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 Geological 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. Listings

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 develocorder 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 evental 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 othe 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 occured, 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
Discrimminators: 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 x 32 + 1 x 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.

Link Name	Resolution File	Description
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 HYPO71 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.

File Name	Description
User defined Default is FAZLST	ASCII random file. Created by Save command. Contains station name, P and S times, and other phase list varibles. 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 inputing 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. It's 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 <u>and</u> 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 milimeters (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 milivolts 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 Tekronix 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.

KEY	DESCRIPTION
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 obtined by subtracting 5 from the entered value. Thus 5 corresponds to a S weight of 0.

E or I — Emergent of Impulsive P arrivale 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 form 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 equavelent 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 opeartor 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 $\overline{\text{p}}$ 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
```

PLCOM2

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

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
```

PART II: PROGRAMMER'S REFERENCE

A. 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 IIth trace.

TZ = - not used -.

S = scale factor for IIth trace.

D = base level of Ith trace from top of plot (absolute amplitude units).

A = Amplitude of IJth sample for ith trace (amplitude units from proper base level).

Z = Amplitude of zero offset for IIth trace (amplitude units from proper base level).

As an illustration of how the plot algroithm 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 use 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 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 vertical screen units.

Example 2

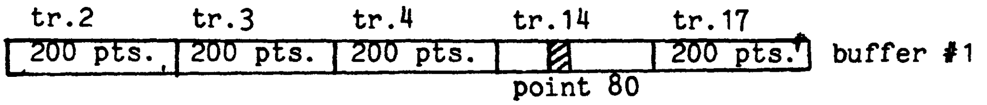
For this example let us choose the first peak of trace number 3 (relative trace 2):

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.

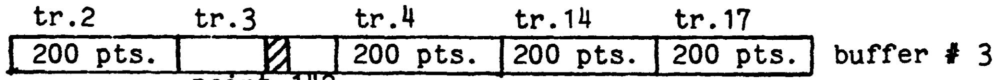
Buffer layouts for the two examples, showing locations of IJth point:

Equation 1:



point 80
within segment;
point 680
within buffer;
point 80 of trace.

Equation 2:



point 142
within segment;
point 342
within buffer;
point 542 of trace.

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 RQEVT).
    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 decimentation 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).
Erasê 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 date (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.

LBSCL 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 Elimates 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 volunarilly 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 RQEVT is called, and each time access to a new digitizing pass is requested, RQPAZ is called. A call to ENAME returnes 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.
RQEVT	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 digitial 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 timeing from single traces: U.S. Geological Survey Open-File Report.

Klein, Fred W., (1978). Hypocenter location program HYPOINVERSE: U.S. Geologcial Survey Open-File Report.

Lee, W. H. K., and Lahr, J. C. , (1972). HYPO71: 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

Common Label	Varible	Type	Description
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 ermergent '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)	Real Array	- not used -
	LIST(L), L = 1, LSZ	Integer Array	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	Integer	- not used -

	ISTA(I, J),
	I = 1, 2,
	J = 1, 32
		Integer array	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, inscreen coordinates.
	LY	Integer	height of plot window, in screen coordinates.

	ITERM	Integer	Terminal type 1 = HOLD, style (1024 x 780) 4 = 4014 style (4096 x 3120)
	ISCAL	Integer	Width of screen in screeen 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(/BPARM/).
	OTWIND	Real	Temporary storage for TWIND(/BPARM/).
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.

APPENDIX F

MENU	LPICK	LBSTA	LBSCL	ISDS7	INFREE	GPICK	FRAME	FAZPU	EVPRC	DEPRC	DEFAULT	DECIM	COMMAND	BOX	ARRAY	Subroutine / Include or Common
	x	x	x	x					x	x	x	x			x	C$BLAN(blank)
	x	x	x	x					x		x	x			x	C$APARM(APARM)
	x	x	x	x					x	x	x	x			x	C$BPARM(BPARM)
	x			x		x		x			x					C$PICK(DPARM)
													x			C$DICT
	x	x		x		x		x	x	x						C$GPARM*
x	x			x		x		x	x	x			x			TVTUNE
				x												RPLAY
			x	x												SFL
				x					x							ZER
				x							x					ZZZ
				x					x		x	x			x	DECI
	x			x		x		x	x							CPARM
	x	x	x	x			x		x	x	x					SCAL
				x					x	x						EPARM
				x												SHCOM
				x					x						x	IOPRM
				x				x								OLD
	x			x					x							MISC

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

APPENDIX F

(Con't)

		TRPRC	TPARS	ZLEV	SWND	SPLOT	SORT	SCALER	RSTAT	RSETD	PSHIFT	PMARK	PLPRC	PICK	NOSHF	Subroutine Include or Common
		x		x				x	x	x	x			x	x	C$BLAN(blank)
				x		x		x		x	x			x	x	C$APARM(APARM)
				x		x			x	x	x			x		C$BPARM(BPARM)
											x			x		C$PICK(DPARM)
																C$DICT
						x							x			C$GPARM*
		x				x		x			x			x		TVTUNE
																RPLAY
						x							x	x		SFL
						x					x		x			ZER
																ZZZ
													x			DECI
		x						x								CPARM
					x	x					x		x	x		SCAL
		x				x			x		x					EPARM
						x										SHCOM
		x				x					x					IOPRM
						x										OLD
						x							x	x		MISC

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

G: ISDS SUBROUTINE CALLS CROSS REFERENCE

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

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	SPLO7
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
RQEVT	-	EVPRC
RQPAZ	FBLM	EVPRC
ENAME	-	EVPRC, PLPRC
FBLM	-	RQPAZ, SEGMO
MCAME	-	ISDS7
FRAMD	MINO(R), SEGIN, SEGMO,	ARRAY, ZLEV
GETFM	DEMUX, MINO(R), SEGMO	ARRAY, ZLEV
SEGIN	DISKI	FRAMD, SEGMO
SEGMO	CHECK(R), FBLM, SEGIN	FRAMD, GETFM
DISKI	RDBLK(R)	SEGIN
DEMUX	-	GETFM

Code: (R) = Runtime routine.

I: PHASE DATA OUTPUT FORMAT

Col.	Format	Explanation
1-4	2A2	Station name. Must by 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 at '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 <u>if</u> colums 1-4 and 70-72 of the terminator cards ar 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. Usuaued except on output (not HYPO71 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 develocorder 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:

Columns	Format	Explanation
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):

Word	Data
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 + number of passes)
= 628 for 12-pass case.

The first record in the directory file (DPO:EVDIR) contians 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 continguous 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*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, RQEVT.

   b) ENAME is different only in the size of the EVDIR records, so parameter REKSIZ varys.

