

United States
Department of the Interior
U.S. Geological Survey

345 Middlefield Road
Menlo Park, California 94025

Program ISDS
An Interactive Seismic Display System For
Displaying and Measuring Seismic Wave Forms

by
Peter R. Stevenson

Open-File Report 79-205

1978

This report is preliminary and has not been
edited or reviewed for conformity with Geo-
logical Survey standards and nomenclature.

Any use of tradename and trademarks in this report
is for descriptive purposes only and does not
constitute endorsement by the U.S. Geological Survey.

Contents

2

Page

Part I: ISDS USERS GUIDE

A. INTRODUCTION

1. Purpose.....	4
2. Process flow.....	4
3. Hardware required	
3.a Digital Hardware.....	5
3.b Analog Hardware.....	5
4. Analog Tape Format.....	5
5. Operation of A/D Program.....	6
6. Communication Between ISDS and EVCON.....	6
7. Communication Between ISDS and Phase files.....	7
8. Command Structure of ISDS.....	7

B. HOW TO USE ISDS

1. Basic Execution Procedures.....	8
2. Summary of ISDS Commands.....	8
3. Description of Pick Commands.....	11
4. Examples.....	12

Part II: PROGRAMMER'S REFERENCE

A. ISDS FROM THE INSIDE

1. Overview.....	15
2. Creation of a Multichannel Data Window.....	16
3. Major Processor Algorithms	
3.a Command Processor.....	19
3.b Event/Pass Processor.....	19
3.c Window Definition Processor.....	20
3.d Trace Definition Processor.....	20
3.e Plot Processor.....	21
3.f Pick Processor.....	21
3.g Shift Processor.....	22
3.h Scale Processor.....	22
4. Brief Description of Subroutines.....	22
5. Access to Digital Data.....	24
6. Interprocess Communication.....	24
7. Support Software.....	25
8. References.....	26

B. APPENDICES

1. Tables

A. Parameters Used by ISDS.....	27
B. Include Files Used by ISDS.....	28
C. Data Files Used by ISDS.....	29
D. Links Used by ISDS.....	29
E. Description of Common Block Variables.....	30
F. Subroutine, Common Block, and Include File Cross Reference Table.....	35
G. ISDS Subroutine Calls Cross Reference Table.....	37

	3
H. Support Subroutine Calls Cross Reference Table.....	39
I. Format of Phase Data Output.....	40
J. Format of 'HYPOUT' File.....	41
K. Format of Auto-Pick 'PIKLST' File.....	42
L. Formats of EVDIR Files.....	43
M. Routines Which Differ Between Three-Pass and Twelve-Pass. Versions.....	48
2. Listings	
N. Listings of ISDS Subroutines.....	
O. Listings of Some Support Software.....	
P. Listings of Twelve-Pass Routines.....	

PART I: USERS GUIDE

A. INTRODUCTION

1. PURPOSE

The specific purpose of ISDS is to display seismic traces on the Tektronix screen and to allow the operator to make meaningful measurements from the display, using the interactive cursor. This program is only one of a collection of programs whose general purpose is to display and process seismic data which have previously been recorded on analog tape.

Historically these functions have been performed manually. The functions include timing the seismic waveforms from microfilm displayed on a developer screen and punching the appropriate data on computer cards for location on a large scale computer. More recently, the timing has been done using a digitizing table which outputs a code on punch cards for further processing and eventual location. In both these instances the time between display of the data and location of the earthquake has been a few days.

In order to shorten the time between the occurrence of an earthquake and the determination of its location, a system has been implemented for performing all of the analysis on a minicomputer which has been interfaced to the analog playback equipment (Figure 1).

2. PROCESS FLOW

The minicomputer positions the analog tape to the correct time by reading signals from a time code translator and an electronic footage counter and sending control signals to the playback transport. After the tape has been positioned, the seismic signals are digitized by the minicomputer and stored on a disk. The data is retrieved from disk and displayed by ISDS. After the operator has picked the P arrival times and made other appropriate measurements from the seismic traces, the measurements are output by ISDS to a disk file in a format compatible with the input requirements of standard earthquake location programs.

One of the outputs of the location program is a file which contains the P and S travel time residuals. These are the differences between the observed and calculated arrival times. It is possible to read this file using ISDS, and to display both the residuals and the original picks together with the seismic waveform. If an obvious error in picking has occurred, it is possible at this point to correct one or more picks and relocate the event again. Other possibilities include modifying the crustal model if systematic errors are apparent. Thus, a powerful tool for refining the analysis is available.

3. HARDWARE REQUIRED

The present system is implemented on a Data General S200 Eclipse minicomputer which is interfaced to a Bell and Howell 1 inch Playback transport. In order to run the system, a minimum of 32 kilowords of memory and a 10 megabyte capacity disk are required. A detailed description of the hardware is given below.

3.a DIGITAL HARDWARE:

- CPU: Eclipse S-200 with ERCC, MAP, 16KW
- Memory: additional 16 Kw minimum
- Floating-point Processor
- Disc System, 10 Mbyte
- A/D Converter and 32 channel Multiplexer
- Tektronix 4014 Terminal
- Parallel and serial interface for Tektronix
- Teletype and Serial Interface
- Real Time Clock
- Programable Interval Timer
- Digital I/O System

3.b ANALOG HARDWARE:

- Analog Tape Drive: Bell and Howell VR3700B
- Discriminators: 39 Model 402P Tri-coms
- Time Code Translator
- Patch Panel - In House Design and Manufacture

4. ANALOG TAPE FORMAT

There are presently two different analog tape formats that can be processed by this system. One is the Geothermal 'Centipede' tapes, the other is the California Network telemetry tapes.

The Geothermal tapes are arranged such that all of the data for a given earthquake or other seismic event are recorded on a single tape. The tape is composed of 14 tape tracks, with each track containing 8 channels which are frequency division multiplexed onto the track. Twelve of the tracks are used for data, one track is reserved for timing signals, and one track for tape speed compensation signals. Since only 32 multiplexer inputs are available with the A/D converter, only 4 tracks can be sampled at a given time. The A/D conversion process requires 3 passes of this analog tape in order to retrieve data from all 12 tape tracks (96 channels). A different version of the A/D program is required for the Geothermal tapes than that used for the California Network tapes. The Geothermal digitizing scheme is also known as the Three Pass System.

The California Network telemetry tapes (Cal-Net Tapes) utilize all 14 tape tracks for data, and record 2 timing signals and a tape speed compensation signal at other frequencies on each of the 14 tracks. The Network is currently recording on 3 tapes simultaneously and these tapes are then copied or 'Dubbed' onto a master "Library Dub Tape" such that each event occupies 3 consecutive dubbed 'patches' on the tape. Since these 3 patches have the same time, uniqueness is obtained by calculating a hexadecimal footage counter address for each of the dubs. Since each dub required four passes at the A/D conversion process ($3 \times 32 + 1 \times 16$) and since there are 3 dubs; a total of 12 passes are required to obtain all possible seismic data for a given event. The version which processes this data is called the Twelve Pass System.

5. OPERATION OF A/D PROGRAM

Two programs, EVCON and EVDIG, are required to perform the A/D conversion process. The first of these, EVCON, is an interactive user interface, which transfers control to EVDIG. Upon receiving control, EVDIG positions the analog tape to the requested time and begins the A/D conversion and memory to disk transfers. Upon completion, control returns to EVCON and the user for subsequent processing. This includes the option of swapping to ISDS to view the data.

6. COMMUNICATION BETWEEN ISDS AND EVCON

There are three files required for communication between ISDS and EVCON. They are EVDIR, the event directory, which contains most of the descriptive information about the digital data; EVDAT, which contains the digital data; and MASTERLIST, which contains the names of the seismic stations. These files reside on DPO, the 5 Mbyte removable disk. Detailed information about the format of these files is contained in Appendix L of Part II of this report. A brief description is given below.

<u>Link Name</u>	<u>Resolution File</u>	<u>Description</u>
DIRECTORY	DPO: EVDIR	Random, binary file, contains event number, title, date and time, sample rate, digitizing duration, I.D. of channels digitized, pointers to information in EVDAT and MASTERLIST.
WHOLE1	DPO: EVDAT	Contiguous binary file. Contains 16 bit integer data.
MASTERLIST	DPO: MASTERLIST	Random, binary file. Usually a copy of HYP071 station file. Contains names of stations and coordinates.

7. COMMUNICATION BETWEEN ISDS AND PHASE FILES

The results of picking P and S phases and the measurement of the usual parameters associated with earthquake locations can be written to an external file (phase list) in the proper format for processing by HYPOINVERSE, HYPO71 or a similar location program. Also, a previously written phase list file can be read back by ISDS, and slightly modified, using the interactive cursor, and then rewritten.

In addition, the file created by HYPOINVERSE which contains P and S residuals, together with the original picks can be read by ISDS and both the picks and residuals can be displayed.

If an automatic phase picker program has generated a file in the proper format, those picks can be edited by ISDS and written as a phaselist file. A brief description of the files is given below. More detailed descriptions are in Appendices of Part II.

<u>File Name</u>	<u>Description</u>
User defined Default is FAZLST	ASCII random file. Created by Save command. Contains station name, P and S times, and other phase list variables. Can be read by picker Y commands.
User defined Default is HYPOUT	ASCII random file. Generated by HYPOINVERSE. Contains station names, P and S times, P and S residuals, etc. Read by picker H command.
PIKLST	ASCII random file. Generated by automatic picker program. Contains P times, first motions, and coda times. Read by picker G command.

8. COMMAND STRUCTURE OF ISDS

There are three modes of commands in ISDS. The primary mode consists of inputting two or four letter commands, in response to the prompt COMMAND?. This mode directs control of the various programs within the program. Within those processors, further control is handled by prompts from the program which are followed by typed in answers at the keyboard. The third mode is the confined to the Picking Process. Following the Command PI, the interactive cursor is activated, and control is via inputs to it.

A brief description of some useful command sequences will be given here. Either upon entry to ISDS, or by typing the command EV, the event number is requested, followed by a request for the pass number. If all of the data is to be displayed, simply type PL. If a more selective display is desired, type

DE to define the data window, and the program will prompt for the start time and length of the display time window. Entering NU allows the user to select which traces, by number, which he/she wishes displayed. Following these selections, type PL to plot the desired traces within the desired time window.

The user may interact with the data by entering PI. This causes the interactive cross-hair cursor to come on. The user positions the cross hair and enters the appropriate keyboard letter P for P-time, S for S-time, F for coda end time, etc. Upon typing '.', the program exits from the picker processor. The user may then enter SA to save the phaselist. The program prompts the user for the phase list filename (FAZLST is the default).

Instead of using the phase list, the user might choose to align the data on the P-times. Entering SH will accomplish this. If some of the arrivals are weak they could be enlarged on the display by using the scale option SC. Finally the scaled and shifted traces can be repicked (PI) and then the phase list saved (SA). If several digitizing passes are required for an event (the usual case), invoking the command PA will transfer control to another pass, and parts of the above sequence may be repeated on that data. Finally one can go to the next event (EV) and process it in a similar manner.

B. HOW TO USE ISDS

1. BASIC EXECUTION PROCEDURES

The following assumes EVCON/EVDIG have already acquired the seismic data.

1. Boot the appropriate system if it is not running on the Eclipse. ISDS wants USMALL, and MISDS (mapped ISDS) wants ALPT.
2. Go to the processor directory:
type: DIR DPOF:PROCESSOR
INIT DPO
3. Execute the program:
type: ISDS (or MISDS)
4. When the program comes up it automatically goes to the event processor (EV), and prompts the user for the event number and pass number. If the user wishes to plot all of the traces from all of the passes, when the "COMMAND?" prompt comes up he/she should type EVPL.
5. If the user wishes to know what commands are available he/she should type ME to display the menu.
6. Following is a summary of the ISDS commands used to invoke the command processor. Individual processors also provide appropriate prompts to the user.

2. SUMMARY OF ISDS COMMANDS

EV	SELECT EVENT: Choose event by number of those which have been digitized on the current disk.
PA	SELECT PASS: Select the particular pass within the current event.
EVPL	SELECT AND PLOT EVENT: The same as EV except that all of the traces from all of the passes are plotted on the screen.

- TR** **SELECT TRACES:** This command allows the user to select a group of traces, append to or delete from the existing group, or list the existing group of traces. The select process has its own command level. Its syntax is `command : argument list ; c :` `a ;` where the commands are SE (select traces), AP (append to traces), DE (delete traces), LI (list existing traces), NA (list existings traces by staton name), and EX (exit trace processor). The argument list is of the form: `I J-K ABCD`, where I refers to an individual sequence number, J-K represents a range of numbers, and ABCD refers to a station name. Example - if BGO = trace #14, from SE: 2-4 13 15 31-29 GBO; LI; EX would select traces 2, 3, 4, 13, 15, 31, 30, 29, 14 and list them, then return to the command level.
- DE** **DEFINE TIME WINDOW:** The user, following the program prompts, specifies the start time and time duration of the plot data window. The first prompt asks for window length, the next prompt asks for the window start-time. Three different types of responses are possible. They are:
- 1) Specify the hour, minute and second of the window beginning.
 - 2) Hit CR, which defaults to the data start time.
 - 3) Type X "n" where n is the number of seconds of offset from the start of digitizing.
- PL** **PLOT TRACES:** Those traces selected by the trace select processor (NU) are plotted in the time window defined by the window definition processor (DE). The border has tick marks every second. The start and stop times, the event and pass numbers; and the event name are plotted at the top. The traces are labeled by name and by sequence number. If the scale factor is not 1.0, the right hand side of the trace is labeled (eg. 10.0x). An amplitude units scale appears opposite each trace with the appropriate amplitude units (see SESC). The number refers to the number of units between the marks on the right hand side. These tic marks correspond to full scale if the scale factor is one.
- SESC** **SELECT SCALE UNITS OPTION:** This option allows the user to choose the units he wishes to use for amplitude measurements. Three options available are 1) digital counts (full range = 1024), 2) volts input into the A/D mux by the discriminators (full range = 5.0 volts), and 3) equivalent develocorder millimeters (full range = 160 mm), where full range of the discriminators corresponds to 160 mm on the develocorder playback machines. These units are displayed as the right hand side of the plot. Also these units are output on the phase cards with the exception that millivolts and not volts are put out for option 2.
- PLDE** **PLOT DECIMATED:** Same as PL except that a decimation factor is calculated and the traces are displayed according to this decimation factor. The factor also appears as a label on the plot. The factor is selected so that only 40 secs of real time are required to plot all of the traces requested.
- SC** **SCALE TRACES:** A scale factor may be applied either to all of the traces or to individual traces. This is global setting, i.e. it does not change from event to event. The command sequence is of the form `nn.nX IJK-L ABCD` where I and J are individual traces to be scaled, K-L is a range of traces, ABCE is a trace station name and nn.n is a scale factor to be applied to these traces.

PI PICK TRACES: A detailed menu of the pick functions may be obtained by typing (?) in response to the COMMAND? prompt. When in the Pick processor, the interactive cursor is turned on. The user may be positioned by two thumbwheels mounted at the upper right-hand corner of the Tektronix keyboard. The horizontal cross-hair is used for selecting the trace and for making amplitude picks. The vertical cross-hair is for timing. Pressing the appropriate key selects the desired function. Most functions require only 1 key input. Some exceptions are Q and A which request amplitude measurements, which must each be followed by 2 space-bar inputs to select the peaks of the peak-peak measurement; and R and *, which then expect either 3 or 1 character entries respectively to follow crosshair key input. In the case of Q and A, the position of the horizontal crosshair at the time Q or A are entered determines which trace is to be measured, and the two succeeding space-bar inputs to the cursor select the peaks being measured. The program accepts only spacebar as a valid entry in this case.

SA SAVE PICKS AS PHASE CARDS: Before proceeding to the next pass or event, this command must be invoked to save the pick information in the Phaselist file. Once per event, the user has the option of naming the Phaselist file. The default is the current file or in the case of the very first use of SA, the file 'FAZLIST'. The pick information only lives for the time you are in a pass, and if you leave the pass and return later, the information is lost if you have not saved it.

SH ALIGN TRACES ON P-TIMES: All of the traces in the current pass for which P-times have been picked are shifted so that the point in the waveform corresponding to that particular time is aligned on a vertical dotted line. This line is one-tenth of the way in from the left-hand side of the plot. The traces are re-ordered such that the earliest P-time is displaced at the top and the next earliest in second place, etc. The traces can be repicked in this display and another SH operation performed until the user is satisfied with his picks. Only those traces with P-picks are plotted so this provides another means of selecting traces for plotting.

SH-S ALIGN TRACES ON S-TIMES: This performs the same as SH with the substitution of S-times for P-times.

ZERO REMOVE DC LEVEL: The processor sets a flag so that at the beginning of every plot, a 100 sample average of each trace is computed. This average is used as a offset by the plotting algorithm. This option must be reinvoked when a new event is requested.

ME PRINT MENU: Displays menu of processor commands.

UNZE DO NOT REMOVE DC LEVEL: This undoes the zero command.

LDCO LOAD COMMANDS FROM REMOTE FILE: This allows the user to switch control of ISDS from the Tektronix keyboard to an ASCII RDOS file. The user must build this file in advance, using either the Editor or XFER/A \$TTI "myfile". "myfile" would contain commands which the user

would ordinarily type in from the keyboard. The following is an example in which the USER prepared two files. The first file, PLCOM1, requests an event and pass and a plot window and then selects eight traces. It plots them and then enters the PICK processor. At this point, the cursor is turned on. The user always has control of the cross-hair. After exiting from the picker, control returns to the file PLCOM1 which saves the picks onto FAZLST, makes a hard copy of the screen, and then calls the command LDCO and supplies the file name PLCOM1 for that processor. At this point control passes to PLCOM2 which prints the menu. Then it requests event two, pass one. It selects 15-31 traces in ascending order, deletes trace #24, scales by 2.5 times, then defines a twenty second window starting at 5 seconds after the beginning of the data. It plots it and then invokes UNCO (see below) to return control to the Tektronix keyboard.

NOTE: You must always provide a path either back to the keyboard (UNCO) or to another file (LDCO, filename). If you don't do this, the program will terminate.

UNCO	DISABLE REMOTE (LDCO) OPTION: Returns control to the Tektronix keyboard.
COPY	MAKE HARD COPY: Useful with LDCO option.
PAUS	PAUSE During execution from remote file. This command allows control to pass temporarily to the console in a way that permits execution to later resume from the remote file. The command is included in the remote file.
RESU	Resume control by remote file. Returns control to remote file. The next command following the PAUS command is executed. Carriage return is used as a resume command. It has this effect only after a PAUS from a remote file.
?	PRINT PICK MENU: Display menu of pick processor functions.
EX	EXIT PROGRAM: A CNTRL Z will also exit the program.

3. DESCRIPTION OF PICK COMMANDS

All pick commands consist of one character keyboard inputs to the cross-hair cursor. Where relevant, the horizontal cross hair selects the trace and the vertical cross hair picks the time. In some cases, character strings must be input following the command character.

<u>KEY</u>	<u>DESCRIPTION</u>
P	Vertical cross-hair picks the P-time for trace selected by horizontal cross-hair.
S	Picks S-time for selected trace.
F	Picks end of coda for selected trace.
C	Picks first zero crossing. The pulse width is determined by C time minus P time.
X	Nullifies previous P, S, and F picks on selected trace.
0 to 4	Enter grade of P-pick for selected station.

5 to 9 The grade of the S-pick for the selected station is obtained by subtracting 5 from the entered value. Thus 5 corresponds to a S weight of 0.

E or I Emergent of Impulsive P arrival selected trace.

U, D +, - First motion for selected trace.

J or K Emergent of Impulsive S arrival for selected trace. Note: This is an artifact to distinguish IS from IP and ES from EP.

; Speedy pick option. This allows the user to follow ';' with a 4 character string which determines P-remark or S-remark while picking either P or S for the selected trace. For example entering ;IPU2 CR would be equivalent to P CR , I CR , U CR , 2 CR and ;ESD1 CR would be equivalent to S CR , J CR , 6 CR . There is no provision for individually entering S first motions.

A Measure P amplitude. Following A CR , position the horizontal cross hair to the peak and then the trough (or vice versa), entering space-bar, carriage return at each position.

Q Measure S or maximum amplitude. Following Q CR , position cursor as above and enter measurements using space bar, carriage return.

R Enter three character remark.

* Enter one character remark. Note, as this goes in Column 71 of the phase list record, it is incompatible with HYPO71 input. Enter FMP value in seconds manually for selected trace. This command is useful when the F time is known from other records but the trace has not been digitized to capture this part of the trace.

L List and display picked values at top of screen.

G Input, list and display picks from PIKLST file, which has been generated by Rex Allen's automatic picker program.

Y Input phase list file, list and display picks. Program prompts user for file name.

H Input HYPOUT file, list and display picks and residuals. Program prompts for file name.

T Enter clock correction manually. Correction is in units of seconds and is global to the event.

4. EXAMPLES

Note: All examples use remote files. In general the commands are the same as those from the keyboard. The RDOS operating system treats terminals as input files.

Example 1.

Complete plots of the digital data for events 3 and 2 are shown in Figures 2 and 6.

(continued)

- A. - Receive instructions from PLCOM1 by following the LDCO command with the response PLCOM1.
- Define the data window as 20 seconds long, beginning 18 seconds into the data file.
 - Invoke the trace processor and select traces 26 thru 32, as well as trace 23. List the trace numbers and exit the trace processor routine.
 - Plot the selected traces in the defined data window (Figure 3).
 - Invoke the pick processor. Control returns to the operator via the thumbwheel cursor. The operator picks the first arrivals etc., and exits by entering '.' at the cursor.
 - control returns to the remote file which issues a SAVE command.
 - Since a new phase list file is not requested, the default file is FAZLST.
 - A copy of the screen is made (Figure 4).
 - Control is handed off to PLCOM2.
- B. (Control by PLCOM2)
- A menu is printed on the screen.
 - Event 2, pass 1 is requested.
 - Traces 15 to 31, except for 24 are selected and their numbers listed.
 - Traces 25 to 27 are scaled up 2.5 times.
 - A data window 20 seconds long is defined starting 5 seconds into the data.
 - A plot is made using the decimation algorithm (Figure 5).
 - Control is returned to the user.

Example 2 - The entire event is shown in Figure 6.

(Controlled by PLCOM6)

- Event 2, pass 1 is requested.
- Traces 21 to 32 and traces 9, 15 and 18 are selected and listed.
- The selected traces are plotted in the selected data window (Figure 7).
- The pick processor is turned on so the user can pick the arrivals.
- Upon exit from the picker, a new data window is defined which is 10 seconds long and starts at the beginning (CR default) of the data.
- The trace whose station name is GMK_ is scaled to 4.0.
- The picked traces are plotted with p arrivals aligned (Figure 8).
- A copy is made.
- The picks are saved on a phase list file named FARRAH.
- Control returns to the user.

REMOTE FILE EXAMPLES

PLCOM1

EV
1
1
DE
20
X 28
TR
6
ALL
N
PL
PI
SA
COPY
N
LDCO
PLCOM2

PLCOM6

EV
2
1
TR
SE:21-32 9 15 18;LI;EX
SC
2.5X 1-32
PL
DE
10

SC
4X GMK
SH
COPY
SA
Y
FARRAH
UNCO

PLCOM2

ME
EV
2
1
TR
SE:15-31;DE:24;LI;EX
SC
2.5X 25-27
DE
20
X 5
PLDE
UNCO

PART II: PROGRAMMER'S REFERENCEA. ISDS FROM THE INSIDE

1. OVERVIEW

As stated at the beginning of Part I, the purpose of ISDS is to display on the Tektronix screen seismic traces, which are stored on disk, in a useful way; and to make meaningful measurements from this display, using the interactive cursor.

In order to accomplish the display, a series of transformations on the data must occur, some of which are under control of the user. The principle transformations are:

1. Move a requested subset of the data from Disk into a data buffer (MCAM modules).
2. Map the data buffer into a multichannel data window (subroutine ARRAY).
3. Map the data window into screen co-ordinates (PLOT10 virtual plot routines).
4. Map the screen window into a display list of hardware plot commands (PLOT10 absolute plot routines).
5. Send the display list to the Tektronix CRT (HITEK DMA driver).

The control of the display is accomplished through the use of a simple command language, augmented by man-machine dialogue. The control segments consists of 1) a call to the subroutine COMMAND, which prompts the user with "Command?"; following which a command is input and decoded and a code is returned to the control routine; 2) a 'computed Go To' which directs control of the program, according to the code returned from COMMAND; and 3) subroutine calls to the appropriate processors which are issued upon receiving control from the computed 'GO TO'. Upon completion, the program returns to the COMMAND subroutine and waits for another command from the user (see Figure 9).

The processors roughly parallel the commands, which are described in the 'brief summary of commands' section of this report. A brief description of a few of the most important will be given here to show how the user can shape the data window. The command TR selects how many traces, which traces and in what order they will be displayed within the multichannel data window. The command DE actually defines the time bounds of the data window. It prompts the user for the start time and the window length in seconds. The SC command allows the user to set the scale factor either individually or for all traces.

These three commands (or processors) have the effect of defining the multichannel data window. Invoking the PLOT processor has the effect of mapping the seismic data into this user-defined window. The user may interact with the data using the interactive cursor (PI command) and then remap the data into another window using the P or S picks as shifts (SH or SH-S). He may then repick (PI) and finally save the data for input into a hypocenter program use the SA command.

2. CREATION OF A MULTI-CHANNEL DATA WINDOW (SUBROUTINE ARRAY)

The conceptual model behind the multi-channel data window is that of a collection of overlapping time series strips, each of which is tied to a particular 'proper' vertical distance from the top of the plot. The data window is sliced into 'N' horizontal strips, where N is the number of requested traces for a data window. The vertical midpoint of each strip represents the proper distance for that trace and also the place where a zero amplitude value would be plotted for that trace (see Figure 10).

Each point within a data window has three co-ordinates: time, base level, and amplitude; where time is a function of x, distance a function of y, and amplitude is a function of y and of base level. A trace is identified by determining the closest base level.

As the data plotted on the screen represents far more information than can be held in the data buffer at a given time, the data is plotted in vertical strips, each strip corresponding to the piece-wise de-multiplexed data segment (see Figure 10). The last data point of each trace is stored in the LFRAM vector in order to facilitate plot continuity between the strips.

3. MAJOR PROCESSOR ALGORITHMS

- . The plot algorithm is as follows:.
 - Compute segment length.
 - Compute starting sample index.
 - Initialize Disk retrieval (call FRAMD).
 - Initialize LFRAM vector.
 - Compute number of segments (NSEG).
 - Define data window.
 - Draw border (call FRAME).
 - A: Loop on segments.
 - Get data segment from disk (call GETFM).
 - B: Loop on traces.
 - C: Loop on samples.
 - Compute index.
 - Compute X, Y.
 - Draw vector to X, Y (call DRAWA).
 - Load LFRAM vector with last y value.
 - If end-of-data, Jump to D.
 - Go to C until all samples done (J = MSCAN).
 - Go to B until all traces done (I = NREQ).
 - Go to A until all segments done (L = NSEG).
 - D: Return.

The plotting equation follows:

$Y = S(II) * (A(IJ) - Z(II)) + D(I).$
 $X = DT(II) * t(IJ) + TZ(II) + SHFT(II) + TZARR.$
 where $t(LJ) = (L - 1) * MSCAN + J - 1.$
 $t(LJ)$ = is a temporary variable which corresponds to the relative time within a plot segment.
 L = segment index.
 I = trace index.
 J = sample index within segment.
 II = true trace index.
 IJ = global sample index.
 $MSCAN$ = trace segment length in samples.
 $TZARR$ = Beginning of display window in seconds.
 $SHFT$ = Shift of trace II (positive is right to left).
 DT = sampling interval of II th trace.
 TZ = - not used -.
 S = scale factor for II th trace.
 D = base level of I th trace from top of plot (absolute amplitude units).
 A = Amplitude of IJ th sample for i th trace (amplitude units from proper base level).
 Z = Amplitude of zero offset for II th trace (amplitude units from proper base level).

As an illustration of how the plot algorithm works, we will give several examples of how points are plotted on the screen. The examples are illustrated by Figure 10.

Example 1

Let us compute the x and y values of the first peak of trace number 14. Let us assume we have 5 traces and a buffer size of 100 words, so that each segment is 100 samples long. The digitizing interval is .01 sec., the scale is 1.0 and the zero level for trace 14 is +5. The data window starts at 41 seconds. Let the first pick be at the 80th point on the trace. The trace order in this example is 2, 3, 4, 14, and 17, so that our relative trace number is 4.

Now:

$X = DT(II) * t(LJ) + SHFT(II) + TZARR.$

$II = 14, TZARR = 41.0, DT(II) = .01, SHFT(II) = 0.$

$t(LJ) = (L - 1) * MSCAN + J - 1; L = 1, MSCAN = 200 + 0 * 200 + J - 1 = 79.$

$X = .01 * 78 + 0. + 41 = 41.79 \text{ sec.}$

$Y = S(II) * (A(IJ) - Z(II)) + D(I).$

$S(II) = 1., Z(II) = 5., I = 4, IJ = MSCAN(I - 1) + J = 680,$

$A(IJ) = 213,$

$D(I) = NREQ * ARNG - (I - .5) * ARNG.$

Where $ARNG = 1024, NREQ = 5, I = 4$

$D(I) = 5 * 1024 - 3.5 * 1024 = 1536.$

$Y = 1.0 * (213 - 5) + 1536 = 1744 \text{ vertical screen units.}$

Example 2

For this example let us choose the first peak of trace number 3 (relative trace 2):

Complete X from given parameters.

II = 3, TZARR = 41.0, DT = .01, SHFT(II) = 0.

MSCAN = 200, L = 3, J = 142.

$t(LJ) = (3.1) * 200 + 142 - 1 = 541$.

$X = .01 * 541 + 0. + 41 = 46.41$ sec.

Now complete Y

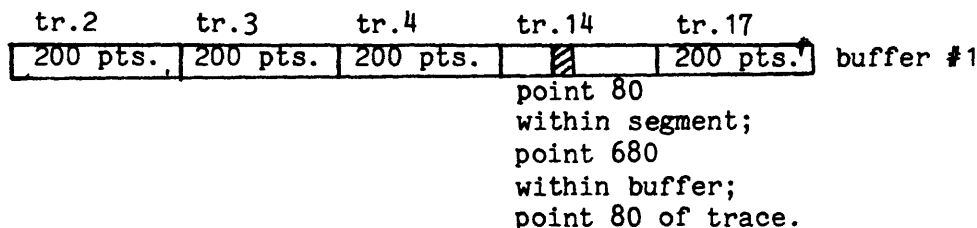
I = 2, IJ = MSCAN * (I - 1) + J = 342, A(IJ) = 123.

S(II) = 2.0, Z(II) = 0., D(I) = $5 * 1024 - 1.5 * 1024 = 3584$.

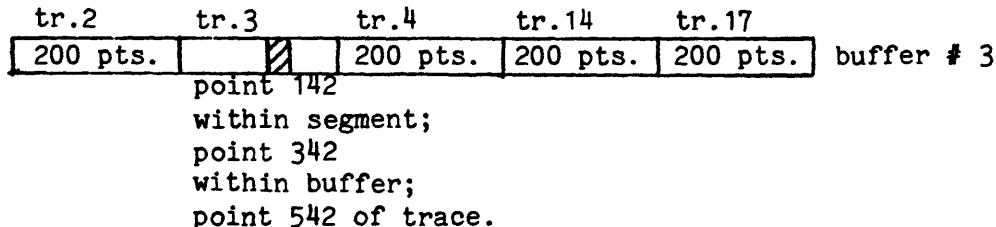
$Y = 2.0 * (123 - 0) + 3584 = 3830$.

Buffer layouts for the two examples, showing locations of IJth point:

Equation 1:



Equation 2:



Note: Computing the index within the buffer can be thought of as running at right angles to the trace direction, i.e.: it is a function of the segment length and the number of traces. The variable t however, which is used to compute the time of the completed point, is a function of the segment length and the number of segments.

3.a COMMAND PROCESSOR ALGORITHM

Prompt user with "Command?"

Input character string: If EOF (CNTRL Z) Go to 5.

Decode as 2A2.

- 1 Do Until J > Number of Dictionary Entries (NCD).
 If Character (1) = Dictionary Entry (J) go to 2.
 Go to 1.
 IC = NCD + 1.
 Go to 6.
- 2 IC = J.
- 3 Do Until J > NCD.
 If Character (2) = Dictionary Entry (J) go to 4.
 Go to 3.
- 4 Go to 6.
- 4 JC = J.
 Go to 6.
- 5 IC = 18 (18 = EX).
- 6 Return (with IC and JC for use by "Computed Go To").

3.b EVENT/PASS PROCESSOR ALGORITHM

If "PA" Go to 2.

Enter EVENT number.

Input Event Directory parameters (call RQEV).T).

If EVENT not on Disk, output message and return.

Reset window start time to beginning of data.

Save date, time, sample rate, and digitizing duration in common blocks.

Compute width of plot windows based on number of digitized passes.

- 1 Do until N number of passes.
 Computer screen co-ordinates of Nth plot window
 Define trace list, from 1 to number of digitized traces.
 Set old trace list = current trace list.
 Set window width = digitizing duration.
 Compute decimation factor (call DECIM).
 Input station names for Nth Pass (call RQPAZ).
 Reset shifts to zero (call NOSHF).
 If not "EVPL" go to 2.
 Reset trace distance (call RSETD).
 If zero flag on, compute zero level (call ZLEV).
 Plot data (call ARRAY).
 Label plot (call LBSTA, ENAME, TEK routines).
 Go to 1.
- 2 Prompt for PASS number.
 Input PASS number.
 Input station names for PASS (call RQPAZ).
 Define trace list for PASS.
 Set backup trace list = current trace list.

Set shifts to zero (call NOSHF).
 Reset time distances (call RSETD).
 Define data window.
 Define physical screen window.

3.c WINDOW DEFINITION PROCESSOR

Enter window width in seconds.
 Echo current window start time.
 Enter start time (hour, minute, second).
 If carriage return, keep current time;
 If "R", use current window + R seconds ($SEC = SEC + R$).
 Compute time window, relative to data.
 If out of bounds, type error message.
 Set old start time = new start time.
 Return.

3.d TRACE DEFINITION PROCESSOR

```

9  Prompt.
   Readline.
14  Extract Command.
   If (SE, AP, DE) then parse arguments.
   If (EX) Return.
   If (SE) Go to 10.
   If (AP) Go to 20.
   If (LI) Go to 30.
   If (ST) Go to 35.
   IF (DE) go to 40.
   Go to 9.
10  Load Trace list.
   Go to 19.
20  APPEND TO TRACE LIST.
   Go to 19.
30  WRITE TRACE LIST.
   Go to 19.
35  WRITE TRACE LIST.
   Go to 19.
40  SQUEEZE LIST IF (;) Go to 14.
   Go to 19.
19  -      -
   Backup trace list (fill ITEMP and NTEMP.
   If (end of command) Go to 14.
   RETURN.
```

3.e PLOT PROCESSOR ALGORITHM

Restore old trace order.
 Rest screen window to default.
 Restore old window start time (call RSTAT).
 Reset shifts to zero (call NOSHF).
 Reset trace distance (call RSETD).
 Erase screen.
 Decimate if requested (call DECIM).
 Compute zero level; if flag set (call ZLEV).
 Plot traces (call ARRAY).
 Label Plot (call LBSTA, LBSCL, ENAME, TEK package).
 Reset decimation factor to 1.
 Return.

3.f PICK PROCESSOR ALGORITHM

1 Enable Cursor

Position Cursor (Manual - thumbwheels).
 Enter character (Manual - strike key).
 Identify trace by horizontal cross-hair (Y-co-ordinate).
 Process input character and perform appropriate action.
 R = enter 3 character remark.
 * = Enter 1 character remark (Col. 71 of phase list).
 E = emergent P-arrival.
 I = impulsive P-arrival.
 J = emergent S-arrival.
 K = impulsive S-arrival.
 C = 1st zero crossing time (y - co-ordinates).
 P = P-arrival time (x coordinates).
 S = S-arrival times (y coordinates).
 F = time of coda die out.
 0 - 4 = Grade of P-pick.
 5 - 9 = Grade +5 of S-pick.
 . = exit pick processor (return).
 A = P amplitude:
 Enable cursor.
 Position cursor, hit space bar.
 Enable cursor.
 Position cursor, hit space bar.
 AMP = peak to peak distance between space-bar hits.
 Q = S amplitude same as A except SAMP = peak to peak distance between space-bar hits.
 ; - Speedy Pick mode: enter P or S remark as a 4 character string with P or S time determined by vertical cross-hair (x co-ordinates).
 = enter FMP manually:
 Enter character string representing F - P. number. Call to INFREE decodes it. Ftime = Ptime + FMP.

U, D, +, - = First motion for P.
 X = rezero picks: Set P, S, and F times to zero.
 L = List picks and draw P, S, F, R and r marks (call LPICK).
 G = get auto picks and list (call GPICK(1), LPICK).
 Y = get phase list and list (call GPICK(2), LPICK).
 H = get "HYPOUT" file and list (call GPICK(3), LPICK).
 T = enter clock correction (in seconds manually).

Go to 1.

3.g SHIFT PROCESSOR ALGORITHM

Compute fiducial mark to be 1/10 window length.
 Round up to next second.
 If < 2 seconds just take 1/10.
 If P-shift:
 Then load time vector with P-times.
 Else load time vector with S-times.
 Sort time vector, return key (call SORT).
 Eliminate null times.
 Save old trace indices.
 Re-order traces using sort key as index.
 Save 'old' window start-time.
 Set window start-time = time (earliest) - Fiducial mark.
 Reset trace distance based on new order (call RSETD).
 Erase screen.
 If zero flag on, compute zero level for traces (call ZLEV).
 Plot data (call ARRAY).
 Label plot (call LBSTA, LBSCL).
 Draw fiducial dashed line.
 Put caption of top of plot.
 Return.

3.h SCALE PROCESSOR ALGORITHM

Prompt.
 Readline.
 Scan for 'x'.
 Scan left and decode scale factor (call INFREE).
 Scan right for argument list (call TPARS).
 Assign scale factor to traces in argument list.
 Return.

4. BRIEF DESCRIPTION OF SUBROUTINES

ARRAY Sets up data window and calls in data from disk. Computes amplitude and time co-ordinates of each selected trace within data window. Calls Tektronix plot routines.

BOX Draws a border around screen window.

COMMAND Prompts user with "Command?". Looks up command in dictionary and returns code to control program.

DECIM Computes decimation function based on total points requested (within a data window) for display.

DEFAULT Sets or resets default values for various variables.

DEPRC Defines data window for a given plot.

EVPRC Selects events for plotting. Plots all passes of event if EVPL option is selected. Selects pass for subsequent plotting or interaction.

FAZPU Writes plase card file based on picks. Entered with SAVE command.

FRAME Calls BOX to draw border, then draws tic marks on border.

GPICK Called from PICK. It opens external files 'PIKLIST', 'FZLINK', or 'HYPOUT', and loads the information into variables in named common block /DPARM/.

INFREE Decodes ASCII string and treats as free form real number input. Returns real variable array.

ISDS7 Main program. Open files and calls various initialization routines including DEFAULT. Calls main control routine SPLOT.

LBSCCL Labels scale factor on each trace as well as amplitude range of each trace in selected units.

LBSTA Labels each trace with station name and index number. Labels each data window with window start and stop times.

LPICK Called from PICK. Calls PMARK to draw P, S, and F time pick marks on traces. Draws residual time makrs if information is present. Displays list of picks and associated selected parameters.

MENU Writes either command menu or Pick menu to CRT Screen, based on control argument.

NOSHF Eliminates trace time shifts by setting variable SHFT(k) equal to zero.

PICK Allows user to 'pick' phase times, peak-to-peak amplitudes, codas, first-motion and pick qualities using the interactive cursor. Calls LPICK to display picks, and calls both GPICK and LPICK to display external picks.

PLPRC Calls a sequence of routines in order to plot a given data window on the screen.

PMARK Draws a vertical line at a given data co-ordinate with a single letter label.

PSHIFT Plots data window with either P or S phase aligned. Only those traces within a pass who have the appropriate picks will be displayed. Traces are displayed in increasing time order.

RSETD Resets base level (vertical spacing) of traces to uniform spacing and in index order.

RSTAT Restores saved trace order into IREQ(J) from ITEMP(J).

SCALER Allows user to specify trace scale factors, either for all traces or individual traces.

SORT Sorts an array of real variables in ascending order and constructs array of indices. It is called by PSHIFT.

SPLOT This is the main control subroutine. It calls COMMAND and routes control to the appropriate processor subroutine calls by means of a 'computed 'go to'.

SWIND Resets screen window bounds.

TPARS Parses and decodes trace select string of form (IJ M-N ABCD) into list of trace numbers.

TRPRC Selects number of traces to be displayed for a given pass, as well as the order of display. Traces may be appended to previous list.

ZLEV If IZER flag in labeled common block /ZER/ is set, this routine is called prior to plotting. If the flag is set, the NZ points per trace are input then used to compute an arithmetic mean for use as a zero (or DC) level for each trace. Variable Z(J) is set to this level for each trace.

5. ACCESS TO DIGITAL DATA

ISDS access the seismic data thru the MCAM library. There are three subroutines in this library which are called by ISDS. They are MCAM1, which initializes common blocks; FRAMD, which defines the number of traces and their length within the data buffer; and GETFM, which inputs a buffer load for each call.

Typically, following each plot command, a call to FRAMD is made to set up the trace order. Then GETFM is called successively until the plot is complete. Details of the calling sequences appear as documentation in the listings of these three routines.

6. INTERPROCESS COMMUNICATION

Interprocess communication is through various files which are accessed by both ISDS and the other processes. These files fall volunarily into two categories, those concerned with the digital data, and those concerned with the P and S picks and the location program.

The files which fall in the first category are the digital data, file EVDAT, which was described above, the data description file EVDIR, the subroutine which access EVDAT and MASTERLIST are organized in a library known as the DIRC library. There are four routines in this library. The first, which is called only once, is the initializing routine DIRC1. Each time a new event is requested, the subroutine RQEV is called, and each time access to a new digitizing pass is requested, RQPAZ is called. A call to ENAME returns the ASCII name of the currently requested event. Details of the calling sequence of these routines appear in the listings in Appendices O and P.

The second category of interprocess files are those which pertain to communication between ISDS and either an earthquake location program such as HYPOINVERSE or an automatic earthquake program.

In both these cases, input to ISDS is through the subroutine GPICK, which is invoked by the Y, G, or H cursor commands. Output from ISDS is through the routine FAZPU, which is invoked by a SAVE command. The phaselist file, which is generated by this command, can be read back latter by using the Y cursor command. The format of these files is described in Appendices I, J and K.

7. SUPPORT SOFTWARE

(1) MCAM modules by P.A. Marshall:

MCAM is an acronym for Multi-Channel Access Method. Three subroutines from this group are directly called by ISDS. They in turn call five other routines in order to bring in data from the disk file 'EVDAT'. The data is put into a user specified buffer in piece-wise multiplexed form (see FRAMD writeup). The three called routines are:

MCAM1	Initialization
FRAMD	Request the number of traces and the buffer size.
GETFM	Read a buffer load of data.

The other routines are:

SEGMO	Reads in next data segment from disk.
SEGIN	Keeps track of previous segment.
DISKI	Input task for disk segments.
DEMUX	Demultiplexes data.

(2) DIRC modules by P. A. Marshall and R. Haken: These routines bring data descriptor information into ISDS from the EVDIR and MASTERLIST files.

There are five routines:

DIRC1	Initialization.
RQEVF	Request event parameters.
RQPAZ	Request pass parameters and station list.
ENAME	Read event name from EVDIR.
FBLM	Fortran callable interface to block move (BAM) instruction (Data General Eclipse Assembly Language).

(3) TEK modules by Tektronix, Inc. (PLOT10), P. A. Marshall (HITEK, PACK1), T. D. Thomas (ADEIN, ADEOUT, ICODE, KSES).

The PLOT10 package is version 3.3 and is fully described in the PLOT10 users manual (1977). The other routines are I/O interfaces and improvements. HITEK is a DMA driver for the parallel interface which was designed by Ellis and Marshall.

(4) RDOS (Real Time Disk Operation System) by Data General Corporation, including Fortran V (A superset of ANSI Standard Fortran 68) and various Fortran Libraries.

(5) EVCON and EVDIG by T. D. Thomas and P. A. Marshall. These programs, which are briefly described in the section on 'Formats and File Structures of the Data', are responsible for providing digital seismic traces and associated data for ISDS to display and manipulate. Unpublished listings of these programs are available for interested persons.

REFERENCES CITED

- Allen, Rex V., (1978). Automatic earthquake recognition and timing from single traces: U.S. Geological Survey Open-File Report.
- Klein, Fred W., (1978). Hypocenter location program HYPOINVERSE: U.S. Geological Survey Open-File Report.
- Lee, W. H. K., and Lahr, J. C. , (1972). HYP071: A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes; U.S. Geological Survey Open-File Report.
- Data General Corporation, (1974). FORTRAN V Users Manual (093-000085-03).
" " " (1976). FORTRAN V Supplement (093-0018S-00).
- Tektronix, Inc., (1977). Plot 10 Terminal Control System, User Manual.

B. APPENDICES

1. Tables

A: PARAMETERS USED IN ISDS*

Name	Include File	Routines	Description
MAXX	C\$BLAN	See cross-ref for C\$BLAN	Dimension of Data buffer A(MAXX).
LSZ1	C\$GPARM	See cross-ref for C\$GPARM	Location within LIST of number of digital passes.
MAXPASS	"	"	Maximum number of passes per event which this version allows.
MAXVP	"	"	Maximum number of view ports (passes) per plot page for event plot.
LSZ	"	"	Dimension of LIST(LSZ); LSZ = LSZ1 + MAXPASS

*Parameters used by the DIRC modules and the MCAM modules are not included in this list.

B: INCLUDE FILES USED BY ISDS*

Include File	Common Blocks	Parameters	Other
C\$BLAN	blank	MAXX	INTEGER A
C\$APARM	APARM		
C\$BPARM	BPARM		
C\$PICK	DPARM		
C\$PICT	none		Command Dictionary
C\$GPARM	KDATE, JDATE, XXX, VEE.	LSZ1, MAXPASS, MAXVP, LSZ	INTEGER EVENT PASS, UNQYR, UNQID, FCA, LCA, VSAVE

* Include files used by the DIRC1 and MCAM modules do not appear in this list.

C: DATA FILES USED BY ISDS

L.U.	File Name	Where opened	Use
1	DIRECTORY	DIRC1	Description of digital data
2	MASTERLIST	DIRC1	Station names
15	PIKLST	GPICK	List of auto picks
	FAZLST	FAZPU	Phase list file
	HYPOUT	GPICK	Station card file
4	\$TTO1	ISDS	Tek tube serial output
5	\$TTI1	ISDS	Tek tube serial input (includes cursor)
3	JTEK	ISDS	File of tek vectors (not used)
9	TEMP	ISDS	Scratch file
13	(Remote File)	SPLOT	Pre-programmed remote user file
0	WHOLE1	MCAM1	Digital data file

D: LINKS USED BY ISDS

Link Name	Resolution File
DIRECTORY	DPO: EVDIR
MASTERLIST	DPO: MASTERLIST
WHOLE1	DPO: EVDAT

E: DESCRIPTION OF COMMON BLOCK VARIABLES

<u>Common Label</u>	<u>Variable</u>	<u>Type</u>	<u>Description</u>
Blank	A(J)	Integer array	Seismic data input buffer, Data stored here by GETFM, used by ARRAY.
	MAXPTS	Integer	Maximum number of data points for A.
	NREQ	"	Number of traces requested for plotting.
	IREQ(I)		
	I = 1, NREQ	Integer array	Index of 'Ith' trace relative to original digitized traces (organized on a 'pass' basis).
APARM	S(J), J = 1, NREQ	Real array	Scale factor of 'Jth' trace.
	Z(J)	" "	Zero (D.C.) level of 'Jth' trace relative to 'distance'.
	DT(J)	" "	Sampling interval of 'Jth' trace.
	TZ(J)	" "	- not used -
	SHFT(J)	" "	Shift (in samples) of 'Jth' trace. Positive value means later time (shift to left).
	D(J)	Real array	Distance (spacing) of 'Jth' trace from top of plot (in array amplitude units).
BPARM	TZARR	Real	Offset of current plot window (in seconds) from start of data.
	TWIND	"	Length of current plot window (in seconds).
	THR	"	Temporary hour of data.
	TMN	"	Temporary minute of data.
	AZ	"	Bottom of screen in array amplitude units.
	ARNG	"	Width of each trace in digital amplitude units.
	NT	Integer	- not used -
	DTG	Real	Global digital sample interval DT(J) = DTG for all J.
	SG	Real	- not used -

DPARM

All references are to JTH trace unless otherwise noted. All time variables are in seconds. All amplitude variables are in those units selected by "SESC" option.

