is typing out information, and the display is being altered by means of the light pen all at once, the display continuing without noticeable flicker. If several interrupts occur at once, they are serviced sequentially in the order: Console, clock, ADC buffer overflow (see section IX), CALCOMP plotter, keyboard, teletype, light pen, magnetic tape, and paper tape reader. The paper tape punch is not used.

VI. CONSOLE INTERRUPT

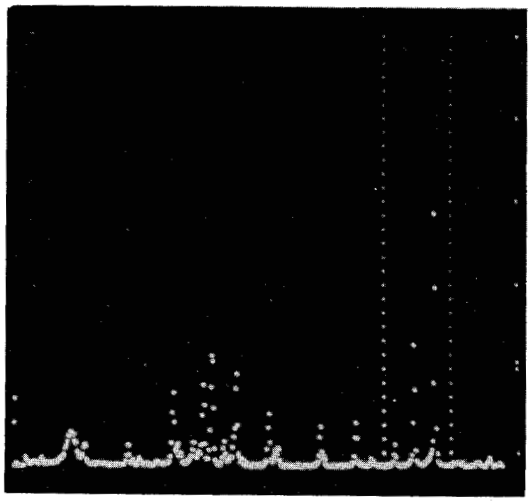
At the request of the buyer, our PDP-7[3] was equipped with one additional interrupt, a push button on the computer control panel which operates on the PIC, called the Console Interrupt. This button is used solely to cause a break in the program. It is programmed in FLOTSAM to be used as a quick and easy way to stop data taking, to shut off the teletype bell in case of a normal halt (timed run) or an error halt (ADC, Scaler) or to continue a run after it has been stopped by one method or another. Note that to start a run; that is, to reinitialize all pertinent counters, buffers, etc., it is necessary to type in an explicit instruction, STR.

VII. DISPLAY ROUTINE

The main program, the display program, is executed continuously if no interrupt occurs. Data can be scaled in the y-coordinate by manipulation of several of the AC switches (see section XIV). Other AC switches are used to indicate which block of data is to be displayed. Particular regions of these blocks can be expanded to fill the entire scope by means of the light pen (Fig. 1), and if precise starting and stopping channels are desired, they can be set from the keyboard, marker lines being displayed at the low and high channels. Conversely, the channel and contents of the low and high-marker lines can be typed out. Teletype commands are also included to change the form of the y-coordinate display. Normal mode is linear; LOG and square-root modes are available. The y-coordinate scale can be wrapped around or not on teletype command. (If the y-coordinate scale is such that the content of a channel would

[3]Digital Equipment Corporation, Maynard, Massachusetts.

a. Full spectrum; lines define region of interest.



b. Expanded region of interest.

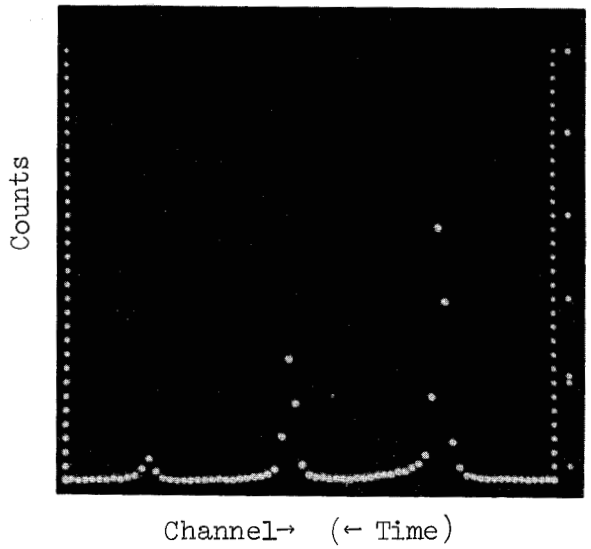


Figure 1. $^{58}Ni(d,n)^{59}Cu$ spectrum at 20° and 34.22 m flight path.

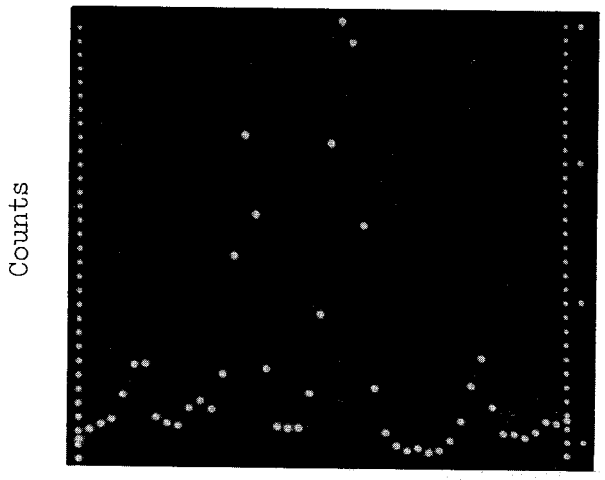
run off scale, it is wrapped around from zero. For no wrap-around this count takes the maximum y-coordinate value on the oscilloscope.) Finally, a teletype command exists to enable data stripping. This sets up the flow of the display loop so that one can also display the difference between a Foreground and the Background, and enables one to alter the spectrum in the Background to extract peak information.

The option of being or not being in Strip Mode exists because there are certain other operations which cannot be exercised while in Strip Mode. Log and Square-Root displays are disabled, and the BCD magnetic tape dump and the CALCOMP plotter routines are disabled. LOG and Square Root displays cannot be used in Strip Mode because the coordinates of oscilloscope points have to be descaled, and no exponential or square routines exist in this program. BCD dump and the plotter routines are stored in the Background spectrum.

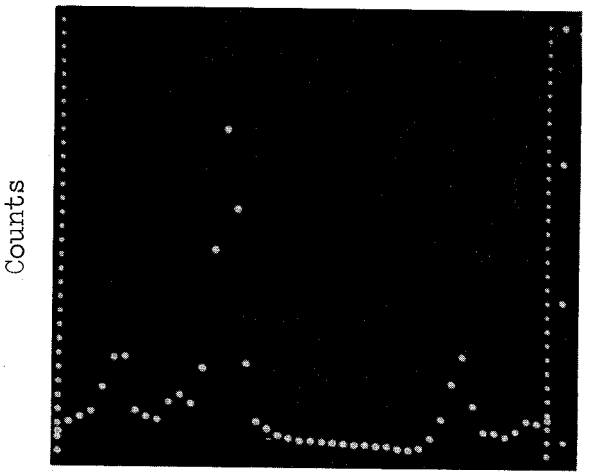
VIII. STRIPPING ROUTINE

If the Stripping Mode is enabled, it now becomes possible to strip away background events from bona fide peaks and to extract the area, centroid, and FWHM of individual peaks. This is done by setting the Background equal to a given Foreground, then altering the Background contents so that the peak of interest is eliminated. The program then calculates the difference between the Foreground and the Background regions and types out the peak information and which Foreground is being displayed. The difference has nonzero counts, of course, only in the peak (Fig. 2). The difference itself can be displayed.

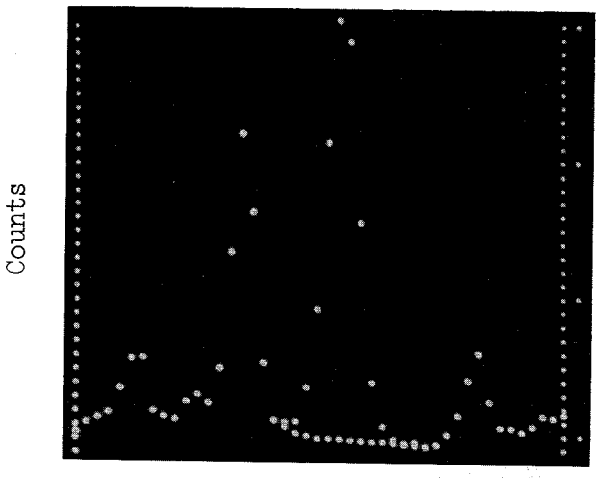
a. Region of interest in Foreground shown in Figure 1.



b. Background set equal to Foreground; peak of interest eliminated with a light pen.



c. Foreground plus Background.



d. Difference.

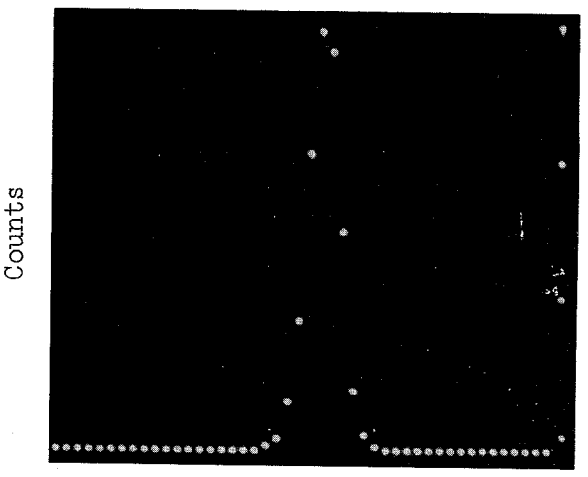


Figure 2. Oscilloscope displays illustrating the sequence of steps in stripping peaks of a spectrum.

Alteration of the Background is accomplished by means of a light pen following routine. A small grid or "strawberry" of 5 dots is programmed in such a way that the center point of the grid is positioned as nearly as possible directly beneath the light pen if the light pen detects any of the 5 points. As the light pen moves, the strawberry follows it. As the strawberry moves through the Background, it alters the counts in the channel nearest the center of the strawberry in such a way that the oscilloscope y-coordinate of the channel contents equals the y-coordinate of the center of the strawberry.

It is possible to strip a peak while counts are being added to the peak. This is for most purposes meaningless, although a person might, for instance, wish to strip a given peak and study its characteristics as a function of time. For most purposes, however, the following procedure is used. The input from the ADC's (see section IX) is so arranged that nothing goes into Group 7, the highest group. After a given run is completed and stored on magnetic tape and a new run is started, the Collapsed Spectrum or Monitor Spectrum from previous runs is read into group 7 on teletype command. These data are then stripped.

IX. DATA TAKING AND PROCESSING

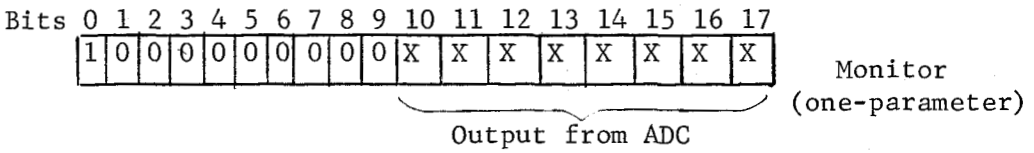
The TOF (Time-of-Flight) data are sorted into groups according to pulse heights in the following manner: the pulse height is converted by an ADC to a number between 0 and 128. Inputted through the keyboard are 8 "bin limits" in ascending order. Pulse heights < the first bin limit will fall in group 0. Pulse heights ≥ first bin limit but < second

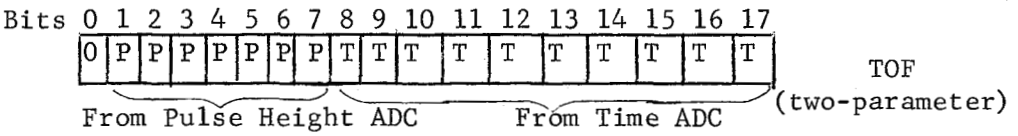
bin limit will fall in group 1, and so forth. There is one limitation. The first group whose bin limit $x \geq 63$ will receive all counts whose pulse height $\geq x$; that is, pulse heights of value y, where $63 \leq y \leq 128$, will fall into one group. It was decided through experience that only events with low pulse heights needed to be separated into groups.

Data are fed directly from the ADC's into the computer by means of the data break "cycle-stealing" feature into one of two buffers through an interface supplied by DEC[3] in the format of Table I. When one buffer is filled (WC overflow), the ADC interface is disabled and a flag to the PIC is set. The data-processing routine immediately switches buffers, re-enables the ADC interface, and starts to process the data in the full buffer event-by-event. The processor separates monitor events from time-of-flight (TOF) events, storing monitor events in the Monitor Spectrum and separating the TOF events according to pulse heights and time, storing events in the correct channels of the correct group. Then, it adds each TOF event to the Collapsed Spectrum, shifting channels for walk correction (the shifts are typed in beforehand), if the group into which the count falls is to be included in the Collapsed Spectrum. If a channel in any spectrum overflows, the channel, the number of times the channel overflowed and the spectrum in which the channel is located are sorted in an overflow buffer, which is stored permanently on magnetic tape and/or is typed out. Meanwhile, of course, the other buffer is being filled with data through the data-break feature.

For moderate counting rates ($\leq$1000 events/second) the computer causes essentially no addition to dead time. Recycling in the ADC's takes an average of 45 μsec for TOF ADC's and 33 μsec for the Monitor ADC.

Table I





The distinguishing feature of the two is ACO = 1 Monitor
= 0 TOF

Once every 64 events, when a buffer is filled, the ADC interface is disabled for about 45 μsecs, the time required to switch buffers. This time is small compared to the total time required to recycle the ADC's 64 times, and for an average of either 33 μsecs (Monitor) or 45 μsecs (TOF) of the 45 μsec used to switch buffers, the ADC's are recycling and cannot accept a new event. The processing time for a TOF event is about 58 μsecs and for a monitor event about 28 μsecs. We have found that the dead time in the TOF ADC's is about 50% when a sufficiently high counting rate has been reached for the computer to contribute to the dead time. Maximum counting rates, during which the dead time of the ADC's is ~100% and the display is blank (all the time is spent switching buffers and processing events), are ~17,000 TOF events/second or 35,000 monitor events/second. One would not, of course, normally run an experiment under these conditions.

Data-taking is initialized from the keyboard and then can be halted and continued by the Console Interrupt.

X. MAGNETIC TAPE HANDLING

Provisions are made to store data in either of two format: odd-parity binary code and even-parity BCD code. In normal usage storage is in binary since it is much more compact than BCD. Binary odd-parity storage consists of, per run:

Record 1: Run number (one 18-bit word)

Record 2: Collapsed Spectrum (512 words)

Record 3: Monitor Spectrum of 256 words plus 6 scalers stored as stripped ASCII (scalers are read in as decimal digits in BCD, the digits are converted to ASCII code, the

6 least significant bits of each ASCII digit are kept, and the stripped ASCII digits are stored three per 18-bit word), 3 words per scaler, the time of the run in seconds, the bin or pulse height limits on each group (8 words), the shifts corresponding to the groups (8 words), the first group to be collapsed, the last group to be collapsed, and the list of channel overflows (see section XIV) which may have occurred (319 words total).

Records 4-19: The 8 groups of uncollapsed data, 256 channels per record.

The runs are separated by an EOF (End of File), and the last run on the tape is terminated by two EOF's.

Runs stored in binary can be read back into the machine entirely or in part at the desire of the user.

BCD even-parity storage consists of, per run, in 120-character records:

Record 1: 1 (one), blank, 5-digit run number, 113 blanks

Records 2-257: 8 groups of 512 channels, each record in the format: (1X, I4, 1617, 3X) where the first number is the number of the first channel of that record, followed by 16 channels of information.

Record 258: 1 (one), 1 (one), 5-digit run number, 113 blanks

Records 259-291: Collapsed Spectrum in the (1X, I4, 1617, 3X) format.

Record 292: 1 (one), 2 (two), 5-digit run number, 113 blanks

Records 293-309: Monitor Spectrum in the (1X, I4, 1617, 3X) format. Runs are separated by an EOF; last run is terminated by 2 EOF's.

Data stored in BCD cannot be read back into the machine. It can, however, be read by standard FORTRAN programs on a variety of other computers. BCD dump uses so many autoindexers and so much temporary storage that no other teletype commands are recognized while the dump is in progress. Furthermore, the BCD dump routines are stored in Background; Strip Mode and INT both modify Background and disable BCD. If this happens, the BCD routines must be reloaded from paper tape on teletype command. This can be done while data are being taken.

XI. TELETYPE INPUT/OUTPUT

The keyboard is used extensively to control the program. With the exception of the light pen and the AC switches, used for manipulating the display, and the Console Interrupt, used to stop and continue data-taking, all commands are inputted through the keyboard. There are 48 legitimate commands to the keyboard, all of which are described in section XIV. Checks are made to ignore illegitimate commands. All data stored permanently on magnetic tape can, with a series of instructions, be typed out on the teletype. Any part or all of the data can be typed out and a running sum given if desired. In addition, there are several instances in which the program will type out information on its own. For instance, if one sets a timed run, the program automatically types out EOT (End of Time) and rings the bell at the proper time. If the program encounters an error occurring in the ADC's or in the scalers (see Data Manipulation and Monitoring) it will automatically

stop data-taking, type out the type of error encountered, and proceed to ring the bell. Finally, if the channel overflow list is filled, data-taking is stopped, the list is typed out and re-initialized, and the run is continued.

XII. CALCOMP PLOTTER

Provisions have been made to allow plotting of data on a CALCOMP plotter. Because of space limitations, plotting cannot take place while stripping peaks in Strip Mode. The plotting package is stored in the Background Spectrum, and both Strip Mode and the instruction INT alter Background. If Background has been altered, one must reload the plotting package. This is done with a teletype instruction, LPT, which loads the paper type containing the plotting package and enables the plotter. This can be done while data are being taken. Furthermore, one can do actual plotting while data are being taken. The routine will plot any Foreground, including group 7. Thus, old data can be read in from magnetic tape and plotted while a new run is in progress.

XIII. HARDWARE CHANGES TO THE PDP-7

Several changes and additions in electronics and interfaces have been made. An interface has been added which enables a read-in of a bank of six 7-digit scalers into memory. Another interface has been added to enable a CALCOMP plotter. The input/output (IOT) skip flags have been modified so that a skip indirect (i.e., KSF I) will cause a skip if the flag is not set. A normal IOT skip will work in the expected manner, i.e., skip if the flag is set. The magnetic tape

drives have been rewired to work properly under all circumstances under PIC control. Magnetic tape control has been altered so that, if desired, only the least-significant 6-bits of a word are dumped on tape. The teletype has been rewired to turn itself on if output is desired and the teletype is turned off.

XIV. OPERATING INSTRUCTIONS

To load program: for an HRI (Hardware Read-In) mode paper tape, set address switches to 20_8, load paper tape, and push READ-IN. For an assembled-version tape, set address switches to 17763_8, load RIM mylar tape, push READ-IN. Set address switches to 17770_8, load program paper tape, push START. After tape is read in, push CONTINUE. To load from the Systems Tape (magnetic tape), load Systems Tape on Tape Drive 0, set address switches to 17763_8, load UGGYCHUG mylar tape, push READ-IN. Type in FLOTSAM!

Normal starting address is 22_8. The HRI paper tape can be reloaded without affecting any spectra which may be in the computer. The others destroy parts of data storage.

All output numbers are in decimal. Input numbers are decimal or octal. The program starts out in decimal mode. The instruction OCT changes to octal mode, and the mode stays in octal until DEC is typed. Similarly, once in decimal mode, it stays there until OCT is typed.

For typing in numbers: A comma ends the number. Numbers larger than 262143_{10} are in error. To restart a number in case of a mistake, type any alphabetic character. The teletype types a ? in case of any such illegal character.

When typing in commands, only the last three characters are used. All commands are three-character mnemonics. End all commands with a carriage return or a space. The carriage return gives a line feed also. If an illegal command is given, computer types "??!".

COMMAND	MEANING
IBS	Initialize. Types out "BINLIMS." Type in eight dynode group limits. Types out "SHFTS,LO,HI." Type in eight channel shifts, lowest group number and highest group number for walk correction. Groups are numbered 0 through 7. New values take effect only on STR.
BIN	Types out "BINLIMS." Type in bin limits only.

COMMAND (con't)	MEANING (con't)
STS	Types out "Shfts, LO, HI." Type in shifts, low and high groups.
BSL	Types out bin limits, shifts, and low and high groups, in that order. This will be typed automatically at the first data-taking command after loading program. Also, any time a complete (GT2, GT7) run is read in from magnetic tape, the BSL is done at the first data-taking instruction, since the BSL contents are stored on tape and are read back in by GT2 or GT7.
STR	Start. Zeros spectra, resets scalers and overflow buffer, and accumulates data until stopped by a console interrupt or until time equals that set by SET. Time can be set at any time during the run. One can continue a run after a timed halt, and type in a new time at which to stop. A set time of zero minutes will run forever. At end of time, teletype types out "EOT" and rings the bell for 15 seconds, after which it acts as a Console Interrupt. One can shut off the bell with an actual Console Interrupt.
ZSG	Types "GP=". Type in a group to be zeroed.
SET	Sets time for a timed run. Types out "TM=". Type in the number of minutes for which run is to be taken.
TS=	Types out running time to the nearest second. If the program is not taking data, the time is that of the previous run taken or that of the last run read in from tape by a GT7 or GT2.
Console Interrupt	Halt. Stops data taking, dumps scalers into memory, dumps time of run in seconds into memory and types out any channel overflows which may have occurred (see OVR). If Interrupt is pushed again, the run is continued, then stopped, etc.
OVR	Types out channel overflows, if any, in the format: XYYYYY Z where X = 0 for collapsed, 1 for monitor, 2 for full, YYYY is the channel, Z = number of times the overflow occurred. If

COMMAND (con't) MEANING (con't)

more than 13 channels overflow, the list is typed out and the list is reinitialized. The first list is not stored on magnetic tape. NOTE: One overflow = 262144_{10} = 1000000_8 counts.

Display

Switches. A display of either the monitor or the dynode groups and/or the Background spectrum and/or the collapsed spectrum is possible. Full scale horizontal is equal to 256 for monitor, 512 for others. The accumulator switches are labeled as: $XXMG_0G_1G_2G_3G_4G_5G_6G_7BCSPPPP$.

AC0, AC1 (XX) are normally inoperative.

SPPPP gives the vertical full scale as $2^{S(PPPP)} \times 2^{10}$.

$S = \begin{matrix} 1 \rightarrow - \\ 0 \rightarrow + \end{matrix}$.

C displays collapsed spectrum.

B displays background spectrum.

G_{0-7} display dynode groups.

Any or all of the above may be displayed simultaneously.

M displays monitor. If this switch is up, the switches G_0 through C are ignored.

LOL Sets low line. Types out "LO=". Type in channel for low line.

HIL Sets high line. Types out "HI=". Type in channel for high line.

KLN Kills lines.

LNS Types out channel and contents of low and high lines. (Use single display for correct contents.)

LO= Types out channel and contents of low line.

HI= Types out channel and contents of high line.

LIN Display linear vertical scale. S(PPPP) applies.

SQT Display square root vertical scale. S(PPPP) applies; full scale = $2^{S(PPPP)} \times 2^2$.

COMMAND (con't)	MEANING (con't)
LOG	Display log vertical scale. S(PPPP) does not apply; numbers <4 suppressed; $\log_2 4$ = base line; $\log_2 1024_{10}$ = full scale.
WAY	Wrap Around Yes. If y-coordinate is such that a channel would run off scale, it is wrapped around from 0.
WAN	Wrap Around No. If y-coordinate of a channel $>1777_8$, it is set to 1777_8.

Light Pen

Seven dots are displayed down the right side of the scope

.6		.6
.5		.5
.4		.4
.3	or	.3
.2		.7
.7		.2
.1		.1

Dots 1 and 2, with Dot 7 next to one of them, indicate which of the lines are active. For instance, if Dot 1 has Dot 7 above it, the low line is active. This means that if one touches with the light pen a data point of any spectra being displayed, the low line will jump to that point. The high line (Dot 2) works similarly. One switches from one to the other by touching the respective dots.

Dot 3 displays full spectrum.

Dot 4 expands between lines. If low line $\geq$ high line, Dot 4 is inoperative.

Dot 5 causes the spectrum or spectra being displayed to rotate to the left, points disappearing on the left and new data points appearing on the right.

Dot 6 causes the spectrum or spectra to rotate to the right.

COMMAND (con't)	MEANING (con't)
REC	Recollapses spectra, puts in Collapsed. Types out "LG=". Type in low group to use. Types out "HG=". Type in high group to use. Shifts used are from IBS or STS.
CRC	Same as REC, but adds to Collapsed.
	Data Output
RSL	Gives channel contents and running sum of channels included by markers for collapsed, Background, dynode groups, and monitor spectra. Be careful to display only ONE spectrum at a time for this. Otherwise it will type out the one corresponding to the first up switch from the left.
TOL	Same as RSL, but without the running sum.
SCA	Types out contents of scalers, updated every 5 minutes. Types out XXXXXXXXX where the first two digits are the number of overflows. If program is not taking data, the scalers are that of last run taken or of last run read in by a GT7 or GT2.
TYO	Types out "N:"; type in the number N. Types out "M:"; type in the number M. Program then types out in decimal N words starting at the machine address M.
TYI	Same as TYO, except after M, type in N words starting at the machine address M. NOTE: for N, M, and type-in, one can type in either octal or decimal.
	Tape Handling
DMP	Dumps in binary onto tape on drive 2. If tape drive 6 is active, the same run is dumped in binary on it also. The program will find the proper starting point on either tape.
BNT	Begin New Tape. Assumes a new tape(s) is on drive 2(6). Program will not search for end of file. ALL AC SWITCHES MUST BE UP, or the program will not dump; it will type ??!. If this happens one must retype BNT.