   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-Pase Routines

Figure 1

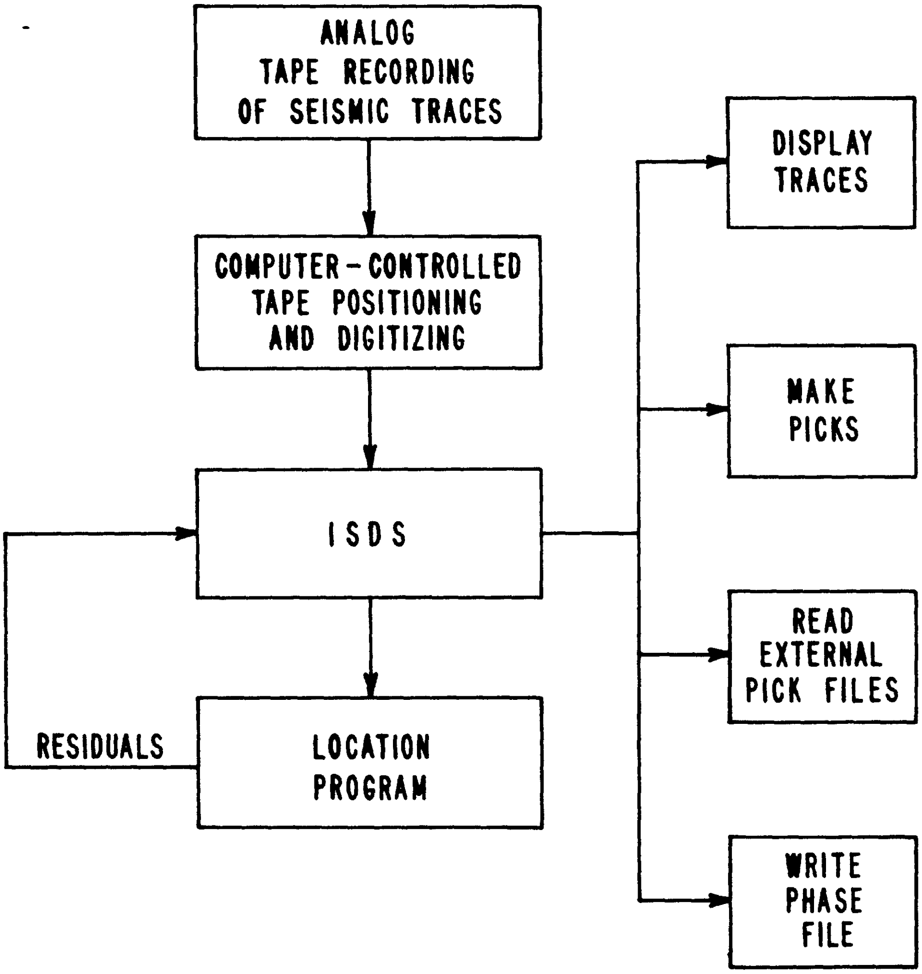

ANALOG TAPE RECORDING OF SEISMIC TRACES
COMPUTER-CONTROLLED TAPE POSITIONING AND DIGITIZING
ISDS
LOCATION PROGRAM
RESIDUALS
DISPLAY TRACES
MAKE PICKS
READ EXTERNAL PICK FILES
WRITE PHASE FILE

figure 2

SELECT PASS NUMBER

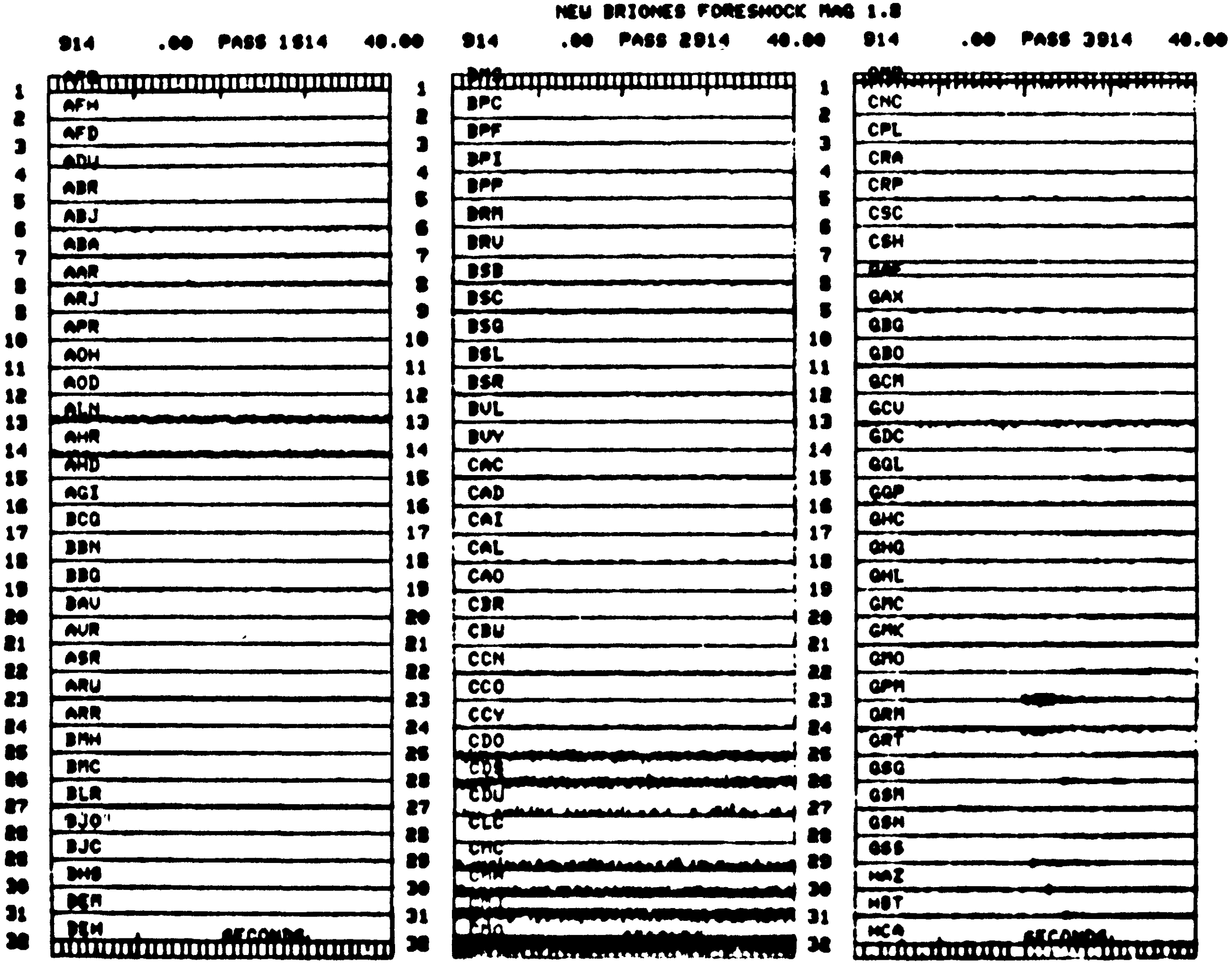


figure 3

COMMAND?