PTIME(J), J = 1, NREQ	Real Array	Time of P-phase.
AMP(J)	"	Peak-to-peak P-amplitude.
STIME(J)	"	Time of S-phase.
FTIME(J)	"	Time of coda cutoff determination.
CROSS(J)	"	Time of first zero crossing.
SMP(J)	"	Time between P and S phases (S-P).
FMP(J)	"	Coda length (F-P).
PCROSS(J)	"	Time between ptime and first zero crossing (Cross-PTIME).
IFMOT(J)	Integer array	First motion of P-phase: U, D, + or -.
IGRADE(J)	" "	Grade of P-arrival (0-4).
SGRADE(J)	" "	Grade of S-arrival (0-4).
SAMP(J)	" "	Peak-to-peak S or maximum amplitude.
RMK (I,J), I = 1, 3; J = 1, NREQ	Integer array	First two words are for three character remark; third word is for 'station remark'.
SRMK(J)	" "	Impulsive 'IS' or emergent 'ES', else 'S' or ' '.
PRMK(J)	" "	Impulse 'IP' or emergent 'IP', else 'P' or ' '.
PRES(J)	" "	P-residual from HYPOUT.
SRES(J)	" "	S-residual from HYPOUT.
EPD(J)	" "	Epicentral distance of trace, as read from HYPOUT.
CLK(J)	" "	Clock correction, add to seconds.

KDATE	KYR	Integer	Year of event, relative to digitizing start time.
	KMO	"	Month of event, relative to digitizing start time.
	KDA	"	Day of event, relative to digitizing start time.
	KHR	"	Hour of event, relative to digitizing start time.
	KMN	"	Minute of event, relative to digitizing start time.

RSEC	Real	Second of event, relative to digitizing start time.
DUR	"	Duration time of digitizing (A/D conversion) in seconds of data time.
EVENT	Integer	Number of the event relative to Data File.
PASS	Integer	Number of the Pass, relative to the event.KDATE
UNQYR	"	The 'year' part of the unique ID number.
UNQID	"	The 'serial number' part of the unique ID number.
TCORR	Real	Clock correction (seconds).
FCA	Integer	First digitized channel 'Mux' address.
LCA	Integer	Last digitized channels 'Mux' address.

JDATE	IHR	Integer	Hour of plot window start time.
	IMN	Integer	Minute of plot window start time.
	TSEC	Real	Second of Plot window start time.

XXX	XX(3) LIST(L), L = 1, LSZ	Real Array Integer Array	- not used - input parameters from file EVDIR by way of subroutine RQEVT. See RQEVT writeup (Appendix C).
-----	---------------------------------	-----------------------------	---

VEE	VSAVE(M, N), M = 1, 4; N = 1, MAXPASS	Integer Array	Screen co-ordinates of windows (view ports) for EVLP option. VSAVE (1, N) = x origin of 'Nth' window VSAVE (2, N) = x length. VSAVE (3, N) = Y origin. VSAVE (4, N) = Y length.
-----	---	---------------	--

TVTUNE	ITV	Integer	Logical unit of CRT serial output.
	ITX	Integer	Logical unit of CRT serial input.
RPLAY	VFILE	Integer	Switch to institute storage of tektube plot vectors in file JTEK, not currently used.
SFL	SFLAG	Integer	Selects scaling option, 1 = digital counts, 2 = volts out of discriminators, 3 = equivalent develocorder mm.
ZER	IZER	Integer	IZER = 1 if ZE option is set, IZER = 0 if UNZE option is set.
ZZZ	IZIT	Integer	- not used -
DECI	IDF	Integer	Decimation factor
	DEMAX	Real	Maximum number of decimated samples. Set in DEFAULT.
	DLIM	Real	- not used -
CPARM	NSTA ISTA(I, J), I = 1, 2, J = 1, 32	Integer Integer array	- not used - ASCII name of 'Jth' station.
SCAL	KX	Integer	x-origin of plot window, in screen coordinates.
	LX	Integer	length of plot window, in screen coordinates.
	KY	Integer	y-origin of plot window, in screen coordinates.
	LY	Integer	height of plot window, in screen coordinates.

ITRM	Integer	Terminal type 1 = HOLD, style (1024 x 780) 4 = 4014 style (4096 x 3120)
ISCAL	Integer	Width of screen in screen coordinates.
JCHAR	Integer	character size: 1 = largest; 2, 3, 4 = smallest.

EPARM	NTEMP	Integer	Temporary storage for NREQ (blank common).
	ITEMP(J), J=1, NTEMP	Integer Array	Temporary storage for IREQ(J) (blank common).
	OTZARR	Real	temporary storage of TZARR(/BPARAM/).
	OTWIND	Real	Temporary storage for TWIND(/BPARAM/).

OLD	OLDEVT	Integer	Previous value of EVENT (/KDATE/).
	PUFLG	Integer	PUFLAG = 1, phase cards have been written.
	PUBLNK	Integer	- not used -
	NFILE	Integer Array	Name of current phaselist file. Initialized in ISDS7 to FAZLST.

* Includes common blocks KDATE, JDATE, XXX, and VEE

(Con't)

Subroutine Include or Common														
														C\$BLAN(blank)
														C\$APARM(APARM)
														C\$BPARM(BPARM)
														C\$PICK(DPARM)
														C\$DICT
														C\$GPARM*
														TVTUNE
														RPLAY
														SFL
														ZER
														ZZZ
														DECI
														CPARM
														SCAL
														EPARM
														SHCOM
														IOPRM
														OLD
														MISC

* Includes common blocks KDATE, JDATE, XXX, and VEE

G: ISDS SUBROUTINE CALLS CROSS REFERENCE

<u>Routine</u>	<u>Calls</u>	<u>Called By</u>
ARRAY	FRAME. FRAMD(M), GETFM(M), AMIN1(R), (TEK)	EVPRC, PLPRC, PSHIFT
BOX	(TEK)	FRAME
COMMAND	-	SPLIT
DECIM	-	EVPRC, PLPRC
DEFAULT	-	EVPRC, ISDS7, PICK, SPLIT
DEPRC	INFREE, SWND, (TEK)	SPLIT
EVPRC	ARRAY, DECIM, DEFAULT, ENAME(D), INFREE, LBSTA, NOSHF, RQEV(D), RQPAZ(D), RSETD, SWND, ZLEV, (TEK)	SPLIT
FAZPU	AMOD(R)	SPLIT
FRAME	BOX, (TEK)	ARRAY
GPICK	-	PICK
INFREE	-	DEPRC, EVPRC, PICK, SCALER, SPLIT
ISDS7	DEFAULT, DIRC1(D), SPLIT, OVOPN(R), MCAM1(M), (TEK)	-
LBSC	(TEK)	PLPRC, PSHIFT
LBSTA	(TEK)	EVPRC, PLPRC, PSHIFT
LPICK	PMARK, (TEK)	PICK
MENU	-	SPLIT
NOSHF	-	EVPRC, PLPRC
PICK	DEFAULT, GPICK, INFREE, LPICK, (TEK)	SPLIT
PLPRC	ARRAY, DECIM, ENAME(D), LBSC, LBSTA, NOSHF, RSETD, ZLEV, (TEK)	SPLIT

PMARK	(TEK)	LPICK
PSHIFT	ARRAY, LBSCL, LBSTA, RSETD, SORT, ZLEV	SPLOT
RSETD	-	EVPRC, PLPRC, PSHIFT
RSTAT	SWND	SPLOT
SCALER	INFREE, TPARS	SPLOT
SORT	-	PSHIFT
SPLOT	COMMAND, DEFAULT, DEPRC, EVPRC, FAZPU, INFREE, MENU, PICK, PLPRC, PSHIFT, RSTAT, SCALER, FDELY(R)	
SWND	-	DEPRC, EVPRC, RSTAT
TPARS	-	SCALER, TRPRC
TRPRC	TPARS	SPLOT
ZLEV	FRAMD(M), GETFM(M)	EVPRC, PLPRC, PSHIFT

Code: (M) = MCAM Library, (D) = DIRC Library, (R) = Run time routine, (TEK) = Tektronix package.

H: SUPPORT SUBROUTINE CALLS CROSS REFERENCE

Routine	Calls	Called by
DIRC1	-	ISDS7
RQEVF	-	EVPRC
RQPAZ	FBLM	EVPRC
ENAME	-	EVPRC, PLPRC
FBLM	-	RQPAZ, SEGMO
MCAME	-	ISDS7
FRAMD	MINO(R), BEGIN, SEGMO,	ARRAY, ZLEV
GETFM	DEMUX, MINO(R), SEGMO	ARRAY, ZLEV
BEGIN	DISKI	FRAMD, SEGMO
SEGMO	CHECK(R), FBLM, BEGIN	FRAMD, GETFM
DISKI	RDBLK(R)	BEGIN
DEMUX	-	GETFM

Code: (R) = Runtime routine.

I: PHASE DATA OUTPUT FORMAT

Col.	Format	Explanation
1-4	2A2	Station name. Must be non-blank, and col. 1 may not be a numeric character.
5-6	A2	Remark such as 'IP'. Must be non-blank for P time to be accepted.
7	A1	P first motion code such as 'C' or 'D'.
8	I1	Assigned weight code for P arrival as follows: 0 or blank--full weight. 1 - 3/4 weight. 2 - 1/2 weight. 3 - 1/4 weight. 4 to 9 - no weight.
9	1X	Blank
10-11	I2	Year. Must be the same as first phase card of event.
12-13	I2	Month. Must be the same as first phase card of event.
14-15	I2	Day. Must be the same as first phase card of event.
16-17	I2	Hour. Used for both P and S arrival.
18-19	I2	Minute. Used for both P and S arrival.
20-24	F5.2	Second of P arrival.
25-29	I5	Peak to Peak P amplitude*
30-31	2X	Blank.
32-36	F5.2	Second of S arrival.
37-38	A2	S remark such as 'ES'. Must be non-blank for S time to be accepted.
39	I2	Assigned weight zone for S arrival (same codes as P).
41-44	4X	Blank.
45-47	I3	Peak to peak amplitude of maximum or S amplitude *.
48-62	15X	Blank.
63-65	A1, A2, A3	Optional remark. An event remark may be entered here. First non-blank field on the series of phase cards will be used as and event remark by HYPOINVERSE if columns 1-4 and 70-72 of the terminator cards are all blank.
66-70	F5.2	Clock correction, to be added to both P and S arrival times.
71	A1	Remark for this station such as 'D' (dead), 'L' (low frequency, 'N' (noisy) etc. Usual except on output (not HYP071 compatible).
72-75	I4	Coda duration (F-P time) in seconds.
76-80	5X	Blank.

*Units determined by SESC option as follows:

- 1) digital counts out of A/D (fullscale = 1024).
- 2) millivolts into /D (fullscale = 5,000 mv).
- 3) equivalent developeorder millimeters (fullscale = 160 mm).

J: STATION DATA FORMAT (HYPOUT)

The station data file contains all of the information in the printed station list, but is in a format compatible with phase data. Station data output for each event consists of one line per station. The event is terminated by a HYPOINVERSE format summary line which acts like a terminator line containing the previous location as a trial hypocenter on input. The format of the station data is as follows:

Columns	Format For Reading	Explanation
1-4	2A2	Station name.
5-6	A2	P remark such as 'IP'.
7	A1	P first motion.
8	I1	Assigned P weight code.
9	1X	Blank.
10-11	I2	Year.
12-13	I2	Month.
14-15	I2	Day.
16-17	I2	Hour.
18-19	I2	Minute.
20-24	F5.2	Second of P arrival.
25-28	F4.2	P travel time residual*.
29-31	F3.2	P weight actually used*.
32-36	F5.2	Second of S arrival.
37-38	A2	S remark such as 'ES'.
39	1X	Blank.
40	I1	Assigned S weight code.
41-44	F4.2	S travel time residual*.
45-47	F3.0	Maximum peak-to-peak amplitude in mm.
48-50	F3.2	S weight actually used*.
51-54	F4.2	P delay*.
55-58	F4.2	S delay*.
59-62	F4.1	Epicentral distance*.
63-65	F3.0	Emergence angle at source, in degrees from nadir. (Replaces optional remark on original phase card.)*
66-70	F5.2	Clock correction.
71	A1	Station remark, such as 'D' for dead.
72-75	F4.0	Duration time in seconds.
76-78	F3.0	Azimuth to station in degrees east of north*.
79-80	F2.1	Duration magnitude for this station*.
81-82	F2.1	Amplitude magnitude for this station*.
83-86	F4.3	Importance of P arrival*.
87-90	F4.3	Importance of S arrival*.

*Additional items not on original phase cards.

K: AUTOMATIC PICKER FORMAT (PIKLST)

The file contains data from multiple events. Each event begins with a header containing the ASCII string 'event' and a two digit event number. The header is followed by a group of records which describe the phases which have been picked for the event. An event may contain several instances of picks for a particular station. Only the first instance is retrieved by ISDS. An event sequence is terminated by a phase record with '\$\$\$\$' in the first four bytes. The format of the phase records is shown below:

<u>Columns</u>	<u>Format</u>	<u>Explanation</u>
1-4	2A2	Station name
5-11	F7.3	P-Pick time (seconds)
12-13	A2	First motion (+, - or blank)
14-20	F7.3	Duration time (seconds)

L: FORMATS OF EVDIR FILES

1. Three-Pass EVDIR

All Records are 160 16-bit words in length. The first record in the directory file (DPO:EVDIR) contains general information. The layout of this record is as follows:

Array Index	Name	Description
1	NEV	Number of events already on disk + 1 (1 means NO EVENTS).
2	NMLST	Number of entries in MASTERLIST file.
3	IOFF	Initial offset for contiguous file "DPO:EVDAT".
4	FILSIZ	Total usable space in "DPO:EVDAT", in disk blocks.
5	NXTBLK	Number of the next Disk block in "DPO:EVDAT" to start store data into (initially equals "IOFF").
6	TIM2AD	The number of the A/D channel receiving a time code. 0 if no time code.
7	IDELT	Time parameter for P.I.T. There are IDELT*6.25 microseconds between sample points.
8	IHEX	Footage count for tape positioning for the current event.
9	NODUB	True if the tape being digitized is not a dubbed tape.
10	SCANS	The (real) number of samples to be digitized on the current event.
11	.	.
12	PNTR	An array used to map tape track-channel to a record in the MASTERLIST file, which is supposed to be a station card image.

RECORD 2-N: EVENT AND PASS INFORMATION
(RECORD J CORRESPONDS TO EVENT J + 1)

Word	Data
1	Event number (= rec # - 1).
2-21	Event name (40 characters).
22	Starting relative block
number.	
23	Size (in blocks).
24-25	DELTA-T (real number).
"	" " "
26-27	Digitizing duration (real number).
"	" "
" "	
28	Year.
29	Month.
30	Day.
31	Hour.
32	Minute.
33-34	Second (real number).
"	" " "
35	Pointer to pass A = 38.
36	Pointer to pass B = 79.
37	Pointer to pass C = 120.

PASS A (CONTINUATION OF RECORD 2):

38	Event number.
39	Pass number (= 1).
40	Starting relative block for
this pass.	
41	Size of pass in blocks.
42	FCA (first Mux channel
address).	
43	LCA (last Mux channel
address).	
44-75	Pointer list to 32 possible station name
	locations in MASTERLIST.
76	Pass switch setting (actual
value in DIGOUT word).	
77	Filter switch setting
(actual value in DIGOUT word).	
78	A/D error word: (bit 0-7 =
IAD); (bit 8-15 = IDSK).	

PASS B (SIMILAR TO PASS A):

79	Event number.
80	Pass number (= 2).
	etc.
	"
	"
	"

2. TWELVE-PASS EVDIR

All records are of length = $(52 + 48 + \text{number of passes})$
 = 628 for 12-pass case.

The first record in the directory file (DPO:EVDIR) contains general information. The layout of this record is as follows:

Index	Name	Description
1	NEV	Number of events already on disk.
2	NMLST	Number of entries in MASTERLIST file.
3	IOFF	Initial offset for contiguous file "DPO:EVDAT".
4	FILSIZ	Total usable space in "DPO:EVDAT", in disk blocks.
5	NXTBLK	Number of the next disk block in "DPO:EVDAT" to start store data into (initially equals "IOFF").
6	TIM2AD	The number of the A/D channel receiving a time code. 0 if no time code.
7	IDELT	Time parameter for P.I.T. There are IDELT*6.25 microseconds between sample points.
8	AUTODIG	True if batch digitizing is to be invoked.
9	NODUB	True if the tape being digitized is not a dubbed tape.
10	SCANS	The (real) number of samples to be digitized on the current event
11	DISKID	ASCII name of physical disk ID (3 words).
12	PNTR	An array used to map table track-channel to a record in the MASTERLIST file, which is supposed to be a station card image. This array is of length MPLSIZE where MPL Size = $32 * \text{NUMPASSES}$. MPLSIZE, 384 for the 12-pass case.

*RECORD 2-N: EVENT INFORMATION (RECORD J CORRESPONDS TO EVENT J + 1)

OFFSET	DATA
0	Event number (record # - 1).
1	Event name (40 characters).
.	.
20	.
21	Starting relative block number.
22	Size (blocks).
23	Delta-T (real number).
24	" " ".
25	Duration (real number).
26	" " ".
27	Year.
28	Month.
29	Day.
30	Hour.
31	Minute.
32	Second (real number).
33	" " ".
34	Time code correction in seconds (real number).
35	" " " " " " " ".
36	Unique ID:YEAR.
37	Unique ID:Serial number.
38	MASTERLIST origin filename (10 characters).
42	.
43	Pointerlist origin filename (10 characters).
47	.
48	Spare words (5).
53	Pointer to pass A = 70.
54	Pointer to pass B = 117.
.	.
68	Pointer to pass P = 775.

Pointer calculation: $IPASS(J) = (J - 1) * PPSIZ + GPSIZ$ for example $PPSIZ = 47$, $GPSIZ = 70$, for 16 passes.

START OF PASS A:

69 Event number.
70 Pass # = 1.
71 Starting relative block # for this pass.
72 Size of pass in blocks.
73 First channel address of A/D (FCA).
74 Last channel address of A/D (LCA).
75 Pointer list (32 words).
106 .
107 Pass switch setting (DIGOUT BITS).
108 Filter switch setting (DIGOUT BITS).
109 A/D error word: BITS 0-7 = IAD; BITS 8-15 = IDSK.
110 HEX start time this pass (Dub tapes only).
111 Spare (5 words).
115 .
116 Start of PASS B:.
.
164 Start of PASS C:.
.
.
775 Start of PASS P (16th PASS):
.
820 (LAST ENTRY OF RECORD).

M: ROUTINES WHICH DIFFER BETWEEN THREE-PASS AND TWELVE-PASS VERSIONS

1. ISDS Routines

- a) The following routines differ only in that they use a different C\$EPARM include file for the 2 versions.
DEPRC, GPICK, ISDS7, LBS7A, LPICK, NUPRC, PLPRC, SPLOT.
- b) The following routines differ substantially:
EVPRC - Different C\$GPARM file, references to TCORR, UNQYR, UNQID. Call to RQPAZ is different between versions.
FAZPU - Different C\$GPARM file, contains labeled common /EDIRC/ which is also in DIRC routines, parameter RECSIZ, references UNQYR, UNQID, TCORR, APASS, FCA, LCA. Call routines MAKNAM and UTEP. Opens EPLOG file, opens FZnnnnn and CCnnnnn where nnnnn refers to 5 digit integer ID(UNQID) read from EPLOG by UTEP, does not open FAZLST.
- c) The following routines occur only in the twelve-pass version:
MAKNAM - Routine to concatenate, two characters.
ASCII Prefix and 5 digit integer number to make ASCII filename.
UTEP - Member of EPLOG (Event progress Log) support software. Given a station code, NCP, and an ID number, UNQID, it updates the status of the EPLOG file.

2. DIRC Routines

- a) The following routines differ that only include file P\$DIR is different between the two versions: DIR1, RQEV1.
- b) ENAME is different only in the size of the EVDIR records, so parameter REKSIZ varies.
- c) RQPAZ differ in the include file P\$DIR and also in the calling sequence. The P 12-pass call is CALL RQPAZ (LIST, FCA, LCA, EVENT, PASS).

2. Listings

- N. Listings of ISDS Subroutines
- O. Listings of Some Support Software
- P. Listings of Twelve-Pass Routines

Figure 1

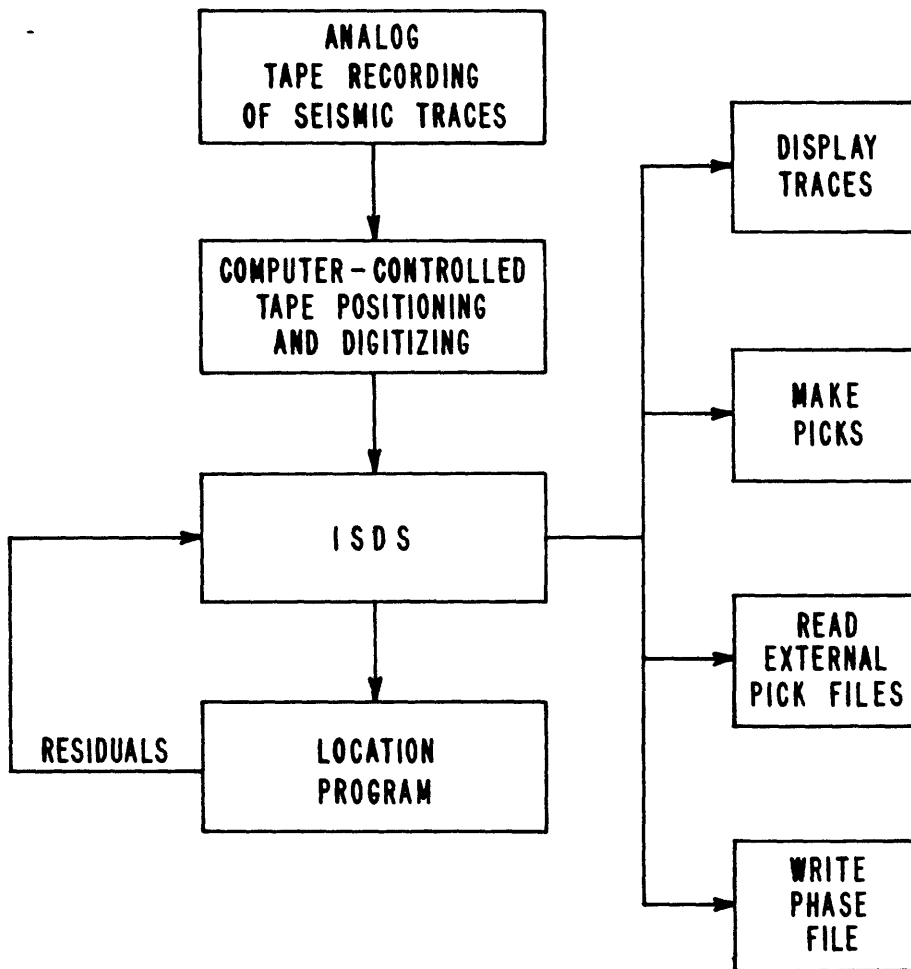


figure 2

SELECT PASS NUMBER

NEW BRIGNES FORESHOCK MAG 1.8			914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00	914	.00	PASS 3814	40.00
1	AFM	1	BPC	1	CNC	1	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
2	AFD	2	BPF	2	CPL	2	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
3	ADU	3	BPI	3	CRA	3	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
4	ABR	4	BPP	4	CRP	4	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
5	ABJ	5	BPM	5	CSC	5	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
6	ABA	6	BRU	6	CBH	6	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
7	AAR	7	BSB	7	CBZ	7	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
8	ARJ	8	BSC	8	GAX	8	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
9	APR	9	BSG	9	GBG	9	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
10	AOH	10	BBL	10	GBD	10	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
11	AOD	11	BSR	11	GBM	11	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
12	ALN	12	BUL	12	GBU	12	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
13	AMR	13	BUY	13	GDC	13	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
14	ARD	14	CAC	14	GDL	14	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
15	AGI	15	CAD	15	GDP	15	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
16	BCQ	16	CAI	16	QAC	16	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
17	BBN	17	CAL	17	QAG	17	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
18	BBQ	18	CAO	18	QAL	18	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
19	BAU	19	CBR	19	QAC	19	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
20	AUR	20	CBU	20	QAC	20	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
21	ASR	21	CCN	21	QAO	21	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
22	ARU	22	CCO	22	QPH	22	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
23	ARR	23	CCV	23	QAM	23	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
24	BHM	24	CDO	24	QRT	24	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
25	BPC	25	CUI	25	QSG	25	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
26	BLR	26	CUU	26	QSH	26	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
27	BJO	27	CLC	27	QSH	27	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
28	BJC	28	CNC	28	QSS	28	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
29	BHG	29	CMH	29	QAZ	29	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
30	BSR	30	CMH	30	QBT	30	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
31	BSM	31	CSO	31	QCA	31	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00
32		32		32		32	914	.00	PASS 1814	40.00	914	.00	PASS 2814	40.00

figure 3

CONTINUED

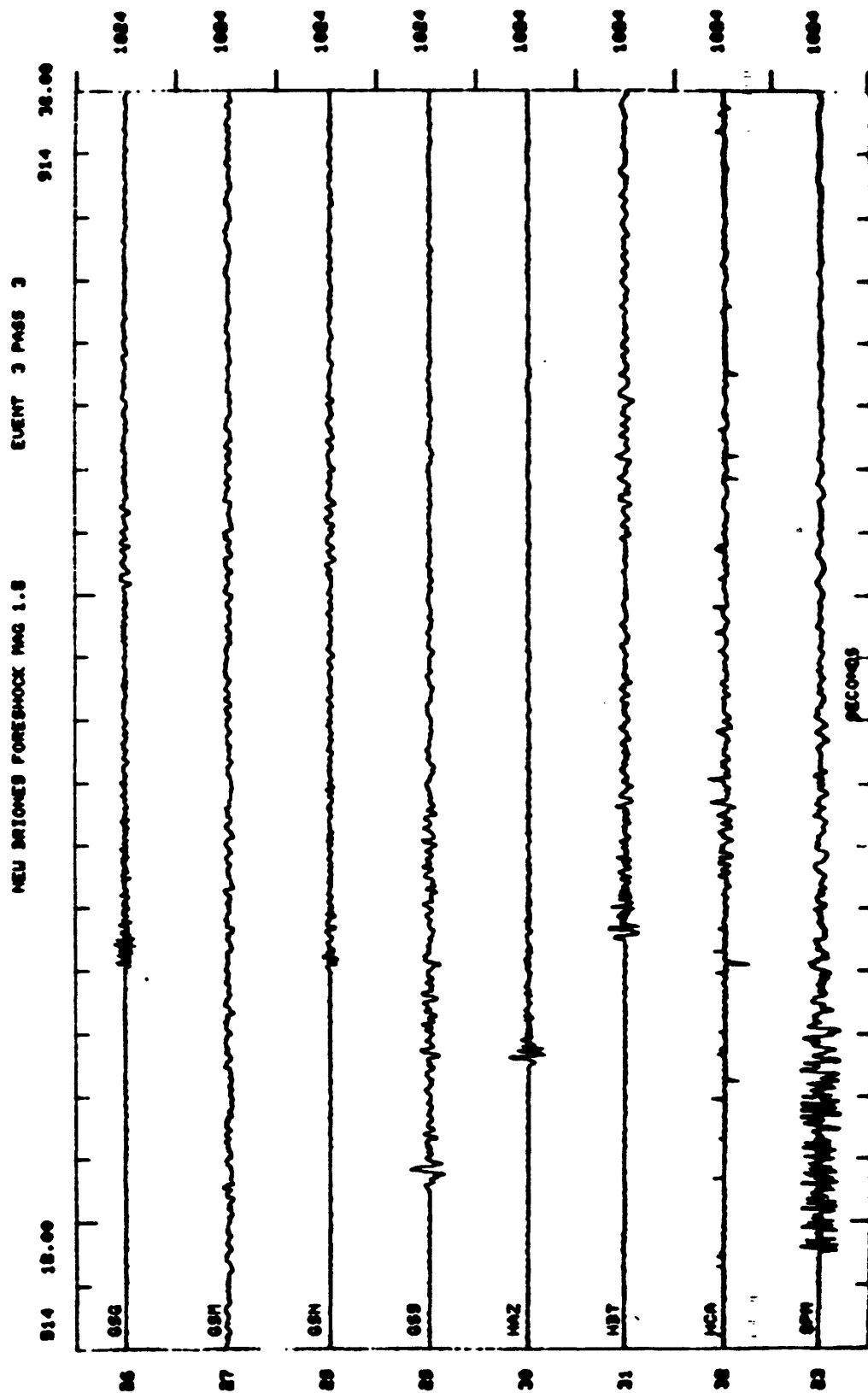
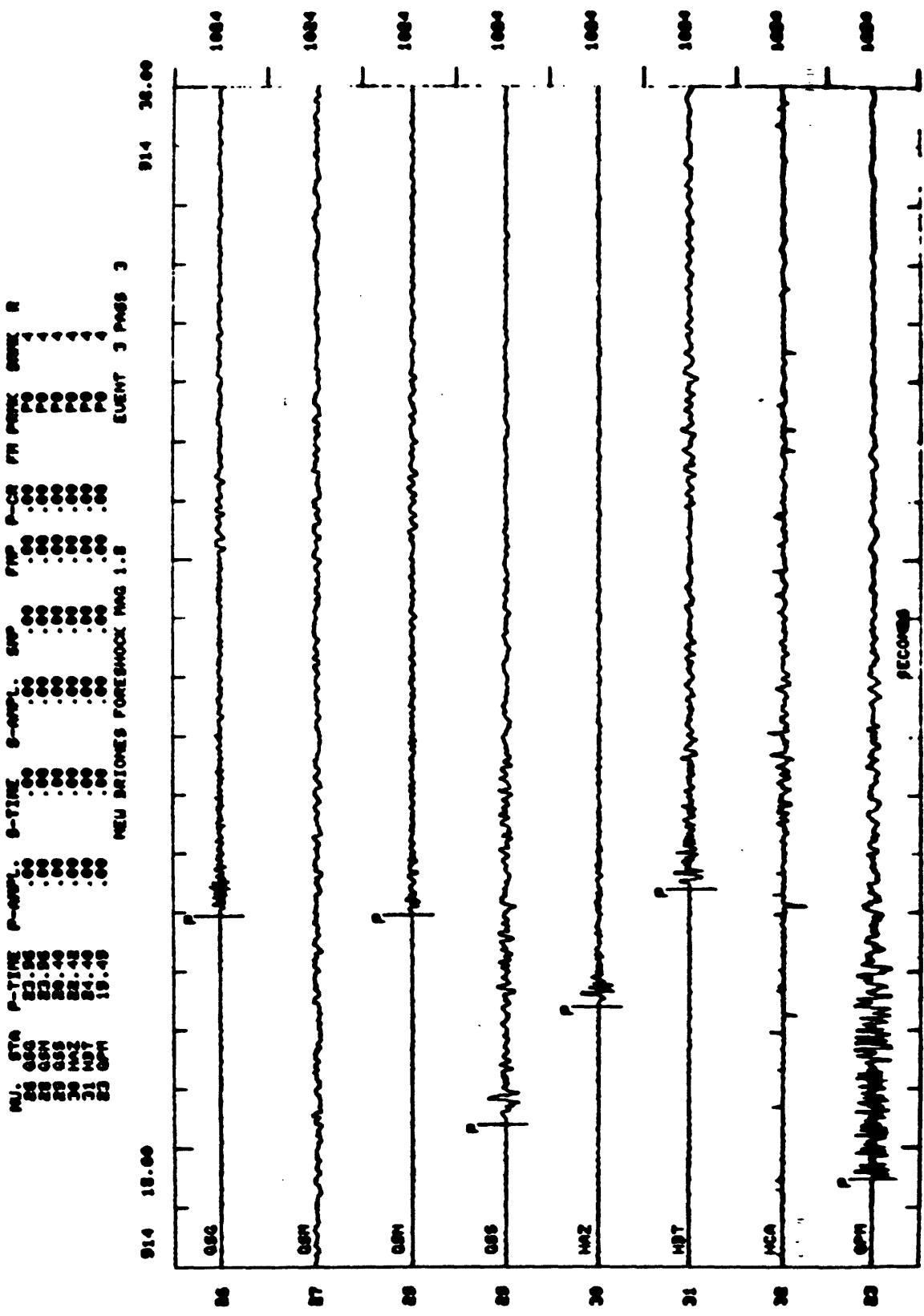


figure 4

CONTINUED



1111

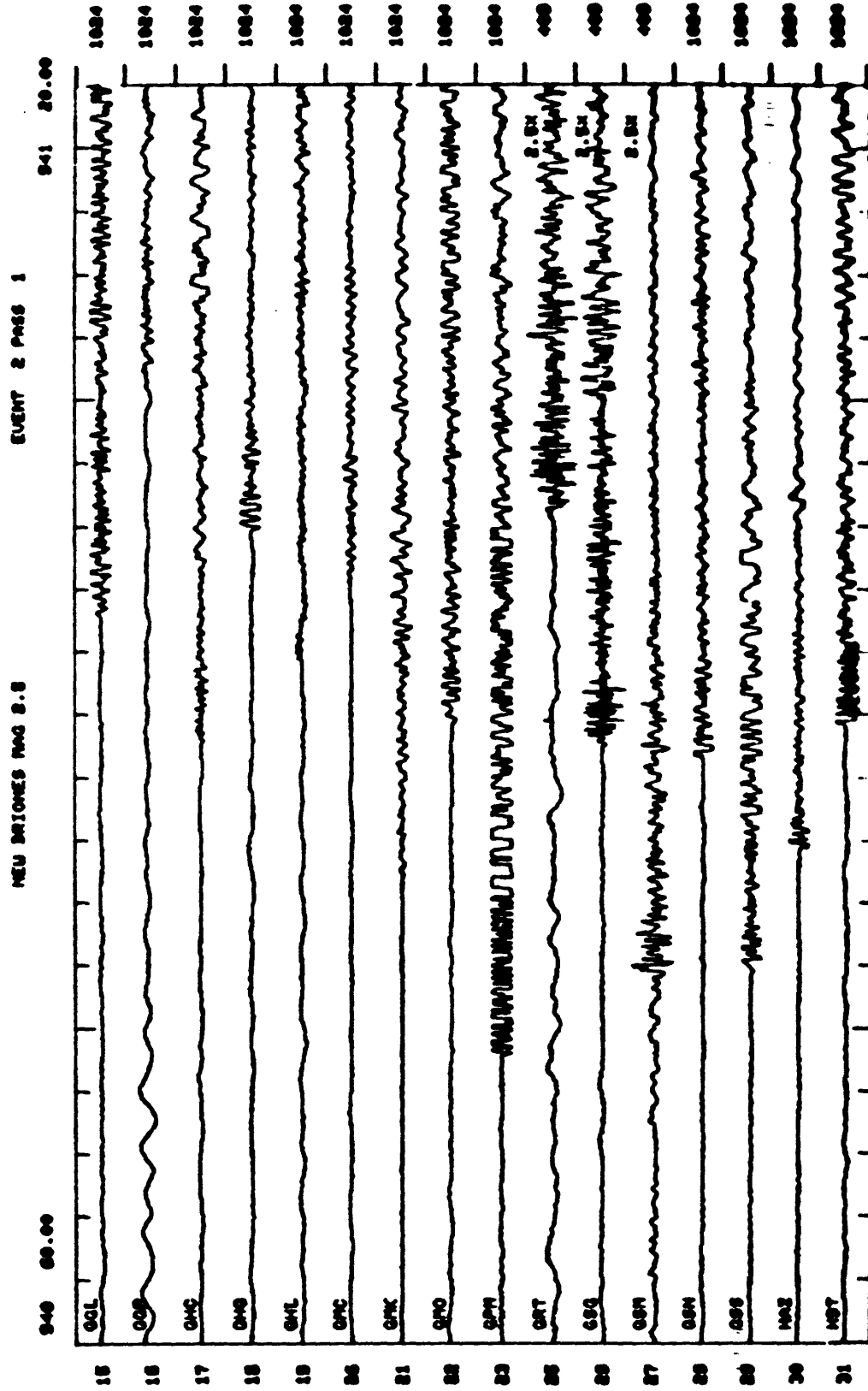


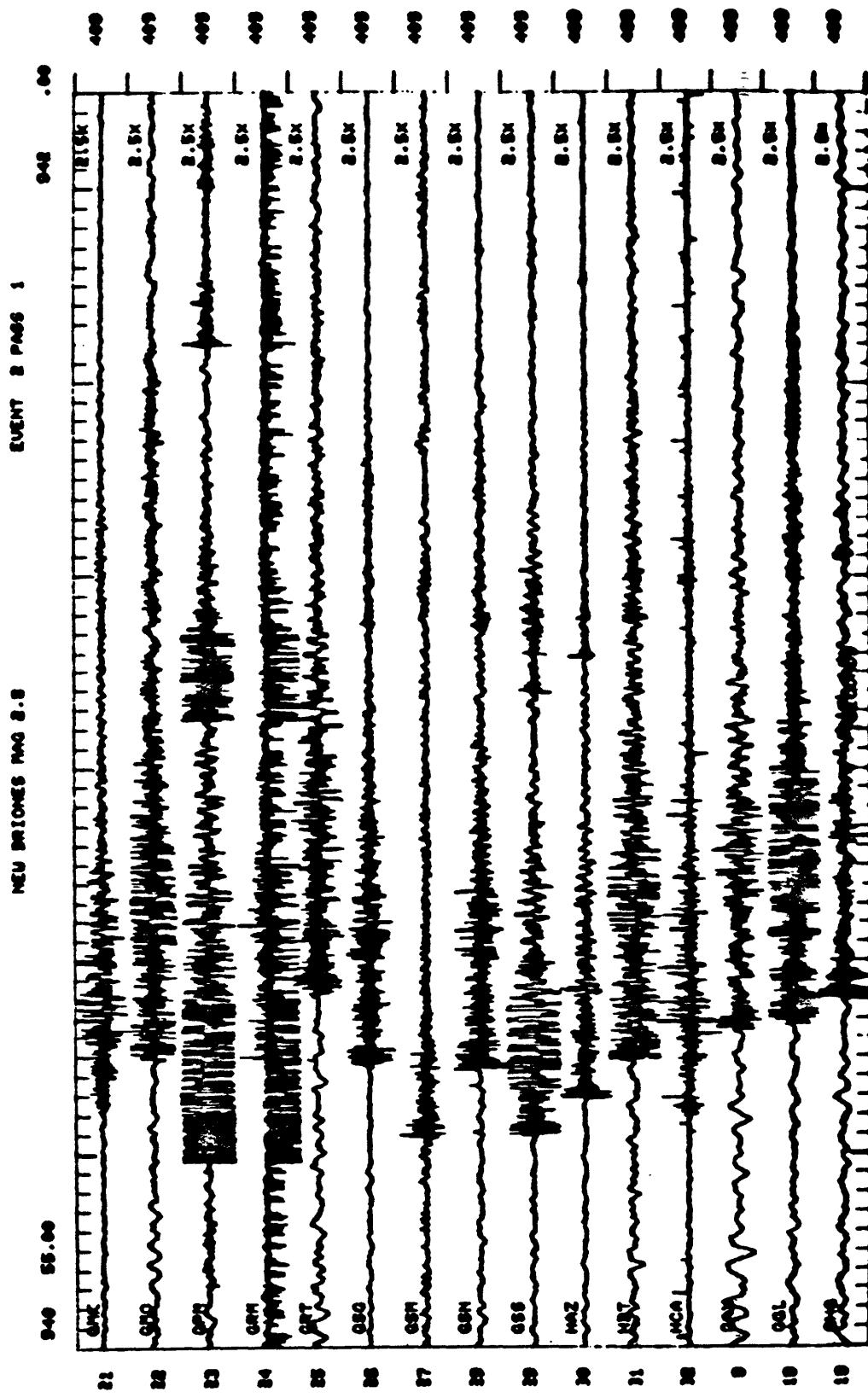
figure 6

SELECT PASS NUMBER

940 55.00		942 .00	
1	CNC		
2	CPL		
3	CRA		
4	CRT		
5	CSC		
6	CSH		
7	DS		
8	GAX		
9	GBD		
10	GBD		
11	GCM		
12	QCU		
13	QDC		
14	QGL		
15	QGP		
16	QMC		
17	QMD		
18	QML		
19	QMC		
20	QMK		
21	QMO		
22	QPN		
23	QRM		
24	QRT		
25	QSO		
26	QSM		
27	QSN		
28	QSS		
29	MAZ		
30	MBT		
31	MCA		
32			

figure 7

COMMENT



TRACES SHIFTED BY P-TIMES

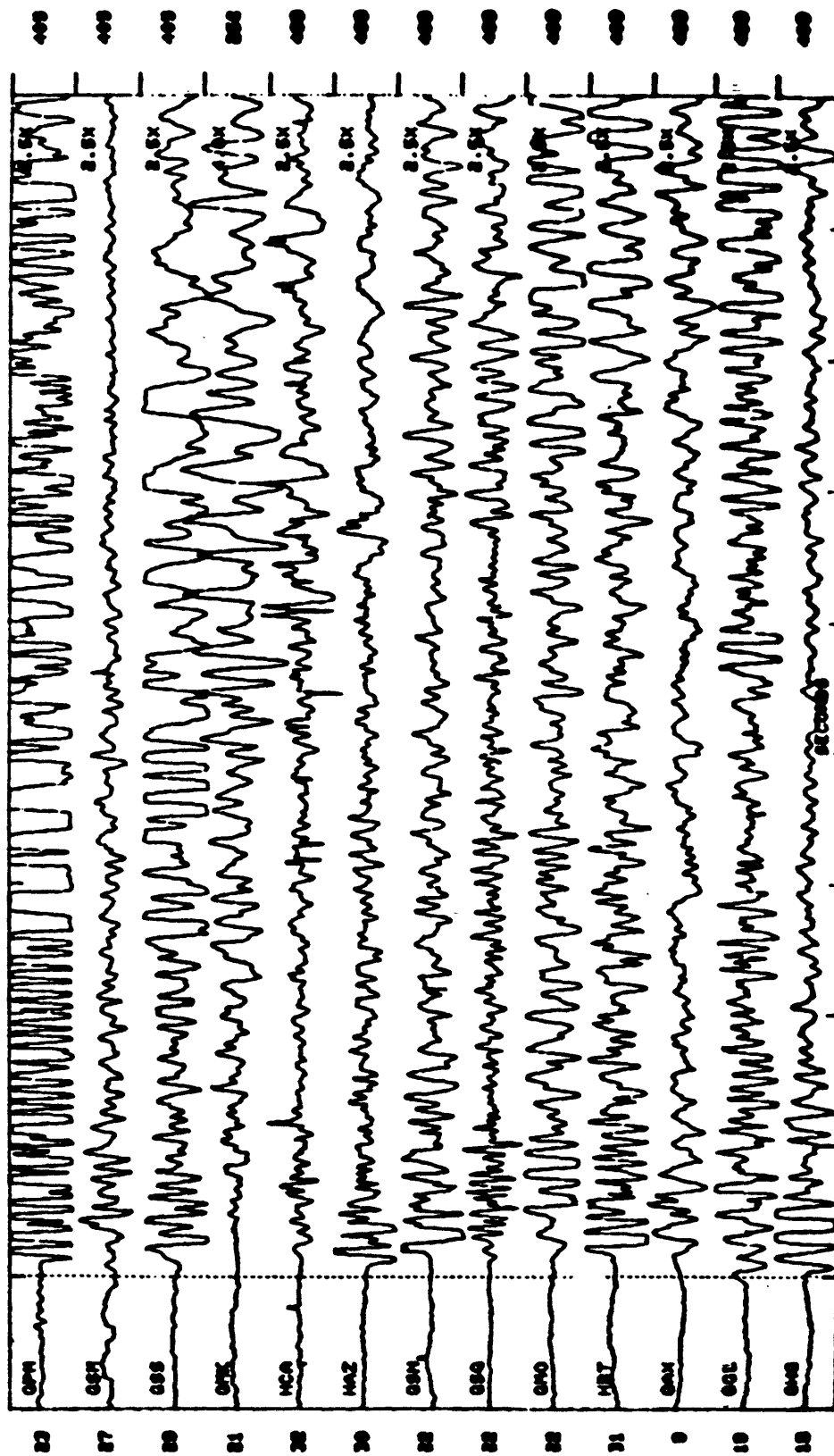
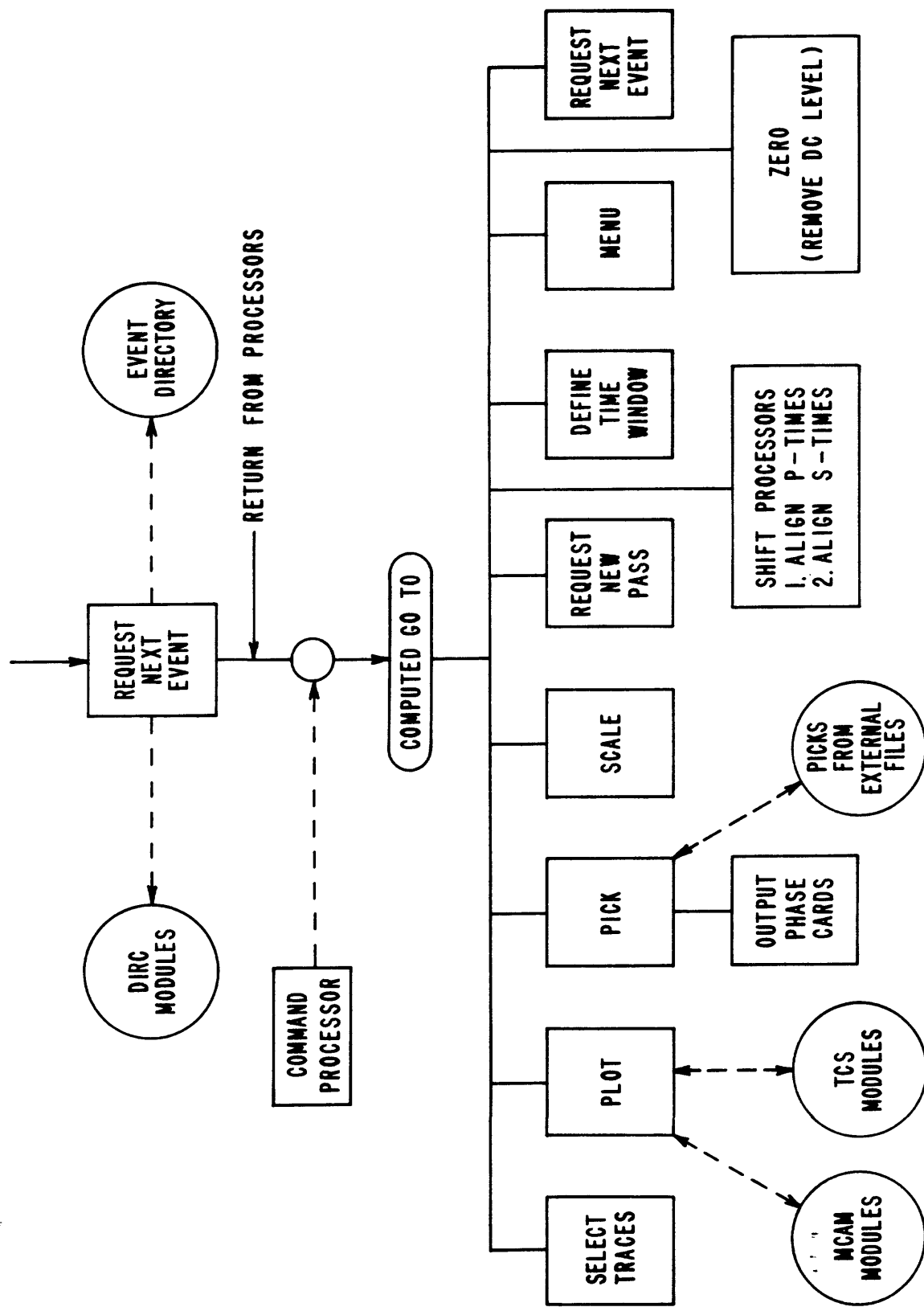
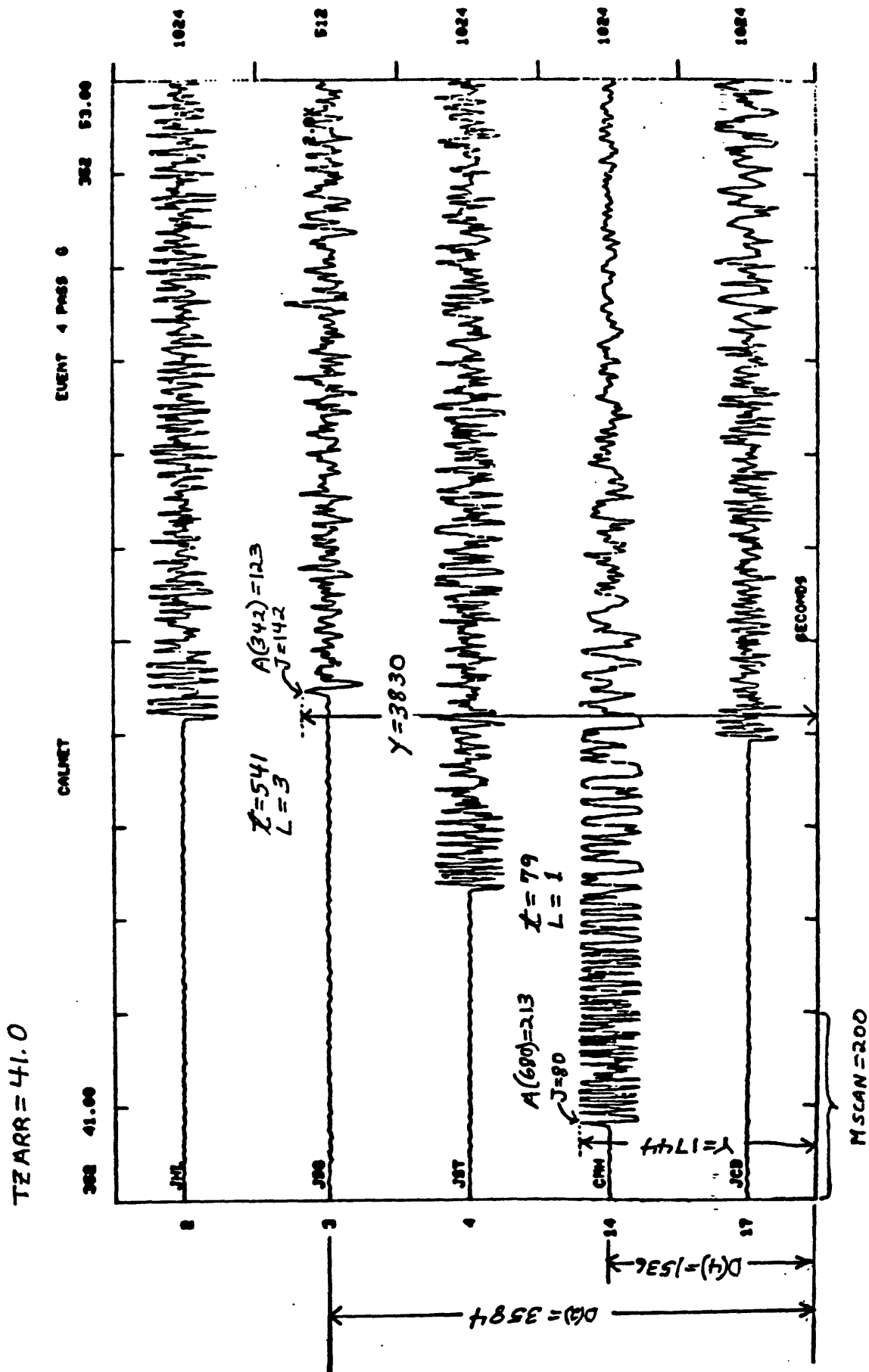