COMMAND (con't)	MEANING (con't)
BCD	Dumps a run in BCD on tape drive 3. Assumes tape is properly positioned. Start new tape at load point. One cannot be in STRIP MODE when dumping in BCD, and when in BCD, keyboard is ignored. See LPT.
EOT	Positions a BCD tape properly, if it has previously been dumped on, and additions are desired. Does not dump.
	NOTE: The procedures for dumping binary and BCD are quite different. BCD assumes the tape is positioned; one needs an explicit instruction, EOT, to position it if it has been unwound. EOT does not dump anything. DMP, however, will always search for the proper starting point. BNT will dump in binary, but will not look for an end of tape.
GT2	Retrieve a run in binary on tape 2. Write ring does not have to be in. Retrieves everything which was dumped.
GT7	Same as GT2 for tape drive 7.
GTC	Retrieve the collapsed spectrum of a run on tape drive 2, store in group 7. Does not affect bins, shifts, etc.
GTM	Retrieve the monitor spectrum of a run on tape drive 2, store in group 7. Does not affect bins, shifts, etc.
RUN	Types out "OLD RUN = XXXXXX, NEW RUN =". Type in new run. Types out "NEW RUN = XXXXXX'". A + gives old run + 1. A - gives old run - 1. A space gives old run. If one makes a mistake (? typed out), the old run number is destroyed; one must type in a number.
BSP	Backspace n runs (files) on tape drive 2. Types "NF="; type in number of runs to backspace.
	If machine halts after receiving any magnetic tape operation instruction, chances are the tapes are not on proper drive, not on remote, or do not have a write ring (if necessary for the operation). Correct the problem and push CONTINUE if computer does not continue on its own.

COMMAND (con't)	MEANING (con't)
	Strip Routine

ESM ENTER STRIP MODE. One can display only in linear (LIN) during this time. It will set linear mode itself. While STRIP is in effect, LOG, SQT, PLT and BCD instructions are inoperative. Switches AC0, AC1, are now operative. AC0 down allows one to draw backgrounds in the background spectrum. AC0 up disables the strawberry mark. AC0 down also displays Background regardless of the "B" switch. If AC1 is up, the normal 7 dots appear across the right of the screen. They work now on any spectrum except Background. If AC1 is down, 4 new dots appear.

.4
.3
.2
.1

Dot 1 if touched will type out the difference between any group, Collapsed, or Monitor, henceforth called Foreground, and the Background, if Foreground is displayed. It types out YYY A = XXXXXX C = ZZZZ.ZZ W = VVVV.VV where

YYY is the Foreground,

A = area,

C = centroid,

W = full width at half maximum, $(8\ell n2\sigma^2)^{1/2}$, where σ^2 is the variance,

XXXXXX is modulus 100000000_{10}.

In the calculation of W, only those points whose value $\geq 1/64$ X maximum point are used. Thus $W_{cal} \leq W_{actual}$. If for some reason the program hangs up in typing out, and refuses to type out again when the dot is touched, type in ESM again.

Dot 2 sets Background equal to the Foreground being displayed, in the region being displayed.

COMMAND (con't) MEANING (con't)

Dot 3 displays the difference between Background and the Foreground being displayed. One can display the difference, Background, and the Foreground all at once, or any combination of the three.

Dot 4 deletes display of the difference.

STRIP works while data are being taken. The binary tape read-in routines work while data are being taken. Thus, if one arranges BIN LIMITS so that nothing goes into group 7, one can take data and read in old data into group 7 and strip it.

LSM Leave STRIP MODE.

INT Works like Dot 1 of Strip Mode, except it sets Background = Foreground in the region being displayed and then zeros the region between the lines. Effectively, then, it gives the number of counts between the lines, their centroid, and their FWHM. Display only one Foreground and no Background for correct answers. This instruction disables PLT and BCD.

CALCOMP Plotter

PLT Plot the Foreground displayed, rounding off first and last channels to nearest unit of 20. Disables ESM and INT. One must be in linear mode to plot. Plots in linear scale, plots the run number and the power of 2 which equals the maximum value on the plot. See LPT.

LPT If BCD or PLT give a ??!, one must load the small paper tape and type LPT, which will load the CALCOMP plotter and the BCD routines into Background. This instruction automatically enables PLT and BCD and leaves Strip Mode.

COMMAND (con't) MEANING (con't)

Error Messages

The program is designed to do simple checks on the electronics. It assumes that, of the scalers, $3 > 4$, $6 > 5$, and $4 > 6$. Every five minutes of running time the program automatically checks these relations. If an error is found, the run is stopped and the program types one of these messages

(the first error it finds)

$3 < 4$ SCALER ERR

$6 < 5$ SCALER ERR

$4 < 6$ SCALER ERR

Then it begins to ring the bell until it is stopped by a Console Interrupt. Pushing Console Interrupt a second time will continue the run, as usual.

The program also checks the ADC's against each other. If either the monitor ADC or the TOF ADC's accept 9 full buffers of data (576 events) while the other does not accept a single event, the run is stopped and one of the following is typed:

TOF ADC ERR	if no TOF events occurred
MON ADC ERR	if no MON events occurred.

Again the bell is rung until stopped by a Console Interrupt, as above.

Sometimes the computer senses an incorrect reading of the scalers. If this happens, the program types

SCALER ERR

and continues the run. No relative scaler checks are made in such a case. If this happens on a Console Interrupt, one must stop the run again, until the scalers read correctly. Similarly, a SET stop would be overruled if an incorrect scaler reading were detected.

SCN	Delete scaler check.
SCY	Restore scaler check.
ACN	Delete ADC check.
ACY	Restore ADC check.

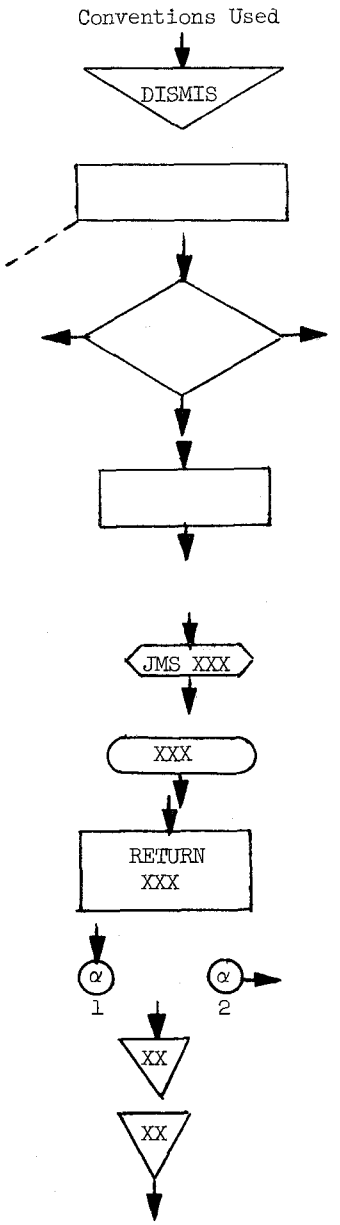
ACKNOWLEDGEMENT

The author wishes to thank W. E. Kinney, who wrote an earlier version of this data-taking code, upon whose work this program is largely based, and who wrote the CALCOMP plotter package, F. G. Perey and J. K. Dickens, who wrote earlier versions of the magnetic tape binary and BCD codes, respectively, J. W. McConnell for his care of and hardware additions to the computer, and finally J. A. Biggerstaff for his continued interest, suggestions, and aid in the writing and debugging of this program. Major ideas for the data processor and light pen routines and many space-saving features are J. A. Biggerstaff's. All of the people mentioned above contributed numerous ideas which were incorporated in the code.

FLOW CHARTS

These are a few flow charts of some of the more important sub-programs; they are meant to illustrate the general form of the program. Symbol definitions can be found in Digital Equipment Corporation publication F-75, "Users Handbook", (unpublished).

CONVENTIONS USED



Return to main program after servicing interrupt

Comment box, connected to flow chart with dotted line

Decision box with variable exits

Function box

Call a subroutine

Entry to a subroutine

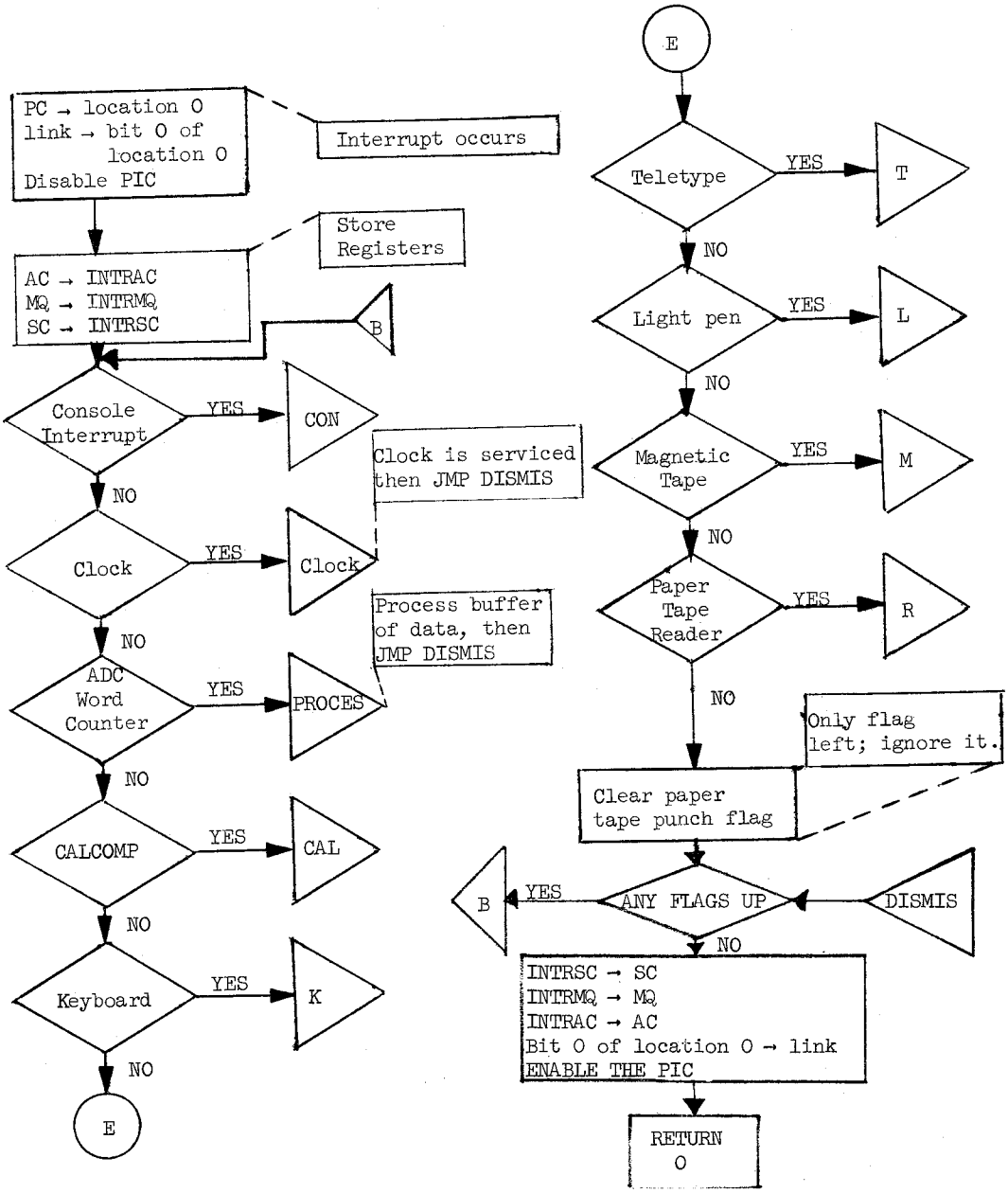
Return from a subroutine to the calling program

Denotes a break in the flow chart or in program; that is, program flows from 1 to 2.