NEW BRIONES FORESHOCK MAG 1.8   EVENT 3 PASS 3

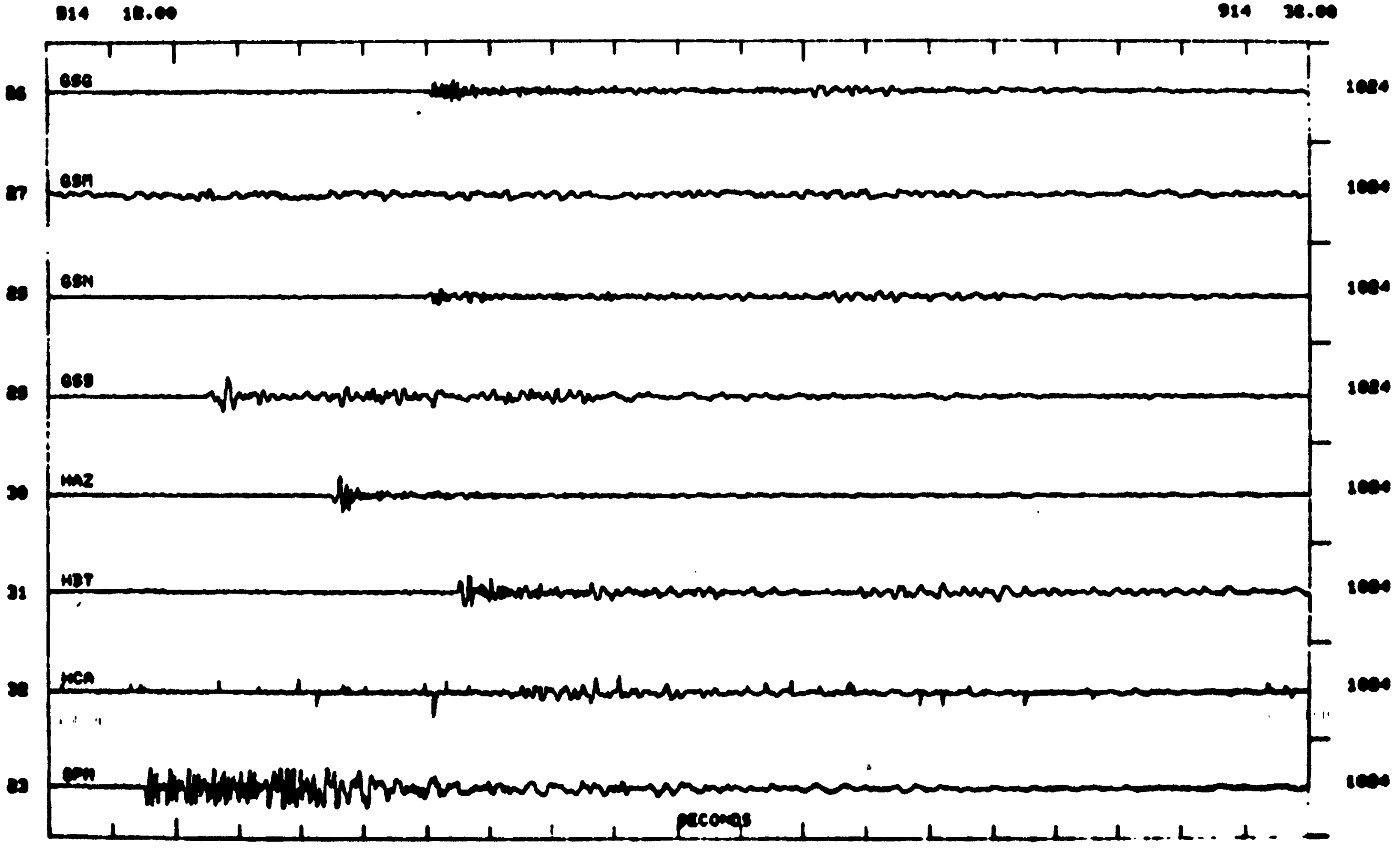


figure 4

COMMAND?

NU.	STA	P-TIME	P-AMPL.	S-TIME	S-AMPL.	SMP	FMP	P-CR	FM	PRNK	SRNK	R
26	QSG	23.96	.00	.00	.00	.00	.00	.00		P0	4	
28	QSN	23.96	.00	.00	.00	.00	.00	.00		P0	4	
29	QSS	20.40	.00	.00	.00	.00	.00	.00		P0	4	
30	HAZ	22.42	.00	.00	.00	.00	.00	.00		P0	4	
31	HDT	24.40	.00	.00	.00	.00	.00	.00		P0	4	
23	QPM	19.49	.00	.00	.00	.00	.00	.00		P0	4	