Figure 9

INTERACTIVE SEISMIC DISPLAY SYSTEM



TZARR=41.0



APPENDIX N: LISTINGS OF ISDS SUBROUTINES

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:05:31 PM

ARRAY.FR

```

1:      SUBROUTINE XARRAY
2:      OVERLAY OARRAY
3:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32)
4:      *,D(32)
5:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
6:      INCLUDE 'CS9LAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
9:      INTEGER A
10:     COMMON /IOPRM/NS,MSCAN,NREM,MUX,IFILE,IUNIT
11:     COMMON /DECI/IDF,DECMAX,DLIM
12:     DIMENSION LFRAM(32)
13:     CC S(I)=AMPLITUDE SCALE FACTOR FOR ITH TRACE
14:     CC Z(I)=ZERO LEVEL OF ITH TRACE RELATIVE TO DISTANCE
15:     CC DT(I)=DATA POINT INTERVAL FOR ITH TRACE
16:     CC TZ(I)=ABSOLUTE TIME (SECONDS) OF FIRST DATA POINT
17:     CC SHFT(I)=TIME SHIFT (SECONDS) FOR ITH TRACE
18:     CC D(I)=DISTANCE OR POSITION OF ITH TRACE ( AMPL. UNITS)
19:     CC TZARR=BEGINNING OF DISPLAY WINDOW (SECONDS)
20:     CC TWIND=LENGTH OF DISPLAY WINDOW (SECONDS)
21:     CC AZ=BOTTOM OF SCREEN IN ARRAY AMPLITUDE UNITS
22:     CC ARNG=WIDTH OF EACH TRACE IN RAW AMPLITUDE UNITS
23:     CC NT=NUMBER OF TRACES TO BE PLOTTED ON SCREEN
24:     CC AL=TOP OF SCREEN IN ARRAY AMPLITUDES UNITS
25:     CC IDF=DECIMATION FACTOR (SET TO 1 FOR NORMAL PLOTS)
26:     CC
27:     C COMPUTE SEGMENT LENGTH AS INTEGER MULTIPLE
28:     C OF DECIMATION FACTOR
29:     LL=MAXPTS/NREQ
30:     NLL=MOD(LL,IDF)
31:     MSCAN=LL-NLL
32:     ISCAN=TZARR/DTG+1
33:     CALL FRAMD(MSCAN,IREQ,NREQ,ISCAN,IERR)
34:     IF(IERR.NE.0)TYPE 'ARRAY-FRAMD: IERR=',IERR
35:     C
36:     C CHECK FOR MAX NEGATIVE SHIFT VALUE
37:     XSHF=1000.
38:     DO 2 J=1,32
39:     XSHF=AMIN1(XSHF,SHFT(J))
40:     C INITIALIZE LFRAM
41:     LFRAM(J)=D(J)
42:     2 CONTINUE
43:     CC DEFINE VIRTUAL SPACE
44:     CC
45:     C MSCAN IS EXPECTED NUMBER OF SCANS/SEGMENT
46:     C NSCAN IS ACTUAL NUMBER OF SCANS/SEGMENT
47:     NSEG=((TWIND-XSHF)/DTG-1)/MSCAN+1
48:     NRQ=NREQ
49:     AR=AZ+AL
50:     CALL VWINDO(TZARR,TWIND,AZ,AR)
51:     C TIC MARK SETUP
52:     TIC=10.
53:     IC=2
54:     C
55:     CALL FRAME(TZARR,TWIND,AZ,AR,TIC,IC)
56:     DO 20 L=1,NSEG
57:     CALL GETFM(A,NSCAN,IEND)
58:     IF(IEND.EQ.1.AND.NSCAN.EQ.0)GO TO 97
59:     DO 10 I=1,NRQ

```

```

60:      II=IREQ(I)
61:      TIME=TZARR+TWIND-SHFT(II)
62:      NPTS=TIME/DT(II)+1.
63:      IF(NPTS.GT.MAXPTS)NPTS=MAXPTS
64:  CC
65:  CC  PLOT TRACE I
66:      IJ=MSCAN*(I-1)+1
67:      Y=(LFRAM(I)-Z(II))*S(II)+D(I)
68:      X=SHFT(II)+TZ(II)+DT(II)*((L-1)*MSCAN-1)+TZARR
69:      CALL MOVEA(X,Y)
70:      NN=NPTS-1
71:      DO 1 J=1,NSCAN,IDF
72:      IJ=MSCAN*(I-1)+J
73:      Y=S(II)*(A(IJ)-Z(II))+D(I)
74:  C  ARG IS ARTIFACT TO GET AROUND 16-BIT INTEGER OVERFLOW
75:      ARG=(L-1)*MSCAN+J-1
76:      X=DT(II)*ARG+TZ(II)+SHFT(II)+TZARR
77:      CALL DRAWA(X,Y)
78:  1  CONTINUE
79:      LFRAM(I)=A(IJ)
80:  10  CONTINUE
81:      IF(IEND.EQ.1)GOTO 97
82:  20  CONTINUE
83:  97  CONTINUE
84:  C-----
85:      CALL TSEND
86:      CALL HITEK(1,0)
87:      CALL HOME
88:  99  RETURN
89:      END

```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:06:41 PM

BOX.FR

```
1:      SUBROUTINE XBOX(IX,LX,IY,LY)
2:      OVERLAY OBOX
3:      CALL SWINDO(IX,LX,IY,LY)
4:      CC  DRAW BOX
5:      CC
6:      CALL MOVABS(IX,IY)
7:      CALL DRWREL(LX,0)
8:      CALL DRWREL(0,LY)
9:      CALL DRWREL(-LX,0)
10:     CALL DRWREL(0,-LY)
11:     RETURN
12:     END
```

COMMAND.FR

```

1:      SUBROUTINE XCOMMAND(IC,JC)
2:      OVERLAY OCOMMAND
3:      COMMON /TVTUNE/ITV,ITX
4:      INCLUDE 'CSDICT'
5:      DIMENSION JCMND(35),ICOMM(41)
6:      C
7:      C      * DICTIONARY CROSS-REF TABLE *
8:      C
9:      C      1 DE    2 NU    3 PL    4 SH    5 PI
10:     C      6 PA    7 RE    8 SC    9 CR   10 PC
11:     C     11 SA   12 ME   13 US   14 EV   15 SP
12:     C     16 WA   17 ST   18 EX   19 TI   20 VI
13:     C     21 SE   22 LE   23 T    24 ZE   25 -P
14:     C     26 -S   27 TU   28 ?    29 UM   30 PR
15:     C     31 LD   32 CO   33 PY   34 SU   35 TR
16:     C
17:     DATA JCMND/2HDE,2HNU,2HPL,2HSH,2HPI,2HPA,
18:           $2HRE,2HSC,2HCR,2HPC,2HSA,2HME,2HUS,
19:           $2HEV,2HSP,2HWA,2HST,2HEX,2HTI,2HVI,2HSE,2HLE,2HT ,
20:           $2HZE,2H-P,2H-S,2HTU,2H? ,2HUN,2HPR,2HLD,2HCO,2HPY,
21:           $2HSU,2HTR/
22:     DATA IQ/63/
23:     MCD=35
24:     C
25:     WRITE(ITV,400)IQ
26:     400  FORMAT(8H COMMAND,R1)
27:     READ(ITX,401,END=500)ICOMM(1)
28:     401  FORMAT(S80)
29:     DECODE(ICOMM,402) (ICOMM(J),J=1,40)
30:     402  FORMAT(40A2)
31:     DO 1 J=1,MCD
32:     IF(ICOMM(1).EQ.JCMND(J))GO TO 2
33:     1    CONTINUE
34:     IC=MCD+1
35:     GO TO 99
36:     2    CONTINUE
37:     IC=J
38:     JC=0
39:     DO 3 J=1,MCD
40:     IF(ICOMM(2).EQ.JCMND(J))GO TO 4
41:     3    CONTINUE
42:     GO TO 99
43:     4    JC=J
44:     GOTO 99
45:     C IC=18 ON EOF MEANS EXIT ON <CNTRL Z>
46:     500  IC=18
47:     99   RETURN
48:     END
49:

```

DECIM.FR

```

1:      SUBROUTINE XDECIM
2:      OVERLAY ODECIM
3:      C INPUT: TWIND (IN /BPARM/) PLOT WINDOW TIME
4:      C DTG (IN /BPARM/) SAMPLE TIME
5:      C DECMAX (IN /DECI/) MAX NUMBER OF DECIMATED SAMPLES
6:      C NREQ (IN / /) NUMBER OF TRACES
7:      C OUTPUT: IDF (IN /DECI/) DECIMATION FACTOR
8:      C
9:      INCLUDE 'CSBLAN'
10:     PARAMETER MAXX=1000
11:     COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
12:     INTEGER A
13:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
14:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
15:     COMMON /DECI/IDF,DECMAX,DLIM
16:     C
17:     C --- COMPUTE APPROPRIATE DECIMATION FACTOR ---
18:     C
19:     C TOTAL=TOTAL NUMBER OF DIGITAL POINTS FOR DISPLAY
20:     C IF THERE WERE NO DECIMATION
21:     C
22:     C TOTAL=(TWIND*NREQ)/DTG
23:     C
24:     C IDF=DECIMATION FACTOR
25:     C IDF = TOTAL/DECMAX
26:     C
27:     C DF=TOTAL/DECMAX
28:     C IDF=DF
29:     C R=DF-IDF
30:     C IF(R.GT.0.)IDF=IDF+1
31:     C -----
32:     C
33:     RETURN
34:     END

```

DEFAULT.FR

```

1:      SUBROUTINE XDEFAULT
2:      C - - - P.R. STEVENSON (8/21/78) - - -
3:      OVERLAY ODEFAULT
4:      INCLUDE 'CSAPARM'
5:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
6:      INCLUDE 'CSBPARM'
7:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
8:      C
9:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
10:     INCLUDE 'CSBLAN'
11:     PARAMETER MAXX=1000
12:     COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
13:     INTEGER A
14:     INCLUDE 'CSPICK'
15:     INTEGER PRMK,SRMK,RMK,SGRADE
16:     COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
17:     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
18:     *IFMOT(32),IGRADE(32),SGRADE(32),
19:     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
20:     *PRES(32),SRES(32),EPD(32),CLK(32)
21:     DATA IBLNK/2H /,IPRMK/2H P/,ISRMK/2H S/
22:     COMMON /DECI/IOF,DECMAX,OLIM
23:     COMMON /ZZZ/IZIT
24:     JMAX=32
25:     C IGO=DEFAULT PICK GRADE ASSIGNMENT
26:     IGO=0
27:     IF(IZIT.GT.0)GO TO 700
28:     C THIS SECTION ON INITIAL CALL ONLY
29:     C IT SETS CERTAIN GLOBAL PARAMETERS
30:     IZIT=1
31:     MAXPTS=MAXX
32:     DECMAX=40000.
33:     OLIM=5.
34:     CC
35:     KX=10
36:     LX=1000
37:     KY=10
38:     LY=600
39:     ITERM=1
40:     ISCAL=1024
41:     JCHAR=4
42:     CC
43:     DTG=.005
44:     SG=1.
45:     C ARNG=1024 CORRESPONDS TO 10-BIT A/D CONVERTER
46:     ARNG=1024.
47:     AZ=0.
48:     TZARR=0.
49:     TWIND=10.
50:     THR=0.
51:     DO 1 J=1,JMAX
52:     S(J)=SG
53:     DT(J)=DTG
54:     TZ(J)=TZARR
55:     SHFT(J)=0.
56:     Z(J)=0.
57:     U(J)=0.
58:     1 CONTINUE
59:     C THIS SECTION(/DPARM/) RE-INITIALIZED ON

```



```

60: C ALL CALLS TO DEFAULT
61: 700 CONTINUE
62: DO 2 J=1,JMAX
63: PTIME(J)=0.
64: AMP(J)=0.
65: STIME(J)=0.
66: FTIME(J)=0.
67: CROSS(J)=0.
68: SMP(J)=0.
69: FMP(J)=0.
70: PCROSS(J)=0.
71: IGRADE(J)=IGD
72: SGRADE(J)=IGD
73: IFMOT(J)=IBLNK
74: PRMK(J)=IPRMK
75: SRMK(J)=ISRMK
76: SAMP(J)=0.
77: AX(J)=0.
78: SX(J)=0.
79: RMK(1,J)=IBLNK
80: RMK(2,J)=IBLNK
81: RMK(3,J)=IBLNK
82: PRES(J)=0.
83: SRES(J)=0.
84: EPD(J)=0.
85: CLK(J)=0.
86: 2 CONTINUE
87: CC
88: CC AL,NT ARE NOT INITIALIZED HERE
89: RETURN
90: END
91:

```

DEPRC.FR

```

1:      SUBROUTINE XDEPRC(IC,JC)
2:      OVERLAY ODEPRC
3:      COMMON /TVTUNE/ITV,ITX
4:      COMMON /BPARM/TZARR,TWIND,THR,TMM,AZ,ARNG,AL,NT,DTG,SG
5:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
6:      INCLUDE 'C$BLAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,MREQ,IREQ(32)
9:      INTEGER A
10:     INCLUDE 'C$GPARM'
11:     C  MODIFIED 5/12/78 FOR 3-PASSES
12:     PARAMETER LSZ1=13
13:     PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
14:     COMMON /KDATE/KYR,KMO,KOA,KHR,KMN,RSEC,DUR,EVENT,PASS
15:     * ,UNQYR,UNQID,TCORR,FCA,LCA
16:     COMMON /JDATE/IHR,IMM,TSEC
17:     COMMON /XXX/XX(3),LIST(LSZ)
18:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
19:     COMMON /VEE/VSAVE(4,MAXPASS)
20:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
21:     C
22:     COMMON /EPARM/NTMP,ITEMP(32),OTZARR,OTWIND
23:     DIMENSION NNN(80),SQ(40)
24:     DATA IQ/63/
25:     DATA IY/1HY/
26:     DATA IX/1HX/
27:     DATA IBL/1H /
28:     INTEGER THR,TMM
29:     C*  DEFINE THE WINDOW
30:     1    CONTINUE
31:     IF(JC.EQ.19)GO TO 13
32:     IF(JC.EQ.20)GO TO 14
33:     C*  DEFINE TIME WINDOW
34:     13   WRITE(ITV,210)
35:     210  FORMAT(19H DEFINE TIME WINDOW)
36:     WRITE(ITV,211)IQ
37:     211  FORMAT(23H WINDOW WIDTH (SECONDS),R1)
38:     C*
39:     C*  READ TWIND
40:     C*  READ TIME WINDOW LENGTH
41:     READ(ITX,234)NNN(1)
42:     234  FORMAT(S80)
43:     CALL INFREE(NNN,RQ,80)
44:     TWIND=RQ
45:     C*
46:     C*
47:     C*  READ HRMIN SEC
48:     WRITE(ITV,222)KHR,KMN,RSEC
49:     222  FORMAT(16H DATA STARTS AT ,I2,1H,,I2,2X,F7.2)
50:     WRITE(ITV,204)IQ
51:     204  FORMAT(23H HR MIN SEC (FREE-FORM),R1,' [CR] GETS DEFAULT')
52:     WRITE(ITV,223)
53:     223  FORMAT(' TYPE 'X R' FOR AN R SEC. OFFSET OF ORIGIN')
54:     READ(ITX,234)NNN(1)
55:     CALL INFREE(NNN,SQ,80)
56:     IF(NNN(1).EQ.IBL)GOTO 55

```

```

57:         IF(NNN(1).EQ.IX)GOTO 55
58:         THR=SQ(1)
59:         TMN=SQ(2)
60:         TSEC=SQ(3)
61:         GOTO 56
62: C   COME HERE FOR CR OR X MN RESPONSE
63: C
64: 55     THR=KHR
65:         TMN=KMN
66:         TSEC=RSEC
67:         IF(NNN(1).EQ.IX)TSEC=RSEC+SQ(1)
68: C
69: 56     CONTINUE
70: C   COMPUTE TIME WINDOW RELATIVE TO DATA
71: C   TTT IS REQUESTED TIME IN SECONDS
72: C   RRR IS START OF DATA WINDOW IN SECONDS
73:         TTT=THR*3600+TMN*60+TSEC
74:         RRR=KHR*3600+KMN*60+RSEC
75:         TZARR=TTT-RRR
76:         IF(TZARR.LT.0.)WRITE(ITV,220)
77:         IF(TZARR.GT.0UR)WRITE(ITV,220)
78:         IF(TZARR.LT.0..OR.TZARR.GT.0UR)TYPE '/JDATE/:'
79:         *,' IHR=',IHR,' IMN=',IMN,' TSEC=',TSEC
80:         IF(TZARR.LT.0..OR.TZARR.GT.0UR)TYPE ' TZARR=',
81:         *TZARR,' TSEC=',TSEC,' RSEC=',RSEC,' TTT=',TTT,
82:         *' RRR=',RRR
83: 220     FORMAT(36H REQUESTED TIME OUT OF RANGE OF DATA)
84: 203     FORMAT(32H STARTING TIME WINDOW (HRMN,SEC),2X,2F4.0,1X,F7.2)
85: C*
86:         WRITE(ITV,203)THR,TMN,TSEC
87:         WRITE(ITV,221)TZARR
88: 221     FORMAT(1H ,F7.2,15H SEC. INTO DATA)
89:         IHR=THR
90:         IMN=TMN
91:         OTZARR=TZARR
92:         RETURN
93: 14     WRITE(ITV,205)IQ
94: 205     FORMAT(10H 4014 TERM,R1,6HY OR N)
95:         READ(ITX,701)IA
96: 701     FORMAT(A1)
97:         CALL TERM(1,1024)
98:         ITERM=1
99:         IF(IA.EQ.IY)CALL TERM(3,4096)
100:        IF(IA.EQ.IY)ITERM=4
101:        LWIDTH=1000
102:        LHY=600
103:        JX=10
104:        JY=10
105:        WRITE(ITV,206)IQ
106: 206     FORMAT(12H RESET FRAME,R1)
107:        READ(ITX,701)IA
108:        IF(IA.NE.IY)GO TO 12
109:        WRITE(ITV,207)
110: 207     FORMAT(33H INPUT WIDTH, HEIGHTH-(FREE FORM))
111:        READ(ITX,234)NNN(1)
112:        CALL INFREE(NNN,SQ,80)
113:        LWIDTH=SQ(1)
114:        LHY=SQ(2)
115:        WRITE(ITV,208)
116: 208     FORMAT(39H INPUT SCREEN ORIGIN-IX,IY--(FREE FORM))

```

```

117:      READ(ITX,234)MNN(1)
118:      CALL INFREE(MNN,SQ,80)
119:      JX=SQ(1)
120:      JY=SQ(2)
121: 12      CONTINUE
122:      CALL SWND(JX,LWIDTH,JY,LHY)
123:      ISCAL=1024*ITERM
124:      CALL CHRISZ(JCHAR)
125: C*
126:      RETURN
127:      END

```

EVPRC.FR

```

1:      SUBROUTINE KEVPRC(IC,JC)
2:      OVERLAY DEVPRC
3:      INCLUDE 'CSGPARM'
4:      C MODIFIED 5/12/78 FOR 3-PASSES
5:      PARAMETER LSZ1=13
6:      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
7:      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
8:      *,UNQYR,UNQID,TCORR,FCA,LCA
9:      COMMON /JDATE/IHR,IMN,TSEC
10:     COMMON /XXX/XX(3),LIST(LSZ)
11:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
12:     COMMON /VEE/VSAVE(4,MAXPASS)
13:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
14:
15:     C
16:     INCLUDE 'CSBLAN'
17:     PARAMETER MAXX=1000
18:     COMMON A(MAXX),MAXPTS,NRED,IREQ(32)
19:     INTEGER A
20:     DIMENSION MSG(6)
21:     DIMENSION NNN(41),SQ(10)
22:     DATA MSG/80,65,83,83,32,48/
23:     COMMON /ZER/IZER
24:     INCLUDE 'CSAPARM'
25:     COMMON /APARM/S(32),Z(32),OT(32),TZ(32),SHFT(32),D(32)
26:     INCLUDE 'CSBPARM'
27:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
28:
29:     C
30:     COMMON /CPARM/NSTA,ISTA(2,32)
31:     COMMON /EPARM/NTEMP,ITEMP(32),UTZARR,OTWIND
32:     COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,ICHAR
33:     COMMON /TVTUNE/ITV,ITX
34:     COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
35:     COMMON /MISC/ZX,ZY,CX,CY
36:     COMMON /DECI/IDF,DECMAX,DLIM
37:     INTEGER CX,CY,ZX,ZY
38:
39:     C EVENT REQUEST PROCESSOR
40:     IF(IC.EQ.6)GO TO 52
41:     WRITE(ITV,302)
42:     302 FORMAT(20H SELECT EVENT NUMBER)
43:     READ(ITX,234)NNN(1)
44:     234 FORMAT(S80)
45:     CALL IVFREE(NNN,SQ,80)
46:     EVENT=SQ(1)
47:     CALL ROEVT(LIST,EVENT)
48:     IF(LIST(1).NE.0)GOTO 50
49:
50:     C OUTPUT ERROR MESSAGE AND RETURN
51:     WRITE(ITV,303)
52:     303 FORMAT(18H EVENT NOT ON DISK)
53:     RETURN
54:
55:     50 CONTINUE
56:
57:     C LOAD BPARM AND KDATE VARIABLES#
58:     KYR=LIST(2)
59:     KMO=LIST(3)
60:     KDA=LIST(4)
61:     KHR=LIST(5)
62:     KMN=LIST(6)
63:     RSEC=SEC
64:     DUR=DURAT
65:     DTG=DELT

```

```

60: C REZERO TZARR (POINT TO START OF DATA)
61:   TZARR=0.
62:   DTZARR=TZARR
63: C TYPE 'SEC,DURAT,DELT=',SEC,DURAT,DELT
64:   NPASS=LIST(LSZ1)
65: C LOAD DT ARRAY
66:   DO 40 J=1,32
67: 40 DT(J)=DTG
68: C
69: C EQUATE JOATE WITH KDATE FOR INITIAL PLOTS
70:   IHR=KHR
71:   IMN=KMN
72:   TSEC=RSEC
73: C
74: C IGAP IS SIZE OF GAP(IN TEKPOINTS) BETWEEN VIEWPORTS
75:   IGAP=120/NPASS
76:   KX1=30
77: C COMPUTE WIDTH OF VIEWPORTS
78: C
79:   LXX=900/NPASS-(NPASS-1)*IGAP
80:   JY=10
81:   LHY=600
82:   CALL ERASE
83: C
84:   DO 51 N=1,NPASS
85: C COMPUTE X-ORIGIN OF NTH VIEWPORT
86:   JX=KX1+(N-1)*(IGAP+LXX)
87:   CALL SAND(JX,LXX,JY,LHY)
88: C SAVE SCREEN CO-ORDINATES OF CURRENT VIEWPORT
89:   VSAVE(1,N)=JX
90:   VSAVE(2,N)=LXX
91:   VSAVE(3,N)=JY
92:   VSAVE(4,N)=LHY
93: C PLOT THIS PASS
94: C SETUP I/O TO PLOT ALL DATA
95:   MUX=LIST(N+LSZ1)
96:   IF(MUX.EQ.0)GOTO 51
97:   NCHAN=MUX
98:   NREQ=MUX
99: C EQUATE PREVIOUS TRACE STATE TO CURRENT STATE
100:   NTEMP=NREQ
101:   DO 30 J=1,NREQ
102:   ITEMP(J)=J
103: 30 IREQ(J)=J
104: C
105: C SET WINDOW TO DURATION TIME OF DATA
106:   T4IND=OUR
107:   CALL DECIM
108: C
109:   CALL RQPAZ(ISTA,EVENT,N)
110: C
111:   CALL NOSHF
112: C SPEED BY-PASS
113: C COMMAND EVSP MEANS GET EVENT BUT DON'T PLOT
114: C COMMAND EVSP CORRESPONDS TO JC=3
115:   IF(JC.NE.3)GOTO 51
116: C
117:   CALL RSETD
118:   IF(IZER.EQ.1)CALL ZLEV
119:   CALL ARRAY
120:   CALL L9STA(1)
121:   MXX=LXX/2+JX
122:   MYY=LHY+JY+20

```

```

123:      CALL MOVABS(MXX,MYX)
124:      MSG(6)=48+N
125:      CALL ANSTR(6,MSG)
126: 51      CONTINUE
127:      IF(JC.NE.3)GOTO 52
128:  C      LABEL PLOT WITH EVENT NAME
129:      CALL ENAME(NNN,EVENT)
130:      MZZ=LHY+JY+40
131:      CALL MOVABS(358,MZZ)
132:      CALL ADUTST(40,NNN)
133:  C
134:  C      PASS REQUEST PROCESSOR
135:      CALL HOME
136:      CALL ANMODE
137:      CALL TSEND
138: 52      CONTINUE
139:      WRITE(ITV,304)
140: 304      FORMAT(19H SELECT PASS NUMBER)
141:      CALL NEWLIN
142:      CALL SEELOC(CX,CY)
143:      READ(ITX,234)NNN(1)
144:      CALL INFREE(NNN,SQ,80)
145:      PASS=SQ(1)
146:      N=PASS
147:      CALL RQPAZ(ISTA,EVENT,PASS)
148:  C      - - - DEFINE TRACES FOR THIS PASS - - -
149:      MUX=LIST(LSZ1+N)
150:      IF(MUX.EQ.0) GOTO 99
151:      NREG=MUX
152:      NTEMP=MUX
153:      DO 1 J=1,NREG
154:      ITEMP(J)=J
155: 1      IREG(J)=J
156:      CALL NOSMF
157:      CALL RSETD
158:  C      DO THINGS USUALLY DONE BY ARRAY
159:      AR=AZ+AL
160:      CALL VWINDO(TZARR,TWIND,AZ,AR)
161:      CALL SRND(VSAVE(1,N),VSAVE(2,N),VSAVE(3,N),VSAVE(4,N))
162:      CALL SWINDO(IX,LX,IY,LY)
163:  C      - - - - -
164: 99      CONTINUE
165:  C      RESET DECIMATION FACTOR
166:      IOF=1
167:  C      NULL OUT OPARM BLOCK
168:      CALL DEFAULT
169:      RETURN
170:      END

```

FAZPU.FR

```

11      SUBROUTINE XFAZPU(18)
12      C      - - - P.R. STEVENSON - - - 8/24/78
13      OVERLAY OFAZPU
14      COMMON /TVTUNE/ITV,ITX
15      INCLUDE 'C6PICK'
16      INTEGER PRNK,SRNK,RNK,SGRADE
17      COMMON /DPARN/ PTIME(32),ANP(32),STIME(32),FTIME(32),
18      *CROSS(32),SNP(32),FNP(32),PCROSS(32),
19      *IFHOT(32),IGRADE(32),SGRADE(32),
20      *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
21      *PRES(32),SRES(32),EPD(32),CLK(32)
22      COMMON /CPARN/HSTA,ISTA(2,32)
23      INCLUDE 'C6GPARN'
24      C      MODIFIED 5/12/78 FOR 3-PASSES
25      PARAMETER LSZ1=13
26      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
27      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
28      *UNQYR,UNQID,TCORR,FCA,LCA
29      COMMON /JDATE/IHR,IMN,TSEC
30      COMMON /XXX/XX(3),LIST(LSZ)
31      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
32      COMMON /VEE/VSAVE(4,MAXPASS)
33      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
34      C
35      INTEGER OLDEVT,PUFLG,PUBLNK
36      DATA IBLANK/2H /,IZZ/2HZZ/,IPRNK/2H P/,ISRNK/2H S/
37      DATA IP/1HP/,IY/1HY/
38      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
39      C
40      C
41      C      PUFLG=0,FIRST TIME THRU FOR THIS PHASE FILE
42      C
43      C      THIS ROUTINE PUNCHES PHASE CARDS DESTINED FOR USGS EARTHQUAKE
44      C      HYPOCENTER ROUTINES
45      C      IF IS=2, PUNCH BLANK AND ZZ DELIMITER
46      C      AND RETURN
47      C      CHECK TO SEE IF SAME EVENT AS BEFORE
48      C
49      C      GOTO(10,20),IS
50      C      IF(PUFLG.EQ.0)RETURN
51      C      OPEN 7,NFIL,ATT='A'
52      C      WRITE(7,701)IBLANK
53      C      WRITE(7,701)IZZ
54      C      RETURN
55      C      CONTINUE
56      C      WRITE(ITV,401)NFIL
57      C      401  FORMAT(' PHASE FILE=',20A2,')
58      C      OPEN 7,NFIL,ATT='A'
59      C      FIRST TIME THRU, JUST PUNCH LIST
60      C      IF(PUFLG.EQ.0)GOTO 40
61      C      SAME EVENT, JUST PUNCH LIST
62      C      IF(OLDEVT.EQ.EVENT)GOTO 30
63      C      OTHERWISE PUNCH BLANK FIRST
64      C      WRITE(7,701)IBLANK
65      C      701  FORMAT(A2,70X)
66      C      40  WRITE(ITV,400)
67      C      400  FORMAT('NEW PHASE FILE?(Y OR N)')

```



```

57:      READ(ITX,200,ERR=30)IYY
58:  200  FORMAT(A1)
59:      IF(IYY.NE.IY)GOTO 30
60:      PUFLG=0
61:      CLOSE 7
62:      WRITE(ITV,402)
63:  402  FORMAT('NAME OF PHASE FILE?')
64:      READ(ITX,234)NFIL(1)
65:  234  FORMAT(S17)
66:      OPEN 7,NFIL,ATT='A'
67:  30   CONTINUE
68:  C   PUNCH PHASE LIST
69:      DO 1 J=1,32
70:  C   CHECK FOR NULL STATIONS
71:      IF(PTIME(J).LE.0..AND.STIME(J).LE.0..AND.
72:      *SAMP(J).LE.0..AND.AMP(J).LE.0.)GO TO 1
73:  C   CHECK FOR NULL LABELS
74:      IF(ISTAC(1,J).EQ.ISBLANK)GOTO 1
75:  C   OUTPUT PHASE CARD FOR THIS STATION
76:  C   THE FOLLOWING ALGORITHM CORRECTS FOR THE PROPER P(OR S) SECOND
77:  C   AND RIPPLES THE CORRECTION THRU THE CLOCK AND CALENDAR.
78:  C   IF ONLY S TIME IS PRESENT IT IS RIPPLED. IF BOTH P AND S
79:  C   ARE PRESENT THEN THE P TIME IS RIPPLED. THE S TIME MUST BE
80:  C   THE SAME MINUTE, HOUR, ETC. AS THE P TIME.
81:      PTA=PTIME(J)+RSEC
82:      ST=0.
83:      PT=0.
84:      IF(PTIME(J).EQ.0..AND.STIME(J).GT.0.)PTA=STIME(J)+RSEC
85:  C
86:      XT=AMOD(PTA,60)
87:      JT=(PTA-XT)/60.
88:      IF(PTIME(J).GT.0.)PT=PTIME(J)+RSEC-JT*60.
89:      IF(STIME(J).GT.0.)ST=STIME(J)+RSEC-JT*60.
90:      JMN=KMN+JT
91:      JHR=KHR
92:      JDA=KDA
93:      IF(JMN.GT.60)JHR=KHR+1
94:      IF(JMN.GT.60)JMN=KMN-60
95:      IF(JHR.GT.24)JDA=KDA+1
96:      IF(JHR.GT.24)JHR=KHR-24
97:      IAMP=AMP(J)
98:      ISAMP=SAMP(J)
99:  C   COMPUTE CODA FROM FTIME AND PTIME
100:      FMP(J)=FTIME(J)-PTIME(J)
101:      IF(FMP(J).LE.0.)FMP(J)=0.
102:  C   IF STIME=0 THEN SGRADE=4 AND SRNK=BLANK
103:  C   IF PTIME=0 THEN IGRADE=4 AND PRNK=BLANK
104:      IF(ST.EQ.0.)SGRADE(J)=4
105:      IF(ST.EQ.0.)SRNK(J)=ISBLANK
106:      IF(PT.EQ.0.)IGRADE(J)=4
107:      IF(PT.EQ.0.)PRNK(J)=ISBLANK
108:  C
109:  C
110:  71   WRITE(7,700)ISTAC(1,J),ISTAC(2,J),PRNK(J),IFNOT(J),IGRADE(J),
111:      *KYR,KNO,JDA,JHR,JMN,PT,IAMP,ST,SRNK(J),SGRADE(J),ISAMP,
112:      *RNK(1,J),RNK(2,J),CLK(J),RNK(3,J),FMP(J)
113:  700  FORMAT(2A2,A2,I1,1X,5I2,F5.2,15,2X,F5.2,A2,1X,I1,4X,13,15X,A2,
114:      *A1,F5.2,A1,I4,5X)
115:  1    CONTINUE
116:      PUFLG=1

```

117: OLDEVT=EVENT
118: CLOSE 7
119: RETURN
120: END
121:

FRAME.FR

```

1:      SUBROUTINE XFRAME(R1,RL,S1,SL,TIC,IC)
2:      OVERLAY OFRAME
3:      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,ICHAR
4:      DIMENSION SXB(2),SXL(2),SBB(2)
5:      CC :
6:      CC IF IC=0, DRAW BOX ONLY
7:      CC IF IC=1, PUT ON TIC MARKS AT TIC INTERVALS
8:      CC STARTING AT FIRST UNIT INTERVAL
9:      CC IF IC=2, ALSO PUT ON FINE TICS AT TIC/10
10:     CC INTERVALS
11:     CC
12:     S2=S1+SL
13:     S2=R1+RL
14:     S=S2/LY
15:     CC
16:     CALL BOX(IX,LX,IY,LY)
17:     IF (IC.EQ.0)GO TO 99
18:     TT=TIC/10
19:     Y=R1/TIC+1
20:     Z=INT(Y)*TIC
21:     NTIC=RL/TIC+1
22:     CC
23:     SX1(1)=-15*S
24:     SX1(2)=15*S
25:     SXL(1)=-10*S
26:     SXL(2)=10*S
27:     SBB(1)=S2
28:     SBB(2)=S1
29:     CC
30:     DO 20 L=1,2
31:     CC
32:     SB=SBB(L)
33:     SIT=SXB(L)
34:     SLT=SXL(L)
35:     IF (IC.EQ.1)GO TO 6
36:     DO 5 K=1,9
37:     XTT=Z-K*TT
38:     IF (XTT.LE.R1)GO TO 6
39:     SLT=SXL(L)
40:     CALL MOVEA(XTT,SB)
41:     CALL DRAWR(0.,SLT)
42:     5 CONTINUE
43:     CC
44:     6 CONTINUE
45:     CC
46:     DO 1 J=1,NTIC
47:     XT=Z+(J-1)*TIC
48:     CALL MOVEA(XT,SB)
49:     CC LARGE TIC DRAWN DOUBLE
50:     CALL DRAWR(0.,SIT)
51:     CALL DRAWR(0.,-SIT)
52:     IF (IC.EQ.1)GO TO 1
53:     CC
54:     CC DRAW SMALL TICS
55:     DO 2 K=1,9
56:     XTT=XT+TT*K
57:     IF (XTT.GE.R2)GO TO 1
58:     CALL MOVEA(XTT,SB)
59:     CALL DRAWR(0.,SLT)

```

```
60: 2    CONTINUE
61: CC
62: 1    CONTINUE
63: 20   CONTINUE
64: CC
65: 99   RETURN
66:      END
```

GPICK.FR

```

11      SUBROUTINE XGPICK(ICC)
12      OVERLAY OGPICK
13      COMMON /TVTUNE/ITV,ITX
14      INCLUDE 'C&CPARN'
15      C MODIFIED 5/12/78 FOR 3-PASSES
16      PARAMETER LSZ1=13
17      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
18      COMMON /KDATE/KYR,KMO,KDA,KHR,KHH,RSEC,DUR,EVENT,PASS
19      * ,UNQYR,UNQID,TCORR,FCA,LCA
20      COMMON /JDATE/IHR,INN,TSEC
21      COMMON /XXX/XX(3),LIST(LSZ)
22      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
23      COMMON /VEE/VSAVE(4,MAXPASS)
24      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
25      C
26      DIMENSION ICARD(41),LAB(3),NTT(3)
27      DATA IDOL/2H$$/
28      DATA NTT/2HEV,2HEM,2HT /
29      COMMON /CPARN/NSTA,ISTA(2,32)
30      INCLUDE 'C&GPICK'
31      INTEGER PRNK,SRNK,RNK,SGRADE
32      COMMON /DPARN/ PTIME(32),AMP(32),STIME(32),FTIME(32),
33      *CROSS(32),SHP(32),FMP(32),PCROSS(32),
34      *IFHOT(32),IGRADE(32),SGRADE(32),
35      *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
36      *PRES(32),SRES(32),EPD(32),CLK(32)
37      DIMENSION IREC(45),MNH(41)
38      DIMENSION NAM(2),IDONE(32),KRNK(2)
39      C ICC=1:READ AUTO PICKS
40      C ICC=2:READ PHASE LIST
41      C ICC=3:READ HYPOUT
42      C
43      WRITE(ITV,150)
44      150 FORMAT(' FILENAME?')
45      C READ '5' RATHER THAN ITX BECAUSE WISH INPUT
46      C FROM USER DURING PICK MODE
47      READ(5,441)MNH(1)
48      441 FORMAT(S17)
49      OPEN 15,MNH
50      GOTO(10,20,30),ICC
51      10 CONTINUE
52      3 READ(15,101,END=500)ICARD(1)
53      101 FORMAT(S80)
54      C LOOK FOR EVENT LABEL
55      DECODE(ICARD,102) (LAB(J),J=1,3)
56      102 FORMAT(3A2)
57      IF(LAB(1).EQ.NTT(1).AND.LAB(2).EQ.NTT(2).AND.LAB(3).EQ.
58      *NTT(3))GOTO 4
59      GOTO 3
60      4 DECODE(ICARD,103)JEVENT
61      103 FORMAT(6X,I2)
62      C - - TEST FOR CORRECT EVENT NUMBER
63      IF (JEVENT.NE.LIST(1))GOTO 3
64      C
65      TYPE 'EVENT NO.',JEVENT
66      C ZERO DONE FLAGS

```

```

57:      DO 5 J=1,32
58:      5      IDONE(J)=0
59:      C LOOP AND READ UNTIL END OF EVENT
60:      2 READ(15,100,END=500)NAM,TIME,IFMT,DRTM
61:      100  FORMAT(2A2,F7.2,A2,F7.2)
62:      C IF NAME = "88", IT IS THE END OF THE EVENT
63:      IF(NAM(1).EQ.IDOL)GOTO 500
64:      DO 1 J=1,32
65:      IF(ISTA(1,J).NE.NAM(1))GOTO 1
66:      IF(ISTA(2,J).NE.NAM(2))GOTO 1
67:      C TAKE FIRST OCCURENCE ONLY
68:      IF(IDONE(J).EQ.1)GOTO 1
69:      C LOAD UP OPARM TABLE
70:      PTIME(J)=TIME
71:      FNP(J)=DRTM
72:      IF(FNP(J).NE.0.)FTIME(J)=PTIME(J)+FNP(J)
73:      IFNOT(J)=IFMT
74:      IDONE(J)=1
75:      GOTO 2
76:      1      CONTINUE
77:      GOTO 2
78:      C COME HERE IF NO FILE
79:      500 CONTINUE
80:      C RETURN
81:      C COME HERE WHEN EOF
82:      502 CLOSE 15
83:      RETURN
84:      C
85:      20      CONTINUE
86:      22      READ(15,200,END=502)NAM,IPR,IFMT,IG,JYR,JNO,
87:      *JDA,JHR,JHM,TIME,IPA,ST,ISR,ISG,ISA,KRMK,CC,JRMK,DRTM
88:      200  FORMAT(3A2,A1,I1,1X,5I2,F5.2,15,2X,F5.2,A2,1X,I1,4X,I3,
89:      *15X,A2,A1,F5.2,A1,F4.0,5X)
90:      GOTO 24
91:      30      CONTINUE
92:      31      READ(15,300,END=502)IREC
93:      300  FORMAT(45A2)
94:      DECODE(IREC,301,ERR=502)NAM,IPR,IFMT,IG,JYR,JNO,
95:      *JDA,JHR,JHM,TIME,PRZ,PVV,ST,ISR,ISG,SRZ,ISA,SWW,
96:      *PD,SD,EPI,ST,ANGLE,CC,JRMK,DRTM,AZ,DMAG,ANAG,PIMP,SIMP
97:      301  FORMAT(3A2,A1,I1,1X,5I2,F5.2,F4.2,F3.2,F5.2,A2,1X,I1,
98:      *F4.2,F3.0,F3.2,2F4.2,F4.1,F3.0,F5.2,A1,F4.0,
99:      *F3.0,2F2.1,2F4.3)
100:      24      CONTINUE
101:      DO 23 J=1,32
102:      C COMMON SECTION FOR HYPOT AND FZLINK
103:      IF(ISTA(1,J).NE.NAM(1))GOTO 23
104:      IF(ISTA(2,J).NE.NAM(2))GOTO 23
105:      PTIME(J)=TIME-RSEC
106:      IF(PTIME(J).LT.0)PTIME(J)=PTIME(J)+60.
107:      FNP(J)=DRTM
108:      IFNOT(J)=IFMT
109:      STIME(J)=ST-RSEC
110:      IF(STIME(J).LT.PTIME(J).AND.ST.NE.0.)
111:      *STIME(J)=STIME(J)+60.
112:      IF(FNP(J).NE.0.)FTIME(J)=PTIME(J)+FNP(J)
113:      CROSS(J)=0.
114:      PCROSS(J)=0.
115:      IGRADE(J)=IG
116:      SGRADE(J)=ISG

```

```

117:      SAMP(J)=ISA
118:      RNK(3,J)=JRNK
119:      SRNK(J)=ISR
120:      PRNK(J)=IPR
121:      C IF NOT FZLINK,SKIP NEXT SECTION
122:      IF(ICC.NE.2)GOTO 25
123:      ANP(J)=IPA
124:      RNK(1,J)=KRNK(1)
125:      RNK(2,J)=KRNK(2)
126:      C IF NOT HYPDUT,SKIP NEXT SECTION
127:      25 IF(ICC.NE.3)GOTO 26
128:      PRES(J)=PRZ
129:      SRES(J)=SRZ
130:      EPD(J)=EPIDST
131:      CLK(J)=CC
132:      GOTO 26
133:      23 CONTINUE
134:      C COME HERE AFTER SUCCESSFUL MATCH
135:      26 CONTINUE
136:      GOTO(502,22,31),ICC
137:      C
138:
139:      END