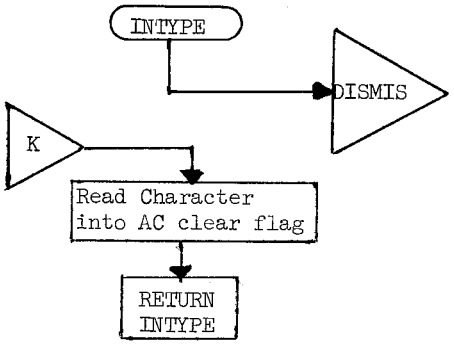
Unconditional jump to XX

Entry point XX

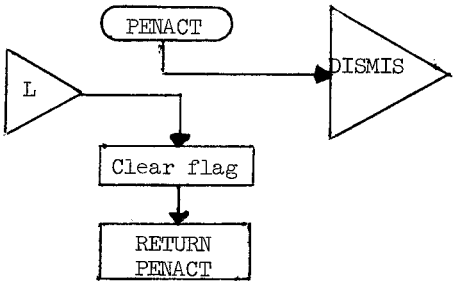
Interrupt Handling - PIC is Enabled



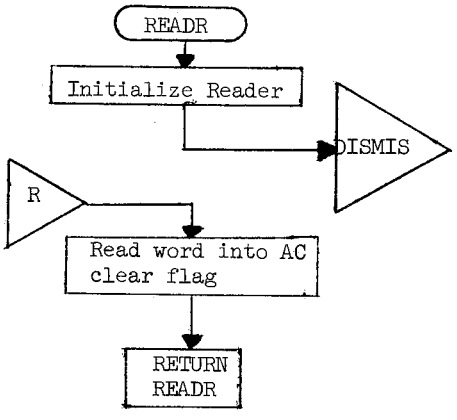
KEYBOARD



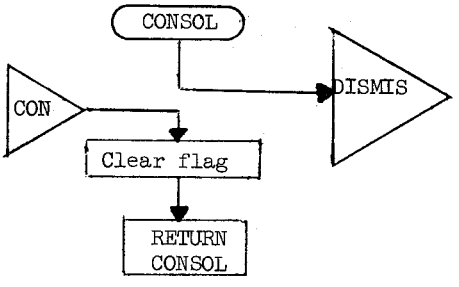
LIGHT PEN



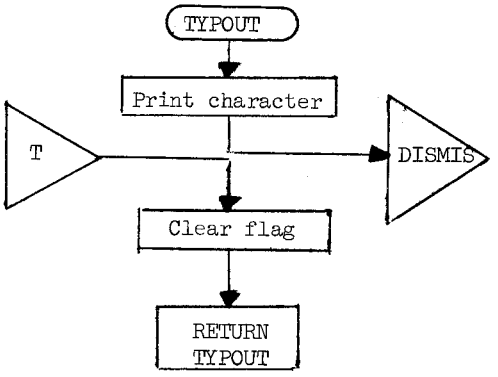
PAPER TAPE READER



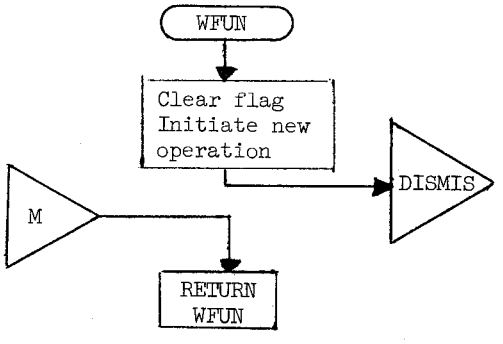
CONSOLE INTERRUPT



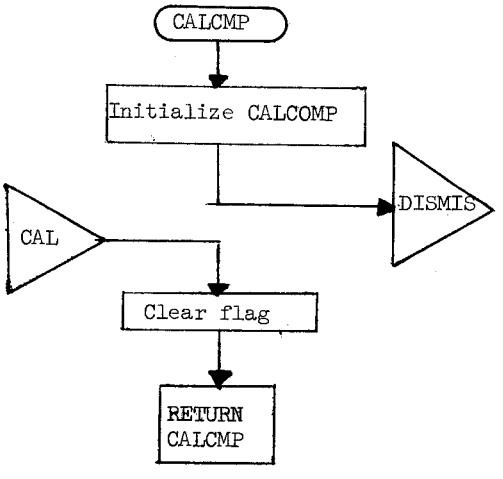
TELETYPE



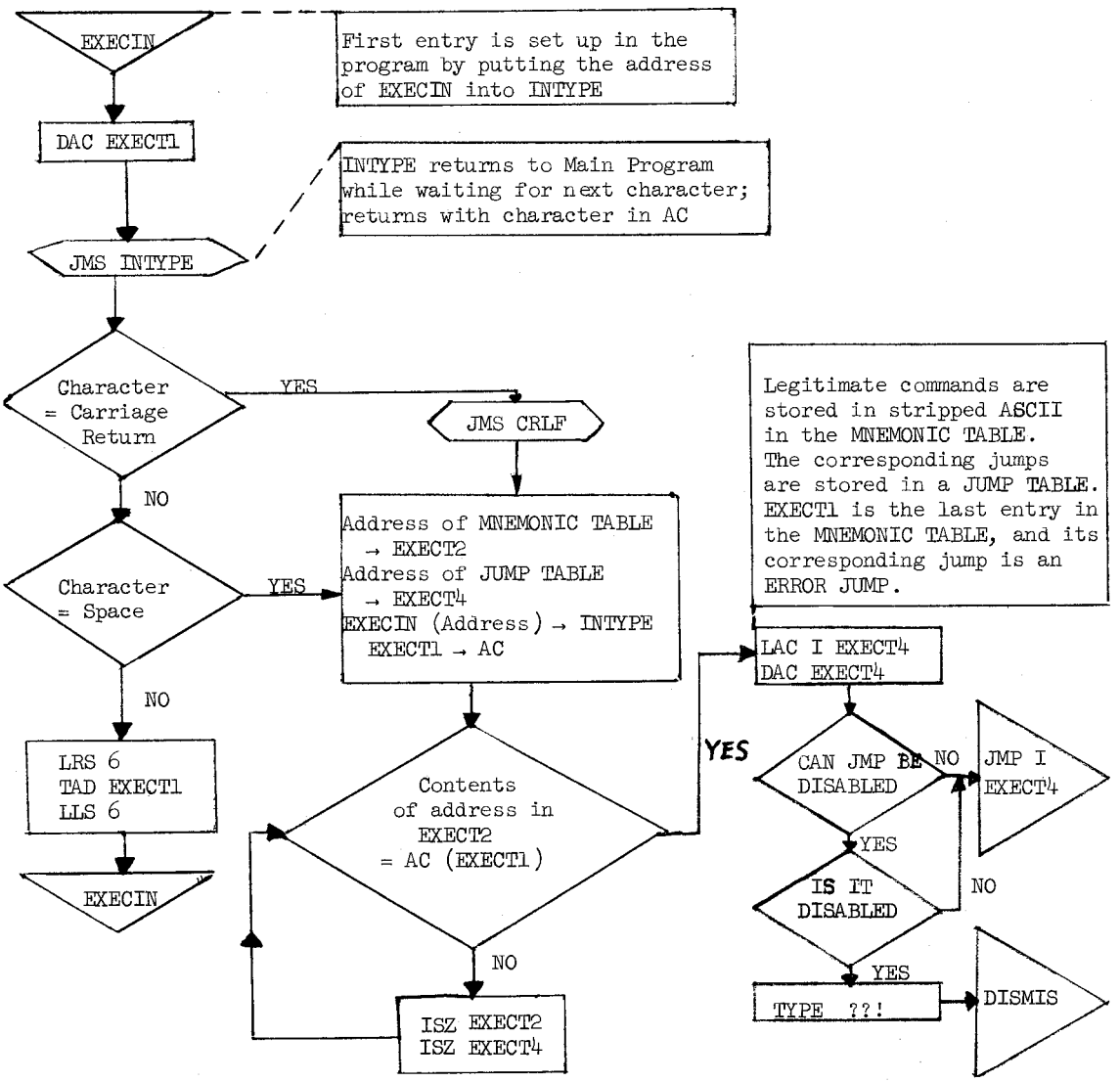
MAGNETIC TAPE



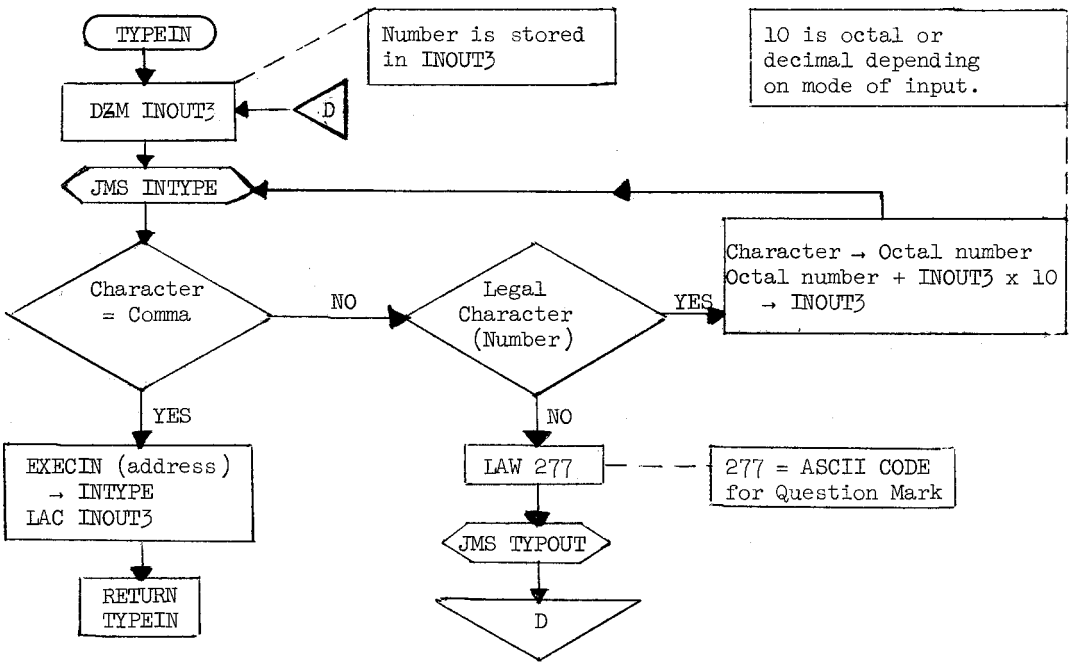
CALCOMP PLOTTER



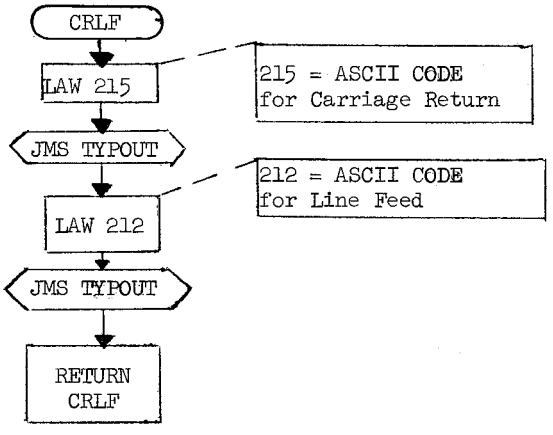
EXECUTIVE ROUTINE TO SERVICE KEYBOARD COMMANDS



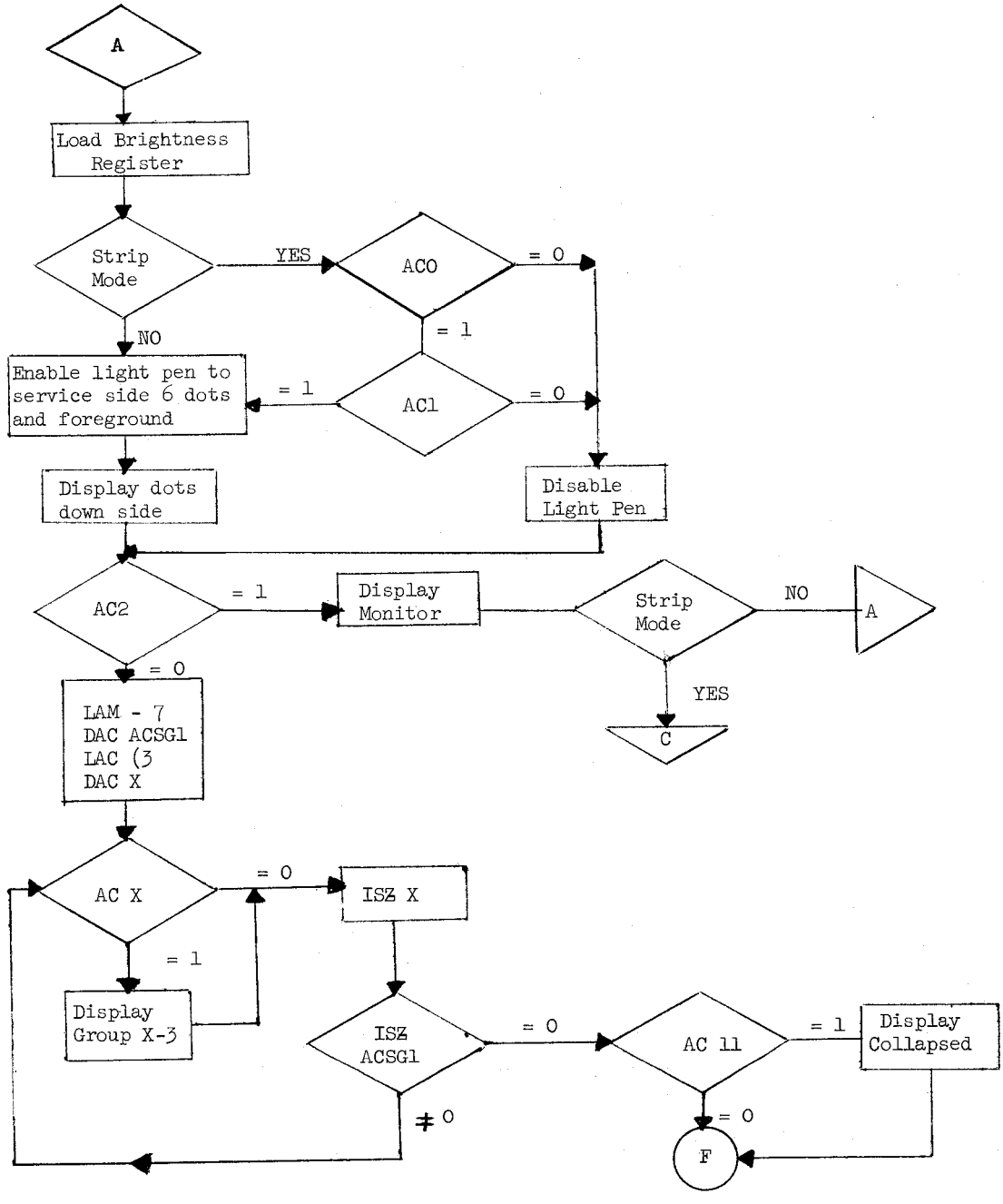
ROUTINE TO TYPE IN A NUMBER

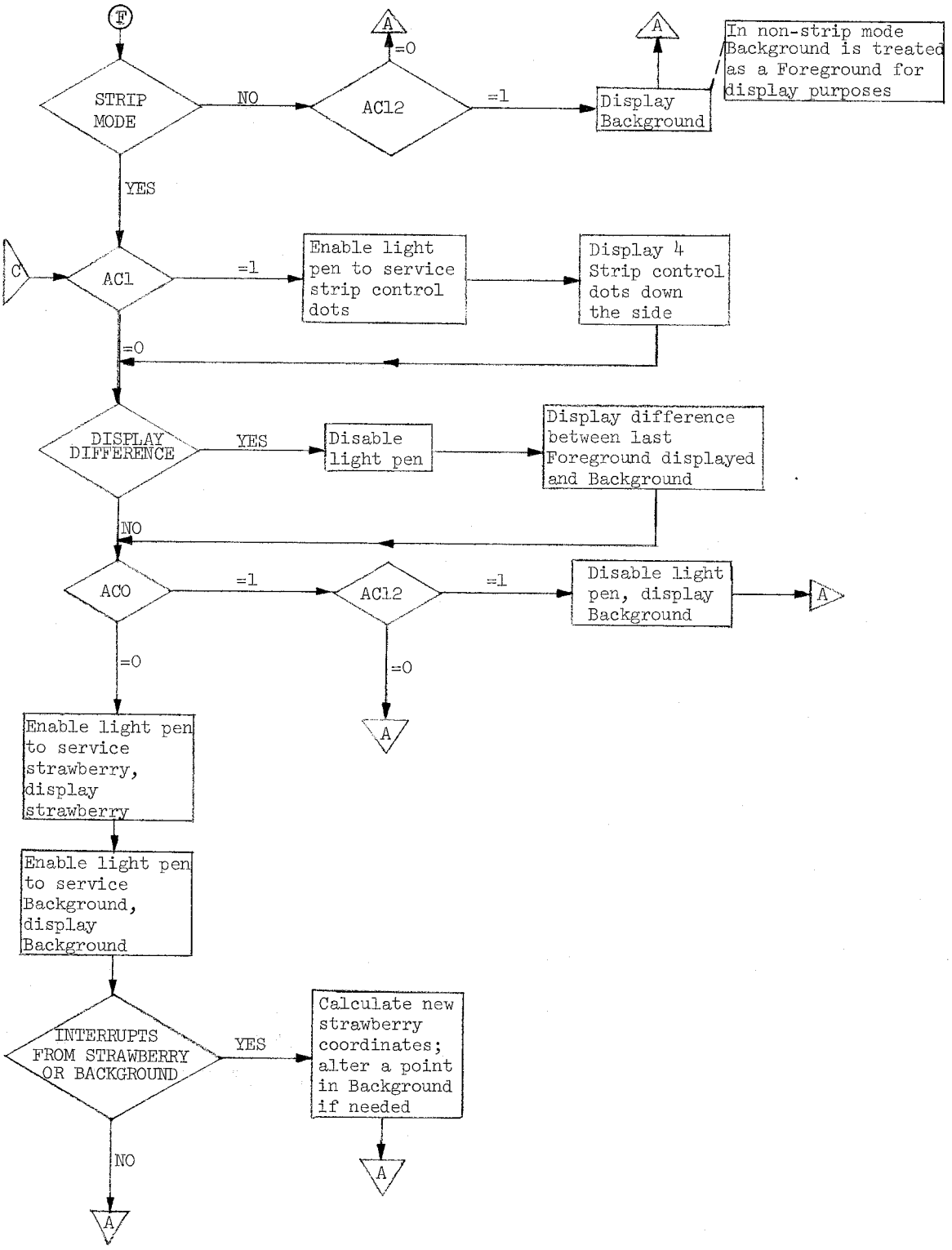


CARRIAGE RETURN, LINE FEED



DISPLAY LOOP - MAIN PROGRAM




F
STRIP MODE
NO
A
=0
AC12
=1
A
Display Background
In non-strip mode Background is treated as a Foreground for display purposes
YES
C
AC1
=1
Enable light pen to service strip control dots
Display 4 Strip control dots down the side
=0
DISPLAY DIFFERENCE
YES
Disable light pen
Display difference between last Foreground displayed and Background
NO
ACO
=1
AC12
=1
Disable light pen, display Background
A
=0
=0
A
Enable light pen to service strawberry, display strawberry
Enable light pen to service Background, display Background
INTERRUPTS FROM STRAWBERRY OR BACKGROUND
YES
Calculate new strawberry coordinates; alter a point in Background if needed
A
NO
A

```
FLOTSAM    FLIGHT OF TIME SYSTEM BY ALEX MARUSAK
/JULY 29, 1968.
/EQUALITIES FOR 8 BY 512 TWO-PARAMETER DATA TAKING
/SOME OF THE FOLLOWING ARE PARAMETERS WHICH MAY BE CHANGED,
/DEPENDING ON THE NUMBER OF GROUPS DESIRED.
/OTHERS DEFINE ABSOLUTE ADDRESSES TO BE USED IN THE PROGRAM.
BEGIN=24
KEYMSK=177640    /174040 FOR 4 GROUPS. MASK THE FOREGROUND SWITCHES.
TOFDIM=1000      /2000 FOR 1024, 400 FOR 256, ETC.
SHFTOF=11        /12 FOR 1024, 10 FOR 256, ETC. (NO. OF BITS IN TOFDIM)
COLLO=6000       /5000 FOR 1024, 6000 FOR 256 (ADDRESS OF COLLAPSED)
RECOLO=5000      /ADDRESS OF RECOLLAPSED SPECTRUM
MONTOR=COLLO TOFDIM              /ADDRESS OF MONITOR
MONSZE=400       /NUMBER OF CHANNELS IN MONITOR
UNCOLL=10000     /ADDRESS OF 4096 CHANNELS
UNCSZE=10000     /NUMBER OF CHANNELS IN 4096
BINGRP=10        /NUMBER OF GROUPS
PULSZE=100       /PULSE SIZE PERMITTED IN ADCS
/TOFMSK IS A MASK USED FOR CHANNELS.
TOFMSK=TOFDIM-1
SCLR=MONTOR MONSZE               /SCALER STORAGE IN STRIPPED ASCII STARTS HERE
CLCK4=SCLR 22    /TIME IN SECONDS STORED HERE
BINLIM=CLCK4 1   /BIN LIMITS STORED HERE
SHFTS=BINLIM BINGRP              /SHIFTS STORED HERE
LLOW=SHFTS BINGRP                /LOWEST GROUP TO BE COLLAPSED
LHI=LLOW 1       /HIGHEST GROUP TO BE COLLAPSED
JSZ=32           /NUMBER OF OCTAL SPACES TO RESERVE IN OVERFLOW LIST
JLIST=LHI 1      /OVERFLOW LIST
PTABLE= JLIST JSZ                /TABLE OF INSTRUCTIONS FOR PROCESSOR
BUFSZE=100       /NUMBER OF WORDS IN DATA BUFFER
BUF00=PTABLE PULSZE              /WORD BUFFER FOR WORDS FROM ADCS
BUF01=BUF00 BUFSZE               /SECOND WORD BUFFER
WGRP=20          /NO GROUPS TO BE DUMPED ON BINARY TAPE FOR UNCOLL.
WGRPN=400        /NO WORDS PER GROUP (WGRP TIMES WGRPN=NO CHAN IN UNCOLL)
JKD1=400         /NO OF RECORDS ON BCD FOR UNCOLLAPSED
JKD2=40          /NO OF RECORDS ON BCD FOR COLLAPSED (100 FOR 1024)
JKD3=20          /NO OF RECORDS ON BCD FOR MONITOR
BONGRP=BINGRP BINGRP
/THE FOLLOWING IS A LIST OF THE SUBROUTINES WHICH USE AUTO INDEXING.
AUTO10=10        /ZERO,NMIND,NMSTAS,SCLRTP,BINBCD,CLOCKS,LPT
AUTO11=11        /PROCESSOR
AUTO12=12        /RSUMQ,BCDPRT,BINBCD
AUTO13=13        /RSUMQ,F1FORM
AUTO14=14        /REC,WOTF1,NOROUT,STRIP2
AUTO15=15        /REC,STRIP3,STRIP4,F1FORM
AUTO16=16        /RED,STRIP3,STRIP4,F1FORM,INRUN,BCDPRT
AUTO17=17        /PEN1,DISPLA,VAL,STRIP2,STRIP5,STRIP6
/THE FOLLOWING DEFINES THE AREA USED AS TEMPORARY STORAGE BY THE CALCOMP.
/DATA MAY BE READ INTO THIS AREA BEFORE, BUT NOT DURING PLOTTING.
CALBUF=17000
/THESE LOCATIONS ARE USED AS TEMPORARY STORAGE BY THE BCD DUMP ROUTINES
BCD4=PTABLE
BCD5=BCD4 1
WLOOP=BCD5 1
BCD2=WLOOP 1
OUTBCD=BCD2 1
BCDRUN=OUTBCD 1
/BCDARY NEEDS 170 OCTAL LOCATIONS
BCDARY=BCDRUN 7
PAUSE BEGIN
```

```
/DISPLAY--MAIN PROGRAM
/BOOTSTRAP ROUTINE TO SET UP INTERRUPT HANDLING
/AND TO INSURE THAT FIRST KEYBOARD INTERRUPT GOES TO THE EXECUTIVE ROUTINE
20/             SKP
                HLT
                LAM-3           /ENABLE EVERY COMMAND BUT PLT,BCD.
                DAC CODEWD
BEGIN,          LAC (JMP INTRUP
                DAC 1
                LAW EXECIN
                DAC INTYPE
                CAF
                JMS STBM        /CAF ZOTS ADC WC; RESET IT
/SET UP AC SWITCH DISPLAY.
ACSRES,         LAS!CLL         /REF FOR LAS!CLL IN STRIP,NOSTRP,VAR3A
                ALS 2
                SPA!CLA
                LAM-TOFDIM MONSZE 1
                TAD (TOFDIM
                JMS DISPFR
/DISPFR STORES NUMBER OF POINTS TO BE DISPLAYED IN DISPFS
/AND INITIALIZES THE ROTATE SUBROUTINES.
                DZM DIS#PLN     /DISPLN CONTAINS THE STARTING CHANNEL
                ION             /READY TO ENTER INTERRUPTABLE PROGRAM
/PRECEDING INSTRUCTION REF AS ACSDIS-1 IN VAR3A
ACSDIS,         LAW 3
                DLB             /LOAD BRIGHTNESS REGISTER
                LAM             /REF AS ACSDIS 2 IN STRIP,NOSTRP
/A LAM IS INTERPRETED AS AC0, AC1 BOTH UP IN NON-STRIP MODE.
/PRECEDING INSTRUCTION CHANGED TO LAS!CLL IF IN STRIP MODE
                SPA!CLL
                RTL
                LAW DISMIS
                SZL
                LAW PENDO
                DAC PENACT
                SNL
                JMP . 11
                LAM-300 1
                JMS MAP         /DOTS DOWN THE SIDE
                LAC LINWHR      /=0 FOR LOW LINE ACTIVE; =1 FOR HIGH LINE
                SZA
                LAW 300
                TAD (60         /ACTIVE LINE INDICATOR
                DYS
                ISZ WCHPNT      /INDICATES WHICH POINT (DOT) WAS DISPLAYED
                LAM-BINGRP 1
                DAC ACS#G1
                LAS!CLL
                ALS 2
                ISZ WCHPNT      /AFTER SUITABLE DELAY FROM LAST ISZ
/NOW WCHPNT INDICATES A LIGHT PEN INTERRUPT CAME FROM A DATA POINT.
                SPA!RAL
                JMP MONDIS      /AC2 UP=> MONITOR
/DISPLAY GROUPS AND COLLAPSED, IF APPROPRIATE SWITCHES ARE UP.
                SMA!RAL
                JMP . 10
                DAC ACS G2      /DISPLAY GROUP IF SWITCH IS UP
                LAW BINGRP
                TAD ACSG1
```

```
                CLQ!LLS SHFTOF
                TAD (UNCOLL
                JMS DISPLA
                LAC ACSG2
                ISZ ACSG1
                JMP .-12
                ALS 11-BINGRP   /PUT COLLAPSED SWITCH INTO AC0
                SMA
                JMP . 3
                LAC (COLLO      /DISPLAY COLLAPSED IF AC12 IS UP
                JMS DISPLA
STRIP1,         LAS!CLL         /CHANGED TO JMP STRIP2 IF IN STRIP MODE
                ALS 13
                SMA
                JMP ACSDIS      /REF AS STRIP1 3 IN NOSTRP
                LAC (RECOLO
                JMP MONDIS 6
/MONITOR DISPLAY
MONDIS,         LAM-MONSZE
                TAD DISPLN
                TAD DISPFS
                SMA
                JMP ACSRES      /REINITIALIZE IF DISPLAY OUT OF RANGE OF MONITOR
                LAC (MONTOR
                JMS DISPLA      /REF AS MONDIS 6 IN STRIP1
                JMP ACSDIS      /REF AS MONDIS 7 IN STRIP, NOSTRP
MAP,            XX              /DRAW DOTS DOWN THE SIDE
                DAC DISPLA      /DISPLA USED AS TEMPORARY STORAGE
                DZM WCH#PNT
                LAW 1777
                DXL
                LAC (1740
                DYS
                ISZ WCHPNT
                SAD (40
                JMP I MAP
                TAD DISPLA
                JMP .-5
```

```
/DISPLAY ONCE DISPFS POINTS, STARTING AT DISPLM.
/THIS IS THE MAIN ROUTINE FOR DISPLAYING DATA.
/DATA IS SCALED IN THE Y-COORDINATE BY THE SWITCHES
/AND IN THE X-COORDINATE BY THE NUMBER OF CHANNELS TO BE DISPLAYED.
/IF IN SQUARE ROOT OR LOG DISPLAY MODE, APPROPRIATE ROUTINE IS CALLED.
/PROPER ROUTINE IS CALLED IF NEEDED FOR NO VERTICAL WRAP-AROUND.
/ROUTINE TO DRAW LINES IS CALLED BY THIS ROUTINE.
/ROUTINE IS ENTERED WITH ADDRESS OF REGION OF INTEREST IN THE AC.
DISPLA,         XX
                TAD DISPLN      /ADD STARTING CHANNEL TO BEGINNING OF REGION
                DAC DIS#PLM     /DISPLM COUTAINS ABSOLUTE STARTING ADDRESS
                TAD LOCLAM      /LAM
                DAC 17
                LAC DISPFC      /REF AS DISPLA 5 IN DISPLB
                DAC DISP4       /SET UP MINUS COUNTER
                CMA!CLL         /FIND NUMBER OF DIVISIONS
                CLQ!NORM
                RCL
                DAC DISP9
                LACS
                TAD (777735
                SPA!CLL!CMA
                JMP . 3
                JMS CMATDD
                TAD (100
                TAD (LRS
                DAC DISP2       /LLS OR LRS, DEPENDING ON DISPFS
                XOR (10300      /CHANGE LRS TO CLQ!LLS;LLS TO CLQ!LRS
                DAC STRI25      /FOR STRIP MODE
                LAC (370000     /NORMALIZED 1740
                FRDIV
DISP9,          XX
                LACQ
                DAC DISP1       /FRACTION TO MULTIPLY BY
                DAC LIN1        /NEEDED TO DRAW LINES
                DAC STRI24      /FOR DIFF DISPLAY IN STRIP MODE
                DAC STRI26      /FOR STRIP MODE
                DAC STRI28      /FOR STRIP MODE
                LAS!CLL         /SET SCALE FOR Y COORDINATE
                LRS 4
                AND (1
                TAD (6505
                RTL
                LLS 4
DISP21,         DAC DISPYS      /REF IN DISDAC
                XOR (300        /CHANGE CLQ!LRS TO CLQ!LLS; VICE VERSA
                DAC STRI27      /FOR STRIP MODE
                DZM DISPLK
                LAC DISPLK
                MUL             /CLL NOT NEEDED FOR LIN,PRESENT LOG,SQT
DISP1,          XX              /FRACTION TO MULTIPLY BY
DISP2,          XX              /SHIFT INSTRUCTION
                DXL
                LAC I 17
DISPYS,         XX              /REF IN LOG,SQRTA
/DISPYS MAY BE A SHIFT (LINEAR DISPLAY), JMP SQRT (SQUARE ROOT), OR
/A JMP LOG2 (LOG DISPLAY).
                DYS             /REF AS DISPYS 1 IN WAN, STRI24
/PRECEDING INSTRUCTION IS CHANGED TO JMS TOPLOP FOR NO WRAP-AROUNDS.
                ISZ DISPLK
```

```
                ISZ DISP4
                JMP .-12
                LAC PENACT
                DAC DISP9       /TEMPORARY STORAGE
                LAW DISMIS
                DAC PENACT      /NO LIGHT PEN INTERRUPT ON LINES
                LAC LINELO
                JMS LINTST
                LAC LINEHI
                JMS LINTST
                LAC DISP9
/RESTORE LIGHT PEN MODE TO THAT WHICH IT WAS BEFORE LINE-DRAWING.
                DAC PENACT
                JMP I DISPLA
DISP4,          XX
DISPLK,         XX
/TEST AND DRAW LINE IF IN RANGE
LINTST,         XX              /CHANNEL OF LINE IN AC AT ENTRY
                DAC LIN#TT1
                SPA!CMA
                JMP I LINTST    /NO CHANNEL MARKER
                TAD DISPLN
                SMA
                JMP I LINTST    /LINE LOWEST CHANNEL DISPLAYED
                TAD DISPFS
                SPA!STL
                JMP I LINTST    /LINE HIGHEST CHANNEL DISPLAYED
                LAC DISPLN
                JMS CMATDD
                TAD LINTT1
                MUL
LIN1,           XX              /FRACTION PUT IN BY DISPLAY
                XCT DISP2       /SHIFT INSTRUCTION
                DXL
                LAM-37
                TAD (40
                DYS             /REF AS LIN1 5 IN WAY
                SAD (1740
                JMP I LINTST
                JMP .-4
/LOG FUNCTION FOR TOF--COURTESY OF JAB
LOG2,           CLL
                CLQ!LRS 2       /CLEAR MQ AND RS2
                SNA
                JMP DISPYS 1    /EXIT WITH 0 IF < 4
                NORM            /AC HAS F*2**17
                DAC LMUL1
                LACS            /23-M-44=-M-21
                TAD (23         /21+NUMBER OF BINADES TO SUPPRESS
                CMA             /M-1
                AND (37         /TIDY UP
                ALS 7           /6 FOR 16 BINADES FULL SCALE,7 FOR 8, ETC.
                DAC LTMP1
                LAC LMUL1
                MUL             /F**2*2**34
LMUL1,          XX
                ALS 1
                MUL             517023          /=1.30874*2**17
                CMA
                DAC LTMP2
```

```
                LAC LMUL1
                ALS 1
                MUL             764762          /=3.91300*2**16
                ADD LTMP2
                RCL
                ADD (145237     /INC. FUDGE FACTOR TO PREVENT OVRFLW
                CLL             /=(LOG+1)*2**17
                LRS 12          /13 FOR 16 BINADES, 12 FOR 8, ETC.
                ADD LTMP1
                JMP DISPYS 1
/SQUARE ROOT, ALSO COURTESY OF JAB
/RESULT IS SCALED BY SWITCHES
SQRT,           CLL
                CLQ!LRS 2
                NORM-21
                DAC SQT
                LACS
                SNA             /A 0 AC ENTRY GIVES A 0 STEP COUNTER
                JMP SQSC        /EXITS WITH 0 IN AC, MQ
                RCR
                TAD (LRS-26
                DAC SQSH
                LAC SQT
                SZL!RCR
                RCR
                DAC SQT
                TAD (LAC
                DAC SQDIV
                LAC SQT
                CLL
                FRDIV
SQDIV,          XX
                LACQ
                TAD SQDIV
                CLL
SQSH,           XX
SQSC,           OPR             /SWITCH SCALING INSTRUCTION
                JMP DISPYS 1
SQT=LMUL1
LTMP1=SQDIV
LTMP2=SQSH
/ROUTINE TO TAKE THE TWO'S COMPLEMENT OF A NUMBER
CMATDD,         XX              /FOR ROUTINES WITH PIC ENABLED
                CMA
                TAD (1
                JMP I CMATDD
/ROUTINE TO AVOID WRAP-AROUND OF Y-COORDINATE DISPLAY
TOPLOP,         XX
                LRS 12
                SNA!CLC
                LLS 12
                DYS
                JMP I TOPLOP
```

```
/KILLS LINES
LINKIL,         CLC
                DAC LINELO
                DAC LINEHI
                JMP DISMIS
/EXPANDS DISPLAY BETWEEN LINES
EXPAND,         LAC LINEHI
                SPA
                JMP DISMIS      /NO HIGH LINE
                LAC LINELO
                SPA!CMA
                JMP DISMIS      /NO LOW LINE
                TAD LINEHI
                SPA
                JMP DISMIS      /LOW LINE = OR > HIGH LINE
                TAD (2
                JMS DISPFR
                LAC LINELO
                DAC DISPLN
                JMP DISMIS
/ROUTINE TO SET UP DISPFS AND A COUNTER FOR ROTATION PURPOSES
DISPFR,         XX
                DAC DISPFS      /DISPFS CONTAINS NUMBER OF POINTS TO DISPLAY
                CMA
                TAD (1
                DAC DIS#PFC     /MINUS COUNTER
                CLL
LIN2,           LAM-1777        /TO ROTATE NICELY
                IDIVS
DISPFS,         TOFDIM
                LACQ
                DAC ROT8IC
                DAC ROT8CT
                JMP I DISPFR
LINWHR,         0               /=0 FOR ACTIVE LOW LINE; =1 FOR HIGH LINE
/ROUTINE TO HANDLE INTERRUPTS OF NON-STRIP MODE SIDE DOTS
PENDO,          LAW PEN1-1
                TAD WCHPNT
                DAC PEN#2
                LAW DISMIS      /ONLY ONE INTERRUPT SERVICED PER PASS
                DAC PENACT
                JMP I PEN2
PEN1,           JMP ROT8L       /ROTATE LEFT
                JMP ROT8R       /ROTATE RIGHT
                JMP EXPAND      /EXPAND DISPLAY BETWEEN LINES
                JMP ACSRES      /DISPLAY FULL SPECTRUM
                JMP PNHI        /SET HIGH LINE INDICATOR
                JMP PNLO        /SET LOW LINE INDICATOR
                JMP DISMIS      /IGNORE INTERRUPT FROM LINE INDICATOR.
/POSITION LOW OR HIGH LINE AT CHANNEL OF DATA POINT CAUSING INTERRUPT.
                LAC LINWHR
                RAR
                LAC DISPLM
                JMS CMATD1
                TAD DISPLN
                TAD 17
                SNL
                DAC LINEHI
                SZL
                DAC LINELO
```

```
                JMP DISMIS
PNLO,           DZM LINWHR      /SET LOW LINE INDICATOR
                JMP DISMIS
PNHI,           LAC (1          /SET HIGH LINE INDICATOR
                DAC LINWHR
                JMP DISMIS
ROT8L,          ISZ ROT8CT      /ROUTINE TO ROTATE LEFT
                JMP DISMIS
                LAM             /SUBTRACT ONE FROM DISPLN,LINELO,LINEHI
                TAD DISPLN
                SPA             /DO NOT ROTATE IF OUT OF RANGE
                JMP ROT8IN      /INITIALIZE COUNTER
                DAC DISPLN
LOCLAM,         LAM             /REF IN DISPLA,LACOM1
                TAD LINELO
                DAC LINELO
                LAM
                TAD LINEHI
                DAC LINEHI
ROT8IN,         LAC ROT8IC      /LAC NUMBER OF SKIPS
                DAC ROT8CT      /BEFORE NEXT ROTATION
                JMP DISMIS
ROT8R,          LAM-TOFDIM 1    /ROTATE RIGHT
                TAD DISPFS
                TAD DISPLN
                SPA
                ISZ ROT8CT
                JMP DISMIS
                ISZ DISPLN      /ADD ONE TO DISPLN,LINELO,LINEHI
                NOP             /IN CASE OF LAM IN DISPLN
                ISZ LINELO
LOCNOP,         NOP             /REF IN STRI17,LOG,BT2,VAR3A
                ISZ LINEHI
                JMP ROT8IN
                JMP ROT8IN      /IN CASE OF LAM IN LINEHI
ROT8CT,         0
ROT8IC,         0
ALO,            LAC LOLINE      /SETS LOW LINE FROM TELETYPE
                JMS STRTYP      /JMS STRASC,JMS TYPEIN,JMS CRLF
                DAC LINELO
                JMP DISMIS
AHI,            LAC HILINE      /SETS HIGH LINE FROM TELETYPE
                JMS STRTYP
                DAC LINEHI
                JMP DISMIS
LINELO,         LAM             /LOCATION OF CHANNEL OF LOW LINE
LINEHI,         LAM             /LOCATION OF CHANNEL OF HIGH LINE
/ROUTINE TO TAKE THE TWO'S COMPLEMENT OF A NUMBER
CMATD1,         XX              /FOR ROUTINES WITH PIC DISABLED
                CMA
                TAD (1
                JMP I CMATD1
/TYPE OUT CONTENTS OF CHANNEL IN AC.
VAL,            XX
                DAC VAL#1
                LAC DISPLN
                JMS CMATD1
                TAD DISPLM
                TAD VAL1
                DAC VAL1
```

```
                LAC I VAL1
                JMS PRNTIN
                JMP I VAL
LOEQ,           XX              /TYPE OUT CHANNEL AND CONTENTS OF LOW LINE
                LAC LINELO
                JMS PRNTIN
                LAC LINELO
                JMS VAL
                JMP I LOEQ
HIEQ,           XX              /TYPE OUT CHANNEL AND CONTENTS OF HIGH LINE
                LAC LINEHI
                JMS PRNTIN
                LAC LINEHI
                JMS VAL
                JMP I HIEQ
LEQ,            LAC LOLINE
                JMS STRASC      /TYPE OUT LOW AND HIGH LINES
                JMS LOEQ
                JMS CRLF
                LAC HILINE
                JMS STRASC
HIEQA,          JMS HIEQ        /HIGH LINE ONLY
                JMP CRLFDM
LOEQA,          JMS LOEQ        /LOW LINE ONLY
                JMP CRLFDM
SQRTA,          LAC (JMP SQRT   /DISPLAY SQUARE ROOT
                DAC DISPYS      /REF IN NDSPLY
                LAC (DAC SQSC
                JMP DISDAC
LOG,            LAC (JMP LOG2   /DISPLAY LOG
                DAC DISPYS
                LAC LOCNOP      /NOP
DISDAC,         DAC DISP21
                LAM-1777
                JMP NDSPLY 3
NDSPLY,         LAC SQRTA 1     /DISPLAY LINEAR SCALE
                DAC DISP21
                LAM-3777
                DAC LIN2        /REF IN DISDAC
                JMP DISMIS
/NO Y-COORDINATE WRAP-AROUND
WAN,            LAC (JMS TOPLOP
                SKP
/RESTORE Y-COORDINATE WRAP-AROUND
WAY,            LAC LIN1 5      /DYS
                DAC DISPYS 1
                JMP DISMIS
PAUSE BEGIN
```

```
/STRIPPING ROUTINE
/REARRANGE FLOW OF PROGRAM SO THAT BACKGROUND DISPLAY IS CONTROLLED BY
/ACO AND B SWITCH, ENABLE STRAWBERRY MARK, ENABLE DIFFERENCE DISPLAY,
/ENABLE STRIPPING CONTROL DOTS, AND DISABLE LOG AND LINEAR DISPLAYS,
/BCD DUMP, AND CALCOMP PLOTTER.
STRIP,          LAC ACSRES      /LAS CLL  INITIALIZE STRIPPING
                DAC ACSDIS 2
                DAC STRIP3 7
                LAC (JMP STRIP2
                DAC STRIP1
                DAC MONDIS 7
                LAC (17         /LOCK OUT BCD,PLT,LOG,SQT
                JMS LOCK
                JMP NDSPLY      /ONLY LINEAR DISPLAY PERMITTED
/FOR STRIP MODE, FLOW OF PROGRAM IS ALTERED SO THAT CONTROL JUMPS HERE
/AFTER LAST FOREGROUND IS DISPLAYED.
STRIP2,         LAW STRIP3      /STRIP3 INTERPRETS DOTS DOWN SIDE
                DAC PENACT
                LAS!CLL
                RTL             /PUT AC1 INTO LINK
                LAM-500 1
                SNL
                JMS MAP         /PUT DOTS DOWN SIDE
STRI21,         JMP DIFNOT      /LAW DISMIS FOR DIFFERENCE DISPLAY
                DAC PENACT
/ROUTINE TO DISPLAY DIFFERENCE OF FOREGROUND AND BACKGROUND
/LIGHT PEN INTERRUPTS ARE IGNORED.
                CLC!STL
                TAD DISPLM
                DAC 17
                LAW RECOLO-1
                TAD DISPLN
                DAC 14
                DZM DISPLK
                LAC DISPFC      /MINUS COUNTER
                DAC DISP4
                LAC DISPLK
                MUL
STRI24,         XX              /FRACTION PUT IN BY DISPLA
                XCT DISP2       /SHIFT INSTRUCTION
                DXL
                LAC I 14
                JMS CMATDD
                TAD I 17
                SNL!CLL
                CLA             /MINUS DIFFERENCES ARE ZEROED.
                XCT DISPYS      /SHIFT INSTRUCTION
                XCT DISPYS 1
/DISPYS 1 IS DYS FOR WRAP-AROUND; JMS TOPLOP FOR NO WRAP-AROUND.
                ISZ DISPLK
                ISZ DISP4
                JMP .-16
/ROUTINE TO DISPLAY STRAWBERRY MARK
DIFNOT,         LAS
                SPA
                JMP STRI23      /SKIP STRAWBERRY MARK IF AC0 UP
                LAW STRIP5      /STRIP5 SERVICES INTERRUPTS FROM STRAWBERRY
                DAC PENACT
                LAW SMK-1       /SMK=STRAWBERRY MARK
                DAC 17
```

```
                LAM-4
                DAC STRIP7
                LAC I 17
                DXL
                LAC I 17
                DYS
                LAC I 100       /STALL
                LAC I 100       /STALL
                ISZ STRIP7
                JMP .-7
                LAW STRIP6      /SERVICES INTERRUPTS FROM ACTIVE BACKGROUND
                DAC PENACT      /REF AS DISPLB-1 BY STRI23
DISPLB,         LAW RECOLO-1
                TAD DISPLN