NEW BRIONES FORESHOCK MAG 1.8 EVENT 3 PASS 3

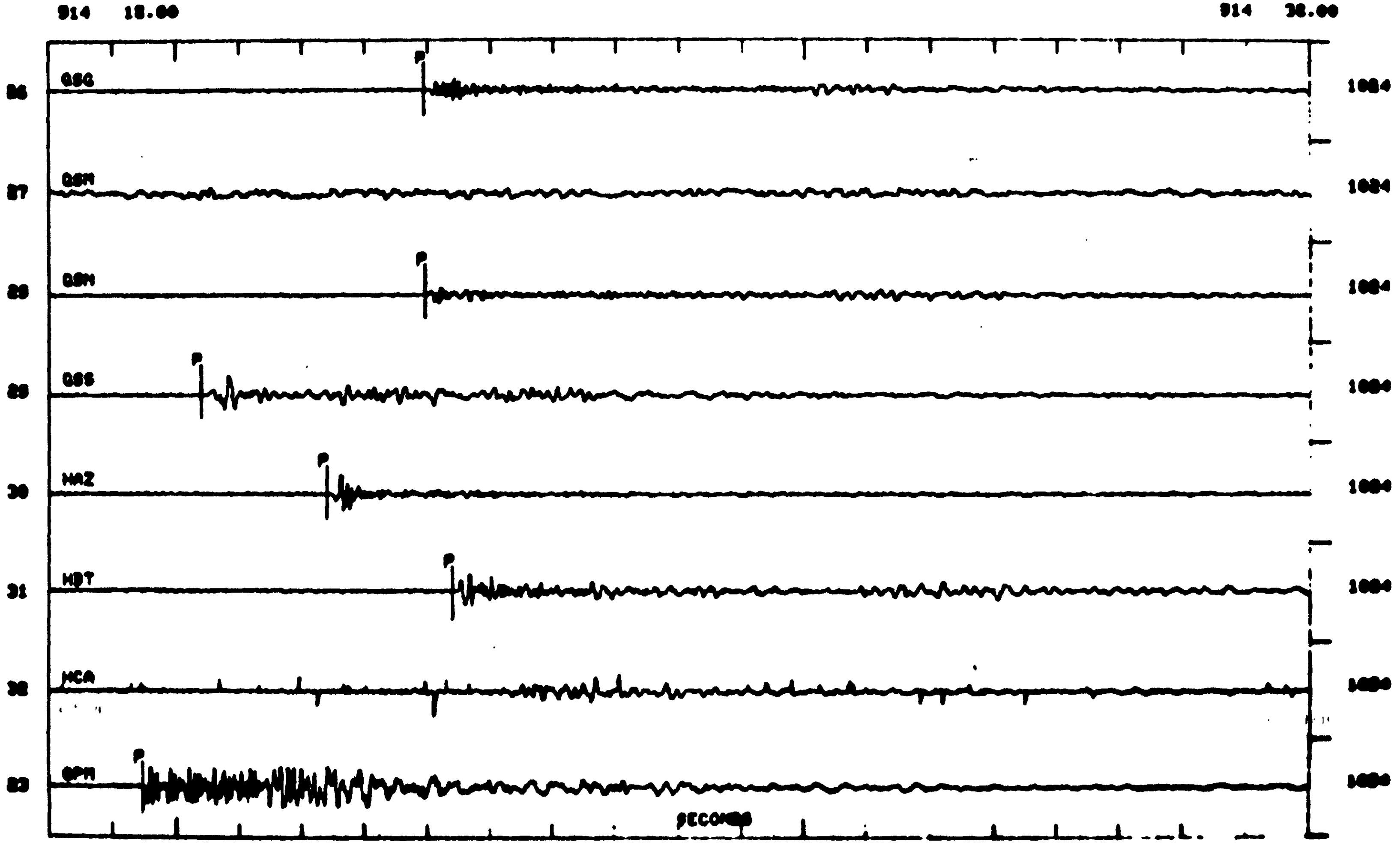


figure 5

COMMAND?
COMMAND?