```

INFREE.FR

```

1:      SUBROUTINE XINFREE(L,R,NC)
2:      OVERLAY OINFREE
3:      INTEGER P,S,D
4:      DIMENSION R(1),L(1),LL(96)
5:      DECODE(L,440)(LL(J),J=1,96)
6:      440  FORMAT(96R1)
7:      MINUS=45
8:      NZER=48
9:      N9=57
10:     NPERI=46
11:     I=0
12:     J=0
13:     6    J=J+1
14:     N=0
15:     P=0
16:     S=0
17:     M=1
18:     F=1.
19:     5    I=I+1
20:     IF(I.GT.NC)RETURN
21:     D=LL(I)-8192
22:     K=4
23:     IF(D.EQ.MINUS)K=2
24:     IF(D.GE.NZER.AND.D.LE.N9)K=1
25:     IF(D.EQ.NPERI)K=3
26:     K=K+S
27:     GOTO(1,2,3,5,1,4,3,4)K
28:     1    N=N*10+D-NZER
29:     S=4
30:     GO TO 5
31:     2    M=-1
32:     S=4
33:     GO TO 5
34:     3    P=I
35:     S=4
36:     GO TO 5
37:     4    IF(P.NE.0)F=10.**(-P-1)
38:     R(J)=N/F*M
39:     GO TO 6
40:     END
41:

```


ISDS7.FR

```

1:  C      PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
2:  C      --WRITTEN BY PETER R. STEVENSON (1976-78)--
3:  C      --U.S.GEOLOGICAL SURVEY--
4:  C      --OFFICE OF EARTHQUAKE STUDIES--
5:  C      --MENLO PARK,CALIF.--
6:      COMMON /TVTUNE/ITV,ITX
7:      COMMON /RPLAY/VFILE
8:      INTEGER VFILE
9:      COMMON/SFL/SFLAG
10:     INTEGER SFLAG
11:     COMMON /ZER/IZER
12:     COMMON/ZZZ/IZIT
13:     COMMON /DEC1/IDF,DECMAX,DLIN
14:     INCLUDE 'C$GPARM'
15:  C      MODIFIED 5/12/78 FOR 3-PASSES
16:     PARAMETER LSZ1=13
17:     PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
18:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
19:     * ,UNQYR,UNQID,TCORR,FCA,LCA
20:     COMMON /JDATE/IHR,INN,TSEC
21:     COMMON /XXX/XX(3),LIST(LSZ)
22:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
23:     COMMON /VEE/VSAVE(4,MAXPASS)
24:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
25:  C
26:     COMMON /CPARM/HSTA,ISTA(2,32)
27:     INCLUDE 'C$APARM'
28:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
29:     INCLUDE 'C$BPARM'
30:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,MT,DTG,SG
31:  C
32:     COMMON /SCAL/ISDUM(7)
33:     INCLUDE 'C$BLAN'
34:     PARAMETER NAXX=1000
35:     COMMON A(NAXX),MAXPTS,NREQ,IREQ(32)
36:     INTEGER A
37:     INCLUDE 'C$PICK'
38:     INTEGER PRNK,SRNK,RNK,SGRADE
39:     COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
40:     *CROSS(32),SNP(32),FNP(32),PCROSS(32),
41:     *IFNOT(32),IGRADE(32),SGRADE(32),
42:     *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
43:     *PRES(32),SRES(32),EPD(32),CLK(32)
44:     COMMON /EPARM/NTMP,ITEMP(32),OTZARR,OTWIND
45:     COMMON /SHCON/ISM
46:     COMMON /IOPRM/IODUM(6)
47:     COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
48:     INTEGER OLDEVT,PUFLG,PUBLNK
49:     DIMENSION MMN(20)/2HFA,2HZL,2HST,17*0/
50:     COMMON /MISC/MDUM(4)
51:  C      LOAD PHASE FILE NAME WITH DEFAULT 'FAZLST'
52:     DO 10 J=1,20
53: 10      NFIL(J)=MMN(J)
54:     IZER=0
55:     IZIT=0
56:     ITV=4

```

```

57:      ITX=5
58:      OPEN 4, '$TT01'
59:      OPEN 5, '$TT11'
60:      OPEN 3, 'JTEK'
61:      VFILE=-1
62:      OPEN 9, 'TEMP'
63:      CALL OVOPH('ISDS.OL',IER)
64:      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER
65:      CALL INITT(0)
66:      CALL SETBUF(3)
67:      CALL CHRSTZ(4)
68:      CALL TSEND
69:      CALL HITEK(1,0)
70:      CALL NCAM1
71:      CALL DIRC1
72:      CALL DEFAULT
73:      CALL SPLOT
74:      STOP
75:      END

```

LBSCL.FR

```

1:      SUBROUTINE XLBSCL
2:      OVERLAY OLBSCL
3:      INCLUDE 'CSBLAN'
4:      PARAMETER MAXX=1000
5:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
6:      INTEGER A
7:      INCLUDE 'CSAPARM'
8:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),O(32)
9:      INCLUDE 'CSBPARM'
10:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
11:
12:     C
13:     COMMON/SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
14:     COMMON/SFL/SFLAG
15:     INTEGER SFLAG
16:     DIMENSION AA(9)
17:     C FIND UPPER LEFT CORNER
18:     PLACE=AL
19:     XX=TZARR+TWIND
20:     C VOLT IS VOLTS INTO A/D
21:     VOLT=5.
22:     C DVCMM IS DEVELOCORDER FULL-SCALE MILLIMETER EQUIVALENTS
23:     DVCMM=160.
24:     C
25:     C DRAW TICS ON RIGHT HAND SIDE,STARTING AT UPPER RIGHT
26:     C HAND CORNER AND MOVING DOWN ARNG UNITS FOR EACH TIC
27:     C
28:     1      CALL MOVEA(XX,PLACE)
29:     JL=15
30:     CALL DR*REL(JL,0)
31:     CALL DR*REL(-JL,0)
32:     PLACE=PLACE-ARNG
33:     IF(PLACE.LT.AZ)GOTO 2
34:     GOTO 1
35:     C
36:     C FIND MIDDLE OF FIRST INTERVAL
37:     2      FIRST=AL-.5*ARNG
38:     C LABEL MIDDLE OF EACH INTERVAL WITH APPROPRIATE
39:     C SCALE FACTOR
40:     C SFLAG INDICATES WHICH SCALING OPTION TO USE
41:     C SFLAG=1 FOR DIGITAL COUNTS
42:     C SFLAG=2 FOR VOLTS OUT
43:     C SFLAG=3 FOR EQUIVALENT DEVELOCORDER MM'S OUT
44:     C
45:     J*=5
46:     DO 3 J=1,NREQ
47:     JJ=IREQ(J)
48:     C
49:     GOTO (10,20,30),SFLAG
50:     10     ISCL=ARNG/S(JJ)
51:     ENCODE(AA,100)ISCL
52:     GOTO 40
53:     C
54:     20     TSCL=VOLT/S(JJ)
55:     ENCODE(AA,101)TSCL
56:     GOTO 40
57:     C
58:     30     TSCL=DVCMM/S(JJ)
59:     ENCODE(AA,102)TSCL
60:     GOTO 40

```

```

60:  C
61:  40  CALL MOVEA(XX,FIRST)
62:      CALL MOVREL(JW,0)
63:      CALL AOUTST(9,AA)
64:  3    FIRST=FIRST-ARNG
65:  C
66:  100  FORMAT(I7," ")
67:  101  :FORMAT(F8.3,"V")
68:  102  FORMAT(F7.1,"MM")
69:  C
70:      RETURN
71:      END

```

```

60: C
61: C PUT WINDOW START TIME ABOVE UPPER RIGHT CORNER
62: C
63: KYY=KY+LY+20
64: CALL MOVABS(KX,KYY)
65: ENCODE(MTIME,100)IHR,IMN,TSEC
66: 100 FORMAT(2I2,2X,F6.2)
67: CALL AOUTST(12,MTIME)
68: C PUT WINDOW END TIME ABOVE UPPER LEFT CORNER
69: C
70: TLAST=TZARR+TWIND+RSEC
71: C CONVERT TO PROPER TIME
72: JJJ=TLAST/60
73: TLAST=TLAST-JJJ*60
74: LMN=IMN+JJJ
75: KKK=LMN/60
76: LMN=LMN-KKK*60
77: LHR=IHR+KKK
78: C
79: KXX=KX+LX-70
80: KYY=KY+LY+20
81: CALL MOVABS(KXX,KYY)
82: ENCODE(MTIME,100)LHR,LMN,TLAST
83: CALL AOUTST(12,MTIME)
84: C
85: C LABEL TIME AXIS
86: 2 MXX=LX/2+KX
87: MYY=KY+10
88: CALL MOVABS(MXX,MY Y)
89: CALL ANSTR(7,MSG)
90: RETURN
91: END

```

LBSTA.FR

```

1:      SUBROUTINE XLBSTA(KC)
2:      OVERLAY OLBSTA
3:      INCLUDE 'CSBLAN'
4:      PARAMETER MAXX=1000
5:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
6:      INTEGER A
7:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
8:      COMMON /CPARM/NSTA,ISTA(2,32)
9:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
10:     *D(32)
11:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
12:      INCLUDE 'CSGPARM'
13:      C MODIFIED 5/12/78 FOR 3-PASSES
14:      PARAMETER LSZ1=13
15:      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
16:      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
17:      *,UNQYR,UNQID,TCORR,FCA,LCA
18:      COMMON /JDATE/IHR,IMN,TSEC
19:      COMMON /XXX/XX(3),LIST(LSZ)
20:      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
21:      COMMON /VEE/VSAVE(4,MAXPASS)
22:      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
23:      C
24:      DIMENSION MTIME(9),NUM(2)
25:      DIMENSION MSG(7)
26:      DATA MSG/83,69,67,79,78,68,83/
27:      DO 1 K=1,NREQ
28:      JJ=IREQ(K)
29:      Y=D(K)
30:      X=TZARR
31:      C* POSITION BEAM TO APROPRIATE TRACE
32:      CALL MOVEA(X,Y)
33:      CALL MOVREL(10,5)
34:      C* WRITE STATION NAME
35:      I1=ISTA(1,JJ)
36:      I2=ISTA(2,JJ)
37:      CALL AOUTST(2,I1)
38:      CALL AOUTST(2,I2)
39:      C PUT OUT TRACE NUMBER BY TRACE START
40:      CALL MOVEA(X,Y)
41:      CALL MOVREL(-30,-4)
42:      ENCODE(NUM,102)JJ
43:      102 FORMAT(I2)
44:      CALL AOUTST(2,NUM)
45:      C
46:      C PUT OUT SCALE FACTOR
47:      IF(S(JJ).EQ.1)GOTO 1
48:      Y=D(K)
49:      X=TZARR+TWIND
50:      CALL MOVEA(X,Y)
51:      CALL MOVREL(-90,10)
52:      ENCODE(MTIME,101)S(JJ)
53:      101 FORMAT(F8.1,'X')
54:      CALL AOUTST(9,MTIME)
55:      1 CONTINUE
56:      C IF KC=2,SKIP TIME LABELS
57:      IF(KC.EQ.2)GOTO 2
58:      C
59:      C LABEL BOX WITH CURRENT TIME

```

LPICK.FR

```

1:      SUBROUTINE XLPICK
2:      OVERLAY OLPICK
3:      CC SUBROUTINE TO DISPLAY PICKS
4:      C P.R.STEVENSON 8/21/78
5:      COMMON /TVTUNE/ITV,ITX
6:      INCLUDE 'C$BLAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,MREQ,IREQ(32)
9:      INTEGER A
10:     INCLUDE 'C$PICK'
11:     INTEGER PRNK,SRNK,RNK,SGRADE
12:     COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
13:     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
14:     *IFMOT(32),IGRADE(32),SGRADE(32),
15:     *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
16:     *PRES(32),GRES(32),EPD(32),CLK(32)
17:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
18:     *D(32)
19:     COMMON /BPARM/TZARR,TWIND,THR,TNN,AZ,ARNG,AL,MT,DTG,
20:     *SG
21:     COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
22:     COMMON /MISC/ZX,ZY,CX,CY
23:     INTEGER ZX,ZY,CX,CY
24:     DIMENSION MSGP(40)
25:     DATA IBLNK/2H /
26:     TRNG=TZARR+TWIND
27:     COMMON /CPARM/HSTA,ISTA(2,32)
28:     INCLUDE 'C$GPARM'
29:     C MODIFIED 5/12/78 FOR 3-PASSES
30:     PARAMETER LSZ1=13
31:     PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
32:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMM,RSEC,DUR,EVENT,PASS
33:     *,UNQYR,UNQID,TCORR,FCA,LCA
34:     COMMON /JDATE/IHR,IMM,TSEC
35:     COMMON /XXX/XX(3),LIST(LSZ)
36:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
37:     COMMON /VEE/VSAVE(4,MAXPASS)
38:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
39:     C
40:     2 CONTINUE
41:     CC
42:     3 CONTINUE
43:     REWIND 9
44:     C SAVE HEADER ON SCRATCH FILE (L.U. 89)
45:     WRITE(9,903)
46:     903 FORMAT(33HNU. STA P-TIME P-AMPL. S-TIME,3X,7HS-AMPL.,2X,
47:     *3HSMP,5X,3HFMP,2X,4HP-CR,2X,3HFM ,4HPRNK,2X,4HSRNK,2X,1HR)
48:     CALL SETNRG(150,1023)
49:     DO 41 K=1,MREQ
50:     J=IREQ(K)
51:     CC DRAW MARK FOR P PICK
52:     IF(PTIME(J)+SHFT(J).LE.TZARR.OR.PTIME(J)+SHFT(J).GT.TRNG)GOTO 20
53:     CALL PHARK(PTIME(J)+SHFT(J),D(K),80)
54:     C
55:     C DRAW DOTTED LINE FOR P RESIDUAL
56:     PRZ=PRES(J)+PTIME(J)+SHFT(J)

```

```

57:         IF(PRZ.LE.TZARR.OR.PRZ.GT.TRNG.OR.PRES(J).EQ.0.)GOTO 20
58:         CALL PHARK(PRZ,D(K),-82)
59:     C
60: 20      CONTINUE
61:     CC
62:     CC    DRAW MARK FOR S PICK
63:         IF(STIME(J)+SHFT(J).LE.TZARR.OR.STIME(J)+SHFT(J).GT.TRNG)GOTO 30
64:         CALL PHARK(STIME(J)+SHFT(J),D(K),83)
65:     C
66:     C    DRAW DOTTED LINE FOR S RESIDUAL
67:         SRZ=SRES(J)+STIME(J)+SHFT(J)
68:         IF(SRZ.LE.TZARR.OR.SRZ.GT.TRNG.OR.SRES(J).EQ.0.)GOTO 30
69:         CALL PHARK(SRZ,D(K),-114)
70:     C
71: 30      CONTINUE
72:         PCROSS(J)=CROSS(J)-PTIME(J)
73:         SMP(J)=STIME(J)-PTIME(J)
74:         FMP(J)=FTIME(J)-PTIME(J)
75:         IF(PCROSS(J).LT.0.)PCROSS(J)=0.
76:         IF(SMP(J).LT.0..OR.STIME(J).LT.0.)SMP(J)=0.
77:         IF(FMP(J).LT.0.)FMP(J)=0.
78:     C    DRAW MARK FOR F PICK
79:         IF(FTIME(J)+SHFT(J).LE.TZARR.OR.FTIME(J)+SHFT(J).GT.TRNG)GOTO 10
80:         IF(FMP(J).GT.0.)CALL PHARK(FTIME(J)+SHFT(J),D(K),70)
81: 10      CONTINUE
82:     C    PRETTY UP NULL S-TIMES
83:         IF(STIME(J).LT.0..AND.PTIME(J) GE.0.)STIME(J)=0.
84:     CC
85:         CALL NEWLIN
86:         IXX=0
87:         IYY=750*ITERM
88:         CALL MOVABS( IXX,IYY)
89:         CALL ANMODE
90:         CALL TSEND
91:         PT=PTIME(J)+RSEC
92:         ST=STIME(J)+RSEC
93:         IF(PTIME(J).EQ.0.)PT=0.
94:         IF(STIME(J).EQ.0.)ST=0.
95:         IF(ISTA(1,J).EQ.0)ISTA(1,J)=IBLNK
96:         IF(ISTA(2,J).EQ.0)ISTA(2,J)=IBLNK
97:         IF(ST.EQ.0.)SGRADE(J)=4
98:         IF(ST.EQ.0.)SRMK(J)=IBLNK
99:         IF(PT.EQ.0.)IGRADE(J)=4
100:        IF(PT.EQ.0.)PRMK(J)=IBLNK
101:     C    SAVE THE JTH ENTRY ON SCRATCH FILE
102:         WRITE(9,900)J,ISTA(1,J),ISTA(2,J),PT,AMP(J),ST,SAMP(J),SMP(J),
103:         *FMP(J),PCROSS(J),IFHOT(J),PRMK(J),IGRADE(J),SRMK(J),SGRADE(J),RMK(3,J)
104: 900      FORMAT(13,1X,2A2,1X,F7.2,2X,F7.2,2X,F7.2,2X,3F7.2,2X,F3.2,2X,A2,
105:         *2X,A2,11,3X,A2,11,2X,A1)
106: 41      CONTINUE
107:     CC
108:     C    READ BACK INFO FROM SCRATCH FILE
109:         REWIND 9
110:         READ(9,905)MSGP
111: 905      FORMAT(40A2)
112:     C    OUTPUT HEADER TO TEK SCREEN
113:         CALL AOUTST(80,MSGP)
114:         DO 44 K=1,NREQ
115:             J=IREQ(K)
116:             READ(9,905)MSGP

```



```
117: 921  CONTINUE
118: C  SKIP IF BOTH P AND S ARE ZERO
119:     IF(PTIME(J).LE.0..AND.STIME(J).LE.0.)GO TO 44
120:     CALL HEVLIN
121: C  OUTPUT ENTRY TO TEK SCREEN
122:     CALL ADUTST(80,MSGP)
123: 44  CONTINUE
124: CC
125: C  SAVE BEAM POSITION
126:     CALL HEVLIN
127:     CALL SEELOC(ZX,ZY)
128:     CALL SETHRG(0,1023)
129:     RETURN
130:     END
```

MENU.FR

```

1:      SUBROUTINE XMENU(JSW)
2:      OVERLAY OMENU
3:      C - - - P.R.STEVENSOM (10/23/78) - - -
4:      COMMON /TVTUNE/ITV,ITX
5:      GOTO (1,2),JSW
6:      1    CONTINUE
7:      WRITE(ITV,700)
8:      700  FORMAT(' COMMAND FUNCTION',/, ' -----')
9:      WRITE(ITV,701)
10:     701  FORMAT(' EV',7X,'SELECT EVENT',/, ' PA',7X,'SELECT PASS',
11:     */, ' EVPL',5X,'SELECT AND PLOT EVENT',
12:     */, ' TR',7X,'SELECT TRACES',/, ' DE',7X,'DEFINE TIME WINDOW',
13:     */, ' PL',7X,'PLOT TRACES',
14:     */, ' SESC',5X,'SELECT SCALE UNITS OPTION',
15:     */, ' PLDE',5X,'PLOT DECINATED',/, ' SC',7X,'SCALE TRACES',
16:     */, ' PI',7X,'PICK TRACES',
17:     */, ' SA',7X,'SAVE PICKS AS PHASE CARDS',
18:     */, ' SH',7X,'ALIGN TRACES ON P-',
19:     *'TIMES',/, ' SH-S',5X,'ALIGN TRACES ON S-',
20:     *'TIMES',/, ' ZERO',5X,'REMOVE DC LEVEL',/, ' NE',7X,'PRINT MENU',
21:     */, ' UNZE',5X,'DO NOT REMOVE DC LEVEL',
22:     */, ' LDCO',5X,'LOAD COMMANDS FROM REMOTE FILE'
23:     */, ' UNCO',5X,'DISABLE REMOTE(LDCO) OPTION'
24:     */, ' COPY',5X,'MAKE HARD COPY',
25:     */, ' PAUSE',4X,'TEMPORARILY RETURN TO MANUAL CONTROL',
26:     */, ' RESU',5X,'RESUME CONTROL WITH REMOTE FILE',
27:     */, ' ?',8X,'PRINT PICK MENU',
28:     */, ' EX',7X,'EXIT PROGRAM')
29:      RETURN
30:      2    CONTINUE
31:      WRITE(ITV,800)
32:      900  FORMAT(' KEY FUNCTION',/, ' ---')
33:      WRITE(ITV,801)
34:     801  FORMAT(' P',4X,'PICK P-TIME',/,
35:     *' S',4X,'PICK S-TIME',/, ' F',4X,'PICK END OF CODA',/,
36:     *' C',4X,'PICK FIRST ZERO-CROSSING',/,
37:     *' A',4X,'MEASURE P-AMPLITUDE(PEAK TO PEAK',/,
38:     *' ',4X,'USE SPACE-BAR TO MEASURE THE PEAKS)',/,
39:     *' Q',4X,'MEASURE S-AMPLITUDE(PEAK TO PEAK',/,
40:     *' ',4X,'USE SPACE-BAR TO MEASURE THE PEAKS)',/,
41:     *' X',4X,'REZERO P,S,F',/,
42:     *' R',4X,'ENTER 3 CHAR. REMARK',/,
43:     *' ',4X,'ENTER 1 CHAR. STATION REMARK(COL.71)',/,
44:     *' 0-4',2X,'GRADE OF P-PICK',/, ' 5-9',2X,'GRADE(+5) OF S-PICK',/
45:     *' E,I',2X,'EMERGENT OR IMPULSIVE',
46:     *' P-ARRIVAL',/, ' +,U',2X,'UP FIRST NOTION',/,
47:     *' -,D',2X,'DOWN FIRST NOTION',/,
48:     *' T',4X,'ENTER CLOCK CORRECTION ',/,
49:     *' ',4X,'ENTER FMP VALUE (CURSOR SELECTS STATION)',/,
50:     *' J,K',2X,'EMERGENT OR IMPULSIVE S-ARRIVAL',/,
51:     *' L',4X,'LIST PICKED VALUES',/, ' G',4X,'GET AUTOMATIC PICKS',/,
52:     *' Y',4X,'GET PHASE LIST',/,
53:     *' H',4X,'GET HYPOT FILE',/,
54:     *' ',4X,'SPEEDY PICK OPTION',/,
55:     *8X,'ENTER 4 CHAR. STRING(PRNK OR SRNK)',/,
56:     *8X,'CROSS-HAIR DETERMINES POSITION OF',/,

```

```
37:      *8X,'P OR 8 PICKS.',/,
38:      *' ',4X,'EXIT PICK ROUTINE')
39:      RETURN
60:      END
```

NIS097.FR

```

11 C PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
12 C --WRITTEN BY PETER R. STEVENSON (1976-78)--
13 C --U.S.GEOLOGICAL SURVEY--
14 C --OFFICE OF EARTHQUAKE STUDIES--
15 C --MENLO PARK,CALIF.--
16 COMMON /TVTUNE/ITV,ITX
17 COMMON /RPLAY/VFILE
18 INTEGER VFILE
19 COMMON/SFL/SFLAG
20 INTEGER SFLAG
21 COMMON /ZER/IZER
22 COMMON/ZZZ/IZIT
23 COMMON /DECI/IDF,DECNAX,DLIN
24 INCLUDE 'C&GPARN'
25 C MODIFIED 5/12/78 FOR 3-PASSES
26 PARAMETER LSZ1=13
27 PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
28 COMMON /KDATE/KYR,KNO,KDA,KHR,KNN,RSEC,DUR,EVENT,PASS
29 *,UNQYR,UNQID,TCORR,FCA,LCA
30 COMMON /JDATE/IHR,IMN,TSEC
31 COMMON /XXX/XX(3),LIST(LSZ)
32 EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
33 COMMON /VEE/VSAVE(4,MAXPASS)
34 INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
35 C
36 COMMON /CPARN/NSTA,ISTA(2,32)
37 INCLUDE 'C&APARN'
38 COMMON /APARN/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
39 INCLUDE 'C&BPARN'
40 COMMON /BPARN/TZARR,TWIND,TWR,TNN,AZ,ARNG,AL,NT,DTG,SG
41 C
42 COMMON /SCAL/ISDUM(7)
43 INCLUDE 'C&BLAN'
44 PARAMETER MAXX=1000
45 COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
46 INTEGER A
47 INCLUDE 'C&PICK'
48 INTEGER PRNK,SRNK,RNK,SGRADE
49 COMMON /DPARN/ PTINE(32),ANP(32),STINE(32),FTIME(32),
50 *CROSS(32),SNP(32),FNP(32),PCROSS(32),
51 *IFNOT(32),IGRADE(32),SGRADE(32),
52 *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
53 *PRES(32),SRES(32),EPD(32),CLK(32)
54 COMMON /EPARN/NTEMP,ITENP(32),OTZARR,OTWIND
55 COMMON /SHCON/ISH
56 COMMON /IOPRN/IODUM(6)
57 COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
58 INTEGER OLDEVT,PUFLG,PUBLNK
59 DIMENSION MNN(20)/2HFA,2HZL,2HST,17*0/
60 COMMON /MISC/MDUM(4)
61 C LOAD PHASE FILE NAME WITH DEFAULT 'FAZLST'
62 DO 10 J=1,20
63 10 NFIL(J)=MNN(J)
64 IZER=0
65 IZIT=0
66 ITV=4

```

```

57:         ITX=5
58:         OPEN 4, '$TT01'
59:         OPEN 5, '$TTI1'
60:         OPEN 3, 'JTEK'
61:         VFILE=-1
62:         OPEN 9, 'TEMP'
63:         CALL OVOPN('MISDS.OL', IER)
64:         IF(IER.NE.1) TYPE ' OVERLAY: IER=', IER
65:         CALL INITT(0)
66:         CALL SETBUF(3)
67:         CALL CHRISZ(4)
68:         CALL TSEND
69:         CALL NITEK(1,0)
70:         CALL NCAM1
71:         CALL DIRC1
72:         CALL DEFAULT
73:         CALL SPLOT
74:         STOP
75:         END

```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:23:44 PM

NOSHF.FR

```
1:      SUBROUTINE XNOSHF
2:      OVERLAY ONOSHF
3:      INCLUDE 'CSAPARM'
4:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
5:      INCLUDE 'CS9LAN'
6:      PARAMETER MAXX=1000
7:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
8:      INTEGER A
9:      DO 1 K=1,NREQ
10:     J=IREQ(K)
11:     SHFT(J)=0.
12:     1  CONTINUE
13:     DO 2 K=1,32
14:     2  Z(K)=0.
15:     RETURN
16:     END
17:
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, OCTOBER 23, 1978 2:20:17 PM

PICK.FR

```
1:      SUBROUTINE XPICK
2:      OVERLAY OPICK
3:      CC SUBROUTINE TO PICK TRACES FROM ARRAY PLOT
4:      C - - - P.R. STEVENSON (8/22/78) - - -
5:      COMMON /TVTUNE/ITV,ITX
6:      INCLUDE 'C$BLAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,MREQ,IREQ(32)
9:      INTEGER A
10:     COMMON /APARN/S(32),Z(32),DT(32),TZ(32),SHFT(32),
11:     *D(32)
12:     COMMON /BPARN/TZARR,TWIND,THR,TMH,AZ,ARNG,AL,NT,DTG,
13:     *SG
14:     COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
15:     COMMON /MISC/ZX,ZY,CX,CY
16:     INTEGER ZX,ZY,CX,CY
17:     INCLUDE 'C$PICK'
18:     INTEGER PRNK,SRNK,RMK,SGRADE
19:     COMMON /DPARN/ PTIME(32),AMP(32),STIME(32),FTIME(32),
20:     *CROSS(32),SNP(32),FNP(32),PCROSS(32),
21:     *IFMOT(32),IGRADE(32),SGRADE(32),
22:     *SAMP(32),RMK(3,32),SRMK(32),PRNK(32),
23:     *PRES(32),SRES(32),EPD(32),CLK(32)
24:     COMMON/SFL/SFLAG
25:     INTEGER SFLAG
26:     DIMENSION NFM(9),ITEMP(4),ASCAL(3),JBUF(11)
27:     DATA NFM/2H+ ,2H- ,2H ,2HD ,2HU ,2HEP,2HIP,2HES,2HIS/
28:     C SCALE FACTORS FOR VARIOUS SCALE SELECT OPTIONS
29:     DATA ASCAL/1.,.2048,6.4/
30:
31:     CC
32:     3 CONTINUE
33:     4 CONTINUE
34:     CALL VCURSR(IB,TP,AP)
35:     CC IDENTIFY TRACE FROM AP
36:     ALST=AL
37:     KK=1
38:     DO 40 K=1,MREQ
39:     ADIST=ABS(AP-D(K))
40:     IF(ADIST.LT.ALST)KK=K
41:     IF(ADIST.LT.ALST)ALST=ADIST
42:     40 CONTINUE
43:     C KK IS RELATIVE INDEX
44:     C J IS ABSOLUTE INDEX
45:     J=IREQ(KK)
46:     DZR=ADIST/S(J)
47:     C* R=ENTER 3 CHAR REMARK
48:     IF(IB.HE.82)GOTO 12
49:     CALL AINST(3,ITEMP)
50:     RMK(1,J)=ITEMP(1)
51:     RMK(2,J)=ITEMP(2)
52:     12 CONTINUE
53:     C* **STATION REMARK IN COL 71(WARNING**NOT HYP071 STANDARD)
54:     IF(IB.HE.42)GOTO 13
55:     CALL AINST(1,ITEMP)
56:     RMK(3,J)=ITEMP(1)
```

```

57: 13 CONTINUE
58: C* E=EMERGENT P-ARRIVAL
59: C* I=IMPULSIVE P-ARRIVAL
60: IF(ID.EQ.69)PRMK(J)=MFM(6)
61: IF(ID.EQ.73)PRMK(J)=MFM(7)
62: C* J=EMERGENT S-ARRIVAL
63: C* K=IMPULSIVE S-ARRIVAL
64: IF(ID.EQ.74)SRMK(J)=MFM(8)
65: IF(ID.EQ.75)SRMK(J)=MFM(9)
66: C* C=CROSS
67: IF(ID.EQ.67)CROSS(J)=TP-SHFT(J)
68: C* P=PTIME
69: IF(ID.EQ.80)PTIME(J)=TP-SHFT(J)
70: C* A=AMPLITUDE
71: IF(ID.NE.65)GOTO 10
72: CALL VCURSR(100,TP1,AP1)
73: CALL VCURSR(100,TP2,AP2)
74: C ONLY SPACE-BAR CAN PICK PEAKS
75: IF(100.NE.32.OR.100.NE.32)GOTO 10
76: AMP(J)=(ABS(AP1-AP2))/(S(J)*ASCAL(SFLAG))
77: 10 CONTINUE
78: C* Q=S-AMPLITUDE
79: IF(ID.NE.81)GOTO 11
80: CALL VCURSR(100,TP1,AP1)
81: CALL VCURSR(100,TP2,AP2)
82: C ONLY SPACE-BAR CAN PICK PEAKS
83: IF(100.NE.32.OR.100.NE.32)GOTO 11
84: SAMP(J)=(ABS(AP1-AP2))/(S(J)*ASCAL(SFLAG))
85: 11 CONTINUE
86: C* S=STIME
87: IF(ID.EQ.83)STIME(J)=TP-SHFT(J)
88: C* F=TIME OF CODA DIE OUT
89: IF(ID.EQ.70)FTIME(J)=TP-SHFT(J)
90: C* Y=GET PICKS FROM PHASE LIST
91: IF(ID.EQ.89)CALL DEFAULT
92: IF(ID.EQ.89)CALL GPICK(2)
93: IF(ID.EQ.89)CALL LPICK
94: C
95: C* H=GET PICKS AND RESIDUALS FROM HYPOT FOR DISPLAY
96: IF(ID.EQ.72)CALL DEFAULT
97: IF(ID.EQ.72)CALL GPICK(3)
98: IF(ID.EQ.72)CALL LPICK
99: C* G=GET AUTO PICKS FOR DISPLAY
100: IF(ID.EQ.71)CALL DEFAULT
101: IF(ID.EQ.71)CALL GPICK(1)
102: IF(ID.EQ.71)CALL LPICK
103: C* NUMBER=ENTER GRADE OF PICK
104: C* 0-4 ENTER GRADE OF P PICK
105: IF(ID.GE.48.AND.ID.LE.52)IGRADE(J)=10-48
106: C* 5-9 ENTER GRADE OF S PICK(GRADE IS NUMBER-5)
107: IF(ID.GE.54.AND.ID.LE.57)SGRADE(J)=10-53
108: C* L=LOOK AT VALUES
109: IF(ID.EQ.76)CALL LPICK
110: C* .-FINISHED WITH THIS ARRAY
111: IF(ID.EQ.46)GO TO 7
112: C
113: C* :=SPEEDY PICK MODE
114: C FOLLOWING ";",ENTER PRMK OR SRMK (4 CHAR. STRING).
115: C THE POSITION OF THE CROSS-HAIR DETERMINES THE
116: C PICK. OTHER ELEMENTS ARE DECODED AND STORED IN

```



```

117: C THE APPROPRIATE VARIABLES IN DPARN.
118: C E.G. "1", <CR>, IPU1 OR ES 2, <CR>.
119: C
120: IF (IB.NE.59)GOTO 14
121: C PRELOAD TEMP TO TAKE CARE OF DEFAULT CASE:
122: C IE. 3 BLANKS AND AN ASCII ZERO
123: C
124: ITEMP(1)=32
125: ITEMP(2)=32
126: ITEMP(3)=32
127: ITEMP(4)=48
128: CALL TIMSTR(4,ITEMP)
129: IF (ITEMP(2).NE.80)GOTO 15
130: PTIME(J)=TP-SHFT(J)
131: C ASSIGN P-REMARK AND P 1ST MOTION
132: IF (ITEMP(1).EQ.69)PRNK(J)=NFM(6)
133: IF (ITEMP(1).EQ.73)PRNK(J)=NFM(7)
134: IF (ITEMP(3).EQ.85)IFNOT(J)=NFM(5)
135: IF (ITEMP(3).EQ.68)IFNOT(J)=NFM(4)
136: IGRADE(J)=ITEMP(4)-48
137: 15 IF (ITEMP(2).NE.83)GOTO 14
138: C ASSIGN S-REMARK
139: STIME(J)=TP-SHFT(J)
140: IF (ITEMP(1).EQ.69)SRNK(J)=NFM(8)
141: IF (ITEMP(1).EQ.73)SRNK(J)=NFM(9)
142: SGRADE(J)=ITEMP(4)-48
143: 14 CONTINUE
144: C* )=ENTER FMP MANUALLY
145: C HORIZONTAL CURSOR SELECTS STATION
146: IF (IB.NE.62)GOTO 16
147: CALL AINST(20,JBUF)
148: CALL INFREE(JBUF,F,20)
149: FMP(J)=F
150: FTIME(J)=PTIME(J)+FMP(J)
151: F=0.
152: 16 CONTINUE
153: C
154: C* T=ENTER CLOCK CORRECTION MANUALLY
155: C HORIZONTAL CURSOR SELECTS STATION
156: IF (IB.NE.84)GOTO 17
157: CALL AINST(20,JBUF)
158: CALL INFREE(JBUF,T,20)
159: C CLOCK CORRECTION IS A GLOBAL FUNCTION
160: DO 5 JJ=1,32
161: 5 CLK(JJ)=T
162: T=0.
163: 17 CONTINUE
164: C
165: C* U,D,+,- = FIRST MOTION
166: C
167: IF (IB.EQ.43)IFNOT(J)=NFM(1)
168: IF (IB.EQ.45)IFNOT(J)=NFM(2)
169: IF (IB.EQ.85)IFNOT(J)=NFM(5)
170: IF (IB.EQ.68)IFNOT(J)=NFM(4)
171: CC
172: C X=REZERO PICKS
173: IF (IB.EQ.88)PTIME(J)=0.
174: IF (IB.EQ.88)STIME(J)=0.
175: IF (IB.EQ.88)FTIME(J)=0.
176: GO TO 4

```

177: 7 CONTINUE
178: RETURN
179: END

PLPRC.FR

```

1:      SUBROUTINE XPLPRC(JC)
2:      OVERLAY OPLPRC
3:      COMMON/SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
4:      COMMON /DECI/IDF,DECMAX,DLIM
5:      COMMON /MISC/ZX,ZY,CX,CY
6:      INCLUDE 'CSGPARM'
7:      C   MODIFIED 5/12/78 FOR 3-PASSES
8:      PARAMETER LSZ1=13
9:      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
10:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
11:     *,UNQYR,UNQID,TCORR,FCA,LCA
12:     COMMON /JOATE/IHR,IMN,TSEC
13:     COMMON /XXX/XX(3),LIST(LSZ)
14:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
15:     COMMON /VEE/VSAVE(4,MAXPASS)
16:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
17:     C
18:     DIMENSION NAM(40),MSG(22)
19:     INTEGER ZX,ZY,CX,CY
20:     COMMON /ZER/IZER
21:     COMMON/SFL/SFLAG
22:     INTEGER SFLAG
23:     C   RESET SHIFTS TO ZERO
24:     CALL NOSHF
25:     C   RESET DISTANCE
26:     CALL RSETD
27:     CALL ERASE
28:     C   DECIMATE IF REQUESTED
29:     IF(JC.EQ.1)CALL DECIM
30:     C   CHECK FOR ZERO-LEVEL FLAG
31:     IF(IZER.EQ.1)CALL ZLEV
32:     C   PLOT TRACES
33:     CALL ARRAY
34:     C   LABEL TRACES
35:     CALL LHSTA(1)
36:     C   PUT EVENT NAME ON PLOT
37:     CALL ENAME(NAM,EVENT)
38:     LHY=600
39:     MZZ=LHY+KY+40
40:     CALL MOVABS(358,MZZ)
41:     CALL AOUTST(40,NAM)
42:     C
43:     C   PUT EVENT # AND PASS # ON THE PLOT
44:     ENCODE(MSG,100)EVENT,PASS
45:     100  FORMAT(' EVENT',I3,' PASS',I3)
46:     CALL AOUTST(17,MSG)
47:     C   PUT DECIMATION FACTOR ON SCREEN IF NOT 1
48:     ENCODE(MSG,101)IDF
49:     101  FORMAT(' DECIMATION FACTOR=',I3)
50:     CALL MOVREL(35,0)
51:     IF(IDF.NE.1)CALL AOUTST(22,MSG)
52:     C
53:     C   PUT SCALE UNITS ON RIGHT SIDE OF PLOT
54:     CALL LBSCL
55:     C
56:     CALL HOME
57:     C   UPDATE BEAM POINTERS
58:     CALL SEELOC(CX,CY)
59:     ZX=0

```

```
60:      ZY=750*ITEM
61:  C   RESTORE DECIMATION FACTOR
62:      IOF=1
63:      RETURN
64:      END
```

PMARK.FR

```
1:      SUBROUTINE XPMARK(X,Y,IA)
2:      OVERLAY OPMARK
3:      CC
4:      C   P.R.STEVENSON 8/21/78
5:      C
6:      CC   PLOT VERTICAL LINE AT (X,Y) AND
7:      CC   LABEL WITH ADE CHARACTER IA
8:      CC
9:      CALL MOVEA(X,Y)
10:     CALL MOVREL(0,-20)
11:     C   IF IA > 0 , DRAW VERTICAL SOLID LINE
12:     IF(IA.LT.0)GOTO 1
13:     CALL DRWREL(0,40)
14:     CALL MOVREL(-5,0)
15:     CALL ANCHO(IA)
16:     GOTO 2
17:     C   ELSE DRAW VERTICAL DOTTED LINE
18:     1   CALL DSHREL(0,40,1)
19:     CALL MOVREL(-5,0)
20:     CALL ANCHO(-IA)
21:     2   CALL MOVEA(X,Y)
22:     RETURN
23:     END
24:
```

PSHIFT.FR

```

11      SUBROUTINE XPSHIFT(ISH)
12      OVERLAY OPSHIFT
13      CC SHIFT PROCESSOR
14      C - - - P.R.STEVENSON (8/22/78) - - -
15      COMMON /TVTUNE/ITV,ITX
16      INCLUDE 'CSBLAN'
17      PARAMETER MAXX=1000
18      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
19      INTEGER A
20      INCLUDE 'CSAPARM'
21      COMMON /APARM/S(32),Z(32),OT(32),TZ(32),SHFT(32),D(32)
22      INCLUDE 'CSBPARM'
23      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
24      C
25      C ZERO-LEVEL CONTROL SWITCH
26      COMMON /ZER/IZER
27      C
28      INCLUDE 'CSPICK'
29      INTEGER PRMK,SRMK,RMK,SGRADE
30      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
31      *CROSS(32),SMP(32),FMP(32),PCROSS(32),
32      *IFMOT(32),IGRADE(32),SGRADE(32),
33      *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
34      *PRES(32),SRES(32),EPD(32),CLK(32)
35      COMMON /EPARM/NTMP,ITEMP(32),OTZARR,OTWIND
36      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
37      COMMON /IOPRM/NS,NSCAN,IDUM(4)
38      DIMENSION KEY(32),X(32)
39      DIMENSION TIME(32),JPNAM(25),JWNAM(19)
40      DATA JPNAM/84,62,65,67,69,83,32,83,72,73,70,
41      *84,69,68,32,66,89,32,80,45,84,73,77,69,83/
42      DATA JWNAM/87,65,68,65,84,73,32,83,69,67,84,
43      *73,79,78,32,80,76,79,84/
44      C COMPUTE FIDUCIAL SHIFT MARK TO BE 1/10 WINDOW LENGTH
45      FIDUC=TWIND/10
46      C ROUND DOWN TO NEXT SECOND
47      IFID=INT(FIDUC)
48      C IF IT'S ZERO,MAKE IT ONE
49      IF(IFID.EQ.0)IFID=1
50      C IF THE WINDOW IS MORE THAN SPCSE SECONDS DO IT
51      ELSE JUST TAKE 10%
52      SPCSE=2.0
53      IF(TWIND.GT.SPCSE)FIDUC=IFID
54      C
55      C
56      C ISH=83 CORRESPONDS TO ALIGN ON S-TIMES
57      IF(ISH.EQ.83)GO TO 9
58      DO 8 J=1,32
59      TIME(J)=PTIME(J)
60      GO TO 11
61      9 CONTINUE
62      DO 10 J=1,32
63      TIME(J)=STIME(J)
64      10 CONTINUE
65      C
66      C LOAD X WITH TIMES,KEY POINTS TO FORMER ORDER
67      DO 1 J=1,NREQ
68      KK=IREQ(J)
69      X(J)=TIME(KK)

```

```

60:      IF (TIME(KK).EQ.0.)X(J)=10000.
61:      1      CONTINUE
62:      C
63:      C  SORT TIMES, KEY POINTS TO FORMER ORDER
64:      CALL SORT(X,KEY,NREQ)
65:      CC  ELIMINATE NULL TIMES(IE.TIME=10000.)
66:      KNT=0
67:      DO 2 J=1,NREQ
68:      IF(X(J).GE.10000.)GO TO 3
69:      2      KNT=KNT+1
70:      3      CONTINUE
71:      C  REORDER TRACES ACCORDING TO SORT KEYS
72:      DO 6 J=1,KNT
73:      JJ=KEY(J)
74:      KK=ITEMP(JJ)
75:      IREQ(J)=KK
76:      6      CONTINUE
77:      NREQ=KNT
78:      CC
79:      C  SET UP WINDOW TO BE CLOSE TO START OF EARLIEST
80:      C  SHIFT TIME (AVOID NEEDLESS CLIPPING TIME)
81:      JR=IREQ(1)
82:      C  SAVE OLD WINDOW TIME-ORIGIN
83:      OTZARR=TZARR
84:      C
85:      IF(ISH.EQ.83)GOTO 5
86:      C  COME HERE TO SHIFT ON P-TIMES
87:      TZARR=PTIME(JR)-FIDUC
88:      IF(TWIND.GT.SPCSE)TZARR=AINT(TZARR)
89:      IF(TZARR.LE.0.)TZARR=0.
90:      DO 7 J=1,NREQ
91:      KK=IREQ(J)
92:      7      SHFT(KK)=TZARR+FIDUC-PTIME(KK)
93:      GOTO 4
94:      C  HERE TO SHIFT ON S-TIMES
95:      5      TZARR=STIME(JR)-FIDUC
96:      IF(TWIND.GT.SPCSE)TZARR=AINT(TZARR)
97:      IF(TZARR.LE.0.)TZARR=0.
98:      DO 12 J=1,NREQ
99:      KK=IREQ(J)
100:      12     SHFT(KK)=TZARR+FIDUC-STIME(KK)
101:      C
102:      4      CALL ERASE
103:      C
104:      C  RECALCULATE TRACE "DISTANCE" FOR P AND S ALIGNMENT
105:      IF(ISH.EQ.80.OR.ISH.EQ.83)CALL RSET0
106:      C  CALL SPECIAL ROUTINE FOR WADATI PLOT
107:      C*  IF(ISH.EQ.87)CALL DWAD(X,NREQ)
108:      C
109:      C  REMOVE DC SHIFT IF ZERO FLAG IS ON
110:      IF(IZER.EQ.1)CALL ZLEV
111:      C  PLOT DATA
112:      CALL ARRAY
113:      C  LABEL OPTION 2 ,NO START AND STOP TIME LABELS
114:      C  (SINCE MEANINGLESS ON A SHIFTED TRACE)
115:      CALL LBSTA(2)
116:      C  PUT AMPLITUDE SCALE FACTORS ON PLOT
117:      CALL LBSCL
118:      C
119:      C  DRAW FIDUCIAL TIC
120:      LXX=(LX+FIDUC)/TWIND
121:      IXX=KX+LXX
122:      IYY=KY+LY

```

```

123:      CALL MOVABS(IXX,IYY)
124:      CALL DSHABS(IXX,KY,1)
125:      C
126:      CC  PLOT TITLE AT TOP OF WINDOW
127:          IXCNT=KX+(2*LX)/5
128:          IYCNT=KY+LY+LY/20
129:          CALL MOVABS(IXCNT,IYCNT)
130:          JPNAM(19)=ISH
131:      C  ALIGN ON P-TIMES
132:          IF(ISH.EQ.80)CALL ANSTR(25,JPNAM)
133:      C  ALIGN ON S-TIMES
134:          IF(ISH.EQ.83)CALL ANSTR(25,JPNAM)
135:      C  WADATI PLOT
136:      C*  IF(ISH.EQ.87)CALL ANSTR(19,JWNAM)
137:      CC
138:      C  RESTORE OLD WINDOW TIME-ORIGIN
139:      C
140:      RETURN
141:      END

```


ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:30:50 PM

RSETD.FR

```
1:      SUBROUTINE XRSETD
2:      OVERLAY ORSETD
3:      INCLUDE 'CSBLAN'
4:      PARAMETER MAXX=1000
5:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
6:      INTEGER A
7:      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
8:      *D(32)
9:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,
10:     *DTG,SG
11:      AL=AZ+NREQ*ARNG
12:      DO 1 J=1,NREQ
13:      D(J)=AL-(J-0.5)*ARNG
14:      1  CONTINUE
15:      RETURN
16:      END
17:
```

RSTAT.FR