                DAC 17          /SET UP 17 WITHOUT CHANGING DISPLM
/DISPLM CONTAINS ABSOLUTE STARTING ADDRESS OF REGION DISPLAYED AND IS USED
/TO CALCULATE ORIGIN OF FOREGROUNDS.
                LAW . 3
                DAC DISPLA
                JMP DISPLA 5
                LAC STRI15      /COME HERE AFTER DISPLAY OF BACKGROUND
                SNA!CLL
                JMP ACSDIS      /TRACK FOLLOWER ROUTINES NOT USED; JUMP TO GO
/STRI15 IS THE NUMBER OF DOTS IN STRAWBERRY OR BACKGROUND SEEN.
/ROUTINE TO CALCULATE NEW STRAWBERRY AND TO ALTER BACKGROUND
                DAC STRI16
                LAC SUMX        /SET NEW STRAWBERRY X-COORDINATE
                IDIV
STRI15,         000000
                LACQ
                AND (1777
                DAC SMK 4       /SMK 4 = CENTER DOT X-COORDINATE
                DAC SMK
                DAC SMK 10
                TAD SUMY 1      /SPACING OF DOTS IN STRAWBERRY
                DAC SMK 6
                TAD SUMY 2      /MINUS TWO TIMES SPACING
                SPA!CLL
                CLA             /PUT ZERO IF MINUS FOR LEFTMOST DOT
                DAC SMK 2
                LAC SUMY        /SET NEW STRAWBERRY Y-COORDINATE
                IDIV
STRI16,         XX
                LACQ
                AND (1777
                DAC SMK 5       /SMK 5 = CENTER DOT Y-COORDINATE
                DAC SMK 3
                DAC SMK 7
                TAD SUMY 1      /SPACING OF DOTS
                DAC SMK 1
                TAD SUMY 2      /MINUS TWO TIMES SPACING
                SPA
                CLA             /PUT ZERO IF MINUS FOR LOWEST DOT
                DAC SMK 11
/THE FOLLOWING INSTRUCTIONS TAKE A POINT IN SCOPE COORDINATES AND CONVERT
/ITS X-COORDINATE INTO A CHANNEL AND ITS Y-COORDINATE INTO A CHANNEL CONTENTS
/THE VALUE OF THE BACKGROUND CHANNEL IS THEN REPLACED BY THAT NUMBER.
/THIS PROCEDURE IS THE INVERSE OF THE SCALING PROCESS IN DISPLA.
                LAC SMK 4
                RCL
```

```
STRI25,         XX              /PUT IN BY DISPLA
                DIV
STRI26,         XX              /PUT IN BY DISPLA
                LACQ
                TAD (1
                RCR
                TAD (RECOLO
                TAD DISPLN
                DAC SUMX
                CLL
                LAC SMK 5
STRI27,         XX              /PUT IN BY DISPLA
                DAC I SUMX
                DZM STRI15
                DZM SUMX
                DZM SUMY
                DZM STRI22
                JMP ACSDIS      /START THE LOOP OVER
STRI23,         LAS!CLL         /STRAWBERRY INACTIVE  CHECK B SWITCH
                ALS 13
                SMA
                JMP ACSDIS      /B SWITCH DOWN; START THE LOOP OVER
                LAW DISMIS
/B SWITCH UP AND ACQ UP  DISPLAY BACKGROUND, BUT IGNORE INTERRUPTS.
                JMP DISPLB-1
```

```
SMK,            1000            /TABLE OF COORDINATES FOR STRAWBERRY
                1040            740             1000
                1000            1000            1040
                1000            1000            740
SUMX,           0
SUMY,           0
                6               /SPACING FOR SMK
                777764          /MINUS TWO TIMES SPACING FOR SMK
CENTRD,         XX              /TWO WORDS USED IN CALCULATION OF CENTROID
                XX
VARANC,         XX              /TWO WORDS USED IN CALCULATION OF VARIANCE
                XX
SETAUT,         XX              /SET UP 15,16 FOR FOREGROUND, BACKGROUND
                LAW RECOLO-1
                TAD DISPLN
                DAC 15
                CLC!STL
                TAD DISPLM
                DAC 16          /LINK IS CLEARED
                JMP I SETAUT
STRIP3,         LAW . 3         /INTERPRET SIDE DOTS
                TAD WCHPNT
                DAC PEN2
                JMP I PEN2
                JMP STRI20      /DO NOT DISPLAY DIFFERENCE
                JMP STRI19      /DISPLAY DIFFERENCE
                JMP STRIP4      /SET BACKGROUND=FOREGROUND
                LAS!CLL         /REF AS STRIP3 7 IN VAR3,STRIP,INT
                AND (KEYMSK
                SNA
                JMP DISMIS      /NO FOREGROUND DISPLAYED; LEAVE ROUTINE
                LAC .-1
                DAC .-5
/PRECEDING INSTRUCTIONS PREVENT A RE-ENTRY OF THIS ROUTINE FROM A LIGHT PEN
/INTERRUPT UNTIL THE END OF THIS PASS IS REACHED.
                LAC DISPFC      /MINUS COUNTER OF NO. OF POINTS ON SCOPE
                DAC AREA1
/THE FOLLOWING PART OF THIS ROUTINE FINDS THE CHANNEL OF THE FIRST NON-ZERO
/DIFFERENCE AND THE VALUE OF THE MAXIMUM DIFFERENCE.
/AREA CONTAINS THE MAXIMUM DIFFERENCE (TEMPORARILY).
/VARSTR CONTAINS THE FIRST NON-ZERO CHANNEL.
                LAC (JMP VARBEG
                DAC VARLOC
                JMS SETAUT      /SET UP 15,16
                DZM AREA
                LAC I 16
                JMS CMATD1
                TAD I 15
                SZL!CLL
                JMP . 15
                LMQ
                LAC AREA1
                TAD DISPFS
                DAC VAREND
VARLOC,         JMP VARBEG
                CLL
                TAD AREA
                SZL!CLL
                JMP . 4
                JMS CMATD1
```

```
                TAD AREA
                DAC AREA
                ISZ AREA1
                JMP .-22
                LAC VARLOC
                SAD (JMP VARBEG
                JMP VAR4-4      /TYPES ONLY A CRLF IF AREA=0
                LAC (JMP . 2
                JMP RSLINE 1    /TYPE WHICH FOREGROUND IS USED
/THE TYPING OF THE FOREGROUND USED IS INSERTED HERE IN ORDER TO ENABLE THE
/PIC AND PROCESS DATA IF NECESSARY.
/THE FOLLOWING SETS UP VARMIN, THE MINIMUM VALUE A CHANNEL MUST HAVE TO BE
/USED IN CALCULATING THE VARIANCE.
/IN THIS CASE MINIMUM = 1/100 (OCTAL) TIMES THE MAXIMUM DIFFERENCE.
                LAC AREA        /AREA1 IS NOW ZERO
                SMA!CLL
                TAD (100
                LRS 7
                DAC VAR#MIN
                DZM AREA
                DZM STRI10
                DZM VAR1
                DZM VARANC
                DZM VARANC 1
                DZM CENTRD
                DZM CENTRD 1
                JMS SETAUT      /SET UP 15,16 AGAIN
                LAC 15
                TAD VARSTR      /ADD ADDRESS OF FIRST NON-ZERO DIFFERENCE
                DAC 15
                LAC 16
                TAD VAR#STR
                DAC 16
                LAC VAR#END
                CMA!CLL
                TAD VARSTR
                DAC STRI18      /DIFFLO(NOT ZERO)-DIFFHI(NOT ZERO)-1 AS COUNTER
/THE PURPOSE OF THE ABOVE INSTRUCTIONS IS TO MAKE SURE ONLY THE NON-ZERO
/DIFFERENCE REGION IS USED IN THE LENGTHY CALCULATION OF CENTROID AND VARIANCE.
STRI12,         LAC I 15
                JMS CMATD1
                TAD I 16
                SNL!CLL
                JMP STRIP9 11
                DAC STRIP9
                MUL
STRI10,         XX
                DAC STRI11
                LACQ
                TAD CENTRD 1
                DAC CENTRD 1
                GLK
                TAD CENTRD
                TAD STRI11
                DAC CENTRD
                LAC STRIP9
                TAD AREA
                DAC AREA
                SPL!CLL
                ISZ AREA1
```

```
                LAC STRIP9      /AREA
                JMS CMATD1
                TAD VARMIN
                SZL!CLL
                JMP STRIP9 11
                LAC STRI10
                MUL
VAR1,           XX              /=STRI10=CHANNEL
                LACQ
                MUL
STRIP9,         XX              /FOREGROUND-BACKGROUND
                DAC STRI11
                LACQ
                TAD VARANC 1
                DAC VARANC 1
                GLK
                TAD VARANC
                TAD STRI11
                DAC VARANC
                ISZ STRI10      /REF AS STRIP9 11 IN STRI12,STRI10
                ISZ VAR1
                ISZ STRI18
                JMP STRI12
                LAC (400175     / A=
                JMS STRASC
                LAC AREA
                DAC DOUB1 1     /LOW ORDER BITS
                LAC AREA1
                JMS PRNTDB      /DOUBLE PRECISION PRINT OUT
                LAC (400375     / C=
                JMS STRASC
                LAC AREA1
                SNA!CLL
                JMP STRI17
/IF THE AREA 262143, THE DOUBLE PRECISION WORD CONTAINING AREA IS NORMALIZED,
/THE 17 MOST SIGNIFICANT BITS ARE KEPT, AND CENTRD AND VARANCE ARE
/SHIFTED ACCORDINGLY.
                LAC AREA
                LMQ
                LAC AREA1
                NORM
                DAC AREA
                LACS
                TAD (44
                AND (77
                CMA!STL
                TAD (LRS 23
                SKP
STRI17,         LAC LOONOP      /NOP
                DAC VAR3
                LAC CENTRD 1
                LMQ
                LAC CENTRD
                XCT VAR3        /NOP OR LRS XX
                DIV
AREA,           XX
                DAC STRIP9      /REMAINDER PART OF CHANNEL
                LACQ
                DAC VAR#2
                TAD VARSTR
```

```
                TAD DISPLN      /PUT CENTROID AS ACTUAL CHANNEL FOR TYPOUT
                JMS PRNTIN      /TYPE OUT INTEGER PART OF CHANNEL
                LAC AREA
                DAC . 6
                DAC VAR3 2
                DAC VAR3 10
                LAC STRIP9
                CLL
                FRDIV
                XX
                LACQ
                DAC STRIP9      /FRACTION OF CHANNEL
                JMS VAR4        /TYPE OUT FRACTION
                LAC (402775     / W=
                JMS STRASC      /CLEARS LINK ALSO
                LAC (JMP VAR3A
/THE PRECEDING SETS UP A RETURN TO THIS ROUTINE FROM SQRT,
/WHICH NORMALLY IS USED ONLY IN DISPLA, TO WHICH IT IS SET TO RETURN.
                DAC SQSC        /CLOBBERS SQRT EXIT
                LAC STRIP9      /CHANNEL, FRACTION OF
                LMQ
                LAC VAR2
                LLS 9           /AC=CHAN*2**9
                DAC . 2
                MUL
                XX              /AC AND MQ=CHAN**2*2**18
                DAC AREA1
                LACQ
                DAC STRIP9      /FRACTIONAL PART OF CHANNEL**2
                LAC VARANC 1
                LMQ
                LAC VARANC      /36-BIT WORD TO BE NORMALIZED IF AREA WAS
VAR3,           XX              /NOP OR LRS XX
                DIV
                XX              /AREA
                DAC VARANC 1
                LACQ            /REF AS VAR3 4 IN VARBEG
                DAC VARANC      /WHOLE PART OF CHAN**2
                LAC VARANC 1
                FRDIV
                XX              /AREA
                LACQ
                DAC VARANC 1    /FRACTIONAL PART OF CHAN**2
                LAC STRIP9      /START TO DO X(I)**2-X(AV)**2
                JMS CMATD1
                TAD VARANC 1
                LMQ
                CML
                GLK
                TAD AREA1
                JMS CMATD1
                TAD VARANC
                CLL
                LLS 11          /ASSUMES VARIANCE 512
                MUL
                542710          /8*LN(2)*2**15
                JMP SQRT        /RETURN SET UP IN AREA
VAR3A,          LRS 13
                DAC VARANC
                LACQ
```

```
                DAC VARANC 1
                LAC LOCNOP      /NOP
                DAC SQSC
                LAC VARANC
                JMS PRNTIN
                LAC VARANC 1
                JMS VAR4        /TYPE OUT FRACTIONAL PART
                JMS CRLF        /REF AS VAR4-4 IN VARLOC
                LAC ACSRES      /LAS!CLL;ENABLE AN ENTRY BACK TO THIS ROUTINE
                DAC STRIP3 7
/SINCE THIS ROUTINE CAN BE CALLED BY INT WHILE DISPLAY IS IN SQUARE ROOT MODE,
/IT IS BEST TO START DISPLAY FROM GO.
                JMP ACSDIS-1    /ION
/TYPE OUT FRACTION STORED IN AC TO TWO DECIMAL PLACES.
VAR4,           XX
                CLL
                MUL
                1750
                TAD (5
                JMS BTDEC       /CONVERT TO DECIMAL STRING OF DIGITS
                LAW 256         /DECIMAL POINT
                JMS TYPOUT
                LAW 260
                TAD INOTTO 4
                JMS TYPOUT
                LAW 260
                TAD INOTTO 5
                JMS TYPOUT
                JMP I VAR4
VARBEG,         DAC VARSTR      /FIRST NON-ZERO DIFFERENCE
                LAC VAR3 4      /LACQ
                DAC VARLOC
                JMP VARLOC
```

```
STRIP4,         JMS STRIP8      /SET BACKGROUND=FOREGROUND
                JMP DISMIS
STRIP8,         XX              /ROUTINE TO SET BACKGROUND=FOREGROUND
                LAC DISPFC      /MINUS COUNTER
                DAC STRI18
                JMS SETAUT      /SET UP 15,16
                LAC I 16
                DAC I 15
                ISZ STRI18
                JMP .-3
                JMP I STRIP8
/ROUTINE TO PROCESS STRAWBERRY MARK INTERRUPT
STRIP5,         LAM             /GET COORDS OF STRAWBERRY MARK
                TAD 17          /ADD TO SUM OF COORDINATES
                DAC STRI14
                LAC I STRI14
                TAD (-1740 1
                SMA
/IGNORE DOT WHOSE SCOPE X-COORDINATE > 1740 (OCTAL).
                JMP DISMIS
                LAC I STRI14
                TAD SUMX
                DAC SUMX
                ISZ STRI14
                LAC I STRI14
                TAD SUMY
                DAC SUMY
                ISZ STRI15      /COUNTER FOR NUMBER OF DOTS SEEN
                ISZ STRI22
                JMP DISMIS
STRI22,         0               / 0 IF ANY DOTS IN STRAWBERRY SEEN
/ROUTINE TO PROCESS ACTIVE BACKGROUND INTERRUPT
STRIP6,         LAC STRI22
                SZA!CLL
                JMP DISMIS      /IGNORE IF ANY DOTS IN STRAWBERRY SEEN FIRST
                ISZ STRI15
                LAC 17
                DAC STRI14
                LAC I STRI14
                XCT DISPYS      /TRANSLATE VALUE OF BACKGROUND POINT
                TAD SUMY        /TO STRAWBERRY COORDINATES
                DAC SUMY
                CLC!STL         /SAME AS LAM,STL;DISPLK>0
                TAD DISPLK
                MUL
STRI28,         XX              /FRACTION PUT IN BY DISPLA
                XCT DISP2       /SHIFT INSTRUCTION
                TAD SUMX
                DAC SUMX
                JMP DISMIS
/SET UP FLOW OF PROGRAM TO DISPLAY DIFFERENCE.
STRI19,         LAW DISMIS
                SKP
/SET UP PROGRAM NOT TO DISPLAY DIFFERENCE.
STRI20,         LAC (JMP DIFNOT
                DAC STRI21
                JMP DISMIS
/LEAVE STRIP MODE.  ENABLE SQUARE ROOT AND LOG DISPLAYS.
NOSTRP,         LAM
                DAC ACSDIS 2
```

```
                LAC ACSRES      /LAS!CLL
                DAC STRIP1
                LAC STRIP1 3    /JMP ACSDIS
                DAC MONDIS 7
                LAC (14
                JMS UNLOCK      /ENABLE LOG,SQUARE ROOT DISPLAY MODES
                JMP ACSRES      /REINITIALIZE DISPLAY.
/INTEGRATE FOREGROUND DISPLAYED BETWEEN THE LINES.
INT,            JMS STRIP8      /SET BACKGROUND=FOREGROUND
                LAC (3          /LOCK OUT PLT,BCD
                JMS LOCK
                LAC LINELO
                SPA
                JMP ERRT        /NO LOW LINE
                LAC LINEHI
                SPA!CMA
                JMP ERRT        /NO HIGH LINE
                TAD LINELO
                SMA
                JMP ERRT        /LOW LINE = OR > HIGH LINE
                DAC ZEROT1
                LAW RECOLO-1
                TAD LINELO
                JMS ZERO        /ZERO BACKGROUND BETWEEN LINES
                JMP STRIP3 7    /TYPE OUT WHICH FOREGROUND, AREA, ETC.
PAUSE BEGIN
```

```
/EXECUTIVE ROUTINE TO INTERPRET TYPEWRITER MESSAGES
/THE BOOTSTRAP PART OF THE PROGRAM SETS UP A JUMP TO EXECIN FOR THE FIRST
/KEYBOARD INTERRUPT; ANY SUBROUTINE NEEDING KEYBOARD INPUT RESETS THIS JUMP
/UPON COMPLETION OF ITS INPUT.
/EXECIN PACKS INPUT CHARACTERS IN STRIPPED ASCII, THREE TO A WORD.
/UPON COMPLETION OF THE INSTRUCTION INPUT BY A CARRIAGE RETURN OR A SPACE,
/EXECIN STORES THE INPUTTED INSTRUCTION AT THE END OF A TABLE OF LEGITIMATE
/INSTRUCTIONS, ALSO STORED IN STRIPPED ASCII.  THEN IT COMPARES THE INPUT
/INSTRUCTION TO THE LIST, JUMPING TO AN ADDRESS IN A CORRESPONDING TABLE,
/IF THAT JUMP HAS NOT BEEN DISABLED.
/FOR JUMPS WHICH CAN BE DISABLED, THE WORD STORING THE JUMP ADDRESS HAS
/AC0=0(NOT JMP).   THOSE WHICH CANNOT BE DISABLED HAVE AC0=1(JMP).
/THOSE WHICH CAN BE DISABLED HAVE IN AC1-4 THE NUMBER OF THE BIT IN CODEWD
/WHICH CORRESPONDS TO THE ENABLE-DISABLE KEY.  IN CODEWD,
/A 1-BIT ENABLES, A 0-BIT DISABLES.
/IF THE INPUT INSTRUCTION AGREES ONLY WITH ITSELF (IT IS STORED AT THE END
/ OF THE TABLE), IT EXECUTES AN ERROR JUMP.
EXECIN,         DAC EXE#CT1
                JMS INTYPE
                SAD (215        /QUIT ON CARRIAGE RETURN
                JMP EXECI2
                SAD (240        /ALSO ON SPACE
                JMP EXECI2 1
                LRS 6
                LAC EXECT1
                LLS 6
                JMP EXECIN
EXECI2,         JMS CRLF
                LAW EXECIN
                DAC INTYPE
                LAW EXECI3
                DAC EXECT2
                LAW EXECI6
                DAC EXE#CT4
                LAC EXECT1
                SAD I EXECT2
                JMP . 4
                ISZ EXECT2
                ISZ EXECT4
                JMP .-4
                LAC I EXECT4
                DAC EXECT4
                SPA!CLL
                JMP I EXECT4    /JUMP WITHOUT CHECKING CODEWD
                LRS 15
                TAD (ALS 2      /FIND BIT IN CODE WORD
                DAC . 2
                LAC CODEWD
                XX
                SPA
                JMP I EXECT4    /JUMP NOT DISABLED
ERRT,           LAM-36          /??!
                JMS STRASC
CRLFDM,         JMS CRLF
                JMP DISMIS      /REF AS CRLFDM 1 IN STOP5,JFND, OTHERS
/STORAGE OF STRIPPED ASCII TYPE-IN INSTRUCTIONS
EXECI3,         243117          /TYO
                243111          /TYI
                232422          /STR
                131416          /KLN
```

```
                110223          /IBS
                230524          /SET
                222314          /RSL
                241714          /TOL
                220503          /REC
                032203          /CRC
                232423          /STS
                141714          /LOL
                101114          /HIL
                021116          /BIN
                020304          /BCD
                051724          /EOT
                021624          /BNT
                041520          /DMP
                072467          /GT7
                072462          /GT2
                072403          /GTC
                072415          /GTM
                022320          /BSP
                040503          /DEC
                170324          /OCT
LOLINE,         141775          /LO=  REF AS LOLINE IN ALO,LOEQ
HILINE,         101175          /HI=  REF AS HILINE IN AHI,HIEQ
                141623          /LNS
                141707          /LOG
                141116          /LIN
                222516          /RUN
                242375          /TS=
                230301          /SCA
                232124          /SQT
                270116          /WAN
                270131          /WAY
BSL,            022314          /BSL  REF AS BSL IN SHFTBN
                172622          /OVR
                052315          /ESM
                142315          /LSM
                111624          /INT
                230316          /SCN
                230331          /SCY
                010316          /ACN
                010331          /ACY
                201424          /PLT
                142024          /LPT
                322307          /ZSG
EXECT1,         HLT             /IF IT GETS HERE, IT'S AN ERROR
/STORAGE OF JUMPS CORRESPONDING TO TYPE-IN INSTRUCTIONS
EXEC16,         DZM I TYPONM    /TYO
                JMP TYPINM      /TYI
                XOR I CLOCKS    /STR
                JMP LINKIL      /KLN
                JMP INISH       /IBS
                JMP SETTIM      /SET
                DZM I RSLINE    /RSL
                DZM I TOLINE    /TOL
                JMP REC         /REC
                JMP CRC         /CRC
                JMP INISH9      /STS
                JMP ALO         /LOL
                JMP AHI         /HIL
                JMP BINSET      /BIN
```

```
TAD I TP1OUT      /BCD
JMP BCDEOT        /EOT
JMP BT2           /BNT
JMP WDUMP         /DMP
JMP GT7           /GT7
JMP GT2           /GT2
JMP GTC           /GTC
JMP GTM           /GTM
JMP BACKSP        /BSP
JMP DEC           /DEC
JMP OCT           /OCT
DZM I LOEQA       /LO=
DZM I HIEQA       /HI=
DZM I LEQ         /LNS
ADD I LOG         /LOG
JMP NDSPLY        /LIN
DZM I NEWRUN      /RUN
DZM I TSEQ        /TS=
JMP SCLRTP        /SCA
ADD SQRTA         /SQT
JMP WAN           /WAN
JMP WAY           /WAY
DZM I SHFTBN 2    /BSL
DZM I STP1        /OVR
LAC I STRIP       /ESM
JMP NOSTRP        /LSM
LAC INT           /INT
JMP SCN           /SCN
JMP SCY           /SCY
JMP ACN           /ACN
JMP ACY           /ACY
TAD PLOTER        /PLT
XOR LPT           /LPT
JMP ZOTGRP        /ZSG
JMP ERRT
```

```
/INTERRUPT PROGRAM
SCON=703701       /SKIP IF CONSOLE INTERRUPT
CLCON=704101      /CLEAR CONSOLE INTERRUPT
SWC=703601        /SKIP ON WORD COUNTER (ADC) OVERFLOW
CLWC=703602       /CLEAR WORD COUNTER FLAG
SKCF=702501       /SKIP IF CALCOMP INTERRUPT
CLCF=702502       /CLEAR CALCOMP FLAG
PLT=702504        /EXECUTE CALCOMP INSTRUCTION
/WHEN PIC IS ENABLED, AN INTERRUPT CAUSES AN EFFECTIVE JMS 0.
/LOCATION 1 CONTAINS JMP INTRUP.
/THE SKIP INSTRUCTIONS HAVE BEEN MODIFIED SO THAT A SKIP INDIRECT
/(KSF I, FOR INSTANCE) SKIPS IF THE FLAG IS NOT SET.
INTRUP,           DAC IN#TRAC
                  LACQ
                  DAC IN#TRMQ
                  LACS
                  DAC IN#TRSC
                  SCON I          /SCON NOT
                  JMP CONSOL 1
                  CLSF I          /SKIP IF CLOCK FLAG=0
                  JMP CLOCK
                  SWC
                  JMP . 6
STORGE,           XX              /SET UP BY STRT AND PROCES
                  LDMAC
                  LAW BUFSZE
                  LDWC            /ADC INTERFACE IS NOW RE-ENABLED.
                  JMP PROCES
                  SKCF I          /SKIP IF CALCOMP FLAG=0
                  JMP CALCMP 3
                  KSF I           /SKIP IF KEYBOARD FLAG=0
                  JMP INTYPE 2
                  TSF I           /SKIP IF TELEPRINTER FLAG=0
                  JMP TYPOUT 3
                  IDVE I          /SKIP IF LIGHT PEN FLAG=0
                  JMP PENACT 1
                  IOT I 601       /SKIP IF MAG. TAPE JOB NOT DONE
                  JMP I WFUN
                  RSF I           /SKIP IF READER FLAG=0
                  JMP READR 3
                  PCF             /PUNCH IS ONLY FLAG LEFT; CLEAR IT