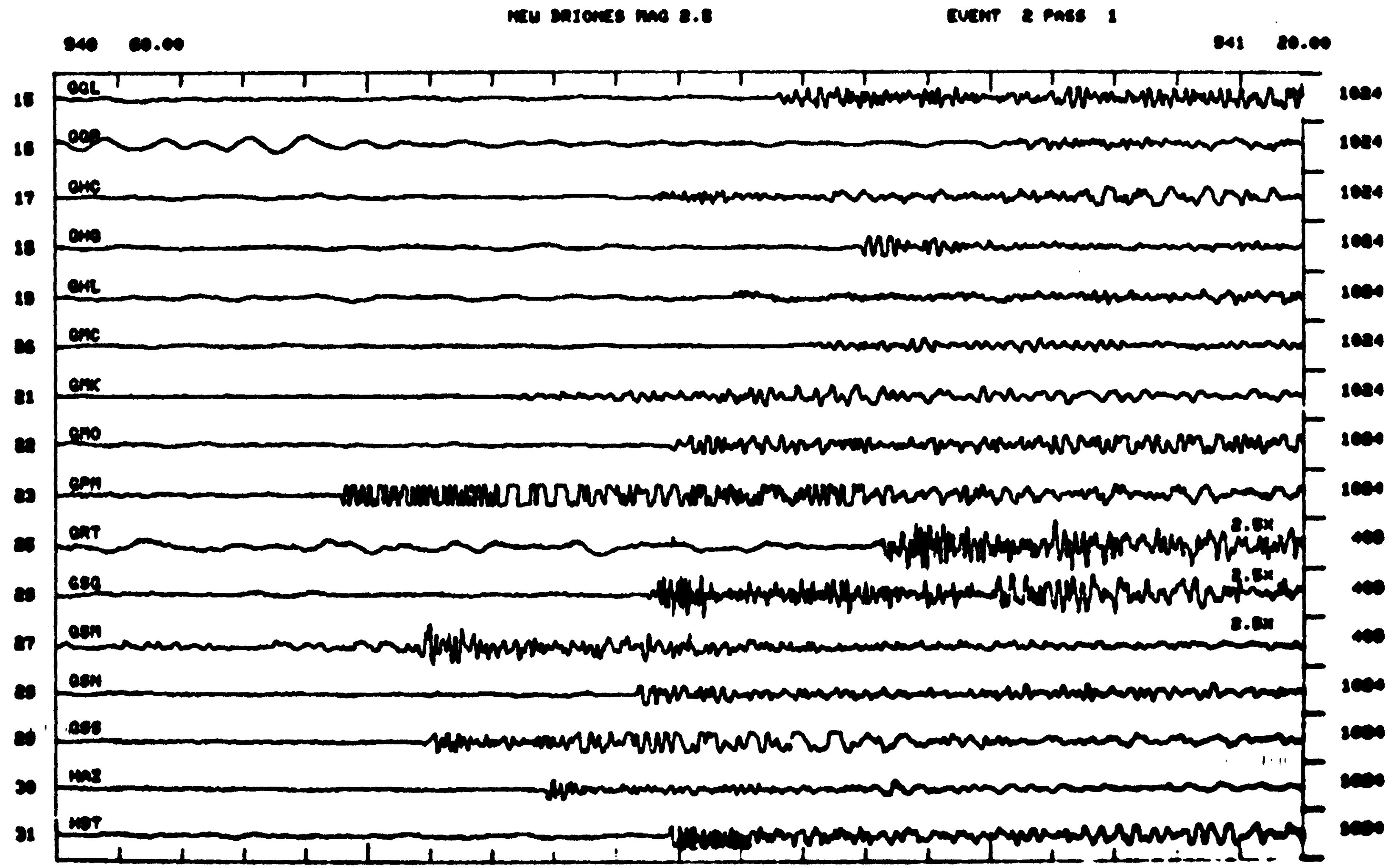


Figure 6

SELECT PASS NUMBER

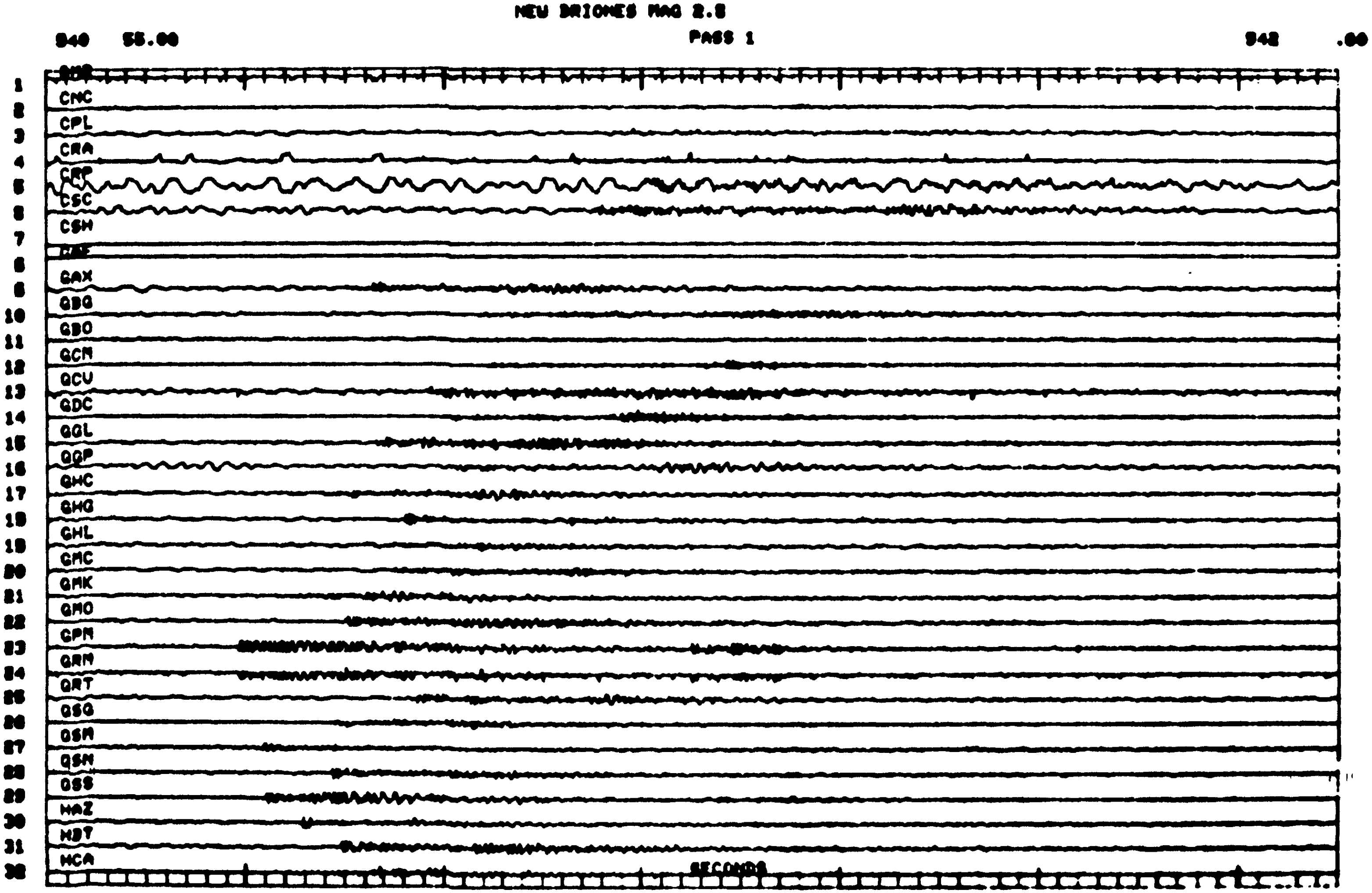
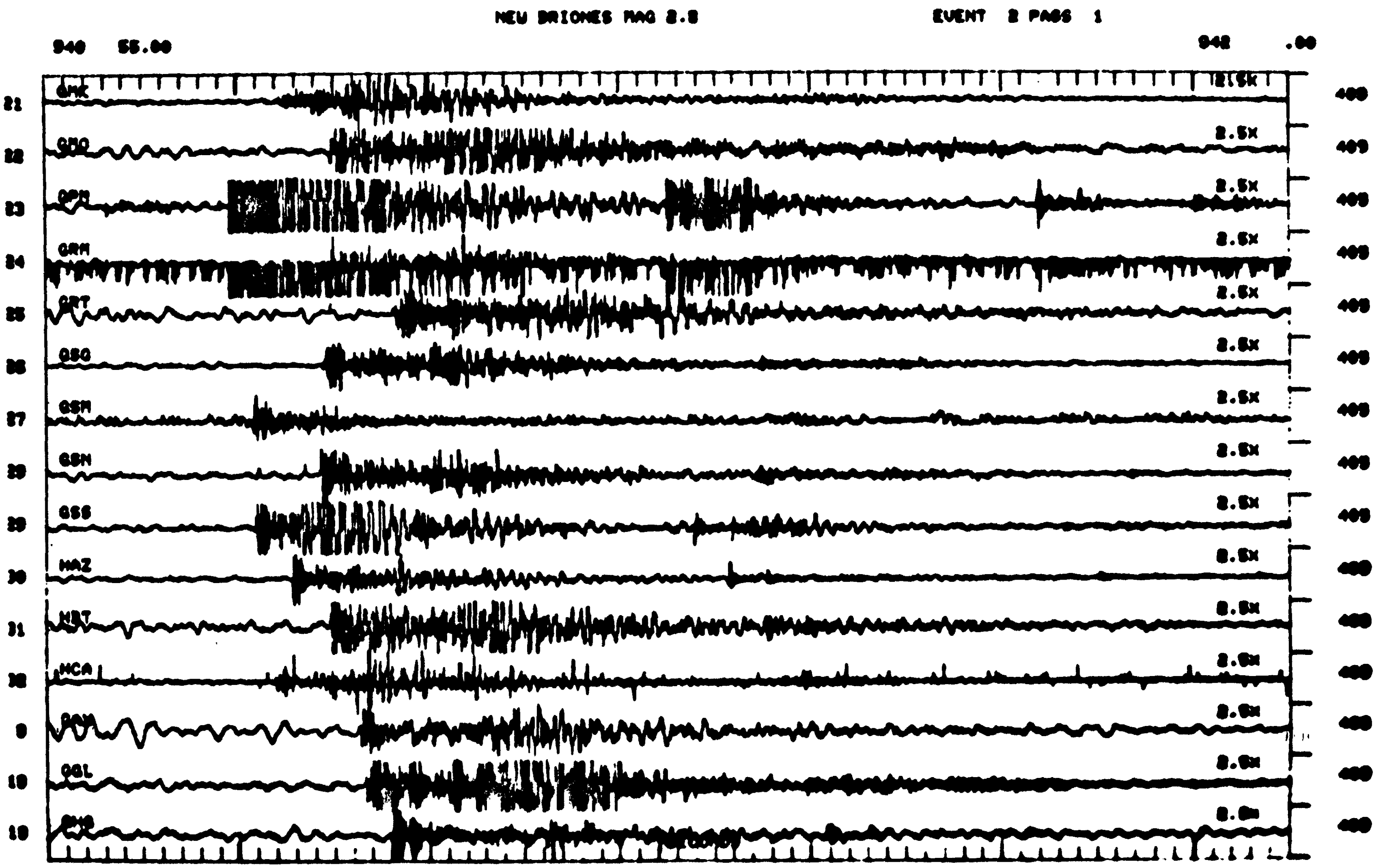


figure 7

COMMAND?



COMMAND?