```

1:      SUBROUTINE XRSTAT
2:      OVERLAY ORSTAT
3:      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
4:      INCLUDE 'CS8LAN'
5:      PARAMETER MAXX=1000
6:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
7:      INTEGER A
8:      INCLUDE 'CS8PARM'
9:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
10:     C
11:     C      THIS ROUTINE RESTORES OLD STATE
12:     C      INTO NREQ AND IREQ
13:     C
14:      NREQ=NTEMP
15:      DO 1 J=1,NTEMP
16:     1   IREQ(J)=ITEMP(J)
17:     C  ALSO RESETS SCREEN WINDOW
18:      CALL SWNO(30,900,10,600)
19:     C  RESTORE OLD WINDOW TIME-ORIGIN
20:      TZARR=OTZARR
21:      RETURN
22:      END

```

SCALER.FR

```

11      SUBROUTINE XSCALER
12      OVERLAY OSCALER
13      INCLUDE 'CSBLAN'
14      PARAMETER MAXX=1000
15      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
16      INTEGER A
17      INCLUDE 'CSAPARN'
18      COMMON /APARN/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
19      INCLUDE 'CSCPARN'
20      COMMON /CPARN/HSTA,ISTA(2,32)
21
22      C
23      COMMON /TVTUNE/ITV,ITX
24      INTEGER X,SEMI,PTR,VEC,NI,BLNK,VCNT,DOT,SP,DSH,EOR
25      DIMENSION VEC(32),LIST(96),ITENP(12),IN(50)
26      PARAMETER X=20130K,HI=20071K,LO=20060K,BLNK=20040K
27      PARAMETER DOT=20056K,SEMI=20073K,DSH=20055K,EOR=20000K
28      PARAMETER LAST=96
29
30      C
31      C PROMPT USER
32      WRITE(ITV,100)
33      100  FORMAT(' ENTER SCALE FACTOR AND TRACE LIST',/,
34            *'EXAMPLE  5.56X 7 5-2 ABCD')
35
36      C
37      C INPUT CHAR STRING
38      READ(ITX,200)IN(1)
39      200  FORMAT($96)
40      DECODE(IN,201,ERR=500)LIST
41      201  FORMAT(96R1)
42      500  CONTINUE
43
44      C
45      IP=0
46      LPTR=LAST
47      SCL=1.
48
49      C SEARCH FOR X
50      DO 1 J=1,LAST
51      IF(LIST(J).EQ.EOR.OR.LIST(J).EQ.SEMI)GOTO 600
52      IF(LIST(J).NE.X)GOTO 1
53      PTR=J+1
54      IP=J-1
55
56      1    CONTINUE
57      600  CONTINUE
58
59      C ERROR IF NO PRECEDING NUMBER
60      IF(IP.LT.1)GOTO 550
61
62      C
63      C EXTRACT SCALE FACTOR
64      DO 2 K=1,IP
65      JK=IP-K+1
66
67      C
68      C CHECK FOR NUMBER,.,OR -
69      IF(LIST(JK).GE.LO.AND.LIST(JK).LE.HI)GOTO 2
70      IF(LIST(JK).EQ.DOT.OR.LIST(JK).EQ.DSH)GOTO 2
71
72      C IF COME HERE AND NOT BLANK, THEN ERROR
73      IF(LIST(JK).NE.BLNK)GOTO 550
74
75      C ALL DONE, NOW DECODE
76      SP=JK+1

```

```

57:      GOTO 3
58:      2      CONTINUE
59:      C
60:      C   COME HERE TO DECODE SCALE FACTOR
61:      C
62:      3      KNT=IP-SP+1
63:      C   MOVE ASCII CHARACTERS
64:      C
65:      C   DECODE SCALE FACTOR
66:      CALL INFREE(IN,SCL,96)
67:      C
68:      C
69:      VCNT=32
70:      CALL TPARS(LIST,PTR,LPTR,ISTA,VEC,VCNT,SENI,IER)
71:      C   ASSIGN SCALE FACTOR
72:      DO 30 K=1,VCNT
73:      II=VEC(K)
74:      30      S(II)=SCL
75:      RETURN
76:      550     WRITE(ITV,510)
77:      510     FORMAT(' SCALE SYNTAX ERROR')
78:      RETURN
79:      END

```

SORT.FR

```

11      SUBROUTINE XSORT(X,KEY,NO)
12      OVERLAY O SORT
13      CC SORT ARRANGES THE ELEMENTS OF X IN ASCENDING ORDER AND
14      CC CONSTRUCTS AN ARRAY KEY* OF SUBSCRIPTS OF X
15      CC
16      CC ARGUMENTS
17      CC X THE ARRAY TO BE ORDERED, REAL
18      CC KEY THE ARRAY OF SUBSCRIPTS OF X FORMED BY SORT
19      CC NO THE NUMBER OF ELEMENTS IN THE X ARRAY
20      CC
21      DIMENSION X(NO),KEY(NO)
22      DO 1 I=1,NO
23      KEY(I)=I
24      1 CONTINUE
25      MO=NO
26      2 IF(MO-15)21,21,23
27      21 IF(MO-1)9,9,22
28      22 MO=2*(MO/4)+1
29      19 GO TO 24
30      23 MO=2*(MO/8)+1
31      21 24 KO=NO-MO
32      22 JO=1
33      23 I=JO
34      24 IF(X(I)-X(I+MO))28,28,27
35      25 27 TEMP=X(I)
36      26 2701 X(I)=X(I+MO)
37      27 2702 X(I+MO)=TEMP
38      28 2703 KEMP=KEY(I)
39      29 2704 KEY(I)=KEY(I+MO)
40      30 2705 KEY(I+MO)=KEMP
41      31 I=I-MO
42      32 IF(I-1)28,26,26
43      33 28 JO=JO+1
44      34 IF(JO-KO)25,25,2
45      35 9 RETURN
46      36 END

```

SPLOT.FR

```

11      SUBROUTINE SPLOT
12      COMMON /TVTUNE/ITV,ITX
13      INCLUDE 'C$APARN'
14      COMMON /APARN/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
15      INCLUDE 'C$BPARN'
16      COMMON /BPARN/TZARR,TWIND,THR,TNN,AZ,ARNG,AL,NT,DTG,SG
17      C
18      COMMON/SFL/SFLAG
19      INTEGER SFLAG
20      COMMON /ZER/IZER
21      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
22      COMMON /EPARN/NTENP,ITEMP(32),DTZARR,DTWIND
23      COMMON /SHCON/ISH
24      COMMON /IOPRN/NS,NSCAN,MREN,MUX,IFILE,IUNIT
25      INCLUDE 'C$GPARN'
26      C MODIFIED 5/12/78 FOR 3-PASSES
27      PARAMETER LSZ1=13
28      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
29      COMMON /KDATE/KYR,KMO,KDA,KHR,KNN,RSEC,DUR,EVENT,PASS
30      *,UNQYR,UNQID,TCORR,FCA,LCA
31      COMMON /JDATE/INR,INN,TSEC
32      COMMON /XXX/XX(3),LIST(LSZ)
33      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
34      COMMON /VEE/VSAVE(4,MAXPASS)
35      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
36      C
37      INTEGER OLDEVT,PUFLG,PUBLNK,CHAIN
38      COMMON /MISC/ZX,ZY,CX,CY
39      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,MFIL(20)
40      INTEGER ZX,ZY,CX,CY
41      DIMENSION MSG1(2),MSG2(3),MSG4(4)
42      DIMENSION MNN(41),SQ(40)
43      DATA IQ/63/
44      DATA IY/1HY/
45      C DEFAULT SCALE UNITS OPTION IS 1
46      IPAUS=0
47      SFLAG=1
48      CHAIN=0
49      MAXCH=32
50      IFILE=1
51      MUX=10
52      PUBLNK=0
53      PUFLG=0
54      OLDEVT=0
55      EVENT=0
56      CX=0
57      CY=780*ITERM
58      ZX=0
59      ZY=750*ITERM
60      C GO IMMEDIATELY TO EVENT PROCESSOR
61      IC=14
62      GO TO 39
63      C
64      C COME HERE TO START ANOTHER COMMAND
65      20 CONTINUE
66      CALL COMMAND(IC,JC)

```

```

57:      GO TO (1,2,3,4,5,44,7,8,9,10,31,32,33,39,35,4,20,90,20,20,99,20,
58:      *20,38,20,20,20,40,42,20,41,43,20,20,2,777), IC
59:      777  CONTINUE
60:      IF(IPAUS.EQ.1)GOTO 45
61:      GO TO 20
62:      C*
63:      C*  DEFINE THE WINDOW
64:      1    CONTINUE
65:      CALL DEPRC(IC,JC)
66:      C*
67:      C*
68:      GO TO 20
69:      2    CONTINUE
70:      CALL TRPRC(IC,JC)
71:      C*
72:      GO TO 20
73:      3    CONTINUE
74:      IF(JC.EQ.13)GO TO 36
75:      IF(JC.EQ.9.OR.JC.EQ.12)GO TO 10
76:      C*  PLOT PROCESSOR
77:      C  RESTORE STATE BEFORE PLOTTING
78:      CALL RSTAT
79:      CALL PLPRC(JC)
80:      900  CONTINUE
81:      CALL MOVABS(CX,CY)
82:      CALL ANMODE
83:      CALL TSEND
84:      GO TO 20
85:      4    CONTINUE
86:      C***SHIFT PROCESSOR
87:      CALL RSTAT
88:      C
89:      C  ALIGN P-PHASES
90:      ISH=80
91:      C  ALIGN S-PHASES
92:      IF(JC.EQ.26)ISH=83
93:      IF(IC.EQ.14)ISH=83
94:      C  WADATI PLOT
95:      IF(IC.EQ.16)ISH=87
96:      C
97:      CALL PSHIFT(ISH)
98:      GO TO 900
99:      5    CONTINUE
100:     C*  PICK PROCESSOR
101:     CALL PICK
102:     GO TO 900
103:     6    CONTINUE
104:     C*
105:     GO TO 20
106:     C*  RESET PLOT PARMS
107:     7    CONTINUE
108:     C  IF 'RESU', RESUME REMOTE FILE CONTROL
109:     IF(JC.EQ.34)GOTO 45
110:     IF(JC.EQ.27)GOTO 90
111:     CALL DEFAULT
112:     IZER=0
113:     GO TO 20
114:     C*  AMPLITUDE SCALING PROCESSOR
115:     8    CONTINUE
116:     CALL SCALER

```

```

117:      GO TO 20
118:      C
119:      C  SELECT SCALING OPTION
120:      89  IF(JC.NE.8)GOTO 20
121:      WRITE(ITV,800)
122:      800  FORMAT(" SELECT INPUT SCALE OPTION",/,
123:      *"1=DIGITAL COUNTS",/,"2=VOLTS OUT",/,
124:      *"3=EQUIVALENT DEVELOCORDER MILLIMETERS")
125:      READ(ITX,234)NNN(1)
126:      234  FORMAT(S81)
127:      CALL INFREE(NNN,R8.96)
128:      SFLAG=R8
129:      WRITE(ITV,801)SFLAG
130:      801  FORMAT(" OPTION ",I1," SELECTED")
131:      GOTO 20
132:      C
133:      C+  CROSS ? AUTO-CORRELATION PROCESSOR
134:      9    CONTINUE
135:      C    CALL SCRSS
136:      GO TO 20
137:      C+  PLOT CROSS & AUTO-CORRELOGRAMS
138:      10   CONTINUE
139:      C    CALL PCRSS
140:      C    GO TO 900
141:      35   CONTINUE
142:      C+  SPECTRA PROCESSOR
143:      C    CALL SFAST
144:      GO TO 20
145:      36   CONTINUE
146:      C    CALL PFAST
147:      C    GO TO 900
148:      C  PUNCH PHASE CARDS
149:      31   CONTINUE
150:      CALL FAZPU(1)
151:      GO TO 20
152:      33   CONTINUE
153:      C  USER ROUTINE
154:      C  ROUTINE SETS KC IN ORDER TO CONTROL ITS RETURN
155:      C    CALL USER(JC,KC)
156:      C    GO TO (20,900)KC
157:      C
158:      C  MENU
159:      32   CONTINUE
160:      CALL MENU(1)
161:      GO TO 20
162:      40   CONTINUE
163:      CALL MENU(2)
164:      GOTO 20
165:      38   CONTINUE
166:      IZER=1
167:      GO TO 20
168:      C  COME HERE FOR 'UN' COMMANDS
169:      42   IF(JC.EQ.24)IZER=0
170:      IF(JC.EQ.32)ITX=5
171:      IF(JC.EQ.32)CLOSE 13
172:      IF(JC.EQ.32)CHAIN=0
173:      GOTO 20
174:      C  COME HERE FOR 'PA'
175:      44   IF(JC.NE.13)GOTO 39
176:      ITX=5

```

```

177:      WRITE(ITV,344)
178: 344  FORMAT('REMOTE PAUSE')
179:      IPAUS=1
180:      GOTO 20
181:  C  COME HERE TO RESUME REMOTE CONTROL
182: 45   ITX=13
183:      IPAUS=0
184:      GOTO 20
185:  C  HARDCOPY HERE
186: 43   IF(JC.NE.33)GOTO 20
187:      CALL HDCOPY
188:      CALL TSEND
189:  C  10 SEC DELAY FOR HARDCOPY UNIT
190:      CALL FDLY(100)
191:      GOTO 20
192:  C  COME HERE FOR COMMANDS FROM REMOTE FILE
193: 41   WRITE(ITV,440)
194: 440  FORMAT(' INPUT REMOTE FILE NAME(10 CHARS. MAX)')
195:      READ(ITX,441)MNH(1)
196: 441  FORMAT(S10)
197:      IF(CHAIN.EQ.1)CLOSE 13
198:  C  MNH IS ARRAY CONTAINING ASCII NAME OF FILE
199:      OPEN 13,MNH
200:      CHAIN=1
201:      ITX=13
202:      GOTO 20
203:  C
204: 39   CONTINUE
205:  C  EVENT/PASS PROCESSOR
206:  C  CHECK ZERO-LEVEL FLAG
207:      IF(IC.EQ.14)IZER=0
208:      IF(JC.EQ.24)IZER=1
209:      CALL EVPRC(IC,JC)
210:      GO TO 20
211: 90   CONTINUE
212:  C  OUTPUT ZZ CARD AT END OF PHASE LIST
213:      CALL FAZPU(2)
214:      WRITE(ITV,990)
215: 990  FORMAT(10H EXIT ISDS)
216:      RETURN
217:      END

```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:35:54 PM

SWND.FR

```
1:      SUBROUTINE SWND(NX,NLX,NY,NLY)
2:  CC  RESET PHYSICAL SCREEN BOUNDS
3:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
4:      KX=NX*ITERM
5:      KY=NY*ITERM
6:      LX=NLX*ITERM
7:      LY=NLY*ITERM
8:      RETURN
9:      END
10:
```

TPARS.FR

```

11      SUBROUTINE XTPARS(LIST, PTR, LPTR, SLIST, VEC, VCNT, EOL, IER)
12      OVERLAY OTPARS
13      C
14      C P.R. STEVENSON 9-28-78
15      C
16      C LIST IS AN ASCII STRING CONTAINING ONLY 1 OR 2 CHAR.
17      C NUMBERS SEPERATED BY EITHER BLANKS OR -.
18      C WHERE N-M MEANS THE RANGE OF NUMBERS FROM N
19      C TO M, AND A SINGLE NUMBER MEANS JUST THAT
20      C NUMBER
21      C
22      C PTR IS INDEX WITHIN LIST WHERE SCAN IS TO BEGIN
23      C THE LIST ENDS WITH THE EOL CHARACTER
24      C
25      C LPTR IS INDEX WITHIN LIST WHERE SCAN IS TO END IF
26      C EOL CHARACTER IS NOT ENCOUNTERED
27      C
28      C VEC IS THE VECTOR CONTAINING THE LIST OF
29      C NUMBERS GENERATED FROM LIST
30      C
31      C VCNT IS THE COUNT OF ELEMENTS IN VEC
32      C
33      C EOL IS A VARIABLE CONTAINING ASCII CODE FOR THE
34      C END OF THE CHARACTER STRING
35      C
36      C
37      C
38      C
39      C
40      C
41      C
42      C
43      C
44      C
45      C
46      C
47      C
48      C
49      C
50      C
51      C
52      C
53      C
54      C
55      C
56      C
57      C
58      C
59      C
60      C
61      C
62      C
63      C
64      C
65      C
66      C
67      C
68      C
69      C
70      C
71      C
72      C
73      C
74      C
75      C
76      C
77      C
78      C
79      C
80      C
81      C
82      C
83      C
84      C
85      C
86      C
87      C
88      C
89      C
90      C
91      C
92      C
93      C
94      C
95      C
96      C
97      C
98      C
99      C
100     C
101     C
102     C
103     C
104     C
105     C
106     C
107     C
108     C
109     C
110     C
111     C
112     C
113     C
114     C
115     C
116     C
117     C
118     C
119     C
120     C
121     C
122     C
123     C
124     C
125     C
126     C
127     C
128     C
129     C
130     C
131     C
132     C
133     C
134     C
135     C
136     C
137     C
138     C
139     C
140     C
141     C
142     C
143     C
144     C
145     C
146     C
147     C
148     C
149     C
150     C
151     C
152     C
153     C
154     C
155     C
156     C
157     C
158     C
159     C
160     C
161     C
162     C
163     C
164     C
165     C
166     C
167     C
168     C
169     C
170     C
171     C
172     C
173     C
174     C
175     C
176     C
177     C
178     C
179     C
180     C
181     C
182     C
183     C
184     C
185     C
186     C
187     C
188     C
189     C
190     C
191     C
192     C
193     C
194     C
195     C
196     C
197     C
198     C
199     C
200     C
201     C
202     C
203     C
204     C
205     C
206     C
207     C
208     C
209     C
210     C
211     C
212     C
213     C
214     C
215     C
216     C
217     C
218     C
219     C
220     C
221     C
222     C
223     C
224     C
225     C
226     C
227     C
228     C
229     C
230     C
231     C
232     C
233     C
234     C
235     C
236     C
237     C
238     C
239     C
240     C
241     C
242     C
243     C
244     C
245     C
246     C
247     C
248     C
249     C
250     C
251     C
252     C
253     C
254     C
255     C
256     C
257     C
258     C
259     C
260     C
261     C
262     C
263     C
264     C
265     C
266     C
267     C
268     C
269     C
270     C
271     C
272     C
273     C
274     C
275     C
276     C
277     C
278     C
279     C
280     C
281     C
282     C
283     C
284     C
285     C
286     C
287     C
288     C
289     C
290     C
291     C
292     C
293     C
294     C
295     C
296     C
297     C
298     C
299     C
300     C
301     C
302     C
303     C
304     C
305     C
306     C
307     C
308     C
309     C
310     C
311     C
312     C
313     C
314     C
315     C
316     C
317     C
318     C
319     C
320     C
321     C
322     C
323     C
324     C
325     C
326     C
327     C
328     C
329     C
330     C
331     C
332     C
333     C
334     C
335     C
336     C
337     C
338     C
339     C
340     C
341     C
342     C
343     C
344     C
345     C
346     C
347     C
348     C
349     C
350     C
351     C
352     C
353     C
354     C
355     C
356     C
357     C
358     C
359     C
360     C
361     C
362     C
363     C
364     C
365     C
366     C
367     C
368     C
369     C
370     C
371     C
372     C
373     C
374     C
375     C
376     C
377     C
378     C
379     C
380     C
381     C
382     C
383     C
384     C
385     C
386     C
387     C
388     C
389     C
390     C
391     C
392     C
393     C
394     C
395     C
396     C
397     C
398     C
399     C
400     C
401     C
402     C
403     C
404     C
405     C
406     C
407     C
408     C
409     C
410     C
411     C
412     C
413     C
414     C
415     C
416     C
417     C
418     C
419     C
420     C
421     C
422     C
423     C
424     C
425     C
426     C
427     C
428     C
429     C
430     C
431     C
432     C
433     C
434     C
435     C
436     C
437     C
438     C
439     C
440     C
441     C
442     C
443     C
444     C
445     C
446     C
447     C
448     C
449     C
450     C
451     C
452     C
453     C
454     C
455     C
456     C
457     C
458     C
459     C
460     C
461     C
462     C
463     C
464     C
465     C
466     C
467     C
468     C
469     C
470     C
471     C
472     C
473     C
474     C
475     C
476     C
477     C
478     C
479     C
480     C
481     C
482     C
483     C
484     C
485     C
486     C
487     C
488     C
489     C
490     C
491     C
492     C
493     C
494     C
495     C
496     C
497     C
498     C
499     C
500     C
501     C
502     C
503     C
504     C
505     C
506     C
507     C
508     C
509     C
510     C
511     C
512     C
513     C
514     C
515     C
516     C
517     C
518     C
519     C
520     C
521     C
522     C
523     C
524     C
525     C
526     C
527     C
528     C
529     C
530     C
531     C
532     C
533     C
534     C
535     C
536     C
537     C
538     C
539     C
540     C
541     C
542     C
543     C
544     C
545     C
546     C
547     C
548     C
549     C
550     C
551     C
552     C
553     C
554     C
555     C
556     C
557     C
558     C
559     C
560     C
561     C
562     C
563     C
564     C
565     C
566     C
567     C
568     C
569     C
570     C
571     C
572     C
573     C
574     C
575     C
576     C
577     C
578     C
579     C
580     C
581     C
582     C
583     C
584     C
585     C
586     C
587     C
588     C
589     C
590     C
591     C
592     C
593     C
594     C
595     C
596     C
597     C
598     C
599     C
600     C
601     C
602     C
603     C
604     C
605     C
606     C
607     C
608     C
609     C
610     C
611     C
612     C
613     C
614     C
615     C
616     C
617     C
618     C
619     C
620     C
621     C
622     C
623     C
624     C
625     C
626     C
627     C
628     C
629     C
630     C
631     C
632     C
633     C
634     C
635     C
636     C
637     C
638     C
639     C
640     C
641     C
642     C
643     C
644     C
645     C
646     C
647     C
648     C
649     C
650     C
651     C
652     C
653     C
654     C
655     C
656     C
657     C
658     C
659     C
660     C
661     C
662     C
663     C
664     C
665     C
666     C
667     C
668     C
669     C
670     C
671     C
672     C
673     C
674     C
675     C
676     C
677     C
678     C
679     C
680     C
681     C
682     C
683     C
684     C
685     C
686     C
687     C
688     C
689     C
690     C
691     C
692     C
693     C
694     C
695     C
696     C
697     C
698     C
699     C
700     C
701     C
702     C
703     C
704     C
705     C
706     C
707     C
708     C
709     C
710     C
711     C
712     C
713     C
714     C
715     C
716     C
717     C
718     C
719     C
720     C
721     C
722     C
723     C
724     C
725     C
726     C
727     C
728     C
729     C
730     C
731     C
732     C
733     C
734     C
735     C
736     C
737     C
738     C
739     C
740     C
741     C
742     C
743     C
744     C
745     C
746     C
747     C
748     C
749     C
750     C
751     C
752     C
753     C
754     C
755     C
756     C
757     C
758     C
759     C
760     C
761     C
762     C
763     C
764     C
765     C
766     C
767     C
768     C
769     C
770     C
771     C
772     C
773     C
774     C
775     C
776     C
777     C
778     C
779     C
780     C
781     C
782     C
783     C
784     C
785     C
786     C
787     C
788     C
789     C
790     C
791     C
792     C
793     C
794     C
795     C
796     C
797     C
798     C
799     C
800     C
801     C
802     C
803     C
804     C
805     C
806     C
807     C
808     C
809     C
810     C
811     C
812     C
813     C
814     C
815     C
816     C
817     C
818     C
819     C
820     C
821     C
822     C
823     C
824     C
825     C
826     C
827     C
828     C
829     C
830     C
831     C
832     C
833     C
834     C
835     C
836     C
837     C
838     C
839     C
840     C
841     C
842     C
843     C
844     C
845     C
846     C
847     C
848     C
849     C
850     C
851     C
852     C
853     C
854     C
855     C
856     C
857     C
858     C
859     C
860     C
861     C
862     C
863     C
864     C
865     C
866     C
867     C
868     C
869     C
870     C
871     C
872     C
873     C
874     C
875     C
876     C
877     C
878     C
879     C
880     C
881     C
882     C
883     C
884     C
885     C
886     C
887     C
888     C
889     C
890     C
891     C
892     C
893     C
894     C
895     C
896     C
897     C
898     C
899     C
900     C
901     C
902     C
903     C
904     C
905     C
906     C
907     C
908     C
909     C
910     C
911     C
912     C
913     C
914     C
915     C
916     C
917     C
918     C
919     C
920     C
921     C
922     C
923     C
924     C
925     C
926     C
927     C
928     C
929     C
930     C
931     C
932     C
933     C
934     C
935     C
936     C
937     C
938     C
939     C
940     C
941     C
942     C
943     C
944     C
945     C
946     C
947     C
948     C
949     C
950     C
951     C
952     C
953     C
954     C
955     C
956     C
957     C
958     C
959     C
960     C
961     C
962     C
963     C
964     C
965     C
966     C
967     C
968     C
969     C
970     C
971     C
972     C
973     C
974     C
975     C
976     C
977     C
978     C
979     C
980     C
981     C
982     C
983     C
984     C
985     C
986     C
987     C
988     C
989     C
990     C
991     C
992     C
993     C
994     C
995     C
996     C
997     C
998     C
999     C
1000    C

```

```

57:         IF(INDEX.GT.LASTX)GOTO 99
58:         PREV=NEW
59:         C
60:         C IS IT BLANK(OR CONNA OR EOL) OR NON-BLANK?
61:         C
62:         NEW=.TRUE.
63:         IF(LIST(INDEX).EQ.BLNK)NEW=.FALSE.
64:         IF(LIST(INDEX).EQ.CONNA.OR.LIST(INDEX).EQ.EOL)NEW=.FALSE.
65:         IF(INDEX.EQ.LPTR)NEW=.FALSE.
66:         C
67:         C IF PREVIOUS IS FALSE AND NEW IS TRUE, THEN
68:         C START NEW STRING
69:         IF((.NOT.PREV).AND.NEW)GOTO 3
70:         C
71:         C IF PREVIOUS IS TRUE AND NEW IS FALSE, THEN
72:         C END OF STRING
73:         IF(PREV.AND.(.NOT.NEW))GOTO 2
74:         C
75:         C ELSE JUST GET ANOTHER CHARACTER
76:         GOTO 1
77:         C
78:         C
79:         C COME HERE FOR END OF STRING
80:         C
81:         2      P2(KNT)=INDEX-1
82:         GOTO 1
83:         C
84:         C COME HERE FOR START OF STRING
85:         C
86:         3      KNT=KNT+1
87:         P1(KNT)=INDEX
88:         GOTO 1
89:         C
90:         C
91:         C - - - - -
92:         C
93:         C END OF LINE
94:         C
95:         9      CONTINUE
96:         C
97:         C NOW EXPAND AND EVALUATE EXPRESSIONS
98:         C
99:         LAST=KNT
100:        JV=0
101:        DO 10 J=1, LAST
102:        C COMPUTE STRING LENGTH
103:        NCHAR=P2(J)-P1(J)+1
104:        C VALIDITY CHECK
105:        C
106:        IF(NCHAR.LT.1.OR.NCHAR.GT.5)GOTO 99
107:        C
108:        C IF LENGTH GREATER THAN 2, IT IS A
109:        C RANGE EXPRESSION
110:        N1=P1(J)
111:        N2=P2(J)
112:        IF(NCHAR.GT.2)GOTO 16
113:        C ELSE IT IS A SINGLE NUMBER
114:        JV=JV+1
115:        C COMPUTE NUMBER FROM ASCII CHARACTER
116:        DO 15 K=N1,N2

```

```

117: 6 CHECK FOR ILLEGAL NUMERALS
118: IF(LIST(K).LT.OFF.OR.LIST(K).GT.NINE)GOTO 98
119: VEC(JV)=LIST(N2)-OFF
120: IF(NCHAR.EQ.2)VEC(JV)=VEC(JV)+(LIST(N1)-OFF)*10
121: 15 CONTINUE
122: GOTO 10
123: C
124: 6 COME HERE TO EVALUATE RANGE EXPRESSIONS
125: 16 N3=0
126: DO 17 K=N1,N2
127: IF(LIST(K).EQ.DSN)N3=K
128: 17 CONTINUE
129: C IF NOT DASH, NOT A RANGE EXPRESSION
130: IF(N3.EQ.0)GOTO 20
131: C CHECK FOR SYNTAX
132: IF((K-N1).GT.5.OR.(K-N1).LT.1)GOTO 98
133: N4=N3-1
134: N5=N3+1
135: 6 CHECK FOR ILLEGAL NUMERALS IN RANGE EXPRESSION
136: DO 18 K=N1,N2
137: IF(K.EQ.N3)GOTO 10
138: IF(LIST(K).LT.OFF.OR.LIST(K).GT.NINE)GOTO 98
139: 18 CONTINUE
140: JSTART=LIST(N4)-OFF
141: IF(N1.LT.N4)JSTART=JSTART+(LIST(N1)-OFF)*10
142: JLAST=LIST(N2)-OFF
143: IF(N5.LT.N2)JLAST=JLAST+(LIST(N5)-OFF)*10
144: C
145: C IS IT 'UP TO' OR 'DOWN TO' ?
146: C
147: IUD=1
148: IF(JLAST.LT.JSTART)IUD=-1
149: IRNG=IABS(JLAST-JSTART)+1
150: C
151: C EXPAND RANGE INTO LIST OF NUMBERS
152: DO 19 K=1,IRNG
153: JV=JV+1
154: 19 VEC(JV)=JSTART+IUD*(K-1)
155: GOTO 10
156: 20 CONTINUE
157: C PACK UP THE CHARACTER STRING FOR COMPARE
158: DO 23 K=1,2
159: K1=N1+(K-1)*2
160: ITEMP=IAND(MASK,LIST(K1))
161: ITEMP=ITEMP*256
162: JTEMP=IAND(MASK,LIST(K1+1))
163: NAN(K)=IOR(ITEMP,JTEMP)
164: 23 CONTINUE
165: NZ=N1+3
166: 8 SEARCH SLIST FOR 4-CHAR NAME
167: DO 21 K=1,SNAX
168: KK=K
169: IF(NAN(1).EQ.SLIST(1,K).AND.NAN(2).EQ.SLIST(2,K))GOTO 22
170: 21 CONTINUE
171: GOTO 98
172: C COME HERE TO ADD SEQ. NUMBER TO STATION TO VEC
173: 22 JV=JV+1
174: VEC(JV)=KK
175: C
176: 10 CONTINUE

```

```
177:      VCNT=JV
178:      RETURN
179: 90      IER=1
180:      VCNT=JV
181:      RETURN
182:      END
```

TRPRC.FR

```

11      SUBROUTINE XTRPRC(JC,JC)
12      OVERLAY OTRPRC
13      C ROUTINE TO SELECT, MANIPULATE AND LIST TRACES
14      C
15      INCLUDE 'C$BLAN'
16      PARAMETER MAXX=1000
17      COMMON A(MAXX), MAXPTS, NREQ, IREQ(32)
18      INTEGER A
19      INCLUDE 'C$CPARM'
20      COMMON /CPARM/HSTA, ISTA(2,32)
21      C
22      COMMON /IOPRM/NS, NSCAN, NREN, NUX, IFILE, IUNIT
23      COMMON /TVTUNE/ITV, ITX
24      COMMON /EPARM/HTEMP, ITEMP(32), OTZARR, OTWIND
25      LOGICAL EOC
26      INTEGER DUN(2,32), VEC(32), VCNT, IN(50), LIST(96)
27      INTEGER SENI, COLON, BLANK, EX, SE, DE, AP, PTR, CC
28      PARAMETER LAST=96, SENI=20073K, COLON=20072K,
29      *BLANK=20040K, EX=20105K, SE=20123K, DE=20104K,
30      *LI=20114K, AP=20101K, NA=20116K, MAXCH=32
31      C
32      C TRACE MANIPULATION COMMANDS MUST BE OF THE FORM:
33      C <COMMAND>: I J-K ABCD ;(<...>); L N ; . . . ETC.
34      C
35      C WHERE ';' SEPERATES MORE THAN ONE COMMAND ON THE SAME LINE.
36      C EITHER SINGLE TRACES MAY BE SPECIFIED, OR RANGES(BOTH
37      C INCREASING OR DECREASING) OF TRACES, OR THE TRACES MAY
38      C BE REFERENCED BY STATION NAME(ABCD).
39      C
40      C PROMPT
41      C
42      C WRITE(ITV,101)
43      C 101 FORMAT(' <SE,AP,DE,LI,NA,EX>; I J-K ABCD ;ETC...')
44      C
45      C READ LINE
46      C
47      ISENI=0
48      LPTR=0
49      EOC=.FALSE.
50      K1=1
51      READ(ITX,200)IN(1)
52      200 FORMAT($96)
53      DECODE(IN,201,ERR=88)LIST
54      201 FORMAT(96R1)
55      88 CONTINUE
56      C
57      C EXTRACT COMMAND
58      C
59      14 JP=0
60      C
61      C SEARCH FOR ';'
62      C
63      DO 70 K=K1, LAST
64      IF(LIST(K).EQ.COLON)GOTO 71
65      IF(LIST(K).EQ.BLANK)GOTO 70
66      IF(LIST(K).EQ.SENI)GOTO 72

```

```

57:      IF(JP.EQ.1)GOTO 70
58:      C FIRST TIME HERE, GET THE COMMAND!
59:      CC=LIST(K)
60:      JP=1
61:      GOTO 70
62:      C GOT THE COLON
63:      71 PTR=K+1
64:      70 IP=K
65:      C IF COME HERE, NO ';'
66:      LPTR=LAST
67:      EOC=.FALSE.
68:      GOTO 73
69:      C
70:      C
71:      C FOUND A SEMICOLON
72:      72 ISEMI=K
73:      C RESET COMMAND SCAN POINTER
74:      K1=K+1
75:      EOC=.TRUE.
76:      C SET END OF ARG FIELD
77:      LPTR=SEMI
78:      C
79:      73 IER=0
80:      C CALL SCANNER/PARSER FOR TRACE LIST
81:      IF(CC.EQ.SE.OR.CC.EQ.AP.OR.CC.EQ.DE)CALL TPARS(LIST,
82:      *PTR,LPTR,ISTA,VEC,VCNT,SEMI,IER)
83:      C
84:      IF(IER.NE.0)GOTO 8
85:      C WHAT COMMAND IS CC?
86:      C
87:      C RETURN IF EXIT
88:      IF(CC.EQ.EX)RETURN
89:      C SELECT?
90:      IF(CC.EQ.SE)GOTO 10
91:      C APPEND?
92:      IF(CC.EQ.AP)GOTO 20
93:      C LIST?
94:      IF(CC.EQ.LI)GOTO 30
95:      C LIST BY STATION NAME?
96:      IF(CC.EQ.NA)GOTO 35
97:      C DELETE?
98:      IF(CC.EQ.DE)GOTO 40
99:      C TRY AGAIN!
100:      GOTO 9
101:      C
102:      C
103:      C LOAD TRACE LIST
104:      10 J1=1
105:      JN=VCNT
106:      IF(JN.GT.NUX)WRITE(ITY,150)NUX
107:      IF(JN.GT.NUX)GOTO 9
108:      GOTO 15
109:      C APPEND TO EXISTING TRACE LIST
110:      20 J1=NREQ+1
111:      JN=NREQ+VCNT
112:      IF(JN.GT.NAXCH)WRITE(ITY,150)NAXCH
113:      IF(JN.GT.NAXCH)GOTO 9
114:      C COMMON PATH
115:      15 JJ=0
116:      150 FORMAT(' MORE THAN ',I3,' TRACES REQUESTED')

```



```

117:         IF(JN.GT.32)JN=32
118:         DO 16 J=J1,JN
119:         JJ=JJ+1
120:     16     IREQ(J)=VEC(JJ)
121:         NREQ=JN
122:         GOTO 19
123:     8     WRITE(ITV,800)
124:     800    FORMAT('*** SYNTAX ERROR *** ')
125:         GOTO 9
126:     C
127:     C WRITE TRACE LIST
128:     30     WRITE(ITV,102) (IREQ(J),J=1,NREQ)
129:     102    FORMAT(' TRACES:',32I3)
130:         GOTO 19
131:     C
132:     C WRITE TRACE LIST BY STATION NAME
133:     35     CONTINUE
134:         DO 36 J=1,NREQ
135:         JJ=IREQ(J)
136:         DUN(1,J)=ISTA(1,JJ)
137:         DUN(2,J)=ISTA(2,JJ)
138:     36     C
139:         WRITE(ITV,103) (DUN(1,J),DUN(2,J),J=1,NREQ)
140:     103    FORMAT(' TRACES:',16(2A2,1X))
141:         GOTO 19
142:     C
143:     C DELETE TRACES
144:     40     CONTINUE
145:     C LOOP ON NUMBER OF ITEMS TO BE DELETED
146:         DO 3 I=1,VCNT
147:         ITOK=VEC(I)
148:         KILL=-1
149:     C LOOK FOR FIRST OCCURRENCE OF ITEM TO BE DELETED
150:     C
151:         DO 31 J=1,NREQ
152:         IF(ITOK.EQ.IREQ(J))KILL=J
153:         IF(ITOK.EQ.IREQ(J))GOTO 34
154:     31     CONTINUE
155:         GOTO 3
156:     34     CONTINUE
157:     C SQUEEZE ITEM FROM LIST
158:     C
159:         IF(NREQ.EQ.1)GOTO 8
160:         NREQ=NREQ-1
161:         DO 33 J=KILL,NREQ
162:         IREQ(J)=IREQ(J+1)
163:     33     CONTINUE
164:         GOTO 19
165:     C BACKUP TRACES
166:     19     NTEMP=NREQ
167:         DO 190 J=1,NREQ
168:     190    ITEMP(J)=IREQ(J)
169:     C
170:     C DO WE HAVE A ')?
171:     C IF YES, THEN GET ANOTHER COMMAND
172:         IF(EOC)GOTO 14
173:     C ELSE GO GIVE ANOTHER PROMPT
174:         GOTO 9
175:     END

```

ZLEV.FR

```

1:      SUBROUTINE XZLEV
2:      OVERLAY OZLEV
3:      INTEGER START
4:      DIMENSION AV(32)
5:      PARAMETER NZ=100
6:      INCLUDE 'CSBLAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
9:      INTEGER A
10:     INCLUDE 'CSAPARM'
11:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
12:     INCLUDE 'CSBPARM'
13:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
14:     C
15:     C
16:     C  COMPUTE START
17:     START=TZARR/DTG+1
18:     C  CLEAR ACCUMULATORS
19:     DO 6 I=1,NREQ
20:     6  AV(I)=0.
21:     C  COMPUTE SEGMENT LENGTH
22:     IB=MAXX/NREQ+1
23:     NBUF=NZ/IB+1
24:     MSCAN=NZ/NBUF
25:     CALL FRAMD(MSCAN,IREQ,NREQ,START,IERR)
26:     IF(IERR.NE.0)TYPE 'ZERO-FRAMD ERROR:IERR=',IERR
27:     DO 1 K=1,NBUF
28:     CALL GETFM(A,NSCAN,IEND)
29:     IF(IEND.EQ.1.AND.IEND.EQ.0)TYPE 'EOF IN ZLEV'
30:     IF(IEND.EQ.1.AND.IEND.EQ.0)RETURN
31:     DO 2 I=1,NREQ
32:     DO 3 J=1,NSCAN
33:     JJ=MSCAN*(I-1)+J
34:     R=A(JJ)
35:     3  AV(I)=R+AV(I)
36:     2  CONTINUE
37:     1  CONTINUE
38:     C  LOAD Z ARRAY
39:     DO 5 I=1,NREQ
40:     II=IREQ(I)
41:     Z(II)=AV(I)/NZ
42:     5  CONTINUE
43:     RETURN
44:     END

```

APPENDIX D: LISTINGS OF SOME SUPPORT SOFTWARE

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 3:59:49 PM

DIRC1

```

1:      SUBROUTINE DIRC1
2:      C
3:      C   INTERPROCESS COMMUNICATIONS DIRECTORY INITIALIZER
4:      C
5:      C   P.A. MARSHALL
6:      C   SEPTEMBER 1976
7:      C-----
8:      C
9:      C   INCLUDE 'PSDIR'
10:     C**** EVENT DIRECTORY OFFSETS
11:     C
12:     C *** DIRECTORY RECORD SIZE
13:     C   PARAMETER RECSIZ = 160
14:     C
15:     C *** RECORD 0:
16:     C
17:     C   PARAMETER ENX = 1           ; NEXT EVENT (LAST+1)
18:     C   PARAMETER DML = ENX+1      ; NO. OF MASTERLIST ENTRIES
19:     C   PARAMETER DOF = DML+1      ; DATABASE FILE OFFSET
20:     C   PARAMETER DSZ = DOF+1      ; FILE SIZE IN SECTORS
21:     C   PARAMETER DNX = DSZ+1      ; NEXT BLOCK NO.
22:     C   PARAMETER CTE = DNX+1      ; CURRENT EVENT SAVE AREA
23:     C   PARAMETER MPL = CTE+5      ; MASTER POINTER LIST
24:     C   PARAMETER EPL = MPL+95     ; END OF POINTER LIST
25:     C
26:     C *** RECORD N:
27:     C
28:     C   PARAMETER ENO = 1           ; EVENT NO.
29:     C   PARAMETER ENA = ENO+1      ; EVENT NAME
30:     C   PARAMETER EIB = ENA+20     ; INITIAL BLOCK NO.
31:     C   PARAMETER ENB = EIB+1      ; TOTAL NUMBER OF BLOCKS
32:     C   PARAMETER DLT = ENB+1      ; DELTA-T (SECS)
33:     C   PARAMETER DRA = DLT+2      ; DURATION (SECS)
34:     C   PARAMETER EYR = DRA+2      ; YEAR
35:     C   PARAMETER EMO = EYR+1      ; MONTH
36:     C   PARAMETER EDA = EMO+1      ; DAY
37:     C   PARAMETER EHR = EDA+1      ; HOUR
38:     C   PARAMETER EMI = EHR+1      ; MINUTE
39:     C   PARAMETER ESE = EMI+1      ; SECONDS
40:     C   PARAMETER EP1 = ESE+2      ; EVENT, PASS 1
41:     C   PARAMETER EP2 = EP1+1      ; EVENT, PASS 2
42:     C   PARAMETER EP3 = EP2+1      ; EVENT, PASS 3
43:     C
44:     C   PARAMETER ETM = EYR         ; EVENT TIME
45:     C   PARAMETER EPP = EP1-1      ; EVENT, PASS POINTER
46:     C
47:     C *** PASS N:
48:     C
49:     C   PARAMETER ENP = 0           ; ASSOCIATED EVENT NO.
50:     C   PARAMETER PNO = ENP+1      ; PASS NO.
51:     C   PARAMETER PIB = PNO+1      ; INITIAL BLOCK NO.
52:     C   PARAMETER PNB = PIB+1      ; NUMBER OF BLOCKS
53:     C   PARAMETER PFC = PNB+1      ; FCN (FIRST CHANNEL NO.)
54:     C   PARAMETER PLC = PFC+1      ; LCN (LAST CHANNEL NO.)
55:     C   PARAMETER PLS = PLC+1      ; POINTER LIST
56:     C   PARAMETER PSW = PLS+32     ; TRACK SELECT SWITCH
57:     C   PARAMETER PFS = PSW+1      ; FILTER SELECT SWITCH
58:     C   PARAMETER PAD = PFS+1      ; A/D ERROR STATUS
59:     C****

```

```

60:      INCLUDE 'PSFFNDIR'
61:      C**** FORTRAN FILE NOS. FOR DIRECTORIES
62:      PARAMETER IED = 1          ; EVENT DIRECTORY
63:      PARAMETER IML = 2          ; MASTERLIST DIRECTORY (CARD IMAGES)
64:      C****
65:      IMPLICIT INTEGER (A-Z)
66:      COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
67:      COMMON /EVPAZ/ EVENT, PASS, CHANNL
68:      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
69:      C
70:      C
71:      C --- OPEN DIRECTORY FILES
72:      OPEN IED,'DIRECTORY',LEN=2*RECSIZ
73:      OPEN IML,'MASTERLIST',LEN=80
74:      C
75:      C --- READ DIRECTORY RECORD ZERO
76:      READ(IED,REC=1) REC
77:      C
78:      C --- INITIALIZE DATABASE PARAMETERS
79:      NEVENT = REC(ENX)-1
80:      OFFSET = REC(DOF)
81:      C
82:      RETURN
83:      END

```

ENAME

```

1:      SUBROUTINE XENAME(EVTNAME,EVTNUM)
2:      OVERLAY OENAME
3:      C
4:      C      EDITED 10/3/77 BY P.R.STEVENSON TO SAVE STACK SPACE
5:      C
6:      C      *****
7:      C
8:      C      HERE IS A ROUTINE TO READ AN EVENT NAME FROM A FILE CALLED "EVDIR"
9:      C      WHICH RESIDES ON REMOVABLE DISK DPO. THIS IS NOT A STAND-ALONE
10:     C      ROUTINE. SUBROUTINE ENAME SHOULD BE USED IN CONJUNCTION WITH 2
11:     C      OTHER SUBROUTINES, ONE OF WHICH MUST BE CALLED PRIOR TO CALLING
12:     C      SUBROUTINE ENAME. SUBROUTINE ENAME READS RECORDS FROM
13:     C      THE EVDIR FILE, BUT ASSUMES THIS FILE HAS BEEN PREVIOUSLY OPENED
14:     C      WITH A CALL TO SUBROUTINE "DIRC1".
15:     C
16:     C      NOTE: IN ORDER TO USE SUBROUTINE DIRC1, THE USER MUST HAVE CRE-
17:     C      ATED A LINK TO FILE DPO:EVDIR USING THE LINK NAME
18:     C      "DIRECTORY".
19:     C
20:     C      FORMAT OF CALL:  CALL ENAME(NAME,K)
21:     C
22:     C      WHERE K IS (UPON CALL) THE INTEGER NUMBER OF THE EVENT FOR WHICH
23:     C      THE NAME (IN ASCII) IS REQUESTED,
24:     C
25:     C      AND NAME IS A 20 WORD INTEGER ARRAY WHICH WILL BE STUFFED WITH
26:     C      THE 40 CHARACTER EVENT NAME, CORRESPONDING TO THE EVENT NUMBER.
27:     C
28:     C      IF THE EVENT NUMBER PASSED IN ARGUMENT "K" IS LESS THAN 1 OR
29:     C      GREATER THAN THE NUMBER OF EVENTS CONTAINED WITHIN THE EVDIR
30:     C      FILE, SUBROUTINE ENAME RETURNS TO THE CALLING ROUTINE WITH
31:     C      ASTERISKS IN THE 20 WORD "NAME" ARRAY.
32:     C
33:     C      IN MOST INSTANCES THE USER WILL ALSO BE USING THE DATA FILE
34:     C      ASSOCIATED WITH A PARTICULAR EVENT. THIS REQUIRES A CALL
35:     C      TO SUBROUTINE "MCAM1", WHICH OPENS FILE "EVDAT" (ALSO ON DPO).
36:     C
37:     C      NOTE: IN NON-TASKING ENVIRONMENT, USER MUST LINK DPO:EVDAT
38:     C      USING LINK NAME "WHOLE1".
39:     C
40:     C      THE RECOMMENDED CALLING SEQUENCE IS:
41:     C
42:     C          CALL MCAM1
43:     C          CALL DIRC1
44:     C          .
45:     C          .
46:     C          .
47:     C          .
48:     C          .
49:     C          CALL ENAME(NAME,K)
50:     C          .
51:     C          .
52:     C
53:     C      NOTE: SUBROUTINE ENAME MAY BE CALLED SEVERAL TIMES DURING EXECU-
54:     C      TION OF A PROGRAM (OR ANOTHER SUBROUTINE). HOWEVER, SUB-
55:     C      ROUTINE MCAM1 AND DIRC1 SHOULD BE CALLED ONCE AND ONLY
56:     C      ONCE DURING THE EXECUTION OF A PROGRAM.
57:     C
58:     C      *****
59:     C

```