DISMIS,           IORS
                  AND (375700     /CHECK IF ANY FLAGS UP
                  SZA
                  JMP INTRUP 5    /YES, SERVICE ANOTHER INTERRUPT
                  LAC INTRSC      /NO, RETURN TO MAIN PROGRAM
                  CMA!CLL
                  ADD (640300
                  DAC . 1
                  HLT
                  0
                  LAC 0
                  RAL
                  LAC INTRMQ
                  LMQ
                  LAC INTRAC
                  ION
                  JMP I 0
CONSOL,           JMP STRT        /ROUTINE TO HANDLE CONSOLE INTERRUPTS
                  CLCON
```

```
                JMP I CONSOL
INTYPE,         JMP DISMIS      /ROUTINE TO HANDLE KEYBOARD INTERRUPTS
                JMP DISMIS
                KRB
                JMP I INTYPE
TYPOUT,         XX              /ROUTINE TO HANDLE TELETYPE INTERRUPTS
                TLS
                JMP DISMIS
                TCF
                JMP I TYPOUT
PENACT,         PENDO           /ROUTINE TO HANDLE LIGHT PEN INTERRUPTS
                DCF             /CLEAR LIGHT PEN FLAG
                JMP I PENACT
CALCMP,         JMP DISMIS      /ROUTINE TO HANDLE CALCOMP INTERRUPTS
                PLT
                JMP DISMIS
                CLCF
                JMP I CALCMP
READR,          JMP DISMIS      /ROUTINE TO HANDLE PAPER TAPE READER INTERRUPTS
                RSB
                JMP DISMIS
                RRB
                JMP I READR
/READ BINARY TAPE INTO BACKGROUND SPECTRUM
LPT,            LAC (43         /LOCK OUT LPT,PLT,BCD
                JMS LOCK
                LAM-1000 1
                DAC INOTTO 1
                LAW RECOLO-1
                DAC 10
                JMS READR
                DAC I 10
                ISZ INOTTO 1
                JMP .-3
                LAC (43         /UNLOCK LPT,PLT,BCD
                JMS UNLOCK
                JMP NOSTRP      /LEAVE STRIP MODE
/A 1-BIT MEANS UNLOCK, A 0-BIT MEANS LOCK.
LOCK,           XX
                CMA
                AND CODEWD
                DAC CODEWD
                JMP I LOCK
UNLOCK,         XX
                LMQ
                LAC CODEWD
                OMQ
                DAC CODEWD
                JMP I UNLOCK
CODEWD,         LAM             /START OFF WITH EVERYTHING UNLOCKED
```

```
/THESE ARE VARIABLES USED BY ROUTINES WHICH INTERRUPT THE MAIN ROUTINE.
/THEY ARE STORED HERE FOR NO PARTICULAR REASON.
EXECT2,         XX
EXECT4,         XX
TOFT4,          XX
TOFT5,          XX
BUFTAG,         XX
BLAST,          XX
ZEROT1,         XX
CLCK2,          XX
BELLT,          XX
RSUMB,          XX
RSUMN,          XX
RSUMT3,         XX
RSUMT2,         XX
                XX              /FOR RSUMT2 1
RSUMT4,         XX
RSUMT1,         XX
RSUM6,          XX
GLO,            XX
GHI,            XX
INOUT2,         XX
INOUT4,         XX
PRNTT2,         XX
NMOUTN,         XX
NMINDN,         XX
NMSTT1,         XX
STRAT1,         XX
STRAT2,         XX
VAL1,           XX
SCL1,           XX
SCL3,           XX
SCL4,           XX
SCL2,           XX
STPVR2,         XX
STPVR1,         XX
STRI14,         XX
STRI11,         XX
AREA1,          XX
STRI18,         XX
STRIP7,         XX
PAUSE BEGIN
```

```
/START AND STOP DATA ACQUISITION
/TYPE IN BIN LIMITS,CHANNEL SHIFTS, LOW AND HIGH GROUPS
/ENTER HERE TO TYPE IN BINS, SHIFTS, AND LOW AND HIGH GROUPS.
INISH,          LAC BINSET 4     /JMS CRLF
                SKP
/ENTER HERE TO TYPE IN BINS ONLY.
BINSET,         LAC INISH1-1     /JMP CRLFDM
                DAC INISH9-1
                LAW INISH1-1
                JMS NMSTAS
                JMS CRLF
                LAW BINLIM
                LMQ
                LAM-BINGRP
                JMS NMIND
                XX               /JMS CRLF OR JMP CRLFDM
/ENTER HERE TO TYPE IN SHIFTS AND LOW AND HIGH GROUPS ONLY.
INISH9,         LAW INISH2-1
                JMS NMSTAS
                JMS CRLF
                LAW SHFTS
                LMQ
                LAM-BINGRP-2
                JMS NMIND
                JMP CRLFDM
INISH1,         021116           /BINLIMS
                141115
                230000
INISH2,         231011           /SHIFTS,LO,HI
                062423
                544014
                175440
                101100
SETTIM,         LAC (241575      /TM=  SET TIME FOR RUN
                JMS STRTYP
                DAC SETTT1
                JMP DISMIS
SETTT1,         0
CLOCKS,         JMS STBM         /START HERE FOR NEW RUN
/STBM STOPS DATA TAKING IF NOT STOPPED, CLEARS ALL FLOPS, AND LOADS WC
                DZM 6            /RESET CLOCK
                LAM-7020 1
                DAC 7
                LAM-4            /REF AS CLOCKS 4 IN CLOCK
                DAC CL#CK6
                IOT 3006         /RESET SCALERS
                JMS STOVER       /INITIALIZE WORD OVERFLOW BUFFER
                LAM-21           /ZERO SCALER STORAGE (STRIPPED ASCII)
                DAC SCL4
                LAW SCLR-1
                DAC 10
                LAC (606060      /ZERO IN ASCII
                DAC I 10
                ISZ SCL4
                JMP .-2
                LAM-TOFDIM-MONSZE 1
                DAC ZER#OT1
                LAW COLLO-1
LOCZOT,         JMS ZERO         /ZERO COLLAPSED AND MONITOR
                LAM-UNCSZE TOFDIM 1
```

```
                DAC ZEROT1
                LAW UNCOLL-1
                JMS ZERO        /ZERO FULL SPECTRUM LESS GROUP 7
/THE FOLLOWING ROUTINE SETS UP A TABLE OF 100 (OCTAL) LOCATIONS IN THE FORMAT
/AC0=1 IF THE GROUP INTO WHICH THE COUNT FALLS IS NOT TO BE COLLAPSED,
/    =0 IF THE GROUP IS TO BE COLLAPSED.
/AC1-4 CONTAIN THE NUMBER OF CHANNELS GROUP IS TO BE SHIFTED BEFORE COLLAPSING.
/AC5-17 CONTAIN THE STARTING ADDRESS OF THE GROUP INTO WHICH A COUNT WOULD FALL.
/HIGH AND LOW GROUP TO USE ARE STORED IN LLOW AND LHI, FOLLOWING SHFTS.
/SHIFTS ARE STORED STARTING AT SHFTS, FOLLOWING BINLIM.
/THE GROUP INTO WHICH A COUNT OF GIVEN PULSE HEIGHT WOULD FALL IS DETERMINED
/BY THE PULSE HEIGHTS STORED STARTING AT BINLIM.
/MAXIMUM PULSE HEIGHT IS 177 (OCTAL).  IF PULSE > 77, IT IS STORED IN THE
/GROUP WHOSE ADDRESS IS STORED IN THE LAST ENTRY IN THE TABLE.
/IF PULSE HEIGHT ≤ 77, IT IS STORED IN THE GROUP WHOSE STARTING ADDRESS
/IS STORED IN THE ENTRY CORRESPONDING TO THE PULSE HEIGHT.
                JMS PTABL1      /START TO SET UP PROCESSOR
                LAC LHI
                TAD (1
                DAC INOTTO 1    /INOTTO 1-6 ARE TEMPORARIES USED ELSEWHERE
                LAW SHFTS
                DAC INOTTO 2
                LAW BINLIM
                DAC INOTTO 3
                DZM INOTTO 4    /PULSE SIZE
                DZM INOTTO 5    /GROUP NUMBER
                LAW PTABLE-1
                DAC 10
                LAM-PULSZE 1    /COUNTER
                DAC INOTTO 6
PTAB11,         LAC INOTTO 5
                SAD LLOW        /LOW GROUP TO USE
                DZM INOTTO 7    /400000 OR 0
                SAD INOTTO 1    /HIGH GROUP+1 TO USE
                JMS PTABL1
                LAC INOTTO 4
                SAD I INOTTO 3
                JMP PTAB10
                LAC INOTTO 5
                CLQ!LRS 14-SHFTOF
                CLC!CLL
                LRS 1
                LAC I INOTTO 2
                AND (17         /SHIFT
                LLS 15
                XOR INOTTO 7
                DAC I 10
                ISZ INOTTO 4
                ISZ INOTTO 6
                JMP PTAB11
                JMP STRT
PTAB10,         ISZ INOTTO 3
                ISZ INOTTO 2
                ISZ INOTTO 5
                JMP PTAB11
PTABL1,         XX
                LAC (400000
                DAC INOTTO 7    /SET NO COLLAPSE
                JMP I PTABL1
STRT,           DZM BUFTAG      /CONTINUE A RUN FROM HERE
```

```
                LAW SCRAM
                DAC CONSOL
                XCT ADC3 10      /LAM-XX
                DAC TES#T1
                LAC (JMP ADC1    /SET UP ADC CHECK
                DAC PROCE3 3
                LAC (JMP ADC2
                DAC PROCE3 4
                LAW BUF01
                DAC STORGE
                LAW BUF00
                LDMAC
/DO NOT CONTINUE DATA-TAKING VIA CONSOLE INTERRUPT IF STR IS DISABLED.
                LAC CODEWD
                ALS 15
                SMA
                JMP ERRT
                ENADC 3
                CLON
                ISZ BINCHK
                JMP DISMIS
/TYPE OUT BIN LIMITS, SHIFTS, AND LOW AND HIGH GROUPS, IN THAT ORDER.
SHFTBN,         LAC BSL
                JMS STRASC
                LAW BINLIM       /REF AS SHFTBN 2 BY XSORTA 3
                LMQ
                LAM-BONGRP-2
NMOUTA,         JMS RSUMQ
/NEXT THREE INSTRUCTIONS ARE PART OF THE CALLING SEQUENCE FOR RSUMQ.
                NOP
                JMP CRLFDM
                JMP RSUM4
BINCHK,         LAM
ZERO,           XX               /ZERO SELECTED REGION
                DAC 10
                DZM I 10
                ISZ ZEROT1
                JMP .-2
                JMP I ZERO
ZOTGRP,         LAM-TOFDIM 1     /ZERO SELECTED GROUP
                DAC ZEROT1
                LAC (072075      /GP=
                JMS GRPCHK
                CLQ!LLS SHFTOF
                TAD (UNCOLL-1
                JMS ZERO
                JMP DISMIS
/ENABLE THE SCALAR CHECK.
SCY,            LAC (ISZ CLCK6
                SKP
/DELETE THE SCALAR CHECK.
SCN,            LAC LOCNOP       /NOP
                DAC CLOCK 6
                JMP DISMIS
/ENABLE THE ADC CHECK.
ACY,            LAC (ISZ TEST1
                SKP
/DELETE THE ADC CHECK.
ACN,            LAC LOCNOP       /NOP
                DAC TEST 2
                JMP DISMIS
```

```
/STOPS DATA ACQUISITION, PROCESES LAST BUFFER
RDMAC=703714     /READ MEMORY ADDRESS COUNTER
CLCK1,           JMS STBM          /END OF TIMED RUN, STOP COUNTING
                 JMS CRLF
                 LAC (051724       /EOT
                 JMS STRASC
                 LAM-200 1
                 DAC BEL#LT        /REF AS CLCK1 5 IN SCLERR
                 JMS CRLF
                 LAW RESTRT
                 DAC CONSOL
                 LAW 207
                 JMS TYPOUT        /RING BELL
                 ISZ BELLT         /NORMAL HALT AFTER 200 BONGS
/AN ERROR HALT WILL RING THE BELL 262144 TIMES OR UNTIL STOPPED BY CONSOLE.
                 JMP .-3
/A CONSOLE INTERRUPT TO STOP A RUN (NORMAL HALT) COMES HERE.
SCRAM,           LAW STRT
/SET UP CONSOL SO THAT A SECOND CONSOLE INTERRUPT CONTINUES DATA TAKING.
                 DAC CONSOL
                 JMS STBM          /STOP THE BUBBLE MACHINE
                 JMS CLCK3         /UPDATE THE TIME
                 LAC BUF#TAG
                 SZA
                 JMP STOP1
                 LAW BUF00-1
STOP2,           DAC 11
                 RDMAC             /READ THE MEMORY ADDRESS COUNTER
                 CMA
                 TAD 11
                 TAD (20002        /GET RID OF STRAY LAW, -2
                 DAC BLA#ST
                 SPA
                 JMS PROCE3        /PROCESS LAST BUFFER
                 JMS SCALAR        /UPDATE THE SCALERS
                 JMS STPOVR        /CHECK OVERFLOWS
                 JMP DISMIS
STOP1,           LAW BUF01-1
                 JMP STOP2
/A CONSOLE INTERRUPT TO SHUT OFF THE BELL COMES HERE.
RESTRT,          LAM               /RING BELL ONCE MORE ONLY
                 DAC BELLT
                 JMP DISMIS
STBM,            XX                /STOP THE BUBBLE MACHINE
                 DISADC 3
                 CLOF
                 CLFLOP            /THESE 3 INSTRUCTIONS PREVENT AN INTERRUPT
                 LAW BUFSZE
                 LDWC              /BY A POSSIBLE FULL BUFFER
                 JMP I STBM
```

```
TSEQ,           IORS            /UPDATE THE TIME IF NEEDED AND TYPE OUT
                ALS 7           /SEE IF CLOCK IS ON
                SPA
                JMS CLCK3       /IF SO, UPDATE THE TIME
                LAC CLCK4
                JMS PRNTIN
                JMP CRLFDM
/A CLOCK INTERRUPT COMES HERE (ALWAYS).
CLOCK,          ISZ 6
                LAM-7020 1      /LOAD -1 MINUTES INTO 7
                DAC 7
                LAC 6
                SAD SETTT1
                JMP CLCK1       /END OF TIMED RUN
                ISZ CLCK6       /REF AS CLOCK 6 IN SCN
/PRECEDING INSTRUCTION = NOP IF SCALAR CHECK IS DISABLED.
                JMP . 4
                XCT CLOCKS 4    /LAM-X
                DAC CLCK6
                JMS SCLRCK      /CHECK SCALERS
                CLON
                JMP DISMIS
/ROUTINE TO UPDATE THE TIME TO THE NEAREST SECOND
CLCK3,          XX
                LAC 7
                SZA!CLL
                ADD (7055
                IDIV
                74
                LACQ
                DAC CLCK4
                LAC 6
                MUL
                74
                LACQ
                TAD CLCK4
                DAC CLCK4       /TIME IS STORED IN SECONDS HERE PERMANENTLY.
                JMP I CLCK3
PAUSE BEGIN
```

```
/PROCESSOR
ENADC=703400     /ENABLE THE ADC'S
DISADC=703204    /DISABLE THE ADC'S
CLFLOP=704102    /CLEAR ALL DATA BREAK FLOPS, ACCEPT NEW FLOP
LDMAC=703702     /CLEAR AND LOAD MAC
LDWC=703606      /LOAD WORD COUNTER,CLEAR WC FLAG
ZAP=CLL!CLA
/DETERMINE WHICH BUFFER IS FULL, SET UP NEW BUFFER SWITCH FOR INTRUP.
PROCES,          LAC BUFTAG
                 SZA
                 JMP .+4
                 LAW BUF00-1
                 ISZ BUFTAG
                 JMP .+3
                 LAW BUF01-1
                 DZM BUFTAG
                 DAC STORGE
                 ISZ STORGE
PROCE2,          DAC 11
                 LAM-BUFSZE 1
                 DAC BLAST
                 JMS PROCE3
                 JMP DISMIS
/MAIN ROUTINE FOR PROCESSING DATA, USING TABLE SET UP BY CLOCKS.
/AC0=0  TWO PARAMETER (TOF) EVENT
/   =1  MONITOR EVENT (ONE PARAMETER)
/AC1-7 CONTAIN PULSE HEIGHT FOR TOF
/AC8-17 CONTAIN CHANNEL (TIME) FOR TOF
/AC10-17 CONTAIN CHANNEL FOR MONITOR
PROCE3,          XX
                 LAC I 11
                 SPA!CLL
                 JMP MON          /REF AS PROCE3 3 IN STRT,ADC1,ADC3(2)
                 LRS 12           /REF AS PROCE3 4 IN STRT,ADC2(2),ADC3(2)
                 TAD (-PULSZE 1
                 SMA
/AN EVENT WITH PULSE HEIGHT > 77 GOES INTO HIGHEST GROUP BEING USED.
                 LAM
                 TAD (TAD PTABLE PULSZE
                 DAC PROCE4
                 LLS SHFTOF 1000
/1000 ADDED TO AN EAE INSTRUCTION CLEARS THE AC.
PROCE4,          HLT
                 DAC TOFL
                 ISZ I TOFL
                 JMP PROCE5-2
                 AND (17777       /TO READ AS 2XXXXX FOR OVERFLOW IN FULL
                 TAD (-UNCOLL 606500
                 JMS OVERB        /ADD THE INFORMATION TO OVERFLOW BUFFER
                 LAC TOFL
                 SPA!CLL
                 JMP TEST         /DO NOT COLLAPSE THIS GROUP
PROCE5,          LRS 15
                 TAD TOFL
                 AND (TOFMSK
                 TAD (ISZ COLLO
                 DAC TOFL
TOFL,            HLT
                 JMP TEST
                 TAD (-COLLO-ISZ 1                /TO READ AS XXXX FOR OVERFLOW
```

```
                JMS OVERB                       /IN COLLAPSED
TEST,           ISZ BLAST
                JMP PROCE3 1
                ISZ TEST1       /REF AS TEST 2 IN ACN
/PRECEDING INSTRUCTION = NOP IF ADC CHECK DISABLED.
                JMP I PROCE3
ADC3,           LAC (JMP ADC1   /TEST ADC'S TO MAKE SURE BOTH ARE COUNTING
                SAD PROCE3 3
                JMP . 11
                DAC PROCE3 3
                LAC (JMP ADC2
                SAD PROCE3 4
                JMP . 10
                DAC PROCE3 4
                LAM-10          /REF AS ADC3 10 IN STRT
                DAC TEST1
                JMP I PROCE3
                JMS STBM
                LAC (151716     /MON
                JMP . 3
                JMS STBM
                LAC (241706     /TOF
                JMS STRASC
                LAW . 1
                JMP SCLERR 4
                400104          / ADC ERR
                034005
                222200
ADC1,           LMQ
                LAC . 3
                DAC PROCE3 3
                LACQ
                JMP MON
ADC2,           LMQ
                LAC (LRS 12
                DAC PROCE3 4
                LACQ
                JMP PROCE3 4
MON,            AND (MONSZE-1
                TAD (ISZ MONTOR
                DAC STORG5
STORG5,         HLT
                JMP TEST
                TAD (-MONTOR-ISZ 303241         /TO READ AS 1XXXXX
                JMS OVERB                       /IN MONITOR OVERFLOW
                JMP TEST
```

```
/RECOLLAPSES FULL SPECTRUM BACK INTO COLLAPSED
REC,            LAC LOCZOT      /JMS ZERO
                SKP
/ADD GROUPS TO COLLAPSED
CRC,            LAC LOCNOP      /NOP
                DAC RECOL2-1
                LAC (140775     /LG=
                JMS GRPCHK      /STAY IN THIS ROUTINE UNTIL LEGAL GROUP IS GIVEN
                DAC GL#O
                LAC (100775     /HG=
                JMS GRPCHK      /STAY IN THIS ROUTINE UNTIL LEGAL GROUP IS GIVEN
                CMA
                TAD GLO
                SMA
                JMP ERRT        /ERROR IF HIGH GROUP LESS THAN LOW GROUP
                DAC GH#I
                LAM-TOFDIM 1    /REF AS RECOL2-4 IN RECOL2
                DAC ZEROT1
                LAW COLLO-1
                JMS ZERO        /ZERO COLLAPSED SPECTRUM
/PRECEDING IS A NOP IF ROUTINE STARTED FROM CRC
RECOL2,         LAW SHFTS       /USE SHIFTS AS TYPED IN FOR DATA TAKING
                TAD GLO
                DAC ZEROT1
                LAC I ZEROT1    /SHIFT COUNT
                AND (TOFMSK     /MAXIMUM SHIFT IS 777 (8)
                LMQ
                TAD (COLLO-1
                DAC 16
                DAC 15
                LACQ
                TAD RECOL2-4    /LAM-TOFDIM 1
                DAC ZEROT1
                LAC GLO
                CLQ!LLS SHFTOF
                TAD (UNCOLL-1
                DAC 14
                LAC I 14
                TAD I 16
                DAC I 15
                ISZ ZEROT1
                JMP .-4
                ISZ GLO
                ISZ GHI
                JMP RECOL2
                JMP DISMIS
GRPCHK,         XX              /ROUTINE TO ACCEPT A LEGAL GROUP NUMBER
                DAC GHI
                LAC GHI
                JMS STRTYP
                SPA
                JMP .-3         /START OVER IF GROUP IS NEGATIVE
                TAD RSLINE 2    /LAM-BINGRP 1
                SMA!CLL
                JMP .-6         /START OVER IF GROUP BIGGER THAN BINGRP-1
                LAC INOUT3      /PICK UP GROUP NUMBER
                JMP I GRPCHK
OVERB,          HLT             /ROUTINE TO CHECK OVERFLOWS
                DAC I JT        /END OF LIST
                LAW JLIST
```

```
                DAC JT 1
                LAC I JT
                SAD I JT 1
                JMP JFND
                ISZ JT 1
                ISZ JT 1
                JMP .-4
JFND,           ISZ JT 1
                ISZ I JT 1      /+1 TO COUNT FOR THIS ONE
                LAM
                TAD JT 1
                SAD JT
                JMP . 2         /=IMPLIES NEW CHANNEL
                JMP I OVERB
                ISZ JT          /INCREMENT END OF LIST
                ISZ JT          /POINTER TWICE
                LAW JLIST JSZ
                SAD JT
                SKP             /BUFFER OF OVERFLOWS FULL
                JMP I OVERB
                JMS STBM        /STOP THE BUBBLE MACHINE
                JMS STPOVR      /TYPE OUT OVERFLOW LIST
                JMS STOVER      /RESET THE OVERFLOW BUFFER
                LAW STRT        /CONTINUE RUN AFTER THIS BUFFER IS PROCESSED
                DAC PROCE3
                JMP I OVERB
STOVER,         HLT             /INITIALIZE OVERFLOW BUFFER
                LAW JLIST
                DAC JT
                DAC JT 1
                LAW JLIST JSZ
                DZM I JT 1
                ISZ JT 1
                SAD JT 1
                JMP I STOVER
                JMP .-4
JT,             0               /END OF LIST POINTER
                0               /MOVING POINTER
STPOVR,         HLT             /CHECK FOR OVERFLOWS, IF ANY,
                LAW JLIST 1     /AND TYPE OUT
                DAC STPVR1
                LAM-JSZ 2       /JSZ IS THE NO. OF WORDS IN JLIST
                DAC STPVR2
                LAC I STPVR1
                SNA
                JMP . 6
                ISZ STPVR1
                ISZ STPVR2
                JMP .-5
                LAM-JSZ
                JMP . 7
                LAC STP#VR2
                TAD (JSZ
                RAR
                RCL
                SNA!CMA
                JMP I STPOVR
                DAC STP#VR1
                LAW STPVR3-1
                JMS NMSTAS
```

```
                LAW JLIST
                LMQ
                LAC STPVR1
                JMS RSUMQ
/NEXT THREE INSTRUCTIONS ARE PART OF THE CALLING SEQUENCE FOR RSUMQ.
                NOP
                JMP . 2
                JMP RSUM4
                JMS CRLF
                JMP I STPOVR
STPVR3,         172622          /OVRFLOWS
                061417
                272300
STP1,           JMS STPOVR      /TYPE OUT OVERFLOWS
                JMP DISMIS
```

```
/ROUTINE TO READ IN SCALERS AND DO RELATIVE CHECKS ON THEM
/IF PROPER INEQUALITIES DO NOT HOLD, HALT RUN AND RING BELL.
SCLRCK,         XX
                DISADC 3
                JMS SCALAR          /READ IN SCALERS
                ENADC 3
                LAW SCLR 6          /SCALER 3
                DAC SCLRK1
                LAW SCLR 11         /SCALER 4
                JMS SCLRSB
                637464              /3 4
                LAW SCLR 17         /SCALER 6
                DAC SCLRK1
                LAW SCLR 14         /SCALER 5
                JMS SCLRSB
                667465              /6 5
                LAW SCLR 11         /SCALER 4
                DAC SCLRK1
                LAW SCLR 17         /SCALER 6
                JMS SCLRSB
                647466              /4 6
                JMP I SCLRCK
SCLRSB,         XX                  /SUBTRACT ONE SCALER FROM ANOTHER
                DAC SCLRK2
                LAM-2
                DAC SCLRK3
                CLL
                LAC I SCLRK2
                JMS CMATD1
                TAD I SCLRK1
                SNL
                JMP SCLERR
                SZA!CLL
                JMP . 5
                ISZ SCLRK1
                ISZ SCLRK2
                ISZ SCLRK3
                JMP .-12
                ISZ SCLRSB
                JMP I SCLRSB
SCLERR,         JMS STBM            /ERROR  STOP THE BUBBLE MACHINE
                LAC I SCLRSB
                JMS STRASC
                LAW SCLER1-1
                JMS NMSTAS          /REF AS SCLERR 4 IN ADC3
                JMP CLCK1 5         /NMSTAS EXITS ONLY WITH A 0 IN AC
SCLER1,         402303              /SCALER ERR