TRACES SHIFTED BY P-TIMES

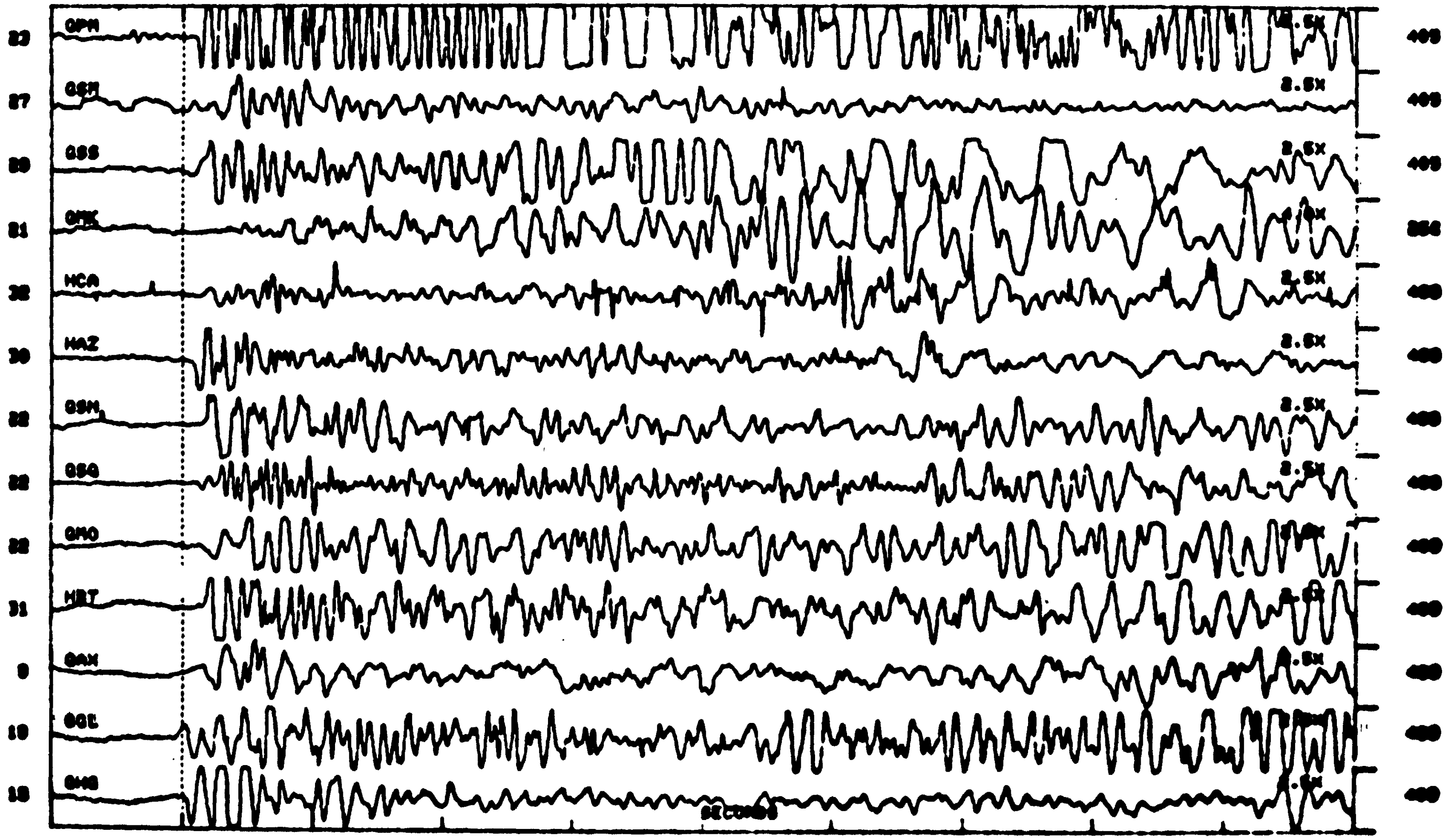


Figure 9

INTERACTIVE SEISMIC DISPLAY SYSTEM

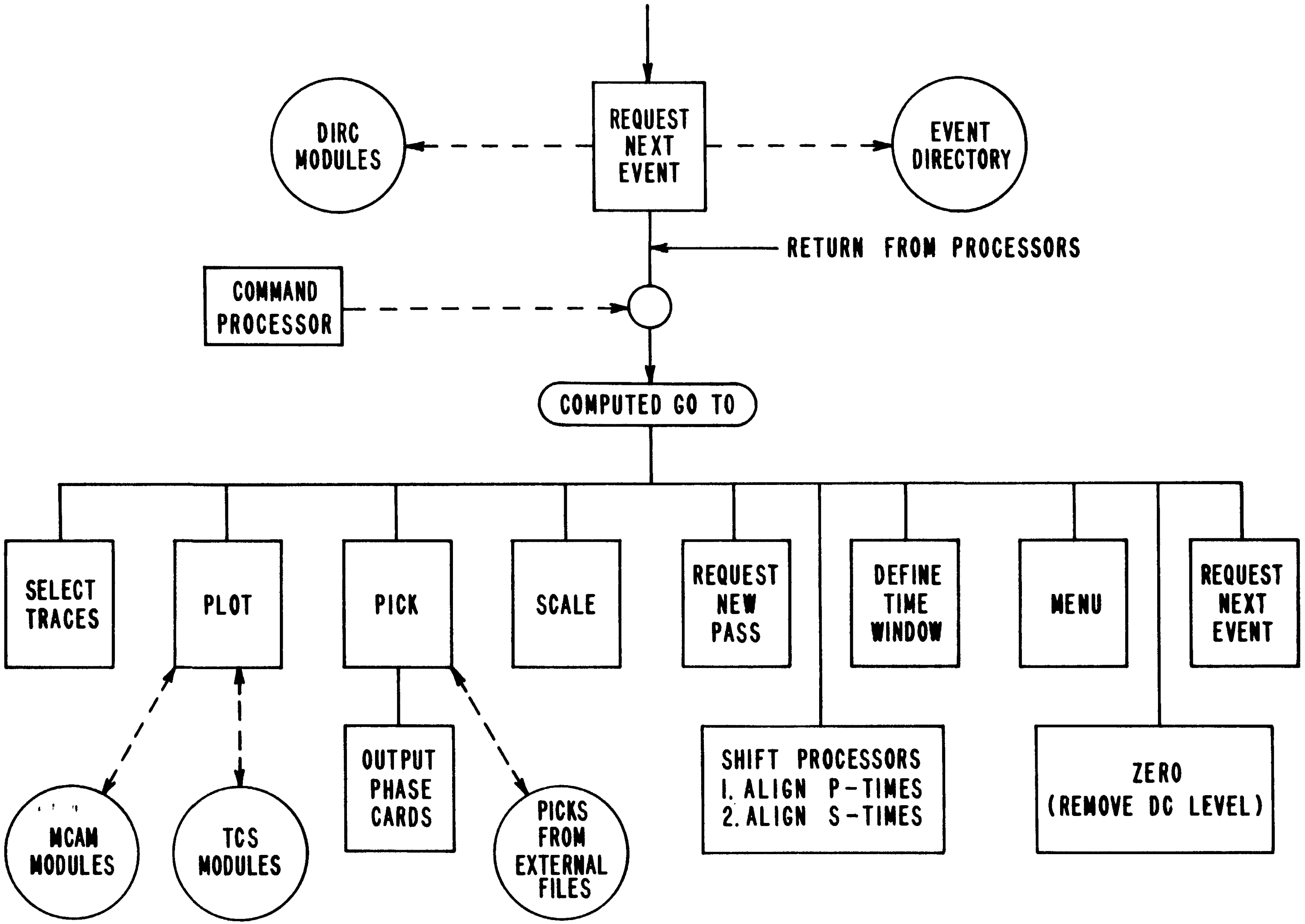
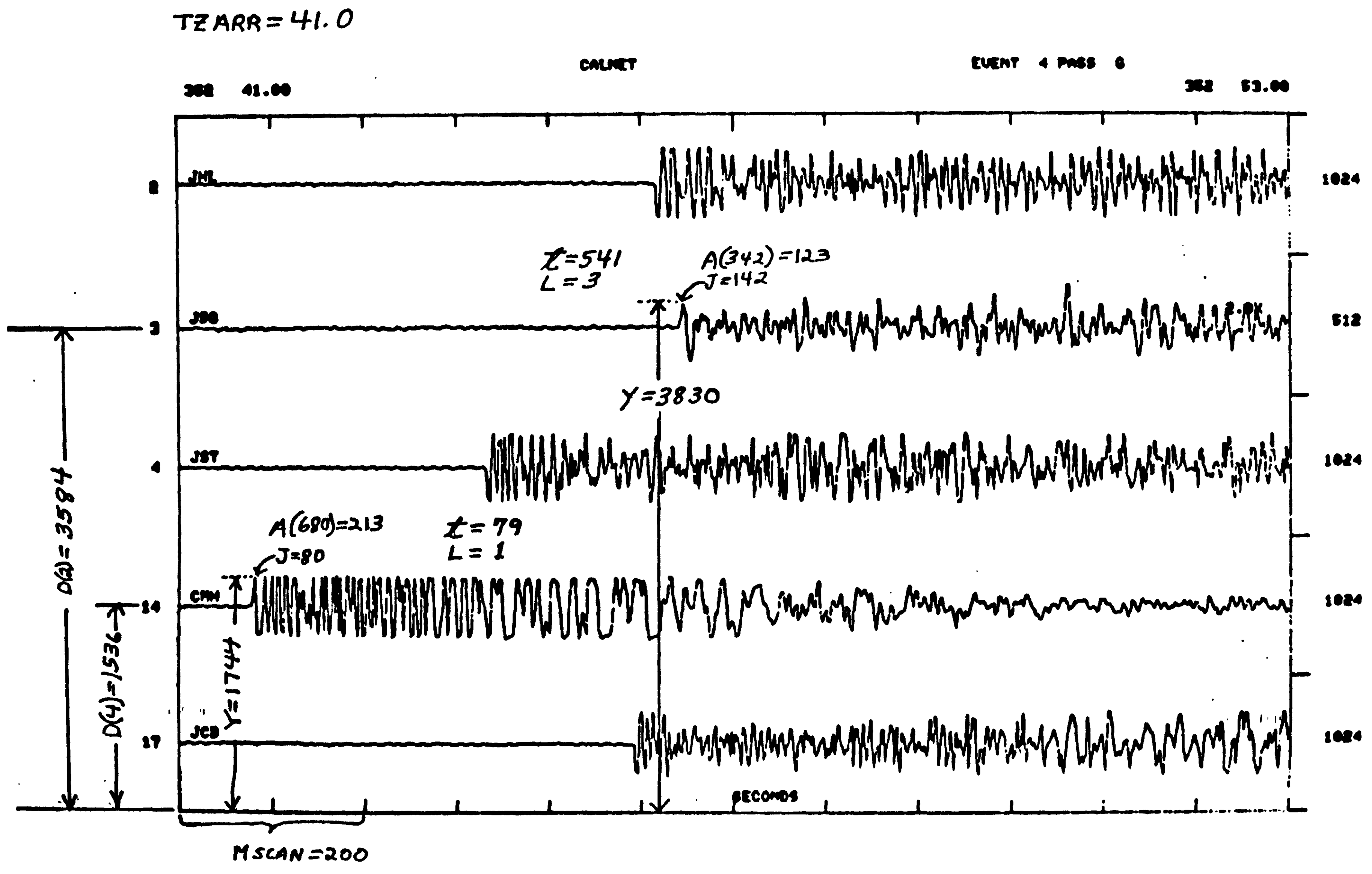


Figure 10


TZARR=41.0
CALNET
EVENT 4 PASS 6
382 41.00
382 53.00
JML
JBG
JST
CMN
JCB
8
3
4
14
17
1024
512
1024
1024
1024
Ɫ=541
L=3
A(342)=123
J=142
Y=3830
A(680)=213
J=80
Ɫ=79
L=1
D(3)=3584
D(4)=1536
Y=1744
SECONDS
MSCAN=200

APPENDIX N: LISTINGS OF ISDS SUBROUTINES

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

                              ARRAY.FR

              SUBROUTINE XARRAY
              OVERLAY OARRAY
              COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32)
             *,D(32)
              COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
              INCLUDE 'CSBLAN'
              PARAMETER MAXX=1000
              COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
              INTEGER A
              COMMON /IOPRM/NS,MSCAN,NREM,MUX,IFILE,IUNIT
              COMMON /DECI/IDF,DECMAX,DLIM
              DIMENSION LFRAM(32)
      CC  S(I)=AMPLITUDE SCALE FACTOR FOR ITH TRACE
      CC  Z(I)=ZERO LEVEL OF ITH TRACE RELATIVE TO DISTANCE
      CC  DT(I)=DATA POINT INTERVAL FOR ITH TRACE
      CC  TZ(I)=ABSOLUTE TIME (SECONDS) OF FIRST DATA POINT
      CC  SHFT(I)=TIME SHIFT (SECONDS) FOR ITH TRACE
      CC  D(I)=DISTANCE OR POSITION OF ITH TRACE ( AMPL. UNITS)
      CC  TZARR=BEGINNING OF DISPLAY WINDOW (SECONDS)
      CC  TWIND=LENGTH OF DISPLAY WINDOW (SECONDS)
      CC  AZ=BOTTOM OF SCREEN IN ARRAY AMPLITUDE UNITS
      CC  ARNG=WIDTH OF EACH TRACE IN RAW AMPLITUDE UNITS
      CC  NT=NUMBER OF TRACES TO BE PLOTTED ON SCREEN
      CC  AL=TOP OF SCREEN IN ARRAY AMPLITUDES UNITS
      CC  IOF=DECIMATION FACTOR (SET TO 1 FOR NORMAL PLOTS)
      CC
      C  COMPUTE SEGMENT LENGTH AS INTEGER MULITIPLE