```

60: CC ** MODIFIED FOR 3 PASSES BY P.STEVENSON (5-12-78)
61:     PARAMETER NPASS=3
62:     PARAMETER REKSIZE = 160
63:     PARAMETER IED = 1
64:     PARAMETER ENO = 1
65:     PARAMETER ENA = ENO+1
66:     IMPLICIT INTEGER (A-Z)
67:     DIMENSION EVTNAME(1)
68:     COMMON /IOBDA/ NCHAN,SEGSZ,REMDR,LTBUP,FIRST
69:     COMMON /EDIRC/NEVENT,PASSN(NPASS),REC(REKSIZE)
70:     C
71:     C --- INITIALIZE THE EVENT NAME ARRAY. PUT ASTERISKS IN ALL
72:     C LOCATIONS.
73:     C
74:         DO 5 K=1,20
75:     5     EVTNAME(K) = 2H**
76:     C
77:     C --- FIND OUT HOW MANY EVENTS ARE ALREADY WITHIN THE DIRECTORY
78:     C
79:     MAXEVTS = NEVENT
80:     C
81:     C --- CHECK THE PASSED PARAMETER "EVTNUM" TO SEE IF IT IS WITHIN
82:     C REASONABLE BOUNDS. IF NOT, RETURN.
83:     C
84:     IF (EVTNUM .LT. 1 .OR. EVTNUM .GT. MAXEVTS) GO TO 20
85:     C
86:     C
87:     C --- READ THE RECORD CORRESPONDING TO THE PARTICULAR EVENT NUMBER
88:     C
89:     READ(IED,REC=EVTNUM+1) REC
90:     C
91:     C
92:     C --- NOW FILL THE EVENT NAME ARRAY WITH THE 40 ASCII CHARACTERS
93:     C STORED IN FILE "EVDIR".
94:     C
95:         DO 10 K=1,20
96:     10     EVTNAME(K) = REC( ENO + K )
97:     C
98:     20     RETURN
99:     C
100:     END

```

RQEVF

```

1:      SUBROUTINE XRQEVF(LIST, PARM1)
2:      OVERLAY ORQEVF
3:      C
4:      C  EVENT REQUEST PROCESSOR
5:      C
6:      C  P.A. MARSHALL
7:      C  SEPTEMBER 1976
8:      C -----
9:      C
10:     INCLUDE 'PSDIR'
11:     C**** EVENT DIRECTORY OFFSETS
12:     C
13:     C *** DIRECTORY RECORD SIZE
14:     C     PARAMETER RECSIZ = 160
15:     C
16:     C *** RECORD 0:
17:     C
18:     C     PARAMETER ENX = 1           ; NEXT EVENT (LAST+1)
19:     C     PARAMETER DML = ENX+1      ; NO. OF MASTERLIST ENTRIES
20:     C     PARAMETER DOF = DML+1      ; DATABASE FILE OFFSET
21:     C     PARAMETER DSZ = DOF+1      ; FILE SIZE IN SECTORS
22:     C     PARAMETER DNX = DSZ+1      ; NEXT BLOCK NO.
23:     C     PARAMETER CTE = DNX+1      ; CURRENT EVENT SAVE AREA
24:     C     PARAMETER MPL = CTE+5      ; MASTER POINTER LIST
25:     C     PARAMETER EPL = MPL+95     ; END OF POINTER LIST
26:     C
27:     C *** RECORD N:
28:     C
29:     C     PARAMETER ENO = 1           ; EVENT NO.
30:     C     PARAMETER ENA = ENO+1      ; EVENT NAME
31:     C     PARAMETER EIB = ENA+20     ; INITIAL BLOCK NO.
32:     C     PARAMETER ENB = EIB+1      ; TOTAL NUMBER OF BLOCKS
33:     C     PARAMETER DLT = ENB+1      ; DELTA-T (SECS)
34:     C     PARAMETER DRA = DLT+2      ; DURATION (SECS)
35:     C     PARAMETER EYR = DRA+2      ; YEAR
36:     C     PARAMETER EMO = EYR+1      ; MONTH
37:     C     PARAMETER EDA = EMO+1      ; DAY
38:     C     PARAMETER EHR = EDA+1      ; HOUR
39:     C     PARAMETER EMI = EHR+1      ; MINUTE
40:     C     PARAMETER ESE = EMI+1      ; SECONDS
41:     C     PARAMETER EP1 = ESE+2      ; EVENT, PASS 1
42:     C     PARAMETER EP2 = EP1+1      ; EVENT, PASS 2
43:     C     PARAMETER EP3 = EP2+1      ; EVENT, PASS 3
44:     C
45:     C     PARAMETER ETM = EYR         ; EVENT TIME
46:     C     PARAMETER EPP = EP1-1      ; EVENT, PASS POINTER
47:     C
48:     C *** PASS N:
49:     C
50:     C     PARAMETER ENP = 0           ; ASSOCIATED EVENT NO.
51:     C     PARAMETER PNO = ENP+1      ; PASS NO.
52:     C     PARAMETER PIB = PNO+1      ; INITIAL BLOCK NO.
53:     C     PARAMETER PNB = PIB+1      ; NUMBER OF BLOCKS
54:     C     PARAMETER PFC = PNB+1      ; FCN (FIRST CHANNEL NO.)
55:     C     PARAMETER PLC = PFC+1      ; LCN (LAST CHANNEL NO.)
56:     C     PARAMETER PLS = PLC+1      ; POINTER LIST
57:     C     PARAMETER PSW = PLS+32     ; TRACK SELECT SWITCH
58:     C     PARAMETER PFS = PSW+1      ; FILTER SELECT SWITCH
59:     C     PARAMETER PAD = PFS+1      ; A/D ERROR STATUS

```

```

60: C****
61: INCLUDE 'PSFFNDIR'
62: C**** FORTRAN FILE NOS. FOR DIRECTORIES
63: PARAMETER IED = 1 ; EVENT DIRECTORY
64: PARAMETER IML = 2 ; MASTERLIST DIRECTORY (CARD IMAGES)
65: C****
66: C**** REQUEST LIST OFFSETS
67: C
68: PARAMETER LEN = 1 ; EVENT NO.
69: PARAMETER LET = LEN+1 ; EVENT TIME
70: PARAMETER LED = LET+7 ; DURATION
71: PARAMETER LDT = LED+2 ; DELTA-T
72: PARAMETER LNP = LDT+2 ; NUMBER OF PASSES
73: PARAMETER LM1 = LNP+1 ; MUX, PASS 1
74: PARAMETER LM2 = LM1+1 ; MUX, PASS 2
75: PARAMETER LM3 = LM2+1 ; MUX, PASS 3
76: C
77: PARAMETER LMP = LM1-1 ; MUX FINDER
78: PARAMETER LSZ = LM3-LEN+1 ; LIST SIZE
79: C****
80: IMPLICIT INTEGER (A-Z)
81: COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
82: COMMON /EVPZ/ EVENT, PASS, CHANNL
83: DIMENSION LIST(LSZ)
84: C
85: C --- CLEAR LIST
86: DO 1 J=1,LSZ
87: 1 LIST(J) = 0
88: C
89: C --- SET EVENT, CHECK FOR VALIDITY
90: EVENT = PARM1
91: IF (EVENT .LE. 0) RETURN
92: C
93: C --- READ IN DIRECTORY RECORD, CHECK VALIDITY
94: READ(IED,REC=EVENT+1) REC
95: IF (REC(ENO) .EQ. 0) RETURN
96: C
97: C --- STUFF PARAMETERS FROM DIRECTORY RECORD INTO REQUEST LIST
98: LIST(LEN) = REC(ENO)
99: LIST(LET) = REC(EYR)
100: LIST(LET+1) = REC(EMD)
101: LIST(LET+2) = REC(EDA)
102: LIST(LET+3) = REC(EHR)
103: LIST(LET+4) = REC(EMI)
104: LIST(LET+5) = REC(ESE)
105: LIST(LET+6) = REC(ESE+1)
106: LIST(LED) = REC(DRA)
107: LIST(LED+1) = REC(DRA+1)
108: LIST(LDT) = REC(DLT)
109: LIST(LDT+1) = REC(DLT+1)
110: C
111: C --- FIND WHICH PASSES EXIST AND INTERPRET AS A PASS COUNT
112: K = 0
113: DO 2 J=1,3
114: C --- CLEAR PASS MEMORY
115: PASSN(J) = 0
116: C --- POINT TO PASS. SEE IF IT EXISTS
117: PNT = REC(EPP+J)
118: NCHAN = REC(PNT+PLC)-REC(PNT+PFC)+1
119: IF (NCHAN .EQ. 1 .AND. REC(PNT+PFC) .EQ. 0) GO TO 2
120: C
121: C --- HAVE PASS. COUNT PASS, SAVE PASS NO., STUFF MUX NO.
122: K = K+1

```



```
123:      PASSN(K) = J
124:      LIST(LMP+K) = NCHAN
125:      2 CONTINUE
126:      C
127:      C --- STUFF PASS COUNT
128:      LIST(LNP) = K
129:      RETURN
130:      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 4:01:58 PM

RQPAZ

```
1:      SUBROUTINE XRQPAZ(LIST, PARM1, PARM2)
2:      OVERLAY ORQPAZ
3:      C
4:      C EVENT/PASS REQUEST PROCESSOR
5:      C
6:      C P.A. MARSHALL
7:      C SEPTEMBER 1976
8:      C-----
9:      C
10:     INCLUDE 'PSDIR'
11:     C**** EVENT DIRECTORY OFFSETS
12:     C
13:     C *** DIRECTORY RECORD SIZE
14:     C     PARAMETER RECSIZ = 160
15:     C
16:     C *** RECORD 0:
17:     C
18:     C     PARAMETER ENX = 1           ; NEXT EVENT (LAST+1)
19:     C     PARAMETER DML = ENX+1      ; NO. OF MASTERLIST ENTRIES
20:     C     PARAMETER DOF = DML+1      ; DATABASE FILE OFFSET
21:     C     PARAMETER DSZ = DOF+1      ; FILE SIZE IN SECTORS
22:     C     PARAMETER DNX = DSZ+1      ; NEXT BLOCK NO.
23:     C     PARAMETER CTE = DNX+1      ; CURRENT EVENT SAVE AREA
24:     C     PARAMETER MPL = CTE+5      ; MASTER POINTER LIST
25:     C     PARAMETER EPL = MPL+95     ; END OF POINTER LIST
26:     C
27:     C *** RECORD N:
28:     C
29:     C     PARAMETER ENO = 1           ; EVENT NO.
30:     C     PARAMETER ENA = ENO+1      ; EVENT NAME
31:     C     PARAMETER EIB = ENA+20     ; INITIAL BLOCK NO.
32:     C     PARAMETER ENB = EIB+1      ; TOTAL NUMBER OF BLOCKS
33:     C     PARAMETER DLT = ENB+1      ; DELTA-T (SECS)
34:     C     PARAMETER ORA = DLT+2      ; DURATION (SECS)
35:     C     PARAMETER EYR = ORA+2      ; YEAR
36:     C     PARAMETER EMO = EYR+1      ; MONTH
37:     C     PARAMETER EDA = EMO+1      ; DAY
38:     C     PARAMETER EHR = EDA+1      ; HOUR
39:     C     PARAMETER EMI = EHR+1      ; MINUTE
40:     C     PARAMETER ESE = EMI+1      ; SECONDS
41:     C     PARAMETER EP1 = ESE+2      ; EVENT, PASS 1
42:     C     PARAMETER EP2 = EP1+1      ; EVENT, PASS 2
43:     C     PARAMETER EP3 = EP2+1      ; EVENT, PASS 3
44:     C
45:     C     PARAMETER ETM = EYR         ; EVENT TIME
46:     C     PARAMETER EPP = EP1-1      ; EVENT, PASS POINTER
47:     C
48:     C *** PASS N:
49:     C
50:     C     PARAMETER ENP = 0           ; ASSOCIATED EVENT NO.
51:     C     PARAMETER PNO = ENP+1      ; PASS NO.
52:     C     PARAMETER PIB = PNO+1      ; INITIAL BLOCK NO.
53:     C     PARAMETER PNB = PIB+1      ; NUMBER OF BLOCKS
54:     C     PARAMETER PFC = PNB+1      ; FCN (FIRST CHANNEL NO.)
55:     C     PARAMETER PLC = PFC+1      ; LCN (LAST CHANNEL NO.)
56:     C     PARAMETER PLS = PLC+1      ; POINTER LIST
57:     C     PARAMETER PSW = PLS+32     ; TRACK SELECT SWITCH
58:     C     PARAMETER PFS = PSW+1      ; FILTER SELECT SWITCH
59:     C     PARAMETER PAD = PFS+1      ; A/D ERROR STATUS
```

```

60:  C****
61:  INCLUDE 'PSFFNDIR'
62:  C**** FORTRAN FILE NOS. FOR DIRECTORIES
63:  PARAMETER IED = 1 ; EVENT DIRECTORY
64:  PARAMETER IML = 2 ; MASTERLIST DIRECTORY (CARD IMAGES)
65:  C****
66:  C**** REQUEST LIST PARAMETERS
67:  PARAMETER NSTAR = 2 ; WIDTH OF LIST ENTRY (IN WORDS)
68:  PARAMETER LSZ = 32*NSTAR ; LIST SIZE
69:  PARAMETER LST = 1 ; START OF LIST
70:  C****
71:  IMPLICIT INTEGER (A-Z)
72:  COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
73:  COMMON /EVPAZ/ EVENT, PASS, CHANNL
74:  COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
75:  COMMON /IOBDA/ NCHAN, IDUM3(4)
76:  DIMENSION LIST(LSZ), STAREC(40)
77:  C
78:  C --- CLEAR LIST
79:  DO 1 J=1,LSZ
80:  1 LIST(J) = 0
81:  C
82:  C --- CHECK EVENT AND PASS FOR VALIDITY
83:  IF (PARM1.NE. EVENT) RETURN
84:  PASS = PARM2
85:  IF (PASS.LT. 1 .OR. PASS.GT. 3) RETURN
86:  IF (PASSN(PASS).EQ. 0) RETURN
87:  C --- POINT TO PASS
88:  PNT = REC(EPP+PASSN(PASS))
89:  C
90:  C --- SET "MCAM" PARAMETERS
91:  EPISEC = REC(PNT+PIB)-OFFSET
92:  LSEC = EPISEC+REC(PNT+PNB)-1
93:  FCN = REC(PNT+PFC)
94:  LCN = REC(PNT+PLC)
95:  NCHAN = LCN-FCN+1
96:  C
97:  C --- SET POINTER LIST, FETCH STATION LIST DATA
98:  M = PNT+PLS-1
99:  K = LST
100:  DO 2 J=FCN,LCN
101:  STAREC(1) = 0
102:  C --- GET STATION RECORD FROM MASTERLIST
103:  MP = REC(M+J)
104:  IF (MP.LE. 0) GO TO 21
105:  READ(IML,REC=MP) STAREC
106:  21 CONTINUE
107:  C --- MOD (4/22/77) BY P.R.S.; IJK FORMERLY 2
108:  C .IJK=1 FOR FORMAT 2, SET IJK=2 FOR FORMAT 1
109:  IJK=1
110:  C --- OR SHOW AS 'IRIG' CHANNEL
111:  IF (MP.EQ. -1) CALL FBLM(STAREC(IJK), 'IRIG', 2)
112:  C --- OR 'WVB' CHANNEL
113:  IF (MP.EQ. -2) CALL FBLM(STAREC(IJK), 'WVB', 2)
114:  C --- OR CLEAR RECORD IF "UNDEFINED" CHANNEL
115:  IF (MP.EQ. 0) CALL FBLM(STAREC(2), STAREC(1), 39)
116:  C
117:  C --- MOVE DATA INTO REQUEST LIST
118:  CALL FBLM(LIST(K), STAREC(IJK), 2)
119:  2 K = K+NSTAR
120:  C
121:  RETURN
122:  END

```

MCAM1

```

1:      SUBROUTINE MCAM1
2:      C
3:      C   M C A M   INITIALIZER MODULE
4:      C
5:      C   P.A. MARSHALL
6:      C   MARCH 1976
7:      C-----
8:      C
9:      C   --- "MULTIPLEXED CHANNEL ACCESS METHOD" (MCAM) IS AN I/O SUBSYSTEM
10:     C   PROVIDING INPUT IN SEGMENTED SERIAL FORM OF MULTIPLEXED CHANNEL
11:     C   DATA STORED ON DISK.
12:     C
13:     C   --- THIS ROUTINE IS THE INITIALIZER FOR THE SUITE OF MCAM MODULES
14:     C   AND MUST BE CALLED ONCE DURING THE EXECUTION OF A PROGRAM BEFORE
15:     C   ANY OF THE OTHER MCAM MODULES ARE USED.
16:     C
17:     C
18:     C   --- USE
19:     C
20:     C       CALL MCAM1
21:     C
22:     C   THE CALL IS MADE BEFORE ANY OTHER MCAM MODULES ARE USED. ONLY ONE
23:     C   SUCH CALL SHOULD BE MADE DURING THE EXECUTION OF A PROGRAM.
24:     C
25:     C   MCAM MODULES:
26:     C
27:     C       MCAM1 - "MCAM" INITIALIZER
28:     C       SEGMO - SEGMENT HEAD-IN
29:     C       SEGIN - SEGMENT BUFFER QUEUING
30:     C       DISKI - SEGMENT INPUT TASK
31:     C
32:     C   USER INTERFACE ROUTINES:
33:     C
34:     C       FRAM0 - INPUT FRAME DEFINITION
35:     C       GETFM - DATA FRAME INPUT
36:     C
37:     C   OTHER SUBPROGRAMS REQUIRED:
38:     C
39:     C       DEMUX - CHANNEL DEMULTIPLEXING
40:     C       BLM   - BLOCK MOVE, DATA IN CORE
41:     C
42:     C-----
43:     C
44:     C   INCLUDE 'MCAMSD'
45:     C**** PARAMETERS - DISK ORGANIZATION
46:     C       PARAMETER NWT = 3072           !NO. OF WORDS PER TRACK
47:     C       PARAMETER NST = 12             !NO. OF SECTORS PER TRACK
48:     C       PARAMETER NWS = 256           !NO. OF WORDS PER SECTOR
49:     C****
50:     C   INCLUDE 'MCAMSP'
51:     C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
52:     C       PARAMETER SEGMENT = 2          !SEGMENT BUFFER SIZE IN SECTORS
53:     C       PARAMETER BUFCBL = 5           !BUFFER CONTROL BLOCK LENGTH
54:     C       PARAMETER MAXCHN = 32          !MAXIMUM MULTIPLEX FACTOR
55:     C       PARAMETER NBUFS = 2            !NO. OF SEGMENT BUFFERS (2 MINIMUM)
56:     C   *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
57:     C       PARAMETER BUFRP = NBUFS*(NWS*SEGMENT+MAXCHN-1)
58:     C       PARAMETER BUFCHP = NBUFS*BUFCBL
59:     C****

```

```

60:      INCLUDE 'MCAMSB'
61:      C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
62:      PARAMETER BUFAD = 0           ;BUFFER ADDRESS (INDEX)
63:      PARAMETER BUFLN = 1           ;BUFFER LENGTH
64:      PARAMETER BUFLAG = 2          ;BUFFER STATUS FLAG
65:      PARAMETER SECNO = 3           ;RELATIVE SECTOR NO.
66:      PARAMETER BUFPNT = 4          ;POINTER TO NEXT BUFFER
67:      C****
68:      INCLUDE 'MCAMSI'
69:      C**** PARAMETERS - MCAM INITIALIZER
70:      PARAMETER IDAFNO = 0           ;FORTRAN FILE NO. FOR DATABASE FILE
71:      PARAMETER EVNDA = 1           ;EVENT FOR 'DISKI' TASK
72:      PARAMETER SECEND = 479        ;LAST SECTOR NO. IN DATABASE FILE
73:      C****
74:      IMPLICIT INTEGER (A-Z)
75:      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
76:      COMMON /IOBDA/ NCHAN, SEGSZ, MEMDR, LTBUF, FIRST
77:      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP), BUFFER(BUFFERP)
78:      C
79:      C
80:      C --- INITIALIZE BUFFER CONTROL BLOCKS
81:      C
82:      JBUFAD = MAXCHN
83:      SEGSZ = NWS*SEGMNT
84:      C
85:      C --- LOOP ONCE FOR EACH BLOCK
86:      BUFP = -BUFCBL+1
87:      DO 1 J=1,NBUFS
88:      BUFP = BUFP+BUFCBL
89:      BUFCB(BUFP+BUFAD) = JBUFAD
90:      BUFCB(BUFP+BUFLN) = SEGSZ
91:      BUFCB(BUFP+BUFLAG) = -1
92:      BUFCB(BUFP+SECNO) = -1
93:      BUFCB(BUFP+BUFPNT) = BUFP+BUFCBL
94:      1 JBUFAD = JBUFAD+SEGSZ+MAXCHN-1
95:      BUFCB(BUFP+BUFPNT) = 1
96:      C
97:      IFNO = IDAFNO
98:      EVENT = EVNOA
99:      NSEC = SEGMNT
100:     C
101:     C --- OPEN THE DATABASE FILE
102:     OPEN IFNO,'WHOLE1',ATT='C',REC=SECEND+1
103:     C
104:     C --- SETUP SOME DEFAULTS
105:     EPISEC = 0
106:     OFFSET = 0
107:     LSEC = SECEND
108:     C
109:     RETURN
110:     END

```

FRAMD

```

1:      SUBROUTINE FRAMD(WIDTH, KSEL, K, ISCAN, IERR)
2:      C
3:      C  FRAME DEFINITION
4:      C
5:      C  P.A. MARSHALL
6:      C  MARCH 1976
7:      C -----
8:      C
9:      C --- THIS ROUTINE INITIALIZES THE FILE POINTERS AND DEFINES THE
10:     C FRAME FOR ACCESSING CHANNELS FROM THE EVENT FILE.  SUBSEQUENT CALLS
11:     C TO SUBROUTINE 'GETFM' WILL ACCESS THE FILE SEQUENTIALLY, BRINGING
12:     C IN "FRAMES" OF DATA AS DEFINED BELOW, AND STARTING WITH THE INITIAL
13:     C SCAN SPECIFIED BY 'ISCAN'.
14:     C
15:     C      A FRAME IS DEFINED TO BE AN ORDERED SET OF CHANNEL SEGMENTS
16:     C OF EQUAL SIZE.  THE SIZE OF A SEGMENT IS CALLED THE FRAME 'WIDTH'.
17:     C DATA IN EACH SEGMENT OCCURS SERIALY AND HAS BEEN DEMULTIPLEXED IN
18:     C PARALLEL WITH EVERY OTHER SEGMENT.  THERE ARE 'K' SEGMENTS IN A
19:     C FRAME, EACH SEGMENT (I) BEING IDENTIFIED WITH THE CHANNEL NUMBER
20:     C GIVEN BY THE ASSOCIATED ELEMENT 'KSEL(I)' FROM THE ARRAY OF
21:     C CHANNEL SELECT ORDINALS 'KSEL'.  ANY CHANNEL NUMBER IDENTIFIED IN
22:     C 'KSEL' NEED NOT BE UNIQUE.
23:     C
24:     C --- STRUCTURE OF A FRAME CONSISTING OF K CHANNELS:
25:     C
26:     C      -----
27:     C      ! CHAN. # !      !_ FRAME WIDTH
28:     C      ! KSEL(1) !      !
29:     C      -----
30:     C      ! CHAN. # !
31:     C      ! KSEL(2) !
32:     C      -----
33:     C      !      :      !
34:     C      !      :      !
35:     C      -----
36:     C      ! CHAN. # !
37:     C      ! KSEL(K) !
38:     C      -----
39:     C
40:     C -- THE FILE IS INITIALLY POSITIONED BY A CALL TO 'FRAMD' AT THE
41:     C SCAN SPECIFIED BY 'ISCAN' WITHIN THE CURRENT EVENT.  THIS SCAN
42:     C WILL BE MADE AVAILABLE AS THE FIRST SCAN OF THE FIRST FRAME
43:     C ACCESSED BY THE NEXT CALL TO 'GETFM'.  SUBSEQUENT CALLS TO 'GETFM'
44:     C WILL RETNIEVE FRAMES SEQUENTIALLY FROM THE INITIAL ONE.
45:     C
46:     C      THE CALL TO 'FRAMD' ALSO EFFECTS INITIALIZATION OF THE MCAM
47:     C I/O MODULES.
48:     C
49:     C
50:     C --- USE
51:     C
52:     C      CALL FRAMD(WIDTH, KSEL, K, ISCAN, IERR)
53:     C
54:     C WHERE, AT CALL
55:     C
56:     C      WIDTH  - FRAME WIDTH IN NO. OF SCANS PER CHANNEL
57:     C
58:     C      KSEL   - THE ARRAY OF CHANNEL SELECT ORDINALS (CHANNEL NOS.)
59:     C

```

```

60: C      K      - THE NO. OF CHANNELS SELECTED (DIMENSION OF 'KSEL')
61: C
62: C      ISCAN  - THE INITIAL SCAN NO. (FIRST SCAN IN FILE = 1)
63: C
64: C      AND UPON RETURN
65: C
66: C      IERR   - FILE STATUS RETURN. THE FOLLOWING VALUES ARE DEFINED:
67: C
68: C      :           IERR = 0 - NORMAL RETURN (FRAME DEF. & FILE OK)
69: C      :           IERR = -1 - INVALID ARGUMENTS IN 'FRAMD' CALL
70: C      :           IERR = 1 - INVALID FILE POSITION (ISCAN NOT THERE)
71: C
72: C-----
73: C
74: C      INCLUDE 'MCAMSD'
75: C**** PARAMETERS - DISK ORGANIZATION
76: C      PARAMETER NWT = 3072           ;NO. OF WORDS PER TRACK
77: C      PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
78: C      PARAMETER NWS = 256           ;NO. OF WORDS PER SECTOR
79: C****
80: C      INCLUDE 'MCAMSP'
81: C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
82: C      PARAMETER SEGMENT = 2          ;SEGMENT BUFFER SIZE IN SECTORS
83: C      PARAMETER BUFCBL = 5           ;BUFFER CONTROL BLOCK LENGTH
84: C      PARAMETER MAXCHN = 32          ;MAXIMUM MULTIPLEX FACTOR
85: C      PARAMETER NBUFS = 2            ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
86: C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
87: C      PARAMETER BUFRP = NBUFS*(NWS*SEGMENT+MAXCHN-1)
88: C      PARAMETER BUFCBP = NBUFS*BUFCBL
89: C****
90: C      INCLUDE 'MCAMSB'
91: C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
92: C      PARAMETER BUFAO = 0             ;BUFFER ADDRESS (INDEX)
93: C      PARAMETER BUFLN = 1             ;BUFFER LENGTH
94: C      PARAMETER BUFLAG = 2            ;BUFFER STATUS FLAG
95: C      PARAMETER SECNO = 3             ;RELATIVE SECTOR NO.
96: C      PARAMETER BUPNT = 4             ;POINTER TO NEXT BUFFER
97: C****
98: C      IMPLICIT INTEGER (A-Z)
99: C      LOGICAL FIRST, INIT
100: C      DIMENSION KSEL(1)
101: C      COMMON /IOSDA/ NCHAN, SEGSZ, REMDR, LTBUF, FIRST
102: C      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC
103: C      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP)
104: C      COMMON /UFRAM/ WID, KCHAN, KEY(MAXCHN)
105: C
106: C
107: C --- CHECK ALL ARGUMENTS FOR VALIDITY
108: C      IERR = 0
109: C      IF (MIN0(WIDTH, K, ISCAN) .LE. 0) GO TO 51
110: C      IF (K .GT. MAXCHN) GO TO 51
111: C      DO 11 J=1,K
112: C      IF (KSEL(J) .LE. 0 .OR. KSEL(J) .GT. NCHAN) GO TO 51
113: C      11 CONTINUE
114: C
115: C --- STORE ARGUMENTS IN FRAME DEFINITION TABLE
116: C      DO 12 J=1,K
117: C      12 KEY(J) = KSEL(J)
118: C      KCHAN = K
119: C      WID = WIDTH
120: C
121: C      FIRST = .TRUE.
122: C      INIT = .TRUE.

```

```

123: C
124: C --- ALGORITHM TO COMPUTE THE INITIAL SECTOR NO. AND REMAINDER
125: C   FOR A SPECIFIED INITIAL SCAN.
126: C
127: C --- THE FOLLOWING VARIABLES ARE DEFINED:
128: C       NCHAN  = NO. OF CHANNELS THIS EVENT
129: C       SEGSZ  = SEGMENT SIZE (IN WORDS)
130: C       NSPT   = NO. OF SCANS PER TRACK
131: C       NWL    = LOGICAL TRACK LENGTH (IN WORDS)
132: C       ISCAN  = INITIAL SCAN NO. (FIRST = 1)
133: C       ITRK   = INITIAL, RELATIVE TRACK NO. (FIRST = 0)
134: C       IPTS   = WORD COUNT OFFSET IN INITIAL TRACK (INCLUDES 1ST SCAN)
135: C       LSEG   = SEGMENT NO. CONTAINING LAST POINT IN INITIAL SCAN
136: C       ISEG   = SEGMENT NO. CONTAINING FIRST POINT IN INITIAL SCAN
137: C       EPISEC = EVENT/PASS FILE OFFSET (RELATIVE SECTOR)
138: C       ISEC   = INITIAL, RELATIVE SECTOR NO. (FILE POSITION)
139: C
140: C       NSPT = NAT/NCHAN
141: C       NWL  = NSPT*NCHAN
142: C       ITRK = (ISCAN-1)/NSPT
143: C       IPTS = NCHAN*ISCAN-ITRK*NWL
144: C       LSEG = (IPTS-1)/SEGSZ
145: C       ISEG = (IPTS-NCHAN)/SEGSZ
146: C
147: C       ISEC = ITRK*NSPT+ISEG*NSECT+EPISEC
148: C --- CHECK FOR VALID FILE POSITION
149: C       IF (ISEC .GT. LSEC) GO TO 52
150: C
151: C --- INITIALIZE MCAM MODULES
152: C
153: C --- FILL THE FIRST 'NBUFS-1' BUFFERS
154: C       BUFP = 1
155: C       N = NBUFS-1
156: C       DO 1 J=1,N
157: C         CALL SEGIN
158: C       1 BUFP = BUFCB(BUFP+BUFPNT)
159: C
160: C --- COMPUTE NEGATIVE REMAINDER TO OFFSET OF INITIAL SCAN
161: C       REMOR = -MOD(IPTS-NCHAN, SEGSZ)
162: C       LTHUF = BUFP
163: C       BUFP = 1
164: C
165: C --- IF FIRST SCAN SPANS A SEGMENT, INPUT NEW BUFFER AND ACCEPT
166: C   NEW REMAINDER
167: C       IF (LSEG .NE. ISEG) CALL SEGMO(952, IDUM, IDUM)
168: C       RETURN
169: C
170: C --- ERRORS
171: C
172: C   51 IERR = -1
173: C       RETURN
174: C
175: C   52 IERR = 1
176: C       RETURN
177: C       END

```


GETFM

```

1:      SUBROUTINE GETFM(FRAME, M, IEND)
2:      C
3:      C   FRAME INPUT MODULE
4:      C
5:      C   P.A. MARSHALL
6:      C   MARCH 1976
7:      C-----
8:      C
9:      C   --- THIS ROUTINE FILLS A FRAME WITH DEMULTIPLEXED DATA FROM THE
10:     C   EVENT FILE. THE FRAME MUST HAVE BEEN PREVIOUSLY DEFINED BY A CALL
11:     C   TO 'FRAMD'. DATA IS LOADED INTO THE FRAME BEGINNING WITH THE NEXT
12:     C   SEQUENTIAL SCAN IN THE FILE.
13:     C
14:     C   THE NUMBER OF SCANS PER CHANNEL LOADED INTO THE FRAME IS
15:     C   RETURNED AS THE ARGUMENT 'M'. THE VALUE OF 'M' IS NORMALLY FIXED
16:     C   AND EQUAL TO THE FRAME "WIDTH," HOWEVER WHEN AN END-OF-FILE
17:     C   CONDITION OCCURS WHILE LOADING THE FRAME THE VALUE RETURNED FOR
18:     C   'M' WILL BE LESS THAN THE WIDTH AND WILL REFLECT THE ACTUAL NUMBER
19:     C   OF SCANS LOADED INTO THE FRAME. (IT IS TO BE NOTED THAT THE FRAME
20:     C   WILL NOT BE COMPRESSED IN THIS LATTER CASE, SO THAT THOSE PORTIONS
21:     C   OF THE FRAME NOT FILLED WILL CONTAIN INVALID DATA.)
22:     C
23:     C
24:     C   --- USE
25:     C
26:     C       CALL GETFM(FRAME, M, IEND)
27:     C
28:     C   WHERE, AT CALL
29:     C
30:     C       FRAME - (INTEGER ARRAY) IS THE DATA AREA WHICH WILL RECEIVE
31:     C               THE FRAME
32:     C
33:     C   AND UPON RETURN
34:     C
35:     C       M       - RECEIVES A VALUE EQUAL TO THE NUMBER OF SCANS PER
36:     C               CHANNEL ACTUALLY LOADED INTO THE FRAME
37:     C
38:     C       IEND    - END OF FILE INDICATOR, AS FOLLOWS:
39:     C
40:     C               IEND = 0 - NORMAL RETURN (M = "WIDTH")
41:     C               IEND = 1 - END-OF-FILE (0 >= M < "WIDTH")
42:     C                       ("WIDTH" HAVING BEEN DEFINED BY A CALL TO 'FRAMD')
43:     C
44:     C-----
45:     C
46:     C   INCLUDE 'MCAMSD'
47:     C**** PARAMETERS - DISK ORGANIZATION
48:     C       PARAMETER NWT = 3072          ;NO. OF WORDS PER TRACK
49:     C       PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
50:     C       PARAMETER NWS = 256          ;NO. OF WORDS PER SECTOR
51:     C****
52:     C   INCLUDE 'MCAMSP'
53:     C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
54:     C       PARAMETER SEGMENT = 2         ;SEGMENT BUFFER SIZE IN SECTORS
55:     C       PARAMETER BUFCBL = 5          ;BUFFER CONTROL BLOCK LENGTH
56:     C       PARAMETER MAXCHN = 32         ;MAXIMUM MULTIPLEX FACTOR
57:     C       PARAMETER NBUFS = 2           ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
58:     C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
59:     C       PARAMETER BUFRP = NBUFS*(NWS*SEGMENT*MAXCHN-1)

```

```

60:      PARAMETER BUFCBP = NBUFS*BUFCBL
61:      C****
62:      IMPLICIT INTEGER (A-Z)
63:      LOGICAL FIRST
64:      COMMON /UFRAM/ WIDTH, KCHAN, KSEL(MAXCHN)
65:      COMMON /IOBDA/ NCHAN, DUM(3), FIRST
66:      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP), BUFFER(BUFFRP)
67:      DIMENSION FRAME(1)
68:      STATIC NSCAN, BUFPT
69:      C
70:      C
71:      C --- INITIALIZE FRAME INDEX AND COUNT
72:      M = 0
73:      C
74:      C --- ON FIRST CALL INITIALIZE SEGMENT BUFFER SCAN COUNT
75:      IF (FIRST) NSCAN = 0
76:      FIRST = .FALSE.
77:      C
78:      C --- LOOP TO LOAD THE FRAME
79:      C
80:      C --- BRING IN NEW SEGMENT AS REQUIRED
81:      10 IF (NSCAN .EQ. 0) CALL SEGMO($50, BUFPT, NSCAN)
82:      C
83:      C --- FAN OUT TO FRAME AS MANY SCANS AS REQUIRED OR AVAILABLE
84:      NMUX = MIN0(NSCAN, WIDTH-M)
85:      CALL DEMUX(BUFFER(BUFPT), NCHAN, NMUX, WIDTH-NMUX, KSEL, KCHAN,
86:      * FRAME(M+1))
87:      C
88:      C --- UPDATE BUFFER POINTER AND COUNTS
89:      BUFPT = BUFPT+(NMUX*NCHAN)
90:      NSCAN = NSCAN-NMUX
91:      M = M+NMUX
92:      C
93:      C --- SEE IF WE HAVE FULLY LOADED FRAME
94:      IF (M .LT. WIDTH) GO TO 10
95:      IEND = 0
96:      RETURN
97:      C
98:      C --- END OF EVENT FILE
99:      50 IEND = 1
100:      RETURN
101:      END

```

BEGIN

```

1:      SUBROUTINE BEGIN
2:      C
3:      C   M C A M SUBSYSTEM - SEGMENT BUFFER QUEUING FOR INPUT
4:      C
5:      C   P.A. MARSHALL
6:      C   FEBRUARY 1976
7:      C-----
8:      C
9:      C --- THE ROUTINE REMEMBERS THE INITIAL SECTOR NO. OF THE LAST
10:     C   SEGMENT INPUT.
11:     C
12:     INCLUDE 'MCAMSD'
13:     C**** PARAMETERS - DISK ORGANIZATION
14:     PARAMETER NWT = 3072           !NO. OF WORDS PER TRACK
15:     PARAMETER NST = 12            !NO. OF SECTORS PER TRACK
16:     PARAMETER NWS = 256           !NO. OF WORDS PER SECTOR
17:     C****
18:     INCLUDE 'MCAMSP'
19:     C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
20:     PARAMETER SEGMENT = 2          !SEGMENT BUFFER SIZE IN SECTORS
21:     PARAMETER BUFCBL = 5           !BUFFER CONTROL BLOCK LENGTH
22:     PARAMETER MAXCHN = 32          !MAXIMUM MULTIPLEX FACTOR
23:     PARAMETER NBUFS = 2            !NO. OF SEGMENT BUFFERS (2 MINIMUM)
24:     C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
25:     PARAMETER BUFRP = NBUFS*(NWS*SEGMENT+MAXCHN-1)
26:     PARAMETER BUFCBP = NBUFS*BUFCBL
27:     C****
28:     INCLUDE 'MCAMSB'
29:     C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
30:     PARAMETER BUFAD = 0            !BUFFER ADDRESS (INDEX)
31:     PARAMETER BUFLN = 1            !BUFFER LENGTH
32:     PARAMETER BUFLAG = 2           !BUFFER STATUS FLAG
33:     PARAMETER SECNO = 3            !RELATIVE SECTOR NO.
34:     PARAMETER BUFPNT = 4           !POINTER TO NEXT BUFFER
35:     C****
36:     IMPLICIT INTEGER (A-Z)
37:     LOGICAL INIT
38:     COMMON /IOPOA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
39:     COMMON /IOBQA/ DUM(2), REMDR
40:     COMMON /BUFDA/ BUFRM, BUFCB(BUFCBP)
41:     STATIC SECTOR
42:     C
43:     C
44:     IF (INIT) SECTOR = ISEC
45:     C
46:     C --- LOG SECTOR NO. FOR SEGMENT
47:     BUFCB(BUFRM+SECNO) = SECTOR
48:     C
49:     C --- CLEAR BUFFER FLAG AND START I/O
50:     BUFCB(BUFRM+BUFLAG) = 0
51:     C**** TASK WAKUP
52:     CALL DISKI
53:     C
54:     SECTOR = SECTOR+NSEC
55:     INIT = .FALSE.
56:     RETURN
57:     END

```

SEGMO

```

1:      SUBROUTINE SEGMO(S, BSTART, NSCAN)
2:      C
3:      C   M C A M' SUBSYSTEM - SEGMENT HEAD-IN
4:      C
5:      C   P.A. MARSHALL
6:      C   FEBRUARY 1976
7:      C-----
8:      C
9:      C --- READ IN NEXT SEGMENT, PREFIX REMAINDER, IDENTIFY START OF
10:     C   BUFFER (BSTART), RETURN NO. OF SCANS (NSCAN) IN THIS BUFFER
11:     C   SEGMENT.
12:     C
13:     C --- DEFINITION: A "SEGMENT" IS THE BUFFERED UNIT OF DATA READ IN
14:     C   FROM DISK, CONSISTING OF "NSEC" CONTIGUOUS SECTORS. THERE ARE
15:     C   AN INTEGRAL NUMBER OF SUCH SEGMENTS PER TRACK.
16:     C
17:     C-----
18:     C
19:     C   INCLUDE 'MCAMSD'
20:     C**** PARAMETERS - DISK ORGANIZATION
21:     C   PARAMETER NWT = 3072          %NO. OF WORDS PER TRACK
22:     C   PARAMETER NST = 12           %NO. OF SECTORS PER TRACK
23:     C   PARAMETER NWS = 256         %NO. OF WORDS PER SECTOR
24:     C****
25:     C   INCLUDE 'MCAMSP'
26:     C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
27:     C   PARAMETER SEGMNT = 2         %SEGMENT BUFFER SIZE IN SECTORS
28:     C   PARAMETER BUFCBL = 5         %BUFFER CONTROL BLOCK LENGTH
29:     C   PARAMETER MAXCHN = 32        %MAXIMUM MULTIPLEX FACTOR
30:     C   PARAMETER NBUFS = 2         %NO. OF SEGMENT BUFFERS (2 MINIMUM)
31:     C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
32:     C   PARAMETER BUFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
33:     C   PARAMETER BUFCBP = NBUFS*BUFCBL
34:     C****
35:     C   INCLUDE 'MCAMSB'
36:     C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
37:     C   PARAMETER BUFPD = 0          %BUFFER ADDRESS (INDEX)
38:     C   PARAMETER BUFLN = 1          %BUFFER LENGTH
39:     C   PARAMETER BUFLAG = 2         %BUFFER STATUS FLAG
40:     C   PARAMETER SECNO = 3          %RELATIVE SECTOR NO.
41:     C   PARAMETER BUFPNT = 4         %POINTER TO NEXT BUFFER
42:     C****
43:     C   IMPLICIT INTEGER (A-Z)
44:     C   COMMON /IOBDA/ NCHAN, SEGSZ, REMDR, LTBUF
45:     C   COMMON /BUFDA/ BUFRP, BUFCB(BUFCBP), BUFB(BUFRP)
46:     C
47:     C
48:     C --- CHECK STATUS OF LAST READ
49:     C**** TASK CALL REC
50:     C   STAT = BUFCB(BUFRP+BUFLAG)
51:     C   IF (STAT.EQ. 9) RETURN 1
52:     C   CALL CHECK(STAT)
53:     C
54:     C --- COMPUTE STARTING ADDRESS OF BUFFER; I.E., SEGMENT ADDRESS
55:     C   MINUS PREFIX
56:     C --- REMAINDER IS ZEROED AT START OF A NEW TRACK
57:     C   IF (REMDR.GT. 0 .AND.
58:     C   * MOD(BUFCB(BUFRP+SECNO), NST).EQ. 0) REMDR = 0
59:     C   BSTART = BUFCB(BUFRP+BUFPD)-REMDR

```

```

60: C
61: C --- IF REMAINDER FROM PRIOR READ IS POSITIVE THEN MOVE IT TO PREFIX
62: IF (REMDR .GT. 0) CALL FBLM(BUFF(BSTART),
63: * BUFF(BUFCB(LTBUF+BUFAD)+SEGSZ-REMDR), REMDR)
64: C
65: C --- COMPUTE CURRENT BUFFER SIZE (INCLUDES PREFIX) AND NO. OF SCANS
66: BUFSZ = SEGSZ+REMDR
67: NSCAN = BUFSZ/NCHAN
68: C
69: C --- COMPUTE NEW REMAINDER
70: REMDR = MOD(BUFSZ, NCHAN)
71: C
72: C --- SAVE CURRENT BUFFER, START INPUT ON LAST BUFFER (NOW AVAILABLE)
73: BUFSV = BUFPRM
74: BUFPRM = LTBUF
75: CALL BEGIN
76: C
77: C --- CHAIN TO NEW SEGMENT BUFFERS
78: LTBUF = BUFSV
79: BUFPRM = BUFCB(BUFSV+BUFPT)
80: RETURN
81: END

```

DISKI

```

11:      SUBROUTINE DISKI
12:      C
13:      C   M C A M SUBSYSTEM - SEGMENT INPUT TASK
14:      C
15:      C   P.A. MARSHALL
16:      C   FEBRUARY 1976
17:      C -----
18:      C
19:      C *** SINGLE TASK VERSION - THIS IS NOT A TASK.
20:      C
21:      C --- THIS IS THE INPUT TASK FOR DISK FILE SEGMENTS. BUFFERS ARE
22:      C   CHAINED VIA POINTERS IN THE BUFFER CONTROL TABLES.
23:      C -----
24:      C
25:      C   INCLUDE 'MCAMSD'
26:      C**** PARAMETERS - DISK ORGANIZATION
27:      C   PARAMETER NWT = 3072           ;NO. OF WORDS PER TRACK
28:      C   PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
29:      C   PARAMETER NWS = 256          ;NO. OF WORDS PER SECTOR
30:      C****
31:      C   INCLUDE 'MCAMSP'
32:      C**** PARAMETERS - CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
33:      C   PARAMETER SEGMENT = 2         ;SEGMENT BUFFER SIZE IN SECTORS
34:      C   PARAMETER BUFCBL = 5          ;BUFFER CONTROL BLOCK LENGTH
35:      C   PARAMETER MAXCHN = 32         ;MAXIMUM MULTIPLEX FACTOR
36:      C   PARAMETER NBUFS = 2           ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
37:      C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
38:      C   PARAMETER BUFRP = NBUFS*(NWS*SEGMENT+MAXCHN-1)
39:      C   PARAMETER BUFCBP = NBUFS*BUFCBL
40:      C****
41:      C   INCLUDE 'MCAMSB'
42:      C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
43:      C   PARAMETER BUFAD = 0           ;BUFFER ADDRESS (INDEX)
44:      C   PARAMETER BUFLN = 1           ;BUFFER LENGTH
45:      C   PARAMETER BUFLAG = 2          ;BUFFER STATUS FLAG
46:      C   PARAMETER SECNO = 3           ;RELATIVE SECTOR NO.
47:      C   PARAMETER BUFPNT = 4          ;POINTER TO NEXT BUFFER
48:      C****
49:      C   IMPLICIT INTEGER (A-Z)
50:      C   COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
51:      C   COMMON /BUFOA/ BUFP, BUFCB(BUFCBP), BUFB(BUFRP)
52:      C
53:      C
54:      C --- I/O IS INITIATED HERE
55:      C**** 9 WAIT EVNDA
56:      C   9 CONTINUE
57:      C   CBUF = BUFP
58:      C
59:      C --- I/O CONTINUES HERE
60:      C**** CHANGE 'RETURN' TO 'GO TO 9'
61:      C   10 IF (BUFCB(CBUF+BUFLAG) .NE. 0) RETURN
62:      C
63:      C --- CHECK FOR EOF; ELSE DO INPUT
64:      C   IERR = 9
65:      C   IF (BUFCB(CBUF+SECNO) .LE. LSEC)
66:      C     * CALL RD9LK(IFNO, BUFCB(CBUF+SECNO)+OFFSET,
67:      C     * BUFB(BUFCB(CBUF+BUFAD)), NSEC, IERR)
68:      C
69:      C --- LOG STATUS

```

```
60: C*** TASK CALL 'XMT'
61: BUFCB(CBUF+BUFLAG) = IERR
62: C
63: C --- CHAIN TO NEXT BUFFER; SEE IF READY
64: CBUF = BUFCB(CBUF+BUFPNT)
65: GO TO 10
66: END
:
```