                011405
                224005
                222200
SCLRK1,         XX
SCLRK2,         XX
SCLRK3,         XX
```

```
/ROUTINES TO READ IN AND TYPE OUT SCALERS
SCALAR,         XX
                IOT 3116        /READ IN OVERFLOW WORD
                ALS 14
                DAC SCL#1
                LAW SCLR
                DAC SCL#3
                LAM-2           /SET FOR THREE SWEEPS OF DUAL SCALERS
                DAC SCL#4
                LAM-3           /READ IN 4 CHARACTERS, DISREGARD
                JMS SCPCKD
                JMS SCPCK1      /READ IN 7 CHARACTERS, STORE IN 3 WORDS
                ISZ SCL3
                LAM             /READ IN 1 CHARACTER, DISREGARD
                JMS SCPCKD
                JMS SCPCK1      /READ IN NEXT 7 CHARACTERS, STORE
                ISZ SCL4
                SKP
                JMP .-5
                ISZ SCL3
                LAM             /READ IN LAST CHARACTER OF LINE DISREGARD
                JMS SCPCKD
                JMP .-15
                IOT 3116        /READ ABSOLUTE LAST CHARACTER
SCPNWK,         DAC SCPACK
                LAM-5
                DAC SCL4
                IOT 3002        /ENABLE SCALER READ IN FOR NEXT PASS
                LAW SCLR
                DAC SCL3
                LAC SCL1
                SPA!CLA
                LAC (000100
                TAD I SCL3
                DAC I SCL3
                AND (007700
                SAD (007200
                JMP SCPACK-2
                LAC SCL3
                TAD (3
                DAC SCL3
                LAC SCL1
                RCL
                DAC SCL1
                ISZ SCL4
                JMP .-17
                LAC SCPACK      /READ LAST CHARACTER INTO AC
                SAD (15
                JMP I SCALAR
                JMS STBM        /SOMETHING WRONG; TURN OFF RUN
                JMS CRLF
                LAW SCLER1-1    /SCALER ERR
                JMS NMSTAS
                JMP STRT        /CONTINUE RUN; GIVE UP ON READING SCALERS
                LAC (006600
                JMP .-26
SCPACK,         XX              /READ IN N CHAR.  PACK INTO ONE WORD
                DAC SCL#2
                CLL
                LAC I SCL3
```

```
                CLQ!LRS 6
                ALS 6
                IOT 3103
                JMP .-1
                IOT 3104
                XOR (60
                DAC I SCL3
                AND (17
                TAD (LAM-11
                SMA!CLA
                JMP SCPNWK      /IF DIGIT>11
                LAC I SCL3
                ISZ SCL2
                JMP .-14
                JMP I SCPACK
SCPCKD,         XX              /READ IN X CHAR.  IGNORE
                DAC SCL2
                IOT 3103
                JMP .-1
                IOT 3104
                ISZ SCL2
                JMP .-4
                JMP I SCPCKD
SCPCK1,         XX              /READ IN 7 CHAR.  PACK INTO 3 WORDS
                LAM
                JMS SCPACK
                ISZ SCL3
                LAM-2
                JMS SCPACK
                ISZ SCL3
                LAM-2
                JMS SCPACK
                JMP I SCPCK1
SCLRTP,         LAW SCLRWD-1    /TYPE OUT SCALERS
                JMS NMSTAS
                JMS CRLF
                LAM-5
                DAC NMSTAS
                LAW SCLR-1
                DAC 10
                LAC I 10
                JMS STRASC
                LAC I 10
                JMS STRASC
                LAC I 10
                JMS STRASC
                JMS CRLF
                ISZ NMSTAS
                JMP .-10
                JMP DISMIS
SCLRWD,         230301          /SCALERS
                140522
                230000
PAUSE BEGIN
```

```
/IN AND OUT TO TELETYPE ROUTINES
/ROUTINES TO TYPE OUT AND READ IN NUMBERS FROM KEYBOARD
DEC,            LAM-12 1        /TYPE-IN IN DECIMAL
                SKP
OCT,            LAM-10 1        /TYPE-IN IN OCTAL
                DAC STORG6
                JMS CMATO1
                DAC STORG7
                JMP DISMIS
/ROUTINE TO ACCEPT FROM KEYBOARD A DECIMAL OR OCTAL INTEGER
TYPEIN,         XX
                DZM INO#UT3
TYPEN1,         JMS INTYPE
                SAD (254        /COMMA ENDS THE INTEGER
                JMP TYPEN3
                TAD (-260 1
                SPA
                JMP TYPEN2      /ERROR; ILLEGAL CHARACTER
                TAD STORG6
                SMA!STL
                JMP TYPEN2      /ERROR; ILLEGAL CHARACTER
                TAD STORG7
                DAC INO#UT2
                LAC INOUT3
                MUL
STORG7,         12              /10 FOR OCTAL
                LACQ
                TAD INOUT2
                DAC INOUT3
                JMP TYPEN1
/ACCEPTABLE NUMBER TYPED IN; SET UP PROGRAM TO ACCEPT FURTHER KEYBOARD
/INTERRUPTS AS EXECUTIVE COMMANDS.
TYPEN3,         LAW EXECIN
                DAC INTYPE
                LAC INOUT3
                JMP I TYPEIN
TYPEN2,         LAW 277         /QUESTION MARK
                JMS TYPOUT      /TYPE QUESTION MARK AND START ROUTINE FROM GO
                JMP TYPEIN 1
STORG6,         LAM-12 1        /LAM-10 1 FOR OCTAL
/CARRIAGE RETURN, LINE FEED
CRLF,           XX
                LAW 215
                JMS TYPOUT
                LAW 212
                JMS TYPOUT
                JMP I CRLF
/ROUTINE FOR SETTING UP N WORDS AT M
NMACQ,          HLT
                LAC (167240     /N:
                JMS STRTYP
                CMA
                DAC INO#UT4
                LAC (157240     /M:
                JMS STRTYP
                LMQ
                LAC INOUT4
                JMP I NMACQ
/ROUTINE TO TYPE OUT A STRIPPED ASCII WORD AND TO ACCEPT AN INTEGER
STRTYP,         XX
```

```
JMS STRASC      /STRIPPED ASCII WORD IN AC
JMS TYPEIN      /TYPE IN A NUMBER
JMS CRLF
LAC INOUT3      /PICK UP TYPED IN NUMBER
JMP I STRTYP
```

```
/TYPE OUT IN DECIMAL THE INTEGER IN AC
/ALL INTEGER TYPE-OUTS USE THE DOUBLE PRECISION ROUTINE
/SINGLE PRECISION ROUTINE ZEROS THE MOST SIGNIFICANT WORD.
PRNTIN,         XX
                DAC DOUB1 1     /PUT INTO LOW ORDER BITS
                CLA!CLL         /ZERO HIGH ORDER BITS
                JMS PRNTDB
                JMP I PRNTIN
/ROUTINE TO TYPE OUT DOUBLE PRECISION INTEGER
/ENTER WITH LOW ORDER BITS IN DOUB1 1, HIGH ORDER BITS IN AC.
PRNTDB,         XX
                DAC DOUB1       /HIGH ORDER BITS IN AC
                LAW INOTTO 10
/CONVERT THE DOUBLE PRECISION OCTAL INTEGER TO A STRING OF DECIMAL DIGITS.
                JMS DBTDEC
                LAM-6
                DAC PRN#TT2
                LAW INOTTO      /REF AS PRNTN3-2 IN DBTDEC
                DAC 10
PRNTN3,         LAC I 10
                SZA
                JMP PRNTN2
                LAC INOTTO      /INOTTO STARTS OFF WITH A 240 (SPACE).
                JMS TYPOUT
                ISZ PRNTT2
                JMP PRNTN3
                LAC I 10
/LAST CHARACTER CANNOT BE A SPACE; MUST BE 0-9.
                TAD PRNTN2 2
                JMS TYPOUT
                LAW 240
                DAC INOTTO
                JMP I PRNTDB
PRNTN2,         TAD . 2
                JMS TYPOUT
                LAW 260
                DAC INOTTO
                JMP PRNTN3 5
INOTTO,         240
INOTTO 10/      /INOTTO 6 IS REF BY BINBCD(2),BTDEC
/INOTTO AREA IS USED AS TEMPORARY STORAGE IN CLOCKS
/BTDEC IS A SINGLE PRECISION OCTAL TO DECIMAL CONVERSION ROUTINE.
BTDEC,          0               /USED AS INOTTO 10 IN PRNTDB
                DAC DOUB1 1     /AC INTO LOW ORDER BITS
                DZM DOUB1       /ZERO HIGH ORDER BITS
                LAW INOTTO 6
                JMS DBTDEC
                JMP I BTDEC
DBTDEC,         XX              /DOUBLE PRECISION BINARY TO
                DAC BTD#ET3     /DECIMAL ROUTINE
                LAC DOUB1
                CLL
                IDIV
                12
                DAC DBT#DE1
                LACQ
                DAC DOUB1
                LAC DOUB1 1
                LMQ
                LAC DBTDE1
```

```
                DIV
                12
                DAC I BTDET3
                LACQ
                DAC DOUB1 1
                LAM
                TAD BTDET3
                SAD PRNTN3-2
                JMP I DBTDEC
                JMP .-24
DOUB1,          XX              /HIGH ORDER BITS
                XX              /LOW ORDER BITS
/TYPE OUT IN DECIMAL N DECIMAL(OCTAL) WORDS START AT M DECIMAL(OCTAL)
TYPONM,         JMS NMACQ
                DZM RSUMB
NMOUTD,         JMS RSUMQ
/NEXT THREE INSTRUCTIONS ARE PART OF THE CALLING SEQUENCE FOR RSUMQ.
                JMS PRNTIN
                JMP CRLFDM
                JMP RSUM4       /NO RUNNING SUM
/TYPE IN N WORDS AT M
TYPINM,         JMS NMACQ
                JMS NMIND
                JMP CRLFDM
```

```
/ROUTINE TO ACCEPT N INTEGERS FROM KEYBOARD; STORE BEGINNING AT M.
NMIND,          XX
                DAC NMI#NDN
                JMS LACQM1
                DAC 10
                ISZ NMINDN
                SKP
                JMP I NMIND
                JMS TYPEIN
                DAC I 10
                JMP .-5
/TYPE N STRIPPED ASCII WORDS STORED BEGINNING AT M.
/ONE EXITS FROM NMSTAS WHEN STRASC, WHICH IS CALLED BY NMSTAS, EXITS WITH A
/ZERO IN THE AC.  STRASC EXITS WITH A ZERO IN THE AC IF IT ENCOUNTERS A
/STRIPPED ASCII CHARACTER CONSISTING OF 00.
NMSTAS,         XX              /CLEARS LINK AND AC ON EXIT
                DAC 10
                LAC I 10
                JMS STRASC
                SZA!CLL
                JMP .-3
                JMP I NMSTAS
/TYPES STRIPPED ASCII, 3 CHARACTERS PER WORD, STORED IN AC ON ENTRY.
STRASC,         XX              /CLEARS LINK ON EXIT
                DAC STR#AT1
                LAM-2
                DAC STR#AT2
STRAS1,         LAC STRAT1
                CLL
                CLQ!LRS 14
                OMQ
                DAC STRAT1
                AND (77
                SNA!CLL
                JMP I STRASC
                TAD (-37
                SMA
                TAD (-77
                TAD (340
STRAS3,         JMS TYPOUT
                ISZ STRAT2
                JMP STRAS1
                CLC!CLL
                JMP I STRASC
LACQM1,         XX
                LACQ
                AND (17777
                TAD LOCLAM      /LAM
                JMP I LACQM1
```

```
/TYPES BETWEEN LINES WITHOUT RUNNING SUM IN THE FORMAT
/REGION WHICH IS BEING DISPLAYED, THEN CHANNEL NUMBER AND 8 CHANNELS PER LINE.
TOLINE,         LAC (JMP NMOUTD
                SKP
/PERFORMS RUNNING SUM BETWEEN LINES
/TYPES OUT SAME INFORMATION AS TOLINE, AND IN ADDITION TYPES THE RUNNING SUM
/UNDER EACH CHANNEL WHOSE CONTENTS ARE TYPED OUT.
RSLINE,         LAC NMOUTD      /JMS RSUMQ
                DAC RSLIT5      /REF AS RSLINE 1 IN STRI10
                LAM-BINGRP 1    /REF AS RSLINE 2 IN GRPCHK
                DAC RSUM6
                LAS             /TYPE FROM FIRST GROUP WHOSE SWITCH IS UP.
                ALS 2
                SPA!RAL
                JMP RSLIN2      /MONITOR
                SPA!RAL
                JMP .-7         /GROUP X
                ISZ RSUM6
                JMP .-3
                ALS 11-BINGRP
                SMA
                JMP RSLIN6      /BACKGROUND IS TYPED OUT IF NO SWITCH IS UP
                JMP RSLIN1      /COLLAPSED
                LAW BINGRP
                TAD RSUM6
                DAC RSUM6
                TAD (117560     /G=X
                JMS STRASC
                LAC RSUM6
                CLQ!LLS SHFTOF
                TAD (UNCOLL
RSLIN5,         DAC RSUMB
                TAD LINELO
                LMQ
                LAC LINEHI
                TAD (1
                CMA
                TAD LINELO
RSLIT5,         XX              /JMS RSUMQ FOR RUNNING SUM, JMP NMOUTD FOR NO
                JMS PRNTIN      /TYPE CHANNEL NUMBER
                JMP RSUM1       /DO A RUNNING SUM
                LAM-7
/ROUTINE TO TYPE OUT A STRING OF INFORMATION, EIGHT PIECES PER LINE
/RSUMQ IS ENTERED WITH THE STARTING MACHINE ADDRESS OF THE INFORMATION IN THE MQ
/LAM- NO. OF PIECES OF INFORMATION (CHANNELS) IN THE AC, AND THE STARTING
/MACHINE ADDRESS OF THE REGION (GROUP, COLLAPSED, MONITOR, ETC.) IN RSUMB.
/IF ONE WANTS THE  CHANNEL  TO BE THE ACTUAL ADDRESS TYPED OUT, ZERO RSUMB.
/IF ONE DOES NOT WANT ANY CHANNEL NUMBER TYPED OUT, IGNORE RSUMB.
/THE ENTRY SEQUENCE IS JMS RSUMQ PLUS THREE MORE ENTRIES.
/1ST ENTRY=JMS PRNTIN TO TYPE CHANNEL OF FIRST BIT OF INFORMATION PER LINE
/         =NOP TO DELETE TYPE OUT OF CHANNEL NUMBER
/2ND ENTRY=JMP RSUM1 TO DO A RUNNING SUM
/         =JMP EXIT TO DELETE RUNNING SUM, WHERE EXIT IS THE ADDRESS TO WHICH
/          ONE WANTS TO EXIT AT FINISH OF TYPE OUT.  A RUNNING SUM WILL ALWAYS
/          JMP DISMIS AT COMPLETION.
/3RD ENTRY=LAM-7 FOR A RUNNING SUM
/         =JMP RSUM4 FOR NO RUNNING SUM
RSUMQ,          XX
                TAD (1
                SMA
```

```
LOCERT,         JMP ERRT        /REF AS RSUMQ 3 IN LOCK1, LOCK2, PLOTER
                DAC RSU#MN
                DAC RSU#MT3
                LAC I RSUMQ
                DAC RSUM4 5
                ISZ RSUMQ
                LAC I RSUMQ
                DAC RSUM2 4
                ISZ RSUMQ
                JMS LACQM1
                DAC 12
                DAC 13
                DZM RSU#MT2     /LOW ORDER BITS
                DZM RSUMT2 1    /HIGH ORDER BITS
                LAC RSUMB
                JMS CMATD1
                DAC RSU#MT4
                JMS CRLF        /REF AS RSUM4-1 IN RSUM3
RSUM4,          LAM-7
                DAC RSU#MT1
                LAC 12
                TAD (1
                TAD RSUMT4
                JMS PRNTIN      /REF AS RSUM4 5 IN RSUMQ
RSUM2,          LAC I 12
                JMS PRNTIN
                ISZ RSUMT3
                SKP
                JMP RSUM1       /REF AS RSUM2 4 IN RSUMQ
                ISZ RSUMT1
                JMP RSUM2
RSUM1,          JMS CRLF
                XCT I RSUMQ
                DAC RSUMT1
                LAW 240
                JMS TYPOUT
                ISZ RSUMT1
                JMP .-3
                LAM-7
                DAC RSUMT1
RSUM3,          LAC RSUMT2
                CLL
                TAD I 13
                DAC RSUMT2
                DAC DOUBI 1     /LOW ORDER BITS HERE
                SZL
                ISZ RSUMT2 1
                LAC RSUMT2 1
                JMS PRNTDB      /HIGH ORDER BITS IN AC
                ISZ RSUMN
                SKP
                JMP CRLFDM
                ISZ RSUMT1
                JMP RSUM3
                JMS CRLF
                JMP RSUM4-1
RSLIN1,         LAC (031714     /COL
                JMS STRASC
                LAC (COLLO
                JMP RSLIN5
```

```
RSLIN2,         LAC (151716     /MON
                JMS STRASC
                LAC (MONTOR
                JMP RSLIN5
RSLIN6,         LAC (020704     /BGD
                JMS STRASC
                LAC (RECOLO
                JMP RSLIN5
PAUSE BEGIN
```

```
/BINARY DUMP, BASED ON WORK OF EGJP
/BEGIN A NEW TAPE IN BINARY
BT2,            LAS
                DAC WSELEC
                ISZ WSELEC
                JMP ERRT        /ALL SWITCHES MUST BE UP
                LAC LOCNOP      /NOP
                SKP
/ADD A RUN TO BINARY TAPE
WDUMP,          LAC BCDEO1      /JMS WEND
                DAC WXDUMP
                IORS            /CHECK IF CLOCK IS ENABLED
                ALS 7
                SPA
                JMP ERRT        /YES  DO NOT DUMP WHILE TAKING DATA
                LAW 22          /SET ODD PARITY, TAPE 2
                DAC WSELEC 3
                LAM
                DAC TEM#P
                LAC (24000      /WRITE RING, EOT CHECK
                DAC WM1
/NEXT INSTRUCTION IS REFERENCED AS WXDUMP-2 IN WDISMI.
                LAC (20         /LOCK OUT STR
                JMS LOCK
WXDUMP,         XX              /JMS WEND FOR END OF TAPE OR NOP FOR NEW TAPE
                LAC (JMP AL1
                DAC AL4-1       /READ COMPARE ON BINARY TAPE
                JMS WONERO      /DUMP THE RUN NUMBER AS ONE RECORD
                LAW RUNUMB-1
                LAM
                JMS WONERO      /DUMP THE COLLAPSED AS ONE RECORD
                LAW COLLO-1
                LAM-TOFDIM 1
                JMS WONERO      /MONITOR+SCALERS+TIME IN SECS+BSL+OVERFLOW LIST
                LAW MONTOR-1
                LAM-MONSZE-24-BONGRP-JSZ
                LAC WSPIT2      /DUMP 4096 AS WGRP RECORDS OF WGRPN CHANNELS
                DAC . 4
                LAM-WGRP 1
                DAC WSPIT
                JMS WONERO      /4096
                XX
                LAM-WGRPN 1
                ISZ WSPIT
                SKP
                JMP . 5
                LAC .-5
                ADD WSPIT3
                DAC .-7
                JMP .-11
                JMS WFIN        /WRITE TWO ENDS OF FILE
WDISMI,         MCD
                LAC (20         /UNLOCK STR
                JMS UNLOCK
                ISZ TEMP
                JMP DISMIS
                LAW 26          /SELECT TAPE 6 ;IF ACTIVE, DUMP ON IT
                DAC WSELEC 3
                MTS
                ISZ TEMP
```

```
                JMP .-1
                MSUR
                JMP DISMIS
                JMP WXDUMP-2
WSPIT,          XX
WSPIT2,         LAW UNCOLL-1
WSPIT3,         WGRPN
/RETREIVE A COMPLETE RUN IN BINARY FROM TAPE DRIVE 7
GT7,            LAW 27
                SKP
/RETREIVE A COMPLETE RUN IN BINARY FROM TAPE DRIVE 2
GT2,            LAW 22
                DAC WSELEC 3
                LAM
                DAC BINCHK
                DZM WM1
                LAC RUNUMB
                JMS WSEAR       /SEARCH FOR A RUN WHOSE NO=RUNUMB
                JMS WINT        /RETREIVE COLLAPSED SPECTRUM
                LAW COLLO-1
                LAM-TOFDIM 1
                JMS WINT        /MONITOR+SCALERS+TIME+BSL+OVERFLOW LIST
                LAW MONTOR-1
                LAM-MONSZE-24-BONGRP-JSZ
                LAC WSPIT2      /RETREIVE THE WGRP RECORDS COMPRISING THE 4096
                DAC . 4
                LAM-WGRP 1
                DAC WSPIT
                JMS WINT        /4096
                XX
                LAM-WGRPN 1
                ISZ WSPIT
                SKP
                JMP WGTEND
                LAC .-5
                ADD WSPIT3
                DAC .-7
                JMP .-11
WGTEND,         MWC-1
                LAW 640         /SKIP TO END OF FILE
                JMS WFUN
                MCD             /CLEAR ALL MAG TAPE FLAGS
/PRECEDING INSTRUCTION IS REFERENCED IN BACKSP, BCDE01.
                JMP DISMIS
WERR,           MCD             /ERROR EXIT
                JMP ERRT
```

```
/ROUTINES TO GET MONITOR OR COLLAPSED OF ANY RUN
GTM,            LAC . 11        /GET MONITOR OF ANY RUN, PUT IN GROUP 7
                SKP
GTC,            LAC . 15        /GET COLLAPSED OF ANY RUN, PUT IN GROUP 7
                DAC . 11
                LAW 22
                DAC WSELEC 3
                DZM WM1
                LAC RUNUMB
                JMS WSEAR       /FIND RUN NUMBER
                JMS WINT        /COLLAPSED
                LAW UNCOLL UNCSZE-TOFDIM-1
                LAM-TOFDIM 1
                XX              /JMS WINT OR JMP WGTEND
                LAW UNCOLL UNCSZE-TOFDIM-1
                LAM-MONSZE 1
                JMP WGTEND
/ROUTINE TO BACKSPACE N FILES ON TELETYPE COMMAND
BACKSP,         LAW 22
                DAC WSELEC 3
                DZM WM1
                LAC (160675     /NF=
                JMS STRTYP
                SNA!CMA
                JMP ERRT        /A COMMAND TO BACKSPACE 0 FILES IS ILLEGAL
                TAD LOCLAM      /LAM
                JMS WBACK
                JMP WGTEND 3
/SELECT TAPE 2 ODD PARITY 200 BITS PER INCH
WSELEC,         0
                MSCR
                JMP .-1
                LAW 22          /REF AS WSELEC 3 IN WDUMP,GT2,TP1OUT,WDISMI
                MTS
                MSUR
                JMP .-1
                MTRS
                AND WM1         /24000 END POINT, WRITE TAPE
                SZA
                HLT
                JMP I WSELEC
/BASIC TAPE FUNCTION SELECTOR
/FUNCTION IN AC
WFUN,           JMP DISMIS
                MCD
                MTC
                JMP DISMIS
/TO BACKSPACE OVER N FILES, IF POSSIBLE, AND POSITION AT
/BEGINNING OF LAST FILE PASSED OVER
/NUMBER OF FILES TO BACKSPACE IS IN AC
WBACK,          0
                DAC WTIMES      /SET COUNTER
                JMP . 4
                MWC-1
                LAW 740         /BACKSPACE ONE FILE
                JMS WFUN
                JMS WSELEC
                MTRS            /IS TAPE AT LOAD POINT
                ALS 5           /10000 LOAD POINT MASK
                SPA
```

```
JMP I WBACK       /AT LOAD POINT
ISZ WTIMES        /COUNT
JMP .-11
LAW 600           /SPACE OVER EOF
JMS WFUN
JMP I WBACK
```

```
/TO SEARCH FOR A FILE STARTING WITH WORD IN AC
WSEAR,          0
                DAC WTEM
                LAM-3           /BACKSPACE 3 FILES
                JMS WBACK
                LAM-1
                DAC WTIMES      /SET 2 TRIES
WFRW,           LAW WTEM-1      /SET CA WC
                MCA
                LAM
                MWC
                LAW 400         /READ COMPARE
                JMS WFUN
                MTRS
                ALS 3           /40000 EOF MASK
                SPA
                JMP WNO 2
                MTRS
                ALS 2           /100000 RDC MASK
                SMA
                JMP I WSEAR     /FOUND IT
                MWC-1
                LAW 640         /SKIP ONE FILE
                JMS WFUN
                JMP WFRW        /TRY NEXT
WNO,            JMS WFUN7       /NOT FOUND BACKSPACE
                JMP WERR        /NO GO RETURN
                ISZ WTIMES      /TEST
                SKP
                JMP WNO         /NO GO
                LAW 100         /REWIND
                JMS WFUN
                JMS WSELEC      /MAKE SURE UNIT IS READY BEFORE CONTINUING
                JMP WFRW        /SCAN FROM BEGINNING
```

```
/TO READ ONE BINARY RECORD.  ENTRY SEQUENCE IS
/JMS WINT
/LAW STARTING ADDRESS-1 OF REGION TO PUT INFORMATION
/LAM-NO OF WORDS+1
WINT,           0
                LAC I WINT      /SET IA
                DAC AL2
                ISZ WINT
                LAC I WINT      /SET WC
                DAC AL3
                ISZ WINT        /NO EXIT SET
                LAM-2
                DAC WTIMES      /SET 3 TRIES
                JMS WSELEC
WHOM,           JMS WHICH
                LAW 500         /READ IT
                JMS WFUN
                MTRS
                ALS 3           /40000 EOF MASK
                SPA
                JMP WERR        /EOF
                MTRS
                AND (600200     /600200 READ OK MASK
                SNA
                JMP I WINT      /OK RETURN
                ISZ WTIMES      /TEST
                SKP