      C  OF DECIMATION FACTOR
              LL=MAXPTS/NREQ
              NLL=MOD(LL,IDF)
              MSCAN=LL-NLL
              ISCAN=TZARR/DTG+1
              CALL FRAMD(MSCAN,IREQ,NREQ,ISCAN,IERR)
              IF(IERR.NE.0)TYPE 'ARRAY-FRAMD: IERR=',IERR
      C
      C   CHECK FOR MAX NEGATIVE SHIFT VALUE
              XSHF=1000.
              DO 2 J=1,32
              XSHF=AMIN1(XSHF,SHFT(J))
      C   INITIALIZE LFRAM
              LFRAM(J)=D(J)
      2       CONTINUE
      CC  DEFINE VIRTUAL SPACE
      CC
      C    MSCAN IS EXPECTED NUMBER OF SCANS/SEGMENT
      C    NSCAN IS ACTUAL NUMBER OF SCANS/SEGMENT
              NSEG=((TWIND-XSHF)/DTG-1)/MSCAN+1
              NRQ=NREQ
              AR=AZ+AL
              CALL VWINDO(TZARR,TWIND,AZ,AR)
      C   TIC MARK SETUP
              TIC=10.
              IC=2
      C
              CALL FRAME(TZARR,TWIND,AZ,AR,TIC,IC)
              DO 20 L=1,NSEG
              CALL GETFM(A,NSCAN,IEND)
              IF(IEND.EQ.1.AND.NSCAN.EQ.0)GO TO 97
              DO 10 I=1,NRQ
      II=IREQ(I)
      TIME=TZARR+TWIND-SHFT(II)
      NPTS=TIME/DT(II)+1.
      IF(NPTS.GT.MAXPTS)NPTS=MAXPTS
CC
CC  PLOT TRACE I
      IJ=MSCAN*(I-1)+1
      Y=(LFRAM(I)-Z(II))*S(II)+D(I)
      X=SHFT(II)+TZ(II)+DT(II)*((L-1)*MSCAN-1)+TZARR
      CALL MOVEA(X,Y)
      NN=NPTS-1
      DO 1 J=1,NSCAN,IDF
      IJ=MSCAN*(I-1)+J
      Y=S(II)*(A(IJ)-Z(II))+D(I)
C  ARG IS ARTIFACT TO GET AROUND 16-BIT INTEGER OVERFLOW
      ARG=(L-1)*MSCAN+J-1
      X=DT(II)*ARG+TZ(II)+SHFT(II)+TZARR
      CALL DRAWA(X,Y)
1     CONTINUE
      LFRAM(I)=A(IJ)
10    CONTINUE
      IF(IEND.EQ.1)GOTO 97
20    CONTINUE
97    CONTINUE
C--------------------------
      CALL TSEND
      CALL HITEK(1,0)
      CALL HOME
99    RETURN
      END
```

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

BOX.FR