DEMUX

```

11      SUBROUTINE OEMUX(MUX, NCHAN, NSCAN, ISPAN, KEY, K, AREA)
12      C
13      C   DIGITAL DATA DEMULTIPLEXING
14      C
15      C   P.A. MARSHALL
16      C   NOVEMBER 1975
17      C-----
18      C
19      C   --- THIS ROUTINE DEMULTIPLEXES DIGITAL CHANNELS FROM THE INPUT "MUX"
20      C   ARRAY TO AN OUTPUT "AREA". INPUT DATA IS A FULLY MULTIPLEXED
21      C   ARRAY OF INTEGER VALUES, CONSISTING OF "NSCAN" SETS OF "NCHAN"
22      C   CHANNELS EACH. CHANNELS ARE NUMBERED FROM ONE TO "NCHAN"
23      C   CONTIGUOUSLY WITHIN EACH SET (SCAN).
24      C
25      C   A SUBSET OF CHANNELS WILL BE DEMULTIPLEXED INTO THE OUTPUT AREA
26      C   ACCORDING TO AN ARRAY ("KEY") OF CHANNEL SELECT ORDINALS. FOR
27      C   EACH CHANNEL NAMED (BY CHANNEL NO.) IN THE KEY ARRAY, "NSCAN"
28      C   POINTS FOR THAT CHANNEL WILL BE DEMULTIPLEXED AND TRANSFERRED
29      C   SEQUENTIALLY INTO THE OUTPUT AREA. THE MULTIPLEXED CHANNELS MAY
30      C   BE SELECTED IN ANY ORDER FOR OUTPUT BY THIS PROCESS AND INDEED,
31      C   ANY CHANNEL MAY BE SELECTED AND TRANSFERRED TO THE OUTPUT AREA
32      C   MORE THAN ONCE.
33      C
34      C   CHANNELS MAY BE SEPARATED FROM EACH OTHER IN THE OUTPUT AREA
35      C   BY SOME FIXED AMOUNT SPECIFIED BY 'ISPAN'. THIS QUANTITY
36      C   SPECIFIES THE SEPARATION IN WORDS BETWEEN THE LAST DATA POINT
37      C   OF ONE CHANNEL AND THE FIRST DATA POINT OF THE SUCCEEDING
38      C   CHANNEL IN THE OUTPUT AREA.
39      C
40      C   UPON RETURN FROM THIS ROUTINE, IF THERE WERE "K" CHANNELS
41      C   SELECTED FOR OUTPUT BY THE KEY ARRAY THEN THE OUTPUT AREA WILL
42      C   CONTAIN "K" SETS OF "NSCAN" SAMPLES EACH, SEPARATED BY AN
43      C   AMOUNT 'ISPAN', WHEREIN EACH SET LIE CONTIGUOUSLY ALL THE DATA
44      C   SAMPLES FOR A PARTICULAR CHANNEL. THE ORDERING OF THE SETS
45      C   WITHIN THE OUTPUT AREA WILL BE THE SAME AS THE ORDERING OF THE
46      C   CHANNELS WITHIN THE KEY ARRAY.
47      C
48      C   USE
49      C
50      C   CALL OEMUX(MUX, NCHAN, NSCAN, ISPAN, KEY, K, AREA)
51      C
52      C   WHERE
53      C
54      C   MUX      - IS THE INPUT ARRAY OF MULTIPLEXED CHANNELS
55      C
56      C   NCHAN    - IS THE NO. OF MULTIPLEXED CHANNELS CONTAINED IN "MUX"
57      C
58      C   NSCAN    - IS THE NO. OF DATA POINTS PER CHANNEL IN "MUX"
59      C
60      C   ISPAN    - IS THE SEPARATION (IN WORDS) BETWEEN CHANNELS IN THE
61      C               OUTPUT AREA. (SET TO ZERO FOR NO SEPARATION.)
62      C
63      C   KEY      - IS THE ARRAY OF CHANNEL SELECT ORDINALS
64      C
65      C   K         - IS THE NO. OF CHANNELS SELECTED (DIMENSION OF "KEY")
66      C
67      C   AREA     - (INTEGER) IS THE OUTPUT ARRAY FOR DEMULTIPLEXED DATA
68      C-----

```



```

      .TITL  DEMUX
;
;  DIGITAL DATA DEMULTIPLEXING, IN CORE, ASSEMBLER ALGORITHM
;
;  P.A. MARSHALL
;  SEPTEMBER 1976
;-----
;
;  --- THIS ROUTINE IS AN EFFICIENT ASSEMBLER CODED ALGORITHM
;  OF THE FORTRAN ROUTINE OF THE SAME NAME. THIS VERSION IS
;  3.5 TIMES MORE CPU EFFICIENT THEN THE FORTRAN 5 COMPILED
;  VERSION.
;
;  USE
;
;  CALL DEMUX(MUX, NCHAN, NSCAN, ISPAN, KEY, K, AREA)
;
;  --- FOR A FULL DESCRIPTION OF THE USE OF THIS ROUTINE SEE THE
;  DOCUMENTATION INCLUDED IN THE FORTRAN LISTING.
;-----

      .EXTU

      ENTRY DEMUX

      MUX      =ARG0
      NCHAN    =ARG1
      NSCAN    =ARG2
      ISPAN    =ARG3
      KEY      =ARG4
      K        =ARG5
      AREA     =ARG6

;  REGISTER USAGE:
;
;      AC0 - TEMPORARY
;      AC1 - "NCHAN"
;      AC2 - "L"
;      AC3 - "ICHAN"
;
;  THESE REGISTER VALUES ARE MAINTAINED WITHIN THE INNER LOOP
;  (JLOOP).  IN THE OUTER LOOP (ILOOP) AC3 IS USED TO HOLD THE
;  FRAME POINTER FOR ARGUMENT ADDRESSING.

      SAVE     0
      LDA      0,MUX,3
      SBI      1,0      ; ADDRESS-1
      STA      0,MUX     ; STORE INTO SECOND WORD OF INSTRUCTION
      LDA      0,KEY,3
      LDA      1,@K,3
      ADD      1,0      ; MAKE ADDRESS (KEY+K)
      STA      0,KEY     ; STORE INTO TWO-WORD INSTRUCTION
      NEG      1,1      ; -K
      STA      1,II      ; "ILOOP" INDEX
      LDA      2,AREA,3
      SBI      1,2      ; ADDRESS-1
      LDA      0,@ISPAN,3
      SUB      0,2      ; INITIAL INDEX, L = -ISPAN
      LDA      1,@NCHAN,3

ILOOP:  LDA      3,.FP    ; RESTORE FRAME POINTER

```

```

        LDA      0,@ISPAN,3
        ADD      0,2      ; L = L+ISPAN
        LDA      0,@NSCAN,3
        LDA      3,II     ; GET "KEY" INDEX
.KEY    =.+1
        ELDA     3,0,3    ; ICHAN = KEY(I)
        STA      0,J      ; "JLOOP" INDEX

JLOOP:  .MUX      =.+1
        ELDA     0,0,3    ; MUX(ICCHAN)
        INC      2,2      ; L = L+1
        STA      0,0,2    ; AREA(L) =
        ADD      1,3      ; ICHAN = ICHAN+NCHAN
        DSZ      J
        JMP      JLOOP    ; END OF JLOOP

        ISZ      II
        JMP      ILOOP    ; END OF ILOOP

        RTN

II:     .BLK      1      ; DEFINE LOOP INDICES LOCALLY
J:      .BLK      1

        .END

```

```

      .TITL   FBLM

;  FORTRAN 5 - BLM INSTRUCTION INTERFACE (BLOCK MOVE)
;
;  P.A. MARSHALL
;  SEPTEMBER 1976
;-----
;
;  --- MOVE A BLOCK OF WORDS FROM ONE STORAGE LOCATION TO ANOTHER
;
;  --- USE:
;
;      CALL FBLM(TO, FROM, COUNT)
;
;  WHERE (ALL PARAMETERS ARE TYPE "INTEGER")
;
;      TO      - IS THE LOCATION TO WHICH DATA WILL BE MOVED
;      FROM    - IS THE LOCATION FROM WHICH DATA WILL BE MOVED
;      COUNT   - IS THE NUMBER OF WORDS TO MOVE
;-----

```

```

      .ENT    FBLM

      TO      =-3      ; ARG 1
      FROM    =-4      ; ARG 2
      COUNT   =-5      ; ARG 3

FBLM:  .NREL
      SAVE    0
      LDA     1,COUNT,3    . ; SETUP FBLM REGISTERS
      LDA     2,FROM,3
      LDA     3,TO,3
      BLM                     ; DO THE BLOCK MOVE
      RTN
      .END

```

APPENDIX P: LISTINGS OF TWELVE-PASS ROUTINES

ECLIPSE FORTRAN 3, VERSION 3.10 -- FRIDAY, NOVEMBER 10, 1978 11:30:20 AM

DEPRC.FR

```

1:      SUBROUTINE XDEPRC(IC,JC)
2:      OVERLAY ODEPRC
3:      COMMON /TVTUNE/ITV,ITX
4:      COMMON /BPARM/TZARR,TWIND,THR,TNN,AZ,ARNG,AL,NT,DTG,SG
5:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
6:      INCLUDE 'CSBLAM'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
9:      INTEGER A
10:     INCLUDE 'C&GPARM'
11:     C MODIFIED 1/12/78 FOR 12-PASSES
12:     PARAMETER LSZ1=17
13:     PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
14:     COMMON /KDATE/KYR,KHO,KDA,KHR,KNN,RSEC,DUR,EVENT,PASS
15:     *,UNQYR,UNQID,TCORR,FCA,LCA
16:     COMMON /JDATE/INR,INN,TSEC
17:     COMMON /XXX/XX(3),LIST(LSZ)
18:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
19:     *,(LIST(13),TINCOR)
20:     COMMON /VEE/VSAVE(4,MAXPASS)
21:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
22:     C
23:     COMMON /EPARM/HTEMP,ITEMP(32),DTZARR,OTWIND
24:     DIMENSION MNN(80),SQ(40)
25:     DATA IQ/63/
26:     DATA IY/1HY/
27:     DATA IX/1HX/
28:     DATA IBL/1H /
29:     INTEGER THR,TNN
30:     C* DEFINE THE WINDOW
31:     1 CONTINUE
32:     IF(JC.EQ.19)GO TO 13
33:     IF(JC.EQ.20)GO TO 14
34:     C* DEFINE TIME WINDOW
35:     13 WRITE(ITV,210)
36:     210 FORMAT(19H DEFINE TIME WINDOW)
37:     WRITE(ITV,211)IQ
38:     211 FORMAT(23H WINDOW WIDTH (SECONDS),R1)
39:     C*
40:     C* READ TWIND
41:     C* READ TIME WINDOW LENGTH
42:     READ(ITX,234)MNN(1)
43:     234 FORMAT(S80)
44:     CALL INFREE(MNN,RQ,80)
45:     TWIND=RQ
46:     C*
47:     C*
48:     C* READ HRMIN SEC
49:     WRITE(ITV,222)KHR,KNN,RSEC
50:     222 FORMAT(16H DATA STARTS AT ,I2,1H,,I2,2X,F7.2)
51:     WRITE(ITV,204)IQ
52:     204 FORMAT(23H HR MIN SEC (FREE-FORM),R1,' (CR) GETS DEFAULT')
53:     WRITE(ITV,223)
54:     223 FORMAT(" TYPE 'X R' FOR AN R SEC. OFFSET OF ORIGIN")
55:     READ(ITX,234)MNN(1)
56:     CALL INFREE(MNN,RQ,80)

```

```

57:         IF(MNH(1).EQ.1BL)GOTO 55
58:         IF(MNH(1).EQ.1X)GOTO 55
59:         THR=SQ(1)
60:         TMN=SQ(2)
61:         TSEC=SQ(3)
62:         GOTO 56
63: C       COME HERE FOR CR OR X MN RESPONSE
64: C
65: 55      THR=KMR
66:         TMN=KMN
67:         TSEC=RSEC
68:         IF(MNH(1).EQ.1X)TSEC=RSEC+SQ(1)
69: C
70: 56      CONTINUE
71: C COMPUTE TIME WINDOW RELATIVE TO DATA
72: C TTT IS REQUESTED TIME IN SECONDS
73: C RRR IS START OF DATA WINDOW IN SECONDS
74:         TTT=THR+3600+TMN*60+TSEC
75:         RRR=KMR+3600+KMN*60+RSEC
76:         TZARR=TTT-RRR
77:         IF(TZARR.LT.0.)WRITE(ITV,220)
78:         IF(TZARR.GT.DUR)WRITE(ITV,220)
79:         IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE '/JDATE/:'
80:         *,' IHR=',IHR,' IMN=',IMN,' TSEC=',TSEC
81:         IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE ' TZARR=',
82:         *TZARR,' TSEC=',TSEC,' RSEC=',RSEC,' TTT=',TTT,
83:         *' RRR=',RRR
84: 220     FORMAT(36H REQUESTED TIME OUT OF RANGE OF DATA)
85: 203     FORMAT(32H STARTING TIME WINDOW (HRMN,SEC),2X,2F4.0,1X,F7.2)
86: C*
87:         WRITE(ITV,203)THR,TMN,TSEC
88:         WRITE(ITV,221)TZARR
89: 221     FORMAT(1H ,F7.2,15H SEC. INTO DATA)
90:         IHR=THR
91:         IMN=TMN
92:         OTZARR=TZARR
93:         RETURN
94: 14      WRITE(ITV,205)IQ
95: 205     FORMAT(10H 4014 TERM,R1,6HY OR N)
96:         READ(ITX,701)IA
97: 701     FORMAT(A1)
98:         CALL TERM(1,1024)
99:         ITERM=1
100:        IF(IA.EQ.IY)CALL TERM(3,4096)
101:        IF(IA.EQ.IY)ITERM=4
102:        LWIDTH=1000
103:        LHY=600
104:        JX=10
105:        JY=10
106:        WRITE(ITV,206)IQ
107: 206     FORMAT(12H RESET FRAME,R1)
108:        READ(ITX,701)IA
109:        IF(IA.NE.IY)GO TO 12
110:        WRITE(ITV,207)
111: 207     FORMAT(33H INPUT WIDTH, HEIGHTN-(FREE FORM))
112:        READ(ITX,234)NMN(1)
113:        CALL INFREE(NMN,SQ,80)
114:        LWIDTH=SQ(1)
115:        LHY=SQ(2)
116:        WRITE(ITV,208)

```

```

117: 209  FORMAT(39H INPUT SCREEN ORIGIN-IX,IY--(FREE FORM))
118:      READ(ITX,234)NNN(1)
119:      CALL INFREE(NNN,SQ,80)
120:      JX=SQ(1)
121:      JY=SQ(2)
122: 12    CONTINUE
123:      CALL SWND(JX,LWIDTH,JY,LHY)
124:      ISCAL=1024*ITERM
125:      CALL CHRISZ(JCHAR)
126:  C*
127:      RETURN
128:      END

```

EVPRC.FR

```

1:      SUBROUTINE XEVPRC(IC,JC)
2:      OVERLAY OEVPRC
3:      INCLUDE 'CSGPARM'
4:      C MODIFIED 1/12/78 FOR 12-PASSES
5:      PARAMETER LSZ1=17
6:      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
7:      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
8:      *,UNQYR,UNQID,TCORR,FCA,LCA
9:      COMMON /JDATE/IHR,IMN,TSEC
10:     COMMON /XXX/XX(3),LIST(LSZ)
11:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
12:     *,(LIST(13),TIMCOR)
13:     COMMON /VEE/VSAVE(4,MAXPASS)
14:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
15:     C
16:     INCLUDE 'CSBLAN'
17:     PARAMETER MAXX=1000
18:     COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
19:     INTEGER A
20:     DIMENSION MSG(7)
21:     DIMENSION NNN(41),SQ(10)
22:     DATA MSG/80,65,83,83,32,32,48/
23:     COMMON /ZER/IZER
24:     INCLUDE 'CSAPARM'
25:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
26:     INCLUDE 'CSBPARM'
27:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
28:     C
29:     COMMON /CPARM/NSTA,ISTA(2,32)
30:     COMMON /EPARM/NTMP,ITEMP(32),OTZARR,OTWIND
31:     COMMON /SCAL/IX,LX,IY,LY,ITEM,ISCAL,ICAR
32:     COMMON /TVTUNE/ITV,ITX
33:     COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
34:     COMMON /MISC/ZX,ZY,CX,CY
35:     COMMON /DECI/IDF,DECMAX,DLIM
36:     INTEGER CX,CY,ZX,ZY
37:     C EVENT REQUEST PROCESSOR
38:     IF(IC.EQ.6)GO TO 52
39:     20 WRITE(ITV,302)
40:     302 FORMAT('**ISDS** = SELECT EVENT NUMBER')
41:     READ(ITX,234)NNN(1)
42:     234 FORMAT(I80)
43:     CALL I\FREE(NNN,SQ,80)
44:     EVENT=SQ(1)
45:     CALL RQEV(TLIST,EVENT)
46:     IF(LIST(1).NE.0)GOTO 50
47:     C OUTPUT ERROR MESSAGE AND RETURN
48:     WRITE(ITV,303)
49:     303 FORMAT(18H EVENT NOT ON DISK)
50:     RETURN
51:     50 CONTINUE
52:     C LOAD BPARM AND KDATE VARIABLES#
53:     KYR=LIST(2)
54:     KMO=LIST(3)
55:     KDA=LIST(4)
56:     KHR=LIST(5)
57:     KMN=LIST(6)
58:     RSEC=SEC
59:     DUR=DURAT

```

```

60:      INTEGER MUX(1), AREA(1), KEY(K)
61:      C
62:      L = -ISPAN
63:      C
64:      DO 10 I=1,K
65:      L = L+ISPAN
66:      ICHAN = KEY(I)
67:      C
68:      DO 10 J=1,NSCAN
69:      L = L+1
70:      AREA(L) = MUX(ICHAN)
71:      10 ICHAN = ICHAN+NCHAN
72:      C
73:      RETURN
74:      END

```



```

601      DTG=DELT
611      TCORR=TIMCOR
621      UNQYR=LIST(15)
631      UNQID=LIST(16)
641      C PRINT OUT UNIQUE ID NUMBER
651      WRITE(ITV,306)UNQYR,UNQID
661      306  FORMAT(' UNIQUE ID:',I3,'-',I5)
671      WRITE(ITV,307)
681      307  :FORMAT(' OK?(Y OR N)')
691      CALL TINPUT(IA)
701      IF(IA.NE.89)GOTO 20
711      C REZERO TZARR (POINT TO START OF DATA)
721      TZARR=0.
731      C TYPE 'SEC,DURAT,DELT=',SEC,DURAT,DELT
741      NPASS=LIST(LSZ1)
751      C LOAD DT ARRAY
761      DO 40 J=1,32
771      40   DT(J)=DTG
781      C
791      C EQUATE JDATE WITH KDATE FOR INITIAL PLOTS
801      IHR=KHR
811      IMN=KMN
821      TSEC=RSEC
831      C
841      C IGAP IS SIZE OF GAP(IN TEKPOINTS) BETWEEN VIEWPORTS
851      C MAXVP IS PARAMETER=MAXIMUM NUMBER OF VIEWPORTS ON
861      C MVP IS MIN OF MAXVP AND NUMBER OF PASSES
871      C
881      MVP=MIN0(MAXVP,NPASS)
891      C SCREEN AT ANY GIVEN TIME
901      IGAP=160/MVP
911      KX1=30
921      C COMPUTE WIDTH OF VIEWPORTS
931      LXX=990/MVP-IGAP
941      JY=10
951      LHY=600
961      CALL ERASE
971      C
981      DO 51 N=1,NPASS
991      NCYCLE=MOD(N,MVP)
1001      IF(NCYCLE.EQ.0)NCYCLE=MVP
1011      C COMPUTE X=ORIGIN OF NTH VIEWPORT
1021      JX=KX1+(NCYCLE-1)*(IGAP+LXX)
1031      CALL SWND(JX,LXX,JY,LHY)
1041      C SAVE SCREEN CO-ORDINATES OF CURRENT VIEWPORT
1051      VSAVE(1,N)=JX
1061      VSAVE(2,N)=LXX
1071      VSAVE(3,N)=JY
1081      VSAVE(4,N)=LHY
1091      C PLOT THIS PASS
1101      C SETUP I/O TO PLOT ALL DATA
1111      MUX=LIST(N+LSZ1)
1121      IF(MUX.EQ.0)GOTO 51
1131      NCHAN=MUX
1141      NREQ=MUX
1151      C EQUATE PREVIOUS TRACE STATE TO CURRENT STATE
1161      NTEMP=NREQ
1171      DO 30 J=1,NREQ
1181      ITEMP(J)=J
1191      30   INEQ(J)=J
1201      C
1211      C SET WINDOW TO DURATION TIME OF DATA
1221      TWIND=DUR

```

```

123:      CALL DECIM
124:      C
125:      CALL RQPAZ(ISTA,FCA,LCA,EVENT,N)
126:      C
127:      CALL NOSHF
128:      C SPEED BY-PASS
129:      C JC=3 MEANS EV (NOT EVPL)
130:      IF(JC.NE.3)GOTO 51
131:      C
132:      CALL RSETD
133:      IF(IZER.EQ.1)CALL ZLFV
134:      CALL ARRAY
135:      C LABEL PLOT ,OPTION 1 MEANS PUT START AND STOP TIMES
136:      C IN ADDITION TO THE OTHER LABELS
137:      CALL LHSTA(1)
138:      C PUT PASS LABEL ON PLOT
139:      MXX=LXX/2+JX
140:      MYY=LHY+JY+40
141:      CALL MOVABS(MXX,MYX)
142:      MSG(7)=48+N
143:      MSG(6)=32
144:      IF(N.GT.9)MSG(6)=49
145:      IF(N.GT.9)MSG(7)=N-10+48
146:      CALL ANSTR(7,MSG)
147:      IF(NCYCLE.NE.MVP)GOTO 51
148:      C LABEL PLOT WITH EVENT NAME AND ID=NUMBER
149:      CALL ENAME(NNN,EVENT)
150:      MZZ=LHY+JY+60
151:      CALL MOVABS(358,MZZ)
152:      CALL ADJUST(40,NNN)
153:      ENCODE(NNN,100)UNGYR,UNQID
154:      100 FORMAT('UNIQUE ID:',I2,'-',I5)
155:      CALL MOVABS(800,MZZ)
156:      CALL ADJUST(18,NNN)
157:      C AFTER MVP WINDOWS ARE PLOTTED,PAUSE
158:      C GIVING OPERATOR CHANCE TO MAKE COPY.
159:      C TO CONTINUE,STRIKE ANY KEY,THEN SCREEN IS ERASED
160:      IF(N.EQ.NPASS)GOTO 51
161:      CALL TINPUT(IA)
162:      C IF YOU INPUT A PERIOD(.),PLOTING WILL BE TERMINATED
163:      IF(IA.EQ.46)JC=15
164:      CALL NEWPAG
165:      51 CONTINUE
166:      IF(JC.NE.3)GOTO 52
167:      C
168:      C PASS REQUEST PROCESSOR
169:      CALL HOME
170:      CALL ANMODE
171:      CALL TSEND
172:      52 CONTINUE
173:      WRITE(ITV,304)
174:      304 FORMAT(19H SELECT PASS NUMBER)
175:      CALL NEWLIN
176:      CALL SEELOC(CX,CY)
177:      READ(ITX,234)NNN(1)
178:      CALL INFREE(NNN,30,80)
179:      PASS=30(1)
180:      N=PASS
181:      CALL RQPAZ(ISTA,FCA,LCA,EVENT,PASS)
182:      C - - - DEFINE TRACES FOR THIS PASS - - -
183:      MUX=LIST(LSZ1+N)
184:      IF(MUX.EQ.0)GOTO 99
185:      NREQ=MUX

```

```

186:      NTEMP=MUX
187:      DO 1 J=1,NREQ
188:      ITEMP(J)=J
189:      1   IREQ(J)=J
190:      CALL NOSHF
191:      CALL RSETD
192:      C DO THINGS USUALLY DONE BY ARRAY
193:      AR=AZ+AL
194:      CALL VWINDO(TZARR,TWIND,AZ,AR)
195:      CALL SWND(VSAVE(1,N),VSAVE(2,N),VSAVE(3,N),VSAVE(4,N))
196:      CALL SWINDO(IX,LX,IY,LY)
197:      C - - - - -
198:      99  CONTINUE
199:      C RESET DECIMATION FACTOR
200:      IOF=1
201:      C NULL OUT DPARM BLOCK
202:      CALL DEFAULT
203:      RETURN
204:      END

```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, NOVEMBER 10, 1978 12:54:39 PM

FAZPU.FR

```

1:      SUBROUTINE XFAZPU(IS)
2:      OVERLAY OFAZPU
3:      COMMON /TVTUNE/ITV,ITX
4:      INCLUDE 'C&PICK'
5:      INTEGER PRNK,SRNK,RNK,SGRADE
6:      COMMON /DPARM/ PTIME(32),ANP(32),STIME(32),FTIME(32),
7:      *CROSS(32),SMP(32),FNP(32),PCROSS(32),
8:      *IFNOT(32),IGRADE(32),SGRADE(32),
9:      *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
10:     *PRES(32),SRES(32),EPD(32),CLK(32)
11:     COMMON /CPARM/MSTA,ISTA(2,32)
12:     INCLUDE 'C&GPARM'
13:     C MODIFIED 1/12/78 FOR 12-PASSES
14:     PARAMETER LSZ1=17
15:     PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
16:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMM,RSEC,DUR,EVENT,PASS
17:     *UNQYR,UNQID,TCORR,FCA,LCA
18:     COMMON /JDATE/IHR,IMN,TSEC
19:     COMMON /XXX/XX(3),LIST(LSZ)
20:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
21:     *,(LIST(13),TINCOR)
22:     COMMON /VEE/VSAVE(4,MAXPASS)
23:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
24:     C
25:     INTEGER OLDEVT,PUFLG,PUBLNK,FZ,CC
26:     INTEGER CLIST(32),DFN(7)
27:     DIMENSION INAME(5)
28:     DATA FZ/2HFZ/,CC/2HCC/
29:     DATA IBLANK/2H /,IZZ/2HZZ/,IPRNK/2H P/,ISRNK/2H S/
30:     DATA IP/1HP/
31:     DATA DFN/2HDP,2H0:,5*2H /
32:     EQUIVALENCE (INAME(1),DFN(3))
33:     COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
34:     PARAMETER ZPICK=4
35:     C GET AT EYDIR RECORD P.R.S. 3/29/78 RECSIZ=628 FOR 12 PASSES
36:     INTEGER REC,PASSN,RECSIZ
37:     PARAMETER RECSIZ=628
38:     COMMON/EDIRC/NEVENT,PASSN(MAXPASS),REC(RECSIZ)
39:     C
40:     C THIS ROUTINE PUNCHES PHASE CARDS DESTINED FOR USGS EARTHQUAKE
41:     C HYPOCENTER ROUTINES
42:     C IF IS=2, PUNCH BLANK AND ZZ DELIMITER
43:     C AND RETURN
44:     C CHECK TO SEE IF SAME EVENT AS BEFORE
45:     GOTO(10,20),IS
46:     20 IF(PUFLG.EQ.0)RETURN
47:     WRITE(7,701)IBLANK
48:     C+ WRITE(7,701)IZZ
49:     701 FORMAT(A2,78X)
50:     CLOSE 7
51:     CLOSE 14
52:     RETURN
53:     10 CONTINUE
54:     C IF NOT NEW EVENT, JUST PUNCH LIST
55:     IF(OLDEVT.EQ.EVENT)GOTO 30
56:     C FIRST TIME THRU?

```

```

57:         IF(PUFLG.EQ.0)GOTO 31
58:     C   IF NOT, CLOSE OUT OLD FILES
59:         WRITE(7,701)IBLANK
60:     C*   WRITE(7,701)I2Z
61:         CLOSE 7
62:         CLOSE 14
63:     31   CONTINUE
64:     C   OPEN NEW FZ AND CC FILES
65:         CALL MAKNAM(FZ,UNQID,INAME)
66:         OPEN 7,DFN
67:         CALL MAKNAM(CC,UNQID,INAME)
68:         OPEN 14,DFN
69:     C   FIRST TIME THIS EVENT:WRITE EVENT INFO ON CC FILE
70:     C   AND UPDATE EPLOG FILE
71:         ISCAN=1
72:         WRITE(14,800)UNQID,UNQID,TCORR,ISCAN,EVENT,MAXPASS
73:     800   FORMAT(I2,I5,1X,F5.3,I5,2I2)
74:     C   UPDATE EPLOG FOR THIS EVENT
75:         OPEN 15,'EPLOG',LEN=512
76:         CALL UTEP(UNQID,ZPICK,0,IER)
77:         CLOSE 15
78:     30   CONTINUE
79:     C   PUNCH PHASE LIST
80:         KMT=0
81:         DO 1 J=1,32
82:     C   CHECK FOR NULL STATIONS
83:         IF(PTIME(J).LE.0..AND.STIME(J).LE.0..AND.
84:         *SAMP(J).LE.0..AND.AMP(J).LE.0.)GO TO 1
85:     C   CHECK FOR NULL LABELS
86:         IF(ISTA(1,J).EQ.IBLANK)GOTO 1
87:     C   OUTPUT PHASE CARD FOR THIS STATION
88:     C   THE FOLLOWING ALGORITHM CORRECTS FOR THE PROPER P(OR S) SECOND
89:     C   AND RIPPLES THE CORRECTION THRU THE CLOCK AND CALENDAR.
90:     C   IF ONLY S TIME IS PRESENT IT IS RIPPLED. IF BOTH P AND S
91:     C   ARE PRESENT THEN THE P TIME IS RIPPLED. THE S TIME MUST BE
92:     C   THE SAME MINUTE, HOUR, ETC. AS THE P TIME.
93:         PTA=PTIME(J)+RSEC
94:         ST=0.
95:         PT=0.
96:         IF(PTIME(J).EQ.0..AND.STIME(J).GT.0.)PTA=STIME(J)+RSEC
97:     C
98:         XT=AMOD(PTA,60)
99:         JT=(PTA-XT)/60.
100:         IF(PTIME(J).GT.0.)PT=PTIME(J)+RSEC-JT*60.
101:         IF(STIME(J).GT.0.)ST=STIME(J)+RSEC-JT*60.
102:         JMN=KMN+JT
103:         JHR=KHR
104:         JDA=KDA
105:         IF(JMN.GT.60)JHR=KHR+1
106:         IF(JMN.GT.60)JMN=KMN-60
107:         IF(JHR.GT.24)JDA=KDA+1
108:         IF(JHR.GT.24)JHR=KHR-24
109:         IAMP=AMP(J)
110:         ISAMP=SAMP(J)
111:     C   COMPUTE CODA FROM FTIME AND PTIME
112:         FMP(J)=FTIME(J)-PTIME(J)
113:         IF(FMP(J).LE.0.)FMP(J)=0.
114:     C   IF STIME=0 THEN SGRADE=4 AND SRMK=BLANK
115:     C   IF PTIME=0 THEN IGRADE=4 AND PRMK=BLANK
116:         IF(ST.EQ.0.)SGRADE(J)=4

```

```

117:         IF(ST.EQ.0.)SRNK(J)=IBLANK
118:         IF(PT.EQ.0.)IGRADE(J)=4
119:         IF(PT.EQ.0.)PRNK(J)=IBLANK
120:     C
121:     C
122: 71      WRITE(7,700)ISTA(1,J),ISTA(2,J),PRNK(J),IFHOT(J),IGRADE(J),
123:      *KYR,KNO,JDA,JNR,JMN,PT,IAMP,ST,SRNK(J),SGRADE(J),ISAMP,
124:      *RNK(1,J),RNK(2,J),CLK(J),RNK(3,J),FHP(J)
125: 700      FORMAT(2A2,A2,A1,I1,1X,5I2,F5.2,I5,2X,F5.2,A2,1X,I1,4X,I3,15X,A2,
126:      *A1,F5.2,A1,I4,5X)
127:         KNT=KNT+1
128:         CLIST(KNT)=J
129:     I
130:     C DETERMINE ABSOLUTE PASS FROM RELATIVE PASS
131:     C APASS IS ABSOLUTE PASS NUMBER(REL. TO ANALOG TAPE)
132:         APASS=PASSN(PASS)
133:         WRITE(14,801)PASS,APASS,FCA,LCA,KNT,(CLIST(J),J=1,KNT)
134: 801      FORMAT(36I2)
135:         PUFLG=1
136:         OLDEVT=EVENT
137:         RETURN
138:         END

```

GPICK.FR

```

11      SUBROUTINE XGPICK(ICC)
12      OVERLAY OGPICK
13      COMMON /TVTUNE/ITV,ITX
14      INCLUDE 'C$CPARM'
15      C MODIFIED 1/12/78 FOR 12-PASSES
16      PARAMETER LSZ1=17
17      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
18      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
19      *,UNQYR,UNQID,TCORR,FCA,LCA
20      COMMON /JDATE/IHR,IMN,TSEC
21      COMMON /XXX/XX(3),LIST(LSZ)
22      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
23      *,(LIST(13),TIMCOR)
24      COMMON /VEE/VSAVE(4,MAXPASS)
25      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
26
27      C
28      DIMENSION ICARD(41),LAB(3),NTT(3)
29      DATA IDOL/2H$/
30      DATA NTT/2HEV,2HEN,2HT /
31      COMMON /CPARM/HSTA,ISTA(2,32)
32      INCLUDE 'C$PICK'
33      INTEGER PRMK,SRMK,RMK,SGRADE
34      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
35      *CROSS(32),SMP(32),FMP(32),PCROSS(32),
36      *IFMOT(32),IGRADE(32),SGRADE(32),
37      *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
38      *PRES(32),SRES(32),EPD(32),CLK(32)
39      DIMENSION IREC(45),NMN(41)
40      DIMENSION NAM(2),IDONE(32),KRMK(2)
41
42      C ICC=1:READ AUTO PICKS
43      C ICC=2:READ PHASE LIST
44      C ICC=3:READ HYPOUT
45      C
46      WRITE(ITV,150)
47      150 FORMAT(' FILENAME?')
48      C READ '5' RATHER THAN ITX BECAUSE WISH INPUT
49      C FROM USER DURING PICK MODE
50      READ(5,441)NMN(1)
51      441 FORMAT(S17)
52      OPEN 15,NMN
53      GOTO(10,20,30),ICC
54      10 CONTINUE
55      3 READ(15,101,END=500)ICARD(1)
56      101 FORMAT(S80)
57      C LOOK FOR EVENT LABEL
58      DECODE(ICARD,102) (LAB(J),J=1,3)
59      102 FORMAT(3A2)
60      IF(LAB(1).EQ.NTT(1).AND.LAB(2).EQ.NTT(2).AND.LAB(3).EQ.
61      *NTT(3))GOTO 4
62      GOTO 3
63      4 DECODE(ICARD,103)JEVENT
64      103 FORMAT(6X,I2)
65      C - - - TEST FOR CORRECT EVENT NUMBER
66      IF (JEVENT.NE.LIST(1))GOTO 3
67      C
68      TYPE 'EVENT NO.',JEVENT

```

```

57: C ZERO DONE FLAGS
58: DO 5 J=1,32
59: IDONE(J)=0
60: C LOOP AND READ UNTIL END OF EVENT
61: 2 READ(15,100,END=500)NAM,TIME,IFMT,DRTM
62: 100 FORMAT(2A2,F7.2,A2,F7.2)
63: C IF NAME = "##", IT IS THE END OF THE EVENT
64: IF(NAM(1).EQ.IDOL)GOTO 500
65: DO 1 J=1,32
66: IF(ISTA(1,J).NE.NAM(1))GOTO 1
67: IF(ISTA(2,J).NE.NAM(2))GOTO 1
68: C TAKE FIRST OCCURENCE ONLY
69: IF(IDONE(J).EQ.1)GOTO 1
70: C LOAD UP DPARM TABLE
71: PTIME(J)=TIME
72: FMP(J)=DRTM
73: IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
74: IFMT(J)=IFMT
75: IDONE(J)=1
76: GOTO 2
77: 1 CONTINUE
78: GOTO 2
79: C COME HERE IF NO FILE
80: 500 CONTINUE
81: C RETURN
82: C COME HERE WHEN EOF
83: 502 CLOSE 15
84: RETURN
85: C
86: 20 CONTINUE
87: 22 READ(15,200,END=502)NAM,IPR,IFMT,IG,JYR,JMO,
88: *JDA,JHR,JHN,TIME,IPA,ST,ISR,ISG,ISA,KRMK,CC,JRMK,DRTM
89: 200 FORMAT(3A2,A1,I1,1X,5I2,F5.2,15,2X,F5.2,A2,1X,I1,4X,I3,
90: *15X,A2,A1,F5.2,A1,F4.0,5X)
91: GOTO 24
92: 30 CONTINUE
93: 31 READ(15,300,END=502)IREC
94: 300 FORMAT(45A2)
95: DECODE(IREC,301,ERR=502)NAM,IPR,IFMT,IG,JYR,JMO,
96: *JDA,JHR,JHN,TIME,PRZ,PVV,ST,ISR,ISG,SRZ,ISA,SWV,
97: *PD,SD,EPI,ST,ANGLE,CC,JRMK,DRTM,AZ,DHAG,AMAG,PIMP,SIMP
98: 301 FORMAT(3A2,A1,I1,1X,5I2,F5.2,F4.2,F3.2,F5.2,A2,1X,I1,
99: *F4.2,F3.0,F3.2,2F4.2,F4.1,F3.0,F5.2,A1,F4.0,
100: *F3.0,2F2.1,2F4.3)
101: 24 CONTINUE
102: DO 23 J=1,32
103: C COMMON SECTION FOR HYPOT AND FZLINK
104: IF(ISTA(1,J).NE.NAM(1))GOTO 23
105: IF(ISTA(2,J).NE.NAM(2))GOTO 23
106: PTIME(J)=TIME-RSEC
107: IF(PTIME(J).LT.0)PTIME(J)=PTIME(J)+60.
108: FMP(J)=DRTM
109: IFMT(J)=IFMT
110: STIME(J)=ST-RSEC
111: IF(STIME(J).LT.PTIME(J).AND.ST.NE.0.)
112: *STIME(J)=STIME(J)+60.
113: IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
114: CROSS(J)=0.
115: PCROSS(J)=0.
116: IGRADE(J)=IG

```



```

117:         SGRADE(J)=ISG
118:         SAMP(J)=ISA
119:         RHK(3,J)=JRNK
120:         SRNK(J)=ISR
121:         PRNK(J)=IPR
122: C IF NOT FZLINK,SKIP NEXT SECTION
123:         IF(ICC.NE.2)GOTO 25
124:         AMP(J)=IPA
125:         RHK(1,J)=KRNK(1)
126:         RHK(2,J)=KRNK(2)
127: C IF NOT HYPOUT,SKIP NEXT SECTION
128: 25         IF(ICC.NE.3)GOTO 26
129:         PRES(J)=PRZ
130:         SRES(J)=SRZ
131:         EPD(J)=EPIDST
132:         CLK(J)=CC
133:         GOTO 26
134: 23         CONTINUE
135: C COME HERE AFTER SUCCESSFUL MATCH
136: 26         CONTINUE
137:         GOTO(502,22,31),ICC
138: C
139:
140:         END

```

LBSTA.FR

```

11:      SUBROUTINE XLBSTA(KC)
12:      OVERLAY OLBSTA
13:      INCLUDE 'CSBLAN'
14:      PARAMETER MAXX=1000
15:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
16:      INTEGER A
17:      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
18:      COMMON /CPARM/NSTA,ISTA(2,32)
19:      COMMON /APARM/S(32),Z(32),OT(32),TZ(32),SHFT(32),
20:      *D(32)
21:      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
22:      INCLUDE 'CSGPARM'
23:      C MODIFIED 1/12/78 FOR 12-PASSES
24:      PARAMETER LSZ1=17
25:      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
26:      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
27:      *,UNQYR,UNQID,TCORR,FCA,LCA
28:      COMMON /JOATE/IHR,IMN,TSEC
29:      COMMON /XXX/XX(3),LIST(LSZ)
30:      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
31:      *,(LIST(13),TIMCOR)
32:      COMMON /VEE/VSAVE(4,MAXPASS)
33:      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
34:      C
35:      DIMENSION MTIME(9),NUM(2)
36:      DIMENSION MSG(7)
37:      DATA MSG/63,69,67,79,78,68,63/
38:      DO 1 K=1,NREQ
39:      JJ=IREQ(K)
40:      Y=D(K)
41:      X=TZARR
42:      C* POSITION BEAM TO APROPRIATE TRACE
43:      CALL MOVEA(X,Y)
44:      CALL MOVREL(10,5)
45:      C* WRITE STATION NAME
46:      I1=ISTA(1,JJ)
47:      I2=ISTA(2,JJ)
48:      CALL AOUTST(2,I1)
49:      CALL AOUTST(2,I2)
50:      C PUT OUT TRACE NUMBER BY TRACE START
51:      CALL MOVEA(X,Y)
52:      CALL MOVREL(-30,-4)
53:      ENCODE(NUM,102)JJ
54:      102 FORMAT(I2)
55:      CALL AOUTST(2,NUM)
56:      C
57:      C PUT OUT SCALE FACTOR
58:      IF(S(JJ).EQ.1)GOTO 1
59:      Y=D(K)
60:      X=TZARR+TWIND
61:      CALL MOVEA(X,Y)
62:      CALL MOVREL(-90,10)
63:      ENCODE(MTIME,101)S(JJ)
64:      101 FORMAT(F8.1,'X')
65:      CALL AOUTST(9,MTIME)
66:      1 CONTINUE
67:      C IF KC=2,SKIP TIME LABELS
68:      IF(KC.EQ.2)GOTO 2
69:      C

```

```

60:  C LABEL BOX WITH CURRENT TIME
61:  C
62:  C PUT WINDOW START TIME ABOVE UPPER RIGHT CORNER
63:  C
64:      KYY=KY+LY+20
65:      CALL MOVARS(KX,KYY)
66:      ENCODE(MTIME,100)IHR,IMN,TSEC
67:  100  FORMAT(2I2,2X,F6.2)
68:      CALL ADUTST(12,MTIME)
69:  C PUT WINDOW END TIME ABOVE UPPER LEFT CORNER
70:  C
71:      TLAST=TZARR+TWIND+RSEC
72:  C CONVERT TO PROPER TIME
73:      JJJ=TLAST/60
74:      TLAST=TLAST-JJJ*60
75:      LMN=IMN+JJJ
76:      KKK=LMN/60
77:      LMN=LMN-KKK*60
78:      LHR=IHR+KKK
79:  C
80:      KXX=KX+LX-90
81:      KYY=KY+LY+20
82:      CALL MOVARS(KXX,KYY)
83:      ENCODE(MTIME,100)LHR,LMN,TLAST
84:      CALL ADUTST(12,MTIME)
85:  C
86:  C LABEL TIME AXIS
87:  2    MXX=LX/2+KX
88:      MYY=KY+10
89:      CALL MOVARS(MXX,MYX)
90:      CALL ANSTR(7,MSG)
91:      RETURN
92:      END

```

ISDS7.FR

```

11 C      PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
12 C      --WRITTEN BY PETER R. STEVENSON (1976-78)--
13 C      --U.S.GEOLOGICAL SURVEY--
14 C      --OFFICE OF EARTHQUAKE STUDIES--
15 C      --MENLO PARK,CALIF.--
16 C      COMMON /VTUNE/ITV,ITX
17 C      COMMON /RPLAY/VFILE
18 C      INTEGER VFILE
19 C      COMMON/SFL/SFLAG
20 C      INTEGER SFLAG
21 C      COMMON /ZER/IZER
22 C      COMMON/ZZZ/IZIT
23 C      COMMON /DECI/IDF,DECMAX,DLIN
24 C      INCLUDE 'C$GPARM'
25 C      MODIFIED 1/12/78 FOR 12-PASSES
26 C      PARAMETER LSZ1=17
27 C      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
28 C      COMMON /KDATE/KYR,KNO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
29 C      *,UNQYR,UNQID,TCORR,FCA,LCA
30 C      COMMON /JDATE/IHR,INM,TSEC
31 C      COMMON /XXX/XX(3),LIST(LSZ)
32 C      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
33 C      *,(LIST(13),TIMCOR)
34 C      COMMON /VEE/VSAVE(4,MAXPASS)
35 C      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
36 C
37 C      COMMON /CPARM/NSTA,ISTA(2,32)
38 C      INCLUDE 'C$APARM'
39 C      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
40 C      INCLUDE 'C$BPARM'
41 C      COMMON /BPARM/TZARR,TWIND,THR,TNN,AZ,ARNG,AL,NT,DTG,SG
42 C
43 C      COMMON /SCAL/ISDUM(7)
44 C      INCLUDE 'C$BLAN'
45 C      PARAMETER MAXX=1000
46 C      COMMON A(MAXX),MAXPTS,MREQ,IREQ(32)
47 C      INTEGER A
48 C      INCLUDE 'C$PICK'
49 C      INTEGER PRNK,SRNK,RNK,SGRADE
50 C      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
51 C      *CROSS(32),SNP(32),FNP(32),PCROSS(32),
52 C      *IFNOT(32),IGRADE(32),SGRADE(32),
53 C      *SAMP(32),RMK(3,32),SRNK(32),PRNK(32),
54 C      *PRES(32),SRES(32),EPD(32),CLK(32)
55 C      COMMON /EPARM/NTMP,ITEMP(32),OTZARR,OTWIND
56 C      COMMON /SHCON/ISH
57 C      COMMON /IOPRM/IODUM(6)
58 C      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
59 C      INTEGER OLDEVT,PUFLG,PUBLNK
60 C      COMMON /MISC/MDUM(4)
61 C      IZER=0
62 C      IZIT=0
63 C      ITV=4
64 C      ITX=5
65 C      OPEN 4,'$TT01'
66 C      OPEN 5,'$TTI1'

```

```
57:      OPEN 3,'JTEK'  
58:      VFILE=-1  
59:      OPEN 9,'TEMP'  
60:      CALL OVOPN('ISDS.OL',IER)  
61:      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER  
62:      CALL INITT(0)  
63:      CALL SETBUF(3)  
64:      CALL CHRSLZ(4)  
65:      CALL TSEND  
66:      CALL HITEK(1,0)  
67:      CALL NCAN1  
68:      CALL DIRC1  
69:      CALL DEFAULT  
70:      CALL SPLOT  
71:      STOP  
72:      END
```

LPICK.FR

```

1:      SUBROUTINE XLPICK
2:      OVERLAY OLPICK
3:      CC SUBROUTINE TO DISPLAY PICKS
4:      C P.R.STEVENSON 8/21/78
5:      COMMON /TVTUNE/ITV,ITX
6:      INCLUDE 'C$BLAN'
7:      PARAMETER MAXX=1000
8:      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
9:      INTEGER A
10:     INCLUDE 'C$PICK'
11:     INTEGER PRNK,SRMK,RMK,SGRADE
12:     COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
13:     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
14:     *IFHOT(32),IGRADE(32),SGRADE(32),
15:     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
16:     *PRES(32),SRES(32),EPD(32),CLK(32)
17:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
18:     *D(32)
19:     COMMON /BPARM/TZARR,TWIND,THR,TMM,AZ,ARNG,AL,MT,DTG,
20:     *SG
21:     COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
22:     COMMON /MISC/ZX,ZY,CX,CY
23:     INTEGER ZX,ZY,CX,CY
24:     DIMENSION MSGP(40)
25:     DATA ISLNK/2H /
26:     TRNG=TZARR+TWIND
27:     COMMON /CPARM/NSTA,ISTA(2,32)
28:     INCLUDE 'C$GPARM'
29:     C MODIFIED 1/12/78 FOR 12-PASSES
30:     PARAMETER LSZ1=17
31:     PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
32:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMM,RSEC,DUR,EVENT,PASS
33:     *,UNQYR,UNQID,TCORR,FCA,LCA
34:     COMMON /JDATE/IHR,IMN,TSEC
35:     COMMON /XXX/XX(3),LIST(LSZ)
36:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
37:     *,(LIST(13),TINCOR)
38:     COMMON /VEE/VSAVE(4,MAXPASS)
39:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
40:     C
41:     2 CONTINUE
42:     CC
43:     3 CONTINUE
44:     REWIND 9
45:     C SAVE HEADER ON SCRATCH FILE (L.U. 89)
46:     WRITE(9,903)
47:     903 FORMAT(33HNU. STA P-TIME P-AMPL. S-TIME,3X,7HS-AMPL.,2X,
48:     *3HSMP,5X,3HFMP,2X,4HP-CR,2X,3HFH,4HPRMK,2X,4HSRMK,2X,1HR)
49:     CALL SETMRG(150,1023)
50:     DO 41 K=1,NREQ
51:     J=IREQ(K)
52:     CC DRAW MARK FOR P PICK
53:     IF(PTIME(J)+SHFT(J).LE.TZARR.OR.PTIME(J)+SHFT(J).GT.TRNG)GOTO 20
54:     CALL PHARK(PTIME(J)+SHFT(J),D(K),80)
55:     C
56:     C DRAW DOTTED LINE FOR P RESIDUAL