                JMP WERR        /FAILED 3 TIMES
                JMS WFUN7       /BACKSPACE TO TRY AGAIN
                JMP WHOM        /TRY AGAIN
/CONSTANTS USED AT VARIOUS PLACES THROUGHOUT MAGNETIC TAPE ROUTINES
WM1,            24000
WTIMES,         0
WTEM,           0
WFUN7,          XX              /BACKSPACE ONE RECORD
                LAW 700
                JMS WFUN
                JMP I WFUN7
WHICH,          XX              /SET UP CA WC AND WC
AL2,            XX              /SET UP BY WONERO OR WINT
                MCA             /SET CA WC
AL3,            XX              /SET UP BY WONERO OR WINT
                MWC             /SET WC
                JMP I WHICH
```

```
/WRITE ONE BINARY RECORD. CALLING SEQUENCE IS
/JMS WONERO
/LAW ADDRESS-1 OF REGION TO DUMP
/LAM-NO OF WORDS+1
WONERO,         0
                LAC I WONERO
                DAC AL2
                ISZ WONERO
                LAC I WONERO
                DAC AL3
                ISZ WONERO      /SET EXIT
                JMS WSELEC      /SELECT  REF AS WHER-1 IN AL4
WHER,           JMS WHICH
                LAW 200         /WRITE
                JMS WFUN
                MTRS
                AND (600000
                SNA             /REQUEST AND PARITY
                JMP AL1         /NOW READ COMPARE  REF AS AL4-1 IN WXDUMP,TP1OUT
/PRECEDING INSTRUCTION IS A JMP I WONERO IF DUMPING IN BCD
AL4,            JMS WFUN7       /BACKSPACE
                LAW 300         /EOF,BAD TAPE
                JMS WFUN
                JMS WFUN7       /BACKSPACE
                JMP WHER-1
AL1,            JMS WFUN7       /BACKSPACE
                JMS WSELEC
                JMS WHICH
                LAW 400         /READ COMPARE
                JMS WFUN
                MTRS
                ALS 2           /CHECK FOR READ COMPARE ERROR
                SMA
AL5,            JMP I WONERO
                JMP AL4
```

```
/WRITE 2 EOF AND BACKSPACE OVER ONE
WFIN,           0
                JMS WSELEC
                LAW 300          /EOF
                JMS WFUN
                LAW 300
                JMS WFUN
                JMS WFUN7        /BACKSPACE
                JMP I WFIN
/TO POSITION AT END OF FILES, I.E., 2 EOF
/AND BACKSPACE OVER ONE
WEND,           0
                LAM-2
                JMS WBACK        /BACKSPACE 2 FILES
WFOR,           MWC-1
                LAW 640          /SKIP 1 FILE
                JMS WFUN
                LAW 600          /SPACE 1 RECORD
                JMS WFUN
                MTRS
                ALS 3            /40000 EOF MASK
                SMA
                JMP WFOR         /NOT YET
                JMS WFUN7        /BACKSPACE OVER EOF
                JMP I WEND
```

```
/ROUTINE TO TYPE IN NEW RUN NUMBER
NEWRUN,         LAW RUN1-1
                JMS NMSTAS
                LAC RUNUMB
                JMS PRNTIN
                JMS CRLF
                LAW RUN2-1
                JMS NMSTAS
                LAW . 22
                DAC TYPEIN      /SET UP TYPEIN FOR NON-JMS ENTRY
                DZM INOUT3
                JMS INTYPE
                SAD (240        /BLANK; RUN NUMBER STAYS THE SAME
                JMP . 11
                SAD (253        /+ ADD 1
                JMP . 5
                SAD (255        /- SUBTRACT 1
                SKP!CLC
                JMP TYPEN1 1    /NONE OF ABOVE; ACCEPT NEW RUN NUMBER
                JMP . 4
                LAC (1
                SKP
                CLA
                TAD RUNUMB
                DAC INOUT3
                JMP TYPEN3
                DAC RUNUMB
                JMP CRLF
                LAW RUN2-1
                JMS NMSTAS
                LAC RUNUMB
                JMS PRNTIN
                JMP CRLFDM
RUN1,           171404          /OLD RUN=
                402225
                167500
RUN2,           160527          /NEW RUN=
                402225
                167500
RUNUMB,         XX
TEMP,           XX
TALLY,          XX
PAUSE BEGIN
```

```
/OUTPUT ON TAPE 3 IN BCD, ADAPTED FROM CODE OF JKD
/FIND END OF BCD TAPE, DO NOT DUMP
BCDEOT,         LAW 3
                DAC WSELEC 3
                LAC (JMP BCDEO1
                JMP . 4
/DUMP THE RUN, DO NOT FIND END OF TAPE
TP1OUT,         LAW 4003        /4000 FOR 6-BIT DUMP
                DAC WSELEC 3
                LAC (JMS INRUN
                DAC . 7
                LAW DISMIS      /IGNORE ALL TELETYPE REQUESTS IN THIS ROUTINE
                DAC INTYPE
                LAC AL5         /JMP I WONERO
                DAC AL4-1       /DELETE THE READ COMPARE
                LAC (24000      /CHECK FOR WRITE RING, EOT
                DAC WM1
                XX              /JMP BCDEO1 OR JMS INRUN
                JMS WOTF1       /START SETTING UP BCD OF DATA
                JKD1            /NUMBER OF RECORDS IN 4096 CHANNELS
                0               /START AT CHANNEL 0
                UNCOLL-1        /ADDRESS-1 OF 4096 CHANNELS
                LAW 1
                DAC BCDRUN 1    /CHANGE RUN NUMBER TO READ 1XXXXX FOR COLLAPSED
                JMS WOTF1       /WRITE COLLAPSED SPECTRUM
                JKD2            /NUMBER OF RECORDS IN COLLAPSED
                0               /START AT CHANNEL 0
                COLL0-1         /ADDRESS-1 OF COLLAPSED SPECTRUM
                ISZ BCDRUN 1    /CHANGE RUN NUMBER TO READ 2XXXXX FOR MONITOR
                JMS WOTF1       /WRITE MONITOR SPECTRUM
                JKD3            /NUMBER OF RECORDS IN MONITOR
                0               /START AT CHANNEL 0
                MONTOR-1        /ADDRESS-1 OF MONITOR
                JMS WFIN        /WRITE 2 EOF AND BACKSPACE OVER ONE
                SKP
BCDEO1,         JMS WEND        /FIND END OF TAPE  REF IN WDUMP FOR JMS WEND
                LAW EXECIN
                DAC INTYPE
                JMP WGTEND 3
```

```
/INPUT RUN NUMBER
/TO INPUT RUN NUMBER OF 5 DECIMAL DIGITS, DECODE,
/STORE IN PROPER PLACE AS BCD INFORMATION
INRUN,          XX
                LAW BCDRUN-1
                DAC 16
                LAW 1
                DAC I 16
                LAW 20
                DAC I 16
                LAC RUNUMB
                JMS BCDPRT
ENDTMP,         JMP I INRUN
RECOLO/
/THE FOLLOWING PARTS OF THE PROGRAM ARE PUT IN 5000 (RECOLO)
/WHICH CAN THEN BE READ IN WITH PLT.
/ROUTINE TO CONVERT OCTAL INTEGER TO A STRING OF DECIMAL DIGITS,
/STORE IN A REGION SELECTED BY CALLING PROGRAM (AUTO INDEXER 16)
BCDPRT,         XX
                JMS BINBCD
                LAM-4           /5 DIGITS ONLY
                DAC BCD4
                LAW INOTTO 1
                DAC 12
                LAC I 12
                DAC I 16        /16 SET UP BY BCDPRT CALLING PROGRAM
                ISZ BCD4
                JMP .-3
                JMP I BCDPRT
```

```
/TO WRITE N RECORDS OF 16 DATA PER RECORD  (FOR 256 CHANNELS,N=16,
/FOR 4096 CHANNELS N=256,ETC)  START WITH ANY CHANNEL
/CALLING SEQUENCE
/JMS WOTF1
/NUMBER OF RECORDS
/VALUE OF FIRST CHANNEL
/ADDRESS OF 1ST OF 16N CONSEQUTIVE DATA
WOTF1,          XX
                JMS NOROUT      /DUMP RUN NUMBER
                LAC I WOTF1
                JMS CMATD1
                DAC WLOOP
                LAC WOTF1
                DAC 14
                LAC I 14
                DAC WLPP        /VALUE OF FIRST CHANNEL
                LAC I 14
                DAC WLPP 1
WLP,            JMS F1FORM
                BCDARY-1        /FOR AUTO INDEXING
WLPP,           XX              /CHANNEL NUMBER
                XX              /ADDRESS OF 1ST OF 16 DIGITS
WLPO,           JMS WONERO
                LAW BCDARY-1
                LAM-167
                ISZ WLOOP       /FINISHED YET
                SKP
                JMP I 14        /YES
                LAC WLPP        /NO
                TAD (20
                DAC WLPP
                LAC WLPP 1
                TAD (20
                DAC WLPP 1
                JMP WLP
```

```
/SET UP 120 BCD WORDS, DUMP THE RUN NUMBER
NOROUT,         XX
                LAW BCDARY
                DAC 14
                LAM-167
                DAC TALLY
                LAW 20
                DAC I 14
                ISZ TALLY
                JMP .-2
                JMS WONERD
                LAW BCDRUN-1
                LAM-167
                JMP I NOROUT
/BINARY TO BCD SUBROUTINE
/SINGLE PRECISION 6 DECIMAL DIGIT OUTPUT
/SUPPRESS LEADING ZEROS
BINBCD,         XX              /18 BIT BINARY IN AC AT JMS
                JMS BTDEC
                LAW 20
                DAC NOROUT
                LAC INOTTO 6
                SNA
                LAW 12
                DAC INOTTO 6
                LAW INOTTO
                DAC 10
                DAC 12
                LAM-5
                DAC BCD2
                LAC I 10
                SZA
                JMP BCD3
                LAC NOROUT
                DAC I 12
                ISZ BCD2
                JMP .-6
                JMP I BINBCD
BCD3,           DAC I 12
                LAW 12
                DAC NOROUT
                JMP .-6
```

```
/WRITE F1FORMAT (I1,I4,16I7,3X)
/ROUTINE TO PREPARE LINE OF DATA OUTPUT IN BCD TO BE WRITTEN
/ON MAGNETIC TAPE.  WRITES A BLANK,CHANNEL NUMBER,
/THEN 16 CHANNELS OF DATA
/CALLING SEQUENCE
/JMS F1FORM
/ADDRESS-1 OF FIRST OF 40 DECIMAL CONSEQ. BCD WORDS TO GO ON TAPE
/CHANNEL NUMBER, 4 DECIMAL DIGITS MAXIMUM
/ADDRESS-1 OF FIRST OF 16 CONSEQUTIVE DATA IN BINARY
F1FORM,         XX
                LAC I F1FORM     /LOAD ADDRESS-1 OF 1ST BCD WORD
                DAC 16
                LAC F1FORM
                DAC 15
                LAC I 15         /LOAD CHANNEL NUMBER
                JMS BCDPRT       /STORE IN I5 FORMAT WITH FIRST CHARACTER A BLANK
                LAC I 15         /GET ADDRESS-1 OF FIRST BINARY WORD
                DAC 13
                LAM-17
                DAC BCD5
                LAC I 13
                JMS BINBCD
                LAM-5
                DAC BCD4
                LAW INOTTO
                DAC 12
                LAW 20
                DAC I 16
                LAC I 12
                DAC I 16
                ISZ BCD4
                JMP .-3
                ISZ BCD5
                JMP .-15
                LAW 20
                DAC I 16
                DAC I 16
                DAC I 16
                JMP I 15
PAUSE BEGIN
```

```
/CALCOMP PLOTTING ROUTINE,COURTESY OF WEK
/PLOTS SCOPE DISPLAY (LINEAR SCALE ONLY)
/USES AREA DEFINED BY CALBUF AS 512 WORD BUFFER.
PLOTER,         LAC DISPYS
                ALS 3
                SMA
                JMP ERRT        /DISPYS IS A JMP (LOG OR SQUARE ROOT MODE)
                LAC N763        /LOCK OUT STRIP,INT,LPT,STR,BCD,PLT,OTHERS
                JMS LOCK
                DZM CALXO       /ZERO CALCOMP X AND Y COORDINATES
                DZM CALYO
                LAW SYMBLX
                DAC CALSYM      /SET X AS PLOTTING SYMBOL
                LAC DISPLN
                JMS MUL20       /NEGATIVE OF CHANNEL NUMBER MOD 20 LEFT IN AC
                LMQ
                TAD DISPLN
                DAC CALCHO      /CALCOMP RELATIVE LOW CHANNEL NUMBER
                LACQ
                TAD DISPLM
                DAC CALADO      /ABSOLUTE STARTING CHANNEL NUMBER
                LAM
                TAD DISPLN
                TAD DISPFS
                DAC CALCHL
                JMS MUL20
LOCSZA,         SZA!CLL         /REF IN BLKNUM
                TAD (24
                TAD CALCHL
                DAC CALCHL      /LAST CALCOMP RELATIVE CHANNEL NUMBER
                TAD (LAM-777    /TEST IF CALCHL 511 (10)
                SMA!CMA
                CLA             /YES, USE -511 (10)
                TAD (LAM-777    /MAXIMUM NO. OF CHANNELS IS 511 (10)
                TAD CALCHO
                DAC CALNO       /NUMBER OF POINTS TO PLOT
                DAC CALNO1
                LAS!CLL         /FIND AC DISPLAY SCALE FACTOR
                AND (37
                LRS 4
                CMA
                DAC CALSKL
                LLS 1004
                ISZ CALSKL
                JMS CMATD1
                TAD (12
                DAC CALSKL      /PUT SCALE FACTOR IN CALSKL
                CLC!STL         /START SET-UP OF CALCOMP BUFFER
                TAD CALADO
                DAC 16
                LAW CALBUF-1    /CALBUF DEFINED IN EQUALITIES
                TAD CALCHO
                DAC 15
                DAC CALADO      /CALADO WILL BE INCREMENTED BEFORE USING
                LAC I 16        /START TO CONVERT TO CALCOMP Y-COORDINATES
                XCT DISPYS      /LRS OR LLS
                LRS 12
                SZA!CLA!CLL     /TEST IF NO. 1023 (10)
                CLQ!CMQ         /YES, PUT LAM IN MQ
                LLS 13          /MAXIMUM SCALE ON CALCOMP IS 2047 (10)
```

```
                DAC I 15
                ISZ CALNO
                JMP .-10
CALAB,          LAC CALSKL      /START GRAPH LABELING
                JMS BTDEC
                LAC (CLO!LLS 2
                DAC LETSZE      /SET LETTER SIZE
                LAC N3740
                LMQ
                CLA
                JMS BLKNUM      /PRINT SCALE FACTOR
                LAC RUNUMB
                JMS BTDEC
                LAC N3740
                LMQ
                ISZ LETSZE
                LAC CALXO
                TAD (100
                JMS BLKNUM      /PRINT RUN NUMBER
                LAM
                TAD LETSZE
                DAC LETSZE      /RESTORE LETTER SIZE
                LAC (-150 1
                DAC CALBF1
                LAC CALCHO
                JMS BTDEC
                LAC (-62 1
                LMQ
                LAC CALBF1
                JMS BLKNUM      /PRINT CHANNEL NUMBER
                LAC CALBF1
                TAD (310
                DAC CALBF1
                LAC CALCHO
                SAD CALCHL
                JMP . 4
                TAD (24
                DAC CALCHO
                JMP .-15
                LAC N421        /UNLOCK STR,BCD,OTHERS(WHICH USE NUMBER OUTPUT)
                JMS UNLOCK
DOPLOT,         CLA             /START POINT PLOTTING
                DAC PLTRX
                ISZ CALADO
                LAC I CALADO
                DAC PLTRY
                LAC I CALSYM
                JMS PTPLOT
PLTRX,          XX
PLTRY,          XX
                LAC PLTRX
                TAD (12
                DAC PLTRX
                ISZ CALNO1
                JMP .-13
                LAC N2246
                TAD PLTRX
                DAC CALXNU
                CLA!CLL
                DAC CALYNU
```

```
                JMS CALIN       /ADVANCE PAPER 6 INCHES
                LAC N763        /UNLOCK EVERYTHING LOCKED OUT BY PLT
                JMS UNLOCK
                JMP DISMIS
/PLOTS POINT.   C(AC)= ADDRESS OF CHARACTER. X,Y AS ARG
PTPLOT,         XX
                DAC PTPL01
                LAC I PTPLOT
                TAD (-5 1
                DAC PTX0        /SET X COORDINATE OF LOWER LEFT OF CHARACTER
                DAC CALXNU
                ISZ PTPLOT
                LAC I PTPLOT
                TAD (-5 1
                DAC PTY0        /SET Y COORDINATE OF LOWER LEFT OF CHARACTER
                DAC CALYNU
                ISZ PTPLOT
                CLL
                JMS CALIN       /MOVE PEN TO CHARACTER LOWER LEFT
                LAM-4 1
                DAC PLTCTR      /SET FOR 3 PAIRS OF RELATIVE COORDS IN CHARACTER
                LAC PTPL01      /SET UP COORDINATES FOR PEN MOTION
                LMQ
                LLS 1002
                JMS PTPL03
                TAD PTX0
                DAC CALXNU
                LLS 1002
                JMS PTPL03
                TAD PTY0
                DAC CALYNU
                LACQ
                DAC PTPL01
                JMS CALIN       /MOVE TO COMPUTED CHARACTER COORDS, PEN DOWN
                ISZ PLTCTR
                JMP .-16
                JMP I PTPLOT
PTPL03,         XX              /MULTIPLIES CHARACTER RELATIVE COORD IN AC BY 5
                JMS CMATD1
                DAC PTPL02
                STL!SNA!CLA
                JMP I PTPL03
                TAD (5
                ISZ PTPL02
                JMP .-2
                JMP I PTPL03
SYMBLX,         512240          /CODE FOR X PLOTTING CHARACTER
MUL20,          XX              /LEAVES -C(AC) MOD 20 IN AC
                CLL
                IDIV
                24
                JMS CMATD1
                JMP I MUL20
```

```
/BLOCK NUMBERS
/PRINTS INOTTO 1=>6 AT X IN AC, Y IN MQ
BLKNUM,         XX
                DAC BLKNXO
                DAC CALXNU
                LAC LOCSZA      /SZA!CLL
                DAC . 7         /SET LEADING ZERO SWITCH
                LAW INOTTO 1
                DAC BLKNU1
                LACQ
                DAC BLKNYO
                DAC CALYNU
                LAC I BLKNU1
                XX              /SWITCH TO SUPPRESS LEADING ZEROS
                JMP . 16
                LAW INOTTO 6
                SAD BLKNU1
                JMP I BLKNUM
                LAC (6
                XCT LETSZE
                TAD BLKNXO
                DAC BLKNXO
                DAC CALXNU
                CLL
                JMS CALIN       /ADVANCE FOR NEXT DIGIT
                ISZ BLKNU1
                LAC BLKNYO
                JMP .-20
                LMQ             /NON-ZERO DIGIT
                LAC . 5
                DAC .-21        /SET SWITCH TO PRINT ZEROS
                LACQ
                JMS BLKPR       /PRINT DIGIT
                JMP .-22
                JMP .-2
/PRINTS BLOCK NUMBER, DECIMAL DIGIT IN AC
BLKPR,          XX
                RCR
                TAD (CODSTR
                DAC BLKPR1
                LAC I BLKPR1
                SZL
                LLS 11          /DIGIT CODE FOR BLOCK PATTERN IN AC
                JMS BLOCKR
                JMP I BLKPR
CODSTR,         770002          /BLOCK PATTERN DIGIT CODE FOR 0-9 PACKED 2/WORD
                664364
                154334
                734160
                774374
/RUNS THROUGH BLOCK PATTERN, LEADING AC BITS CONTROL PEN
BLOCKR,         XX
                DAC BLOCK1      /STORE BLOCK PATTERN CODE
                LAC CALYNU
                DAC BLOCYO
                LAC (3
                XCT LETSZE
                TAD CALYNU
                DAC CALYNU
                CLL
```

```
                JMS CALIN       /MOVE TO LOCATION OF DIGIT
                LAC BLOCYO
                DAC CALYNU
                LAC BLOCK1
                SNA
                JMP I BLOCKR
                AND (774000
                SNA!CLL
                JMP NUMONE      /TREAT THE DIGIT "1" SPECIALLY
                LAW BLOCOD-1    /START RUN THROUGH BLOCK PATTERN
                DAC BLOCK2
                LAC BLOCK1
                RAL
                DAC BLOCK1
                JMS CALIN       /EXECUTE CURRENT LINE OF PATTERN
                ISZ BLOCK2
                LAW BLOCOD 14
                SAD BLOCK2
                JMP I BLOCKR
                LAC I BLOCK2
                XCT LETSZE
                TAD CALXNU
                DAC CALXNU
                ISZ BLOCK2
                LAC I BLOCK2
                XCT LETSZE
                TAD CALYNU
                DAC CALYNU
                JMP .-21
NUMONE,         LAC (1          /SPECIAL TREATMENT OF "1"
                XCT LETSZE
                TAD CALXNU
                DAC CALXNU
                CLL
                LAC (6
                XCT LETSZE
                TAD CALYNU
                DAC CALYNU
                JMS CALIN
                LAC BLOCYO
                DAC CALYNU
                STL
                JMS CALIN
                LAC (-2 1
                XCT LETSZE
                TAD BLKNXO
                DAC BLKNXO
                JMP I BLOCKR
LETSZE,         CLQ!LLS 2       /CONTROLS SIZE OF DIGITS
BLOCOD,         4               /STORAGE OF RELATIVE (X,Y) BLOCK PATTERN COORDS
                0               0               3
                0               3               -4 1
                0               0               -3 1
                4               0
/MOVES PEN TO X,Y IN CALXNU,CALYNU IN 2'S COMPLEMENT
/LINK=1=>PEN DOWN;LINK=0=>PEN UP
CALIN,          XX
                GLK             /FIND OUT IF PEN IS ALREADY SET CORRECTLY
                SAD LINKR
                JMP . 7
```

```
DAC LINKR        /IF PEN NOT SET CORRECTLY, SET IT
RCR
LAW 40
SNL!CLL
LAW 20
JMS CALCMP       /SET PEN
LAC CALXO
JMS CMATD1
TAD CALXNU
DAC CALDX        /DELTA X
SZA
JMP . 3
LAW
JMP . 10
SMA              /TEST FOR ● OR - DELTA X
JMP . 5
JMS CMATD1       /DELTA X IS -
DAC CALDX        /-DELTA X => DELTA X
LAW 1            /-X MOTION
SKP
LAW 2            /+ X MOTION
DAC CALXER       /SET X MOTION
LMQ
LAC CALXNU
DAC CALXO
LAC CALYO
JMS CMATD1
TAD CALYNU
DAC CALDY        /DELTA Y
SNA
JMP . 10
SMA              /TEST + OR - Y
JMP . 5
JMS CMATD1       /DELTA Y IS -
DAC CALDY        /- DELTA Y => DELTA Y
LAW 4            /-Y MOTION
SKP
LAW 10           /+Y MOTION
OMQ
DAC CALYER       /SET COMBINED X AND Y MOTION
LAC CALYNU
DAC CALYO
LAC CALDY
JMS CMATD1
TAD CALDX
SMA!CLL          /TEST DELTA X > DELTA Y
JMP . 12         /IF DELTA X>DELTA Y, JMP
LAC CALDX        /IF DELTA Y>DELTA X, INTERCHANGE
LMQ
LAC CALDY
DAC CALDX
LACQ
DAC CALDY
LAW 14
AND CALYER
DAC CALXER
LAC CALDX        /START MOTION SET UP
SNA
JMP CALXER 5
JMS CMATD1
```

```
                DAC CALCTR      /SET NUMBER OF STEPS
                RCL
                DAC CALNT       /-(2 DELTA X) =>CALNT
                LAC CALDY
                RCL
                DAC CALDY       /(2 DELTA Y) => DELTA Y
                LAC CALDX
                TAD CALDY
                DAC CALNA       /MOTION LOOP START
                TAD CALNT
                SPA
                JMP . 4
                DAC CALNA
CALYER,         XX              /LAW FOR X,Y. SETS AC BITS FOR COMBINED MOTION
                SKP
CALXER,         XX              /LAW FOR X. SETS AC BITS FOR X MOTION ONLY
                JMS CALCMP      /TELLS CALCOMP TO DO ITS THING ON INTERRUPT
                LAC CALNA
                ISZ CALCTR
                JMP .-14
                JMP I CALIN
LINKR,          0               /CONTAINS CURRENT STATE OF PEN; UP OR DOWN
N763,           763
N421,           421
N3740,          3740
N2246,          2246
CALNA,          XX
CALNT,          XX
CALCTR,         XX
CALDY,          XX
CALDX,          XX
BLOCK2,         XX
BLOCY0,         XX
BLOCK1,         XX
BLKPR1,         XX
BLKNY0,         XX
BLKNU1,         XX
BLKNX0,         XX
PTPL02,         XX
PLTCTR,         XX
PTY0,           XX
PTX0,           XX
PTPL01,         XX
CALYNU,         XX
CALXNU,         XX
CALBF1,         XX
CALSKL,         XX
CALN01,         XX
CALN0,          XX
CALCHL,         XX
CALAD0,         XX
CALCH0,         XX
CALSYM,         XX
CALY0,          XX
CALX0,          XX
ENDTMP 1/       /PUT ALL TEMPORARY STORAGE AND CONSTANTS BEFORE 5000 (RECOLO)
PAUSE BEGIN
```