```

```

57:      PRZ=PRES(J)+PTIME(J)+SHFT(J)
58:      IF(PRZ.LE.TZARR.OR.PRZ.GT.TRNG.OR.PRES(J).EQ.0.)GOTO 20
59:      CALL PHARK(PRZ,D(K),-82)
60:      C
61:      20      CONTINUE
62:      CC
63:      CC      DRAW MARK FOR S PICK
64:      IF(STIME(J)+SHFT(J).LE.TZARR.OR.STIME(J)+SHFT(J).GT.TRNG)GOTO 30
65:      CALL PHARK(STIME(J)+SHFT(J),D(K),83)
66:      C
67:      C      DRAW DOTTED LINE FOR S RESIDUAL
68:      SRZ=SRES(J)+STIME(J)+SHFT(J)
69:      IF(SRZ.LE.TZARR.OR.SRZ.GT.TRNG.OR.SRES(J).EQ.0.)GOTO 30
70:      CALL PHARK(SRZ,D(K),-114)
71:      C
72:      30      CONTINUE
73:      PCROSS(J)=CROSS(J)-PTIME(J)
74:      SHP(J)=STIME(J)-PTIME(J)
75:      FMP(J)=FTIME(J)-PTIME(J)
76:      IF(PCROSS(J).LT.0.)PCROSS(J)=0.
77:      IF(SHP(J).LT.0..OR.STIME(J).LT.0.)SHP(J)=0.
78:      IF(FMP(J).LT.0.)FMP(J)=0.
79:      C      DRAW MARK FOR F PICK
80:      IF(FTIME(J)+SHFT(J).LE.TZARR.OR.FTIME(J)+SHFT(J).GT.TRNG)GOTO 10
81:      IF(FMP(J).GT.0.)CALL PHARK(FTIME(J)+SHFT(J),D(K),70)
82:      10      CONTINUE
83:      C      PRETTY UP NULL S-TIMES
84:      IF(STIME(J).LT.0..AND.PTIME(J).GE.0.)STIME(J)=0.
85:      CC
86:      CALL NEWLIN
87:      IXX=0
88:      IYY=750*ITERM
89:      CALL MOVABS(IXX,IYY)
90:      CALL ANMODE
91:      CALL TSEND
92:      PT=PTIME(J)+RSEC
93:      ST=STIME(J)+RSEC
94:      IF(PTIME(J).EQ.0.)PT=0.
95:      IF(STIME(J).EQ.0.)ST=0.
96:      IF(ISTA(1,J).EQ.0)ISTA(1,J)=IBLNK
97:      IF(ISTA(2,J).EQ.0)ISTA(2,J)=IBLNK
98:      IF(ST.EQ.0.)SGRADE(J)=4
99:      IF(ST.EQ.0.)SRMK(J)=IBLNK
100:      IF(PT.EQ.0.)IGRADE(J)=4
101:      IF(PT.EQ.0.)PRMK(J)=IBLNK
102:      C      SAVE THE JTH ENTRY ON SCRATCH FILE
103:      WRITE(9,900)J,ISTA(1,J),ISTA(2,J),PT,AMP(J),ST,SAMP(J),SMP(J),
104:      *FMP(J),PCROSS(J),IFHOT(J),PRMK(J),IGRADE(J),SRMK(J),SGRADE(J),RMK(3,J)
105:      900      FORMAT(13,1X,2A2,1X,F7.2,2X,F7.2,2X,F7.2,2X,3F7.2,2X,F3.2,2X,A2,
106:      *2X,A2,I1,3X,A2,I1,2X,A1)
107:      41      CONTINUE
108:      CC
109:      C      READ BACK INFO FROM SCRATCH FILE
110:      REWIND 9
111:      READ(9,905)MSGP
112:      905      FORMAT(40A2)
113:      C      OUTPUT HEADER TO TEK SCREEN
114:      CALL AOUTST(80,MSGP)
115:      DO 44 K=1,NREQ
116:      J=IREQ(K)

```

```

117:      READ(9,905)MSGP
118:  921  CONTINUE
119:  C  SKIP IF BOTH P AND S ARE ZERO
120:      IF(PTIME(J).LE.0..AND.STIME(J).LE.0.)GO TO 44
121:      CALL NEWLIN
122:  C  OUTPUT ENTRY TO TEK SCREEN
123:      CALL ADUTST(80,MSGP)
124:  44   CONTINUE
125:  CC
126:  C  SAVE BEAM POSITION
127:      CALL NEWLIN
128:      CALL SEELOC(ZX,ZY)
129:      CALL SETHRG(0,1023)
130:      RETURN
131:      END

```


MISDS7.FR

```

1:  C      PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
2:  C      --WRITTEN BY PETER R. STEVENSON (1976-78)--
3:  C      --U.S.GEOLOGICAL SURVEY--
4:  C      --OFFICE OF EARTHQUAKE STUDIES--
5:  C      --MENLO PARK,CALIF.--
6:      COMMON /TVTUNE/ITV,ITX
7:      COMMON /RPLAY/VFILE
8:      INTEGER VFILE
9:      COMMON/SFL/SFLAG
10:     INTEGER SFLAG
11:     COMMON /ZER/IZER
12:     COMMON/ZZZ/IZIT
13:     COMMON /DECI/IDF,DECMAX,DLIN
14:     INCLUDE 'C$GPARM'
15:  C      MODIFIED 1/12/78 FOR 12-PASSES
16:     PARAMETER LSZ1=17
17:     PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
18:     COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
19:     *,UNQYR,UNQID,TCORR,FCA,LCA
20:     COMMON /JDATE/INR,IMN,TSEC
21:     COMMON /XXX/XX(3),LIST(LSZ)
22:     EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
23:     *,(LIST(13),TIMCOR)
24:     COMMON /VEE/VSAVE(4,MAXPASS)
25:     INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
26:  C
27:     COMMON /CPARM/NSTA,ISTA(2,32)
28:     INCLUDE 'C$APARM'
29:     COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
30:     INCLUDE 'C$BPARM'
31:     COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,MT,DTG,SG
32:  C
33:     COMMON /SCAL/ISDUM(7)
34:     INCLUDE 'C$BLAN'
35:     PARAMETER MAXX=1000
36:     COMMON A(MAXX),MAXPTS,MREQ,IREQ(32)
37:     INTEGER A
38:     INCLUDE 'C$PICK'
39:     INTEGER PRNK,SRNK,RNK,SGRADE
40:     COMMON /DPARM/ PTIME(32),ANP(32),STIME(32),FTIME(32),
41:     *CROSS(32),SNP(32),FNP(32),PCROSS(32),
42:     *IFNOT(32),IGRADE(32),SGRADE(32),
43:     *SAMP(32),RMK(3,32),SRNK(32),PRNK(32),
44:     *PRES(32),SRES(32),EPD(32),CLK(32)
45:     COMMON /EPARM/NTENP,ITEMP(32),OTZARR,OTWIND
46:     COMMON /SHCOM/ISH
47:     COMMON /IOPRM/IODUM(6)
48:     COMMON /OLD/OLDEVT,PUFLG,PUBLNK,MFIL(20)
49:     INTEGER OLDEVT,PUFLG,PUBLNK
50:     COMMON /MISC/MDUM(4)
51:     IZER=0
52:     IZIT=0
53:     ITV=4
54:     ITX=5
55:     OPEN 4,'$TT01'
56:     OPEN 5,'$TTI1'

```

```
57:      OPEN 3,'JTEK'  
58:      VFILE=-1  
59:      OPEN 9,'TEMP'  
60:      CALL OVOPN("NISDS.OL",IER)  
61:      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER  
62:      CALL INITT(0)  
63:      CALL SETBUF(3)  
64:      CALL CHRSLZ(4)  
65:      CALL TSEND  
66:      CALL NITEK(1,0)  
67:      CALL NCAM1  
68:      CALL DIRC1  
69:      CALL DEFAULT  
70:      CALL SPLOT  
71:      STOP  
72:      END
```

SPLOT.FR

```

11      SUBROUTINE SPLOT
12      COMMON /TVTUNE/ITV,ITX
13      INCLUDE 'C$APARM'
14      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
15      INCLUDE 'C$BPARM'
16      COMMON /BPARM/TZARR,TWIND,THR,TMM,AZ,ARNG,AL,MT,DTG,SG
17      C
18      COMMON/SFL/SFLAG
19      INTEGER SFLAG
20      COMMON /ZER/IZER
21      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
22      COMMON /EPARM/HTEMP,ITEMP(32),OTZARR,OTWIND
23      COMMON /SHCOM/ISH
24      COMMON /IOPRM/MS,MSCAN,MREN,MUX,IFILE,IUNIT
25      INCLUDE 'C$GPARM'
26      C MODIFIED 1/12/78 FOR 12-PASSES
27      PARAMETER LSZ1=17
28      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
29      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
30      *,UNQYR,UNQID,TCORR,FCA,LCA
31      COMMON /JDATE/IMR,IMM,TSEC
32      COMMON /XXX/XX(3),LIST(LSZ)
33      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
34      *,(LIST(13),TIMCOR)
35      COMMON /VEE/VSAVE(4,MAXPASS)
36      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
37      C
38      INTEGER OLDEVT,PUFLG,PUBLNK,CHAIN
39      COMMON /MISC/ZX,ZY,CX,CY
40      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
41      INTEGER ZX,ZY,CX,CY
42      DIMENSION MSG1(2),MSG2(3),MSG4(4)
43      DIMENSION MMN(41),SQ(40)
44      DATA IQ/63/
45      DATA IY/1HY/
46      C DEFAULT SCALE UNITS OPTION IS 1
47      IPAUS=0
48      SFLAG=1
49      CHAIN=0
50      MAXCH=32
51      IFILE=1
52      MUX=10
53      PUBLNK=0
54      PUFLG=0
55      OLDEVT=0
56      EVENT=0
57      CX=0
58      CY=780*ITERM
59      ZX=0
60      ZY=750*ITERM
61      C GO IMMEDIATELY TO EVENT PROCESSOR
62      IC=14
63      GO TO 39
64      C
65      C COME HERE TO START ANOTHER COMMAND
66      20 CONTINUE

```

```

57:      CALL COMMAND(IC,JC)
58:      GO TO (1,2,3,4,5,44,7,8,9,10,31,32,33,39,35,4,20,90,20,20,98,20,
59:      *20,39,20,20,20,40,42,20,41,43,20,20,2,777),IC
60: 777   CONTINUE
61:      IF(IPAUS.EQ.1)GOTO 45
62:      GO TO 20
63:  C*
64:  C*   DEFINE THE WINDOW
65:  1     CONTINUE
66:      CALL DEPRC(IC,JC)
67:  C*
68:  C*
69:      GO TO 20
70:  2     CONTINUE
71:      CALL TRPRC(IC,JC)
72:  C*
73:      GO TO 20
74:  3     CONTINUE
75:      IF(JC.EQ.15)GO TO 36
76:      IF(JC.EQ.9.OR.JC.EQ.12)GO TO 10
77:  C*   PLOT PROCESSOR
78:  C     RESTORE STATE BEFORE PLOTTING
79:      CALL RSTAT
80:      CALL PLPRC(JC)
81:  900   CONTINUE
82:      CALL MOVABS(CX,CY)
83:      CALL ANMODE
84:      CALL TSEND
85:      GO TO 20
86:  4     CONTINUE
87:  C***SHIFT PROCESSOR
88:      CALL RSTAT
89:  C
90:  C     ALIGN P-PHASES
91:      ISH=80
92:  C     ALIGN S-PHASES
93:      IF(JC.EQ.26)ISH=93
94:      IF(IC.EQ.14)ISH=83
95:  C     MADATI PLOT
96:      IF(IC.EQ.16)ISH=87
97:  C
98:      CALL PSHIFT(ISH)
99:      GO TO 900
100:  5     CONTINUE
101:  C*   PICK PROCESSOR
102:      CALL PICK
103:      GO TO 900
104:  6     CONTINUE
105:  C*
106:      GO TO 20
107:  C*   RESET PLOT PARMS
108:  7     CONTINUE
109:  C     IF 'RESU', RESUME REMOTE FILE CONTROL
110:      IF(JC.EQ.34)GOTO 45
111:      IF(JC.EQ.27)GOTO 90
112:      CALL DEFAULT
113:      IZER=0
114:      GO TO 20
115:  C*   AMPLITUDE SCALING PROCESSOR
116:  8     CONTINUE

```

```

117:      CALL SCALER
118:      GO TO 20
119:      C
120:      C SELECT SCALING OPTION
121:      88 IF(JC.NE.8)GOTO 20
122:      WRITE(ITV,800)
123:      800 FORMAT(" SELECT INPUT SCALE OPTION",/,
124:      *"1=DIGITAL COUNTS",/,"2=VOLTS OUT",/,
125:      *"3=EQUIVALENT DEVELOCORDER MILLIMETERS")
126:      READ(ITX,234)NNH(1)
127:      234 FORMAT(S81)
128:      CALL INFREE(NNH,RQ,96)
129:      SFLAG=RQ
130:      WRITE(ITV,801)SFLAG
131:      801 FORMAT(" OPTION ",I1," SELECTED")
132:      GOTO 20
133:      C
134:      C+ CROSS ? AUTO-CORRELATION PROCESSOR
135:      9 CONTINUE
136:      C CALL SCRSS
137:      GO TO 20
138:      C+ PLOT CROSS & AUTO-CORRELOGRAMS
139:      10 CONTINUE
140:      C CALL PCRSS
141:      C GO TO 900
142:      35 CONTINUE
143:      C+ SPECTRA PROCESSOR
144:      C CALL SFAST
145:      GO TO 20
146:      36 CONTINUE
147:      C CALL PFAST
148:      C GO TO 900
149:      C PUNCH PHASE CARDS
150:      31 CONTINUE
151:      CALL FAZPU(1)
152:      GO TO 20
153:      33 CONTINUE
154:      C USER ROUTINE
155:      C ROUTINE SETS KC IN ORDER TO CONTROL ITS RETURN
156:      C CALL USER(JC,KC)
157:      C GO TO (20,900)KC
158:      C
159:      C MENU
160:      32 CONTINUE
161:      CALL MENU(1)
162:      GO TO 20
163:      40 CONTINUE
164:      CALL MENU(2)
165:      GOTO 20
166:      39 CONTINUE
167:      IZER=1
168:      GO TO 20
169:      C COME HERE FOR 'UH' COMMANDS
170:      42 IF(JC.EQ.24)IZER=0
171:      IF(JC.EQ.32)ITX=5
172:      IF(JC.EQ.32)CLOSE 13
173:      IF(JC.EQ.32)CHAIN=0
174:      GOTO 20
175:      C COME HERE FOR 'PA'
176:      44 IF(JC.NE.13)GOTO 39

```

```

177:         ITX=5
178:         WRITE(ITV,544)
179: 544      FORMAT('REMOTE PAUSE')
180:         IPAUS=1
181:         GOTO 20
182: C COME HERE TO RESUME REMOTE CONTROL
183: 45      ITX=13
184:         IPAUS=0
185:         GOTO 20
186: C HARDCOPY HERE
187: 43      IF(JC.EQ.33)GOTO 20
188:         CALL HBCOPY
189:         CALL TSEND
190: C 10 SEC DELAY FOR HARDCOPY UNIT
191:         CALL FDLY(100)
192:         GOTO 20
193: C COME HERE FOR COMMANDS FROM REMOTE FILE
194: 41      WRITE(ITV,440)
195: 440     FORMAT(' INPUT REMOTE FILE NAME(10 CHARS. MAX)')
196:         READ(ITX,441)MNM(1)
197: 441     FORMAT($10)
198:         IF(CHAIN.EQ.1)CLOSE 13
199: C MNM IS ARRAY CONTAINING ASCII NAME OF FILE
200:         OPEN 13,MNM
201:         CHAIN=1
202:         ITX=13
203:         GOTO 20
204: C
205: 39      CONTINUE
206: C EVENT/PASS PROCESSOR
207: C CHECK ZERO-LEVEL FLAG
208:         IF(IC.EQ.14)IZER=0
209:         IF(JC.EQ.24)IZER=1
210:         CALL EVPRC(IC,JC)
211:         GO TO 20
212: 90      CONTINUE
213: C OUTPUT ZZ CARD AT END OF PHASE LIST
214:         CALL FAZPU(2)
215:         WRITE(ITV,990)
216: 990     FORMAT(10H EXIT ISDS)
217:         RETURN
218:         END

```

MAKNAM.FR

```

1:      SUBROUTINE XMAKNAM(PREFIX,POSTFIX,NAME)
2:      OVERLAY OMAKNAM
3:      DIMENSION NAME(5),II(6)
4:      INTEGER PREFIX,POSTFIX
5:      C PREFIX CONTAINS A2 NAME PREFIX
6:      C POSTFIX IS AN INTEGER NUMBER TO BE APPENDED
7:      C IN ASCII FORM TO PREFIX
8:      C NAME IS THE RESULTS OF THE CONCATENATION
9:      N=POSTFIX
10:     NAME(1)=PREFIX
11:     DO 1 J=1,5
12:     NN=N/10
13:     II(J)=N-NN*10
14:     N=NN
15: 1    CONTINUE
16:     II(6)=-48
17:     DO 2 J=1,3
18:     JJ=J+1
19:     JL=5-2*(J-1)
20:     JR=JL-1
21:     IF(JR.EQ.0)JR=6
22:     NAME(JJ)=((II(JL)+60K)*256)+((II(JR)+60K)
23: 2    CONTINUE
24:     NAME(5)=0
25:     RETURN
26:     END

```

UTEP.FR

```

1:      SUBROUTINE XUTEP(IDNUM,NCP,NRDP,IERC)
2:      OVERLAY DUTEP
3:      DIMENSION IH(256),IB(256),LANG(4,64)
4:      C  MODIFIED 6-30-78 BY P.STEVENSON (L.U.24-L.U.15)
5:      :  EQUIVALENCE (IB(1),LANG(1,1))
6:      DATA NEVPB/64/,IDIDGE/3/,MAXER/4/
7:      DATA KOPY/7/,IDROP/8/,ING/9/
8:      DATA MINLCP/0/,MAXLCP/31/
9:      IERC=MAXER
10:     IF (NCP .LT. MINLCP .OR. NCP .GT. MAXLCP) RETURN
11:     IF (IDNUM .LT. 0) RETURN
12:     C  ERROR RETURN CODES
13:     C  4=> NEW PROCESS ID IS OUT OF BOUNDS
14:     C  3=> ID IS NEGATIVE
15:     C  2=> ID IS NOT IN LOG
16:     C  1=> TRIED TO READ A DISK RECORD OUTSIDE OF EPLOG
17:     C  0=> OK
18:     C
19:     IERC=IERC-1
20:     NREK=1
21:     READ(15,REC=NREK) IH
22:     IFURST=IH(1)
23:     ILAST=IH(2)
24:     IF (IDNUM .LT. IFURST .OR. IDNUM .GT. ILAST) RETURN
25:     IERC=IERC-1
26:     IOFF=IDNUM-IFURST
27:     NREK=(IOFF/NEVPB)+2
28:     IF (NREK .GT. IH(5)) RETURN
29:     IERC=IERC-1
30:     READ(15,REC=NREK) IB
31:     NK=NREK-2
32:     INDEKS=IOFF-(NK*NEVPB)+1
33:     J=INDEKS
34:     ILBACT=16
35:     IRBACT=16
36:     INACT=1
37:     ILBLCP=7
38:     IRBLCP=11
39:     ILBDIO=12
40:     IRBDIO=15
41:     IF (NCP .EQ. IDROP .OR. NCP .EQ. KOPY .OR. NCP .EQ. ING) INACT=0
42:     IF (NCP .EQ. IDIDGE) FLD(LANG(3,J),ILBDIO,IRBDIO)=NRDP
43:     FLD(LANG(3,J),ILBLCP,IRBLCP)=NCP
44:     FLD(LANG(3,J),ILBACT,IRBACT)=INACT
45:     WRITE(15,REC=NREK) IB
46:     IH(3)=IH(3)+INACT
47:     INACT=INACT-1
48:     IH(4)=IH(4)+INACT
49:     IH(225+NCP)=IH(225+NCP)+1
50:     NREK=1
51:     WRITE(15,REC=NREK) IH
52:     IERC=0
53:     RETURN
54:     END

```


DIRC1

```

1:      SUBROUTINE DIRC1
2:      C
3:      C INTERPROCESS COMMUNICATIONS DIRECTORY INITIALIZER
4:      C
5:      C P.A. MARSHALL
6:      C SEPTEMBER 1976
7:      C-----
8:      C
9:      C INCLUDE 'PSDIR'
10:     C*** EVENT DIRECTORY OFFSETS
11:     C MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
12:     C
13:     C *** NUMBER OF POSSIBLE PASSES
14:     C PARAMETER NPASS=12
15:     CC *** DIRECTORY RECORD SIZE
16:     CC ** RECSIZ=52+NPASS+47*NPASS
17:     C PARAMETER RECSIZ = 628
18:     C
19:     C *** RECORD 0:
20:     C
21:     C PARAMETER ENX = 1 ; NEXT EVENT (LAST+1)
22:     C PARAMETER DML = ENX+1 ; NO. OF MASTERLIST ENTRIES
23:     C PARAMETER DOF = DML+1 ; DATABASE FILE OFFSET
24:     C PARAMETER DSZ = DOF+1 ; FILE SIZE IN SECTORS
25:     C PARAMETER DNX = DSZ+1 ; NEXT BLOCK NO.
26:     C PARAMETER CTE = DNX+1 ; CURRENT EVENT SAVE AREA
27:     C PARAMETER MPL = CTE+19 ; MASTER POINTER LIST
28:     CC ** EPL=MPL+NPASS*32-1
29:     C PARAMETER EPL = MPL+383 ; END OF POINTER LIST
30:     C
31:     C *** RECORD N:
32:     C
33:     C PARAMETER ENO = 1 ; EVENT NO.
34:     C PARAMETER ENA = ENO+1 ; EVENT NAME
35:     C PARAMETER EIB = ENA+20 ; INITIAL BLOCK NO.
36:     C PARAMETER ENB = EIB+1 ; TOTAL NUMBER OF BLOCKS
37:     C PARAMETER DLT = ENB+1 ; DELTA-T (SECS)
38:     C PARAMETER DRA = DLT+2 ; DURATION (SECS)
39:     C PARAMETER EYR = DRA+2 ; YEAR
40:     C PARAMETER EMO = EYR+1 ; MONTH
41:     C PARAMETER EDA = EMO+1 ; DAY
42:     C PARAMETER EHR = EDA+1 ; HOUR
43:     C PARAMETER EMI = EHR+1 ; MINUTE
44:     C PARAMETER ESE = EMI+1 ; SECONDS
45:     C PARAMETER TCR = ESE+2 ; TIME CORRECTION (SECS)
46:     C PARAMETER UYR = ESE+4 ; UNIQUE ID:YEAR
47:     C PARAMETER UID = ESE+5 ; UNIQUE ID:SERIAL NUMBER
48:     C PARAMETER EP1 = ESE+21 ; EVENT, PASS 1
49:     C PARAMETER EP2 = EP1+1 ; EVENT, PASS 2
50:     C PARAMETER EP3 = EP2+1 ; EVENT, PASS 3
51:     C PARAMETER EP4 = EP3+1 ; EVENT, PASS 4
52:     C
53:     C PARAMETER ETM = EYR ; EVENT TIME
54:     C PARAMETER EPP = EP1-1 ; EVENT, PASS POINTER
55:     C
56:     C *** PASS N:
57:     C
58:     C PARAMETER ENP = 0 ; ASSOCIATED EVENT NO.
59:     C PARAMETER PNO = ENP+1 ; PASS NO.

```

```

60:      PARAMETER PIB = PNO+1          ; INITIAL BLOCK NO.
61:      PARAMETER PNB = PIB+1          ; NUMBER OF BLOCKS
62:      PARAMETER PFC = PNB+1          ; FCN (FIRST CHANNEL NO.)
63:      PARAMETER PLC = PFC+1          ; LCN (LAST CHANNEL NO.)
64:      PARAMETER PLS = PLC+1          ; POINTER LIST
65:      PARAMETER PSW = PLS+32         ; TRACK SELECT SWITCH
66:      PARAMETER PFS = PSW+1          ; FILTER SELECT SWITCH
67:      PARAMETER PAD = PFS+1          ; A/D ERROR STATUS
68:      PARAMETER HEX = PAD+1          ; HEX COUNTER FOR THIS PASS
69:      C****
70:      INCLUDE 'PSFFNDIR'
71:      C**** FORTRAN FILE NOS. FOR DIRECTORIES
72:      PARAMETER IED = 1              ; EVENT DIRECTORY
73:      PARAMETER IML = 2              ; MASTERLIST DIRECTORY (CARD IMAGES)
74:      C****
75:      IMPLICIT INTEGER (A-Z)
76:      C MODIFIED 8-5-77 P. STEVENSON
77:      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
78:      COMMON /EVPAZ/ EVENT, PASS, CHANNL
79:      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
80:      C
81:      C
82:      C --- OPEN DIRECTORY FILES
83:      OPEN IED,'DIRECTORY',LEN=2*RECSIZ
84:      OPEN IML,'MASTERLIST',LEN=80
85:      C
86:      C --- READ DIRECTORY RECORD ZERO
87:      READ(IED,REC=1) REC
88:      C
89:      C --- INITIALIZE DATABASE PARAMETERS
90:      NEVENT = REC(ENX)-1
91:      OFFSET = REC(DOF)
92:      C
93:      RETURN
94:      END

```

ENAME

```

1:      SUBROUTINE XENAME(EVTNAME,EVTNUM)
2:      OVERLAY OENAME
3:      C
4:      C      EDITED 10/3/77 BY P.R.STEVENSON TO SAVE STACK SPACE
5:      C
6:      C      *****
7:      C
8:      C      HERE IS A ROUTINE TO READ AN EVENT NAME FROM A FILE CALLED "EVDIR"
9:      C      WHICH RESIDES ON REMOVABLE DISK DPO. THIS IS NOT A STAND-ALONE
10:     C      ROUTINE. SUBROUTINE ENAME SHOULD BE USED IN CONJUNCTION WITH 2
11:     C      OTHER SUBROUTINES, ONE OF WHICH MUST BE CALLED PRIOR TO CALLING
12:     C      SUBROUTINE ENAME. SUBROUTINE ENAME READS RECORDS FROM
13:     C      THE EVDIR FILE, BUT ASSUMES THIS FILE HAS BEEN PREVIOUSLY OPENED
14:     C      WITH A CALL TO SUBROUTINE "DIRC1".
15:     C
16:     C      NOTE: IN ORDER TO USE SUBROUTINE DIRC1, THE USER MUST HAVE CRE-
17:     C      ATED A LINK TO FILE DPO:EVDIR USING THE LINK NAME
18:     C      "DIRECTORY".
19:     C
20:     C      FORMAT OF CALL:  CALL ENAME(NAME,K)
21:     C
22:     C      WHERE K IS (UPON CALL) THE INTEGER NUMBER OF THE EVENT FOR WHICH
23:     C      THE NAME (IN ASCII) IS REQUESTED,
24:     C
25:     C      AND NAME IS A 20 WORD INTEGER ARRAY WHICH WILL BE STUFFED WITH
26:     C      THE 40 CHARACTER EVENT NAME, CORRESPONDING TO THE EVENT NUMBER.
27:     C
28:     C      IF THE EVENT NUMBER PASSED IN ARGUMENT "K" IS LESS THAN 1 OR
29:     C      GREATER THAN THE NUMBER OF EVENTS CONTAINED WITHIN THE EVDIR
30:     C      FILE, SUBROUTINE ENAME RETURNS TO THE CALLING ROUTINE WITH
31:     C      ASTERISKS IN THE 20 WORD "NAME" ARRAY.
32:     C
33:     C      IN MOST INSTANCES THE USER WILL ALSO BE USING THE DATA FILE
34:     C      ASSOCIATED WITH A PARTICULAR EVENT. THIS REQUIRES A CALL
35:     C      TO SUBROUTINE "MCAM1", WHICH OPENS FILE "EV DAT" (ALSO ON DPO).
36:     C
37:     C      NOTE: IN NON-TASKING ENVIRONMENT, USER MUST LINK DPO:EV DAT
38:     C      USING LINK NAME "WHOLE1".
39:     C
40:     C      THE RECOMMENDED CALLING SEQUENCE IS:
41:     C
42:     C          CALL MCAM1
43:     C          CALL DIRC1
44:     C          .
45:     C          .
46:     C          .
47:     C          .
48:     C          .
49:     C          CALL ENAME(NAME,K)
50:     C          .
51:     C          .
52:     C
53:     C      NOTE: SUBROUTINE ENAME MAY BE CALLED SEVERAL TIMES DURING EXECU-
54:     C      TION OF A PROGRAM (OR ANOTHER SUBROUTINE). HOWEVER, SUB-
55:     C      ROUTINE MCAM1 AND DIRC1 SHOULD BE CALLED ONCE AND ONLY
56:     C      ONCE DURING THE EXECUTION OF A PROGRAM.
57:     C
58:     C      *****
59:     C

```

```

60: CC ** MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
61:   PARAMETER NPASS=12
62:   PARAMETER REKSIZE = 52+NPASS+47*NPASS
63:   PARAMETER IED = 1
64:   PARAMETER ENO = 1
65:   PARAMETER ENA = ENO+1
66:   IMPLICIT INTEGER (A-Z)
67:   DIMENSION EVTNAME(1)
68:   COMMON /IOBDA/ NCHAN,SEGSZ,REMDR,LTBUF,FIRST
69:   COMMON /EDIRC/NEVENT,PASSN(NPASS),REC(REKSIZE)
70:   C
71:   C --- INITIALIZE THE EVENT NAME ARRAY. PUT ASTERISKS IN ALL
72:   C   LOCATIONS.
73:   C
74:   DO 5 K=1,20
75:     S   EVTNAME(K) = 2H**
76:   C
77:   C --- FIND OUT HOW MANY EVENTS ARE ALREADY WITHIN THE DIRECTORY
78:   C
79:   MAXEVTS = NEVENT
80:   C
81:   C --- CHECK THE PASSED PARAMETER "EVTNUM" TO SEE IF IT IS WITHIN
82:   C   REASONABLE BOUNDS. IF NOT, RETURN.
83:   C
84:   IF (EVTNUM .LT. 1 .OR. EVTNUM .GT. MAXEVTS) GO TO 20
85:   C
86:   C
87:   C --- READ THE RECORD CORRESPONDING TO THE PARTICULAR EVENT NUMBER
88:   C
89:   READ(IED,REC=EVTNUM+1) REC
90:   C
91:   C
92:   C --- NOW FILL THE EVENT NAME ARRAY WITH THE 40 ASCII CHARACTERS
93:   C   STORED IN FILE "EVDIR".
94:   C
95:   DO 10 K=1,20
96:     10   EVTNAME(K) = REC( ENO + K )
97:   C
98:   20   RETURN
99:   C
100:  END

```

RGEVT

```

11:      SUBROUTINE XRGEVT(LIST, PARM1)
12:      OVERLAY ORGEVT
13:      C
14:      C  EVENT REQUEST PROCESSOR
15:      C
16:      C  P.A. MARSHALL
17:      C  SEPTEMBER 1976
18:      C-----
19:      C
101:     INCLUDE 'PSDIR'
111:     C*** EVENT DIRECTORY OFFSETS
121:     C  MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
131:     C
141:     C *** NUMBER OF POSSIBLE PASSES
151:     C  PARAMETER NPASS=12
161:     CC *** DIRECTORY RECORD SIZE
171:     CC ** RECSIZ=52+NPASS*47*NPASS
181:     C  PARAMETER RECSIZ = 628
191:     C
201:     C *** RECORD 0:
211:     C
221:     C  PARAMETER ENX = 1          ; NEXT EVENT (LAST+1)
231:     C  PARAMETER DML = ENX+1    ; NO. OF MASTERLIST ENTRIES
241:     C  PARAMETER DOF = DML+1    ; DATABASE FILE OFFSET
251:     C  PARAMETER OSZ = DOF+1    ; FILE SIZE IN SECTORS
261:     C  PARAMETER ONX = OSZ+1    ; NEXT BLOCK NO.
271:     C  PARAMETER CTE = ONX+1    ; CURRENT EVENT SAVE AREA
281:     C  PARAMETER MPL = CTE+19   ; MASTER POINTER LIST
291:     CC ** EPL=MPL+NPASS*32-1
301:     C  PARAMETER EPL = MPL+383 ; END OF POINTER LIST
311:     C
321:     C *** RECORD N:
331:     C
341:     C  PARAMETER ENO = 1          ; EVENT NO.
351:     C  PARAMETER ENA = ENO+1     ; EVENT NAME
361:     C  PARAMETER EIB = ENA+20   ; INITIAL BLOCK NO.
371:     C  PARAMETER ENB = EIB+1    ; TOTAL NUMBER OF BLOCKS
381:     C  PARAMETER DLT = ENB+1    ; DELTA-T (SECS)
391:     C  PARAMETER ORA = DLT+2    ; DURATION (SECS)
401:     C  PARAMETER EYR = ORA+2    ; YEAR
411:     C  PARAMETER EMO = EYR+1    ; MONTH
421:     C  PARAMETER EDA = EMO+1    ; DAY
431:     C  PARAMETER EHR = EDA+1    ; HOUR
441:     C  PARAMETER EMI = EHR+1    ; MINUTE
451:     C  PARAMETER ESE = EMI+1    ; SECONDS
461:     C  PARAMETER TCR = ESE+2    ; TIME CORRECTION (SECS)
471:     C  PARAMETER UYR = ESE+4    ; UNIQUE ID:YEAR
481:     C  PARAMETER UIO = ESE+5    ; UNIQUE ID:SERIAL NUMBER
491:     C  PARAMETER EP1 = ESE+21   ; EVENT, PASS 1
501:     C  PARAMETER EP2 = EP1+1    ; EVENT, PASS 2
511:     C  PARAMETER EP3 = EP2+1    ; EVENT, PASS 3
521:     C  PARAMETER EP4 = EP3+1    ; EVENT, PASS 4
531:     C
541:     C  PARAMETER ETM = EYR      ; EVENT TIME
551:     C  PARAMETER EPP = EP1-1    ; EVENT, PASS POINTER
561:     C
571:     C *** PASS N:
581:     C
591:     C  PARAMETER ENP = 0          ; ASSOCIATED EVENT NO.

```

```

60:      PARAMETER PNO = ENP+1          ; PASS NO.
61:      PARAMETER PIB = PNO+1          ; INITIAL BLOCK NO.
62:      PARAMETER PNB = PIB+1          ; NUMBER OF BLOCKS
63:      PARAMETER PFC = PNB+1          ; FCN (FIRST CHANNEL NO.)
64:      PARAMETER PLC = PFC+1          ; LCN (LAST CHANNEL NO.)
65:      PARAMETER PLS = PLC+1          ; POINTER LIST
66:      PARAMETER PSW = PLS+32         ; TRACK SELECT SWITCH
67:      PARAMETER PFS = PSW+1          ; FILTER SELECT SWITCH
68:      PARAMETER PAD = PFS+1          ; A/D ERROR STATUS
69:      PARAMETER HEX = PAD+1          ; HEX COUNTER FOR THIS PASS
70:      C****
71:      INCLUDE 'PSFFNDIR'
72:      C**** FORTRAN FILE NOS. FOR DIRECTORIES
73:      PARAMETER IED = 1                ; EVENT DIRECTORY
74:      PARAMETER IML = 2                ; MASTERLIST DIRECTORY (CARD IMAGES)
75:      C****
76:      C MODIFIED 9-24-77 P. STEVENSON
77:      C**** REQUEST LIST OFFSETS
78:      C
79:      PARAMETER LEN = 1                ; EVENT NO.
80:      PARAMETER LET = LEN+1            ; EVENT TIME
81:      PARAMETER LED = LET+7            ; DURATION
82:      PARAMETER LOT = LED+2            ; DELTA-T
83:      PARAMETER LTC = LOT+2            ; TIME CORRECTION
84:      PARAMETER LUY = LTC+2            ; UNIQUE ID:YEAR
85:      PARAMETER LUI = LUY+1            ; UNIQUE ID:SERIAL NO.
86:      PARAMETER LNP = LUI+1            ; NUMBER OF PASSES
87:      PARAMETER LM1 = LNP+1            ; MUX, PASS 1
88:      PARAMETER LMN = LM1+NPASS-1      ; MUX, PASS N
89:      C
90:      PARAMETER LMP = LM1-1            ; MUX FINDER
91:      PARAMETER LSZ = LMN-LEN+1        ; LIST SIZE
92:      C****
93:      IMPLICIT INTEGER (A-Z)
94:      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
95:      COMMON /EVPZ/ EVENT, PASS, CHANNL
96:      DIMENSION LIST(LSZ)
97:      C
98:      C --- CLEAR LIST
99:      DO 1 J=1,LSZ
100:      1 LIST(J) = 0
101:      C
102:      C --- SET EVENT, CHECK FOR VALIDITY
103:      EVENT = PARM1
104:      IF (EVENT.LE. 0) RETURN
105:      C
106:      C --- READ IN DIRECTORY RECORD, CHECK VALIDITY
107:      READ(IED,REC=EVENT+1,END=99) REC
108:      IF (REC(ENO).EQ. 0) RETURN
109:      C
110:      C --- STUFF PARAMETERS FROM DIRECTORY RECORD INTO REQUEST LIST
111:      LIST(LEN) = REC(ENO)
112:      LIST(LET) = REC(EYR)
113:      LIST(LET+1) = REC(EMO)
114:      LIST(LET+2) = REC(EDA)
115:      LIST(LET+3) = REC(EHR)
116:      LIST(LET+4) = REC(EMI)
117:      LIST(LET+5) = REC(ESE)
118:      LIST(LET+6) = REC(ESE+1)
119:      LIST(LED) = REC(DRA)
120:      LIST(LED+1) = REC(DRA+1)
121:      LIST(LOT) = REC(DLT)
122:      LIST(LOT+1) = REC(DLT+1)

```

```

123:      LIST(LTC) = REC(TCR)
124:      LIST(LTC+1) = REC(TCR+1)
125:      LIST(LUY) = REC(UYR)
126:      LIST(LUI) = REC(UID)
127:      C
128:      C --- FIND WHICH PASSES EXIST AND INTERPRET AS A PASS COUNT
129:      K = 0
130:      DO 2 J=1,NPASS
131:      C --- CLEAR PASS MEMORY
132:      PASSN(J) = 0
133:      C --- POINT TO PASS. SEE IF IT EXISTS
134:      PNT = REC(EPP+J)
135:      NCHAN = REC(PNT+PLC)-REC(PNT+PFC)+1
136:      IF (NCHAN .EQ. 1 .AND. REC(PNT+PFC) .EQ. 0) GO TO 2
137:      C
138:      C --- HAVE PASS. COUNT PASS, SAVE PASS NO., STUFF MUX NO.
139:      K = K+1
140:      PASSN(K) = J
141:      LIST(LMP+K) = NCHAN
142:      2 CONTINUE
143:      C
144:      C --- STUFF PASS COUNT
145:      LIST(LNP) = K
146:      99 RETURN
147:      END

```

RQPAZ

```

1:      SUBROUTINE XQPAZ(LIST,FCA,LCA,PARM1,PARM2)
2:      OVERLAY ORQPAZ
3:      C
4:      C EVENT/PASS REQUEST PROCESSOR
5:      C
6:      C P.A. MARSHALL
7:      C SEPTSMHR 1976
8:      C-----
9:      C
10:     INCLUDE 'PSDIR'
11:     C*** EVENT DIRECTORY OFFSETS
12:     C MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
13:     C
14:     C *** NUMBER OF POSSIBLE PASSES
15:     C PARAMETER NPASS=12
16:     CC *** DIRECTORY RECORD SIZE
17:     CC ** RECSIZ=52+NPASS+47*NPASS
18:     C PARAMETER RECSIZ = 628
19:     C
20:     C *** RECORD 0:
21:     C
22:     C PARAMETER ENX = 1 ; NEXT EVENT (LAST+1)
23:     C PARAMETER DML = ENX+1 ; NO. OF MASTERLIST ENTRIES
24:     C PARAMETER DOF = DML+1 ; DATABASE FILE OFFSET
25:     C PARAMETER DSZ = DOF+1 ; FILE SIZE IN SECTORS
26:     C PARAMETER DNX = DSZ+1 ; NEXT BLOCK NO.
27:     C PARAMETER CTE = DNX+1 ; CURRENT EVENT SAVE AREA
28:     C PARAMETER MPL = CTE+19 ; MASTER POINTER LIST
29:     CC ** EPL=MPL+NPASS*32-1
30:     C PARAMETER EPL = MPL+383 ; END OF POINTER LIST
31:     C
32:     C *** RECORD N:
33:     C
34:     C PARAMETER ENO = 1 ; EVENT NO.
35:     C PARAMETER ENA = ENO+1 ; EVENT NAME
36:     C PARAMETER EIB = ENA+20 ; INITIAL BLOCK NO.
37:     C PARAMETER ENB = EIB+1 ; TOTAL NUMBER OF BLOCKS
38:     C PARAMETER DLT = ENB+1 ; DELTA-T (SECS)
39:     C PARAMETER DRA = DLT+2 ; DURATION (SECS)
40:     C PARAMETER EYR = DRA+2 ; YEAR
41:     C PARAMETER EMO = EYR+1 ; MONTH
42:     C PARAMETER EDA = EMO+1 ; DAY
43:     C PARAMETER EHR = EDA+1 ; HOUR
44:     C PARAMETER EMI = EHR+1 ; MINUTE
45:     C PARAMETER ESE = EMI+1 ; SECONDS
46:     C PARAMETER TCR = ESE+2 ; TIME CORRECTION (SECS)
47:     C PARAMETER UYR = TCR+4 ; UNIQUE ID:YEAR
48:     C PARAMETER UID = ESE+5 ; UNIQUE ID:SERIAL NUMBER
49:     C PARAMETER EP1 = ESE+21 ; EVENT, PASS 1
50:     C PARAMETER EP2 = EP1+1 ; EVENT, PASS 2
51:     C PARAMETER EP3 = EP2+1 ; EVENT, PASS 3
52:     C PARAMETER EP4 = EP3+1 ; EVENT, PASS 4
53:     C
54:     C PARAMETER ETM = EYR ; EVENT TIME
55:     C PARAMETER EPP = EP1-1 ; EVENT, PASS POINTER
56:     C
57:     C *** PASS N:
58:     C
59:     C PARAMETER ENP = 0 ; ASSOCIATED EVENT NO.

```



```

60:      PARAMETER PNO = ENP+1      ; PASS NO.
61:      PARAMETER PIB = PNO+1      ; INITIAL BLOCK NO.
62:      PARAMETER PNB = PIB+1      ; NUMBER OF BLOCKS
63:      PARAMETER PFC = PNB+1      ; FCN (FIRST CHANNEL NO.)
64:      PARAMETER PLC = PFC+1      ; LCN (LAST CHANNEL NO.)
65:      PARAMETER PLS = PLC+1      ; POINTER LIST
66:      PARAMETER PSW = PLS+32     ; TRACK SELECT SWITCH
67:      PARAMETER PFS = PSW+1      ; FILTER SELECT SWITCH
68:      PARAMETER PAD = PFS+1      ; A/D ERROR STATUS
69:      PARAMETER HEX = PAD+1      ; HEX COUNTER FOR THIS PASS
70:      C****
71:      INCLUDE 'PSFFNDIR'
72:      C**** FORTRAN FILE NOS. FOR DIRECTORIES
73:      PARAMETER IED = 1          ; EVENT DIRECTORY
74:      PARAMETER IML = 2          ; MASTERLIST DIRECTORY (CARD IMAGES)
75:      C****
76:      C**** REQUEST LIST PARAMETERS
77:      PARAMETER NSTAR = 2        ; WIDTH OF LIST ENTRY (IN WORDS)
78:      PARAMETER LSZ = 32*NSTAR   ; LIST SIZE
79:      PARAMETER LST = 1          ; START OF LIST
80:      C****
81:      IMPLICIT INTEGER (A-Z)
82:      C MODIFIED 8-5-77 P. STEVENSON
83:      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
84:      COMMON /EVPZ/ EVENT, PASS, CHANNL
85:      COMMON /IOPOA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
86:      COMMON /IOBDA/ NCHAN, IDUM3(4)
87:      DIMENSION LIST(LSZ), STAREC(40)
88:      C
89:      C --- CLEAR LIST
90:      DO 1 J=1,LSZ
91:      1 LIST(J) = 0
92:      C
93:      C --- CHECK EVENT AND PASS FOR VALIDITY
94:      IF (PARM1 .NE. EVENT) RETURN
95:      PASS = PARM2
96:      IF (PASS .LT. 1 .OR. PASS .GT. NPASS) RETURN
97:      IF (PASSN(PASS) .EQ. 0) RETURN
98:      C --- POINT TO PASS
99:      PNT = REC(EPP+PASSN(PASS))
100:      C
101:      C --- SET "MCAM" PARAMETERS
102:      EPISEC = REC(PNT+PIB)-OFFSET
103:      LSEC = EPISEC+REC(PNT+PNB)-1
104:      FCN = REC(PNT+PFC)
105:      LCN = REC(PNT+PLC)
106:      NCHAN = LCN-FCN+1
107:      C
108:      C LOAD FCA AND LCA
109:      FCA=FCN
110:      LCA=LCN
111:      C
112:      C --- SET POINTER LIST, FETCH STATION LIST DATA
113:      M = PNT+PLS-1
114:      K = LST
115:      DO 2 J=FCN,LCN
116:      STAREC(1) = 0
117:      C --- GET STATION RECORD FROM MASTERLIST
118:      MP = REC(M+J)
119:      IF (MP .LE. 0) GO TO 21
120:      READ(IML,REC=MP) STAREC
121:      21 CONTINUE
122:      C *** MOD (4/22/77) BY P.R.S. ;IJK FORMERLY 2

```

```

123: C   IJK=1 FOR FORMAT 2, SET IJK=2 FOR FORMAT 1
124:     IJK=1
125: C --- OR SHOW AS 'IRIG' CHANNEL
126:     IF (MP.EQ. -1) CALL FBLM(STAREC(IJK), 'IRIG', 2)
127: C --- OR 'WWVB' CHANNEL
128:     IF (MP.EQ. -2) CALL FBLM(STAREC(IJK), 'WWVB', 2)
129: C --- OR CLEAR RECORD IF "UNDEFINED" CHANNEL
130:     IF (MP.EQ. 0) CALL FBLM(STAREC(2), STAREC(1), 39)
131: C   :
132: C --- MOVE DATA INTO REQUEST LIST
133:     CALL FBLM(LIST(K), STAREC(IJK), 2)
134:     2 K = K+NSTAR
135: C
136:     RETURN
137:     END

```