EQUALITIES

ACN	2422	ACSDIS	42	ACSG1	4602	ACSG2	4601	ACSRES	32
ACY	2420	ADC1	2653	ADC2	2660	ADC3	2625	AHI	540
AL0	534	AL1	4372	AL2	434|	AL3	4343	AL4	4365
AL5	44₀2	AREA	1300	AREA1	2167	AUT010	10	AUT011	11
AUT012	12	AUT013	13	AUT014	14	AUT015	15	AUT016	16
AUT017	17	BACKSP	4164	BCDARY	7513	BCDEOT	4503	BCDO1	4543
BCDPRT	5000	BCDRUN	7504	BCD2	7502	BCD3	511|	BCD4	7477
BCD5	7500	BEGIN	24	BELLT	2133	BINBCD	5064	BINCHK	2374
BINGRP	10	BINLIM	7423	BINSET	2174	BLAST	2130	BLKNUM	5424
BLKNU1	5755	BLKNX0	5756	BLKNY0	5754	BLKPR	5465	BLKPR1	5753
BLOCKR	5503	BLOCK1	5752	BLOCK2	5750	BTDEC	3503	BTDEY0	5751
BONGRP	20	BSL	1665	BTDEC	3503	BTDET3	4562	BT2	4001
BUFSZE	100	BUFTAG	2127	BUF00	7577	BUF01	7677	CALAB	5247
CALAD0	5773	CALBF1	5766	CALBUF	17000	CALCHL	5772	CALCH0	5774
CALCMP	2060	CALCTR	5745	CALDX	5747	CALDY	5746	CALIN	5611
CALNA	5743	CALNB	5771	CALNB1	5770	CALNT	5744	CALSKL	5767
CALSYM	5775	CALXER	5730	CALXNU	5765	CALX0	5777	CALYER	5726
CALYNU	5764	CALY0	5776	CENTRD	1051	CLCF	7025₀2	CLCK1	2425
CLCK2	2132	CLCK3	2525	CLCK4	7422	CLCK6	4565	CLCON	7041₀1
CLFLOP	7041₀2	CLOCK	2510	CLOCKS	2233	CLWC	7036₀2	CMATDD	366
CMATD1	546	CODEWD	2122	CODSTR	5476	COLL0	6000	CONSOL	2041
CRC	2677	CRLF	3406	CRLFDM	1617	DBTDEC	3511	DBTDE1	4561
DEC	3343	DIFNOT	710	DISADC	7032₀4	DISDAC	621	DISMIS	2020
DISPFC	4577	DISPFR	422	DISPFS	432	DISPLA	146	DISPLB	733
DISPLK	245	DISPLM	4600	DISPLN	4604	DISPYS	224	DISP1	220
DISP2	221	DISP21	212	DISP4	244	DISP9	175	DOPLOT	5315
DOUB1	3537	ENADC	7034₀0	ENDTMP	4560	ERRT	1615	EXECIN	1553
EXECI2	1565	EXECI3	1621	EXECI6	1702	EXECT1	1701	EXECT2	2123
EXECT4	2124	EXPAND	4004	F1FORM	5115	GHI	2145	GLO	2144
GRPCHK	2750	GTC	4146	GTM	4144	GT2	4102	GT7	4100
HIEQ	572	HIEQA	6006	HILINE	1653	INISH	2172	INISH1	2216
INISH2	2221	INISH9	2206	INOTT0	3473	INOUT2	2146	INOUT3	4563
INOUT4	2147	INRUN	4547	INT	1532	INTRAC	4570	INTRMQ	4567
INTRSC	4566	INTRUP	1753	INTYPE	2044	JFND	2775	JKD1	400
JKD2	40	JKD3	20	JLIST	7445	JSZ	32	JT	3032
KEYMSK	17764₀	LACOM1	3620	LDMAC	7037₀2	LDWC	7036₀6	LEQ	6000
LETSZE	5574	LHI	7444	LINEHI	545	LINELO	544	LINKIL	4000
LINKR	5736	LINTST	246	LINTT1	4576	LINWHR	437	LIN1	264
LIN2	430	LLOW	7443	LMUL1	313	LOCERT	3675	LOCK	2107
LOCLAM	504	LOCNOP	526	LOCSZA	5202	LOCZOT	2256	LOEQ	564
LOEQA	610	LOG	616	LOG2	275	LOLINE	1652	LPT	2072
LTMP1	357	LTMP2	363	MAP	152	MON	2665	MONDIS	122
MONSZE	400	MONTOR	7000	MUL20	5416	NOSPLY	624	NEWRUN	4432
NMACQ	3414	NMIND	3552	NMINUN	2152	NMOUTA	2370	NMOUTD	3543
NMOUTN	2151	NMSTAS	3564	NMSTT1	2153	NOROUT	5047	NOSTRP	1521
NUMONE	5551	N2246	5742	N374₀	5741	N421	5740	N763	5737
OCT	3345	OUTBCD	7503	OVERB	2763	PENACT	2055	PENDO	440
PEN1	446	PEN2	4575	PLOTER	5153	PLT	7025₀4	PLTCTR	5760
PLTRX	5324	PLTRY	5325	PNHI	472	PNLO	470	PRNTDB	3441
PRNTIN	3434	PRNTN2	3466	PRNTN3	3451	PRNTT2	2150	PROCES	2544
PROCE2	2556	PROCE3	2563	PROCE4	2576	PROCE5	2610	PTABLE	7477
PTABL1	2332	PTAB10	2326	PTAB11	2301	PTPLOT	5344	PTPLO1	5763
PTPLO2	5757	PTPLO3	5404	PTXO	5762	PTYO	5761	PULSZE	100
RDMAC	7037₀4	READR	2065	REC	2675	RECOL0	5000	RECOL2	2717
RESTRT	2467	ROT8CT	532	ROT8IC	533	ROT8IN	512	ROT8L	475
ROT8R	515	RSLINE	3627	RSLIN1	3765	RSLIN2	3771	RSLIN5	3657
RSLIN6	3775	RSLIT5	3666	RSUMB	2134	RSUMN	2135	RSUMQ	3672
RSUMT1	2142	RSUMT2	2137	RSUMT3	2136	RSUMT4	2141	RSUM1	3734
RSUM2	3725	RSUM3	3745	RSUM4	3717	RSUM6	2143	RUNUMB	4500

RUN1	4472	RUN2	4475	SCALAR	3163	SCLERR	3146	SCLER1	3154
SCLR	7400	SCLRCK	3160	SCLRK1	3161	SCLRK2	3161	SCLRK3	3162
SCLRSB	3124	SCLRTP	3317	SCLRWD	3340	SCL1	2157	SCL2	2162
SCL3	2160	SCL4	2161	SCN	2415	SCON	703701	SCPACK	3252
SCPCKD	3275	SCPCK1	3305	SCPNWK	3212	SCRAM	2442	SCY	2413
SETAUT	1055	SETTIM	2226	SETT1	2232	SHFTBN	2363	SHFTOF	11
SHFTS	7433	SKOF	702501	SMK	1033	SQDIV	357	SQRT	334
SQRTA	612	SQSC	364	SQSH	363	SQT	313	STBM	2472
STBP1	2465	STBP2	2432	STBRGE	1776	STBRG5	2670	STBRG6	3405
STBRG7	3371	STBVER	3020	STPBVR	3034	STPVR1	2164	STPVR2	2163
STPVR3	3073	STP1	3076	STRASC	3573	STRAS1	3577	STRAS3	3613
STRAT1	2154	STRAT2	2155	STRIP	636	STRIP1	114	STRIP2	647
STRIP3	1065	STRIP4	1435	STRIP5	1450	STRIP6	1472	STRIP7	2171
STRIP8	1437	STRIP9	1226	STRI10	1176	STRI11	2166	STRI12	1167
STRI14	2165	STRI15	747	STRI16	765	STRI17	127	STRI18	2170
STRI19	1514	STRI20	1516	STRI21	656	STRI22	1471	STRI23	1025
STRI24	673	STRI25	1003	STRI26	1005	STRI27	1016	STRI28	1507
STRT	2336	STRTYP	3426	SUMX	1045	SUMY	1046	SWC	703601
SYMBLX	5415	TALLY	4502	TEMP	4501	TEST	2621	TEST1	4564
TBFDIM	1000	TBFL	2615	TBFMSK	777	TBFT4	2125	TBFT5	2126
TBLINE	3625	TBPLBP	372	TP1BUT	4507	TSEQ	2501	TYPEIN	3352
TYPEN1	3354	TYPEN2	3402	TYPEN3	3376	TYPINM	3547	TYPBNM	3541
TYPBUT	2050	UNBBLL	100000	UNBSZE	100000	UNLBCK	2114	VAL	552
VAL1	2156	VARANC	1053	VARBEG	1431	VAREND	4574	VARLBC	1121
VARMIN	4573	VARSTR	4572	VAR1	1225	VAR2	4571	VAR3	1343
VAR3A	1374	VAR4	1412	WAN	631	WAY	633	WBACK	4216
WCHPNT	4603	WDISM1	4060	WDUMP	4007	WEND	4414	WERR	4142
WFIN	4404	WFBR	4417	WFRW	4244	WFUN	4212	WFUN7	4334
WGRP	20	WGRPN	400	WGTEND	4135	WHER	4356	WHICH	4340
WHBM	4311	WINT	4277	WLBBP	7501	WLP	5026	WLPP	5030
WLP0	5032	WM1	4331	WNB	4266	WBNERB	4346	WBTF1	5013
WSEAR	4236	WSELEC	4176	WSPIT	4075	WSPIT2	4076	WSPIT3	4077
WTEM	4333	WTIMES	4332	WXDUMP	4025	ZAP	754000	ZERB	2375
ZERBT1	2131	ZBTGRP	2403						

ORNL-4299
UC-32 — Mathematics and Computers

INTERNAL DISTRIBUTION

1. Biology Library
2-4. Central Research Library
5-6. ORNL — Y-12 Technical Library Document Reference Section
7-26. Laboratory Records Department
27. Laboratory Records, ORNL R.C.
28. L. S. Abbott
29. R. Auble
30. J. K. Bair
31. J. A. Biggerstaff
32. J. K. Dickens
33. J. S. Eldridge
34. R. L. Ferguson
35. R. Gwin
36. J. A. Harvey
37. R. W. Hendricks
38. K. M. Henry
39. W. H. Jordan
40. C. E. Larson
41. M. E. LaVerne
42. H. G. MacPherson
43-44. F. C. Maienschein
45-54. A. L. Marusak
55. J. W. McConnell
56. F. K. McGowan
57. C. D. Martin
58. A. H. Narten
59. G. D. O'Kelley
60. J. S. Olson
61. D. J. Nelson
62. F. G. Perey
63. J. W. Reynolds
64. R. T. Santoro
65. H. W. Schmitt
66. M. J. Skinner
67. G. G. Slaughter
68. P. H. Stelson
69. E. A. Straker
70. D. A. Sundberg
71. J. W. Wachter
72. A. M. Weinberg
73. H. L. Yakel
74. E. R. Cohen (consultant)
75. B. C. Diven (consultant)
76. W. N. Hess (consultant)
77. M. H. Kalos (consultant)
78. L. V. Spencer (consultant)

EXTERNAL DISTRIBUTION

79. P. M. Aebersold, Tennecomp, Inc., Oak Ridge, Tennessee
80. J. H. Barrett, Mathematics Department, University of Tennessee
81. P. B. Hemmig, Division of Reactor Development and Technology, U.S. Atomic Energy Commission, Washington, D.C.
82. J. A. Swartout, Union Carbide Corporation, New York
83. I. F. Zartman, Division of Reactor Development and Technology, U.S. Atomic Energy Commission, Washington, D.C.
84. Documents Room, Computing Center, University of Notre Dame, Notre Dame, Indiana
85. Laboratory and University Division, AEC, ORO
86-373. Given distribution as shown in TID-4500 under Mathematics and Computers category (25 copies — CFSTI)