```
      SUBROUTINE XBOX(IX,LX,IY,LY)
      OVERLAY OBOX
      CALL SWINDO(IX,LX,IY,LY)
CC  DRAW BOX
CC
      CALL MOVABS(IX,IY)
      CALL DRWREL(LX,0)
      CALL DRWREL(0,LY)
      CALL DRWREL(-LX,0)
      CALL DRWREL(0,-LY)
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- THURSDAY, NOVEMBER 9, 1978 3:48:20 PM

COMMAND.FR

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

```
ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978  2:08:28 PM

                               DECIM.FR
```

```
      SUBROUTINE XDECIM
      OVERLAY ODECIM
C  INPUT: TWIND  (IN /BPARM/) PLOT WINDOW TIME
C         DTG    (IN /BPARM/) SAMPLE TIME
C         DECMAX (IN /DECI/) MAX NUMBER OF DECIMATED SAMPLES
C         NREQ   (IN / /) NUMBER OF TRACES
C  OUTPUT: IDF   (IN /DECI/) DECIMATION FACTOR
C
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      COMMON /DECI/IDF,DECMAX,DLIM
C
C  --- COMPUTE APPROPRIATE DECIMATION FACTOR ---
C
C    TOTAL=TOTAL NUMBER OF DIGITAL POINTS FOR DISPLAY
C    IF THERE WERE NO DECIMATION
C
      TOTAL=(TWIND*NREQ)/DTG
C
C    IDF=DECIMATION FACTOR
C    IDF >= TOTAL/DECMAX
C
      DF=TOTAL/DECMAX
      IDF=DF
      R=DF-IDF
      IF(R.GT.0.)IDF=IDF+1
C -------------------------------
C
      RETURN
      END
      SUBROUTINE XDEFAULT
C  - - - P.R.STEVENSON (8/21/78) - - -
      OVERLAY ODEFAULT
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'CSPICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      DATA IBLNK/2H  /,IPRMK/2H P/,ISRMK/2H S/
      COMMON /DECI/IDF,DECMAX,DLIM
      COMMON /ZZZ/IZIT
      JMAX=32
C  IGD=DEFAULT PICK GRADE ASSIGNMENT
      IGD=0
      IF(IZIT.GT.0)GO TO 700
C THIS SECTION ON INITIAL CALL ONLY
C  IT SETS CERTAIN GLOBAL PARAMETERS
      IZIT=1
      MAXPTS=MAXX
      DECMAX=40000.
      DLIM=5.
CC
      KX=10
      LX=1000
      KY=10
      LY=600
      ITERM=1
      ISCAL=1024
      JCHAR=4
CC
      DTG=.005
      SG=1.
C  ARNG=1024 CORRESPONDS TO 10-BIT A/D CONVERTER
      ARNG=1024.
      AZ=0.
      TZARR=0.
      TWIND=10.
      THR=0.
      DO 1 J=1,JMAX
      S(J)=SG
      DT(J)=DTG
      TZ(J)=TZARR
      SHFT(J)=0.
      Z(J)=0.
      D(J)=0.
1     CONTINUE
C THIS SECTION(/DPARM/) RE-INITIALIZED ON
```

```
C  ALL CALLS TO DEFAULT
700   CONTINUE
      DO 2 J=1,JMAX
      PTIME(J)=0.
      AMP(J)=0.
      STIME(J)=0.
      FTIME(J)=0.
      CROSS(J)=0.
      SMP(J)=0.
      FMP(J)=0.
      PCROSS(J)=0.
      IGRADE(J)=IGD
      SGRADE(J)=IGD
      IFMOT(J)=IBLNK
      PRMK(J)=IPRMK
      SRMK(J)=ISRMK
      SAMP(J)=0.
      AX(J)=0.
      SX(J)=0.
      RMK(1,J)=IBLNK
      RMK(2,J)=IBLNK
      RMK(3,J)=IBLNK
      PRES(J)=0.
      SRES(J)=0.
      EPD(J)=0.
      CLK(J)=0.
2     CONTINUE
CC
CC   AL,NT ARE NOT INITIALIZED HERE
      RETURN
      END
```

DEPRC.FR

```
      SUBROUTINE XDEPRC(IC,JC)
      OVERLAY ODEPRC
      COMMON /TVTUNE/ITV,ITX
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      DIMENSION NNN(80),SQ(40)
      DATA IQ/63/
      DATA IY/1HY/
      DATA IX/1HX/
      DATA IBL/1H /
      INTEGER THR,TMN
C*  DEFINE THE WINDOW
1     CONTINUE
      IF(JC.EQ.19)GO TO 13
      IF(JC.EQ.20)GO TO 14
C*  DEFINE TIME WINDOW
13    WRITE(ITV,210)
210   FORMAT(19H DEFINE TIME WINDOW)
      WRITE(ITV,211)IQ
211   FORMAT(23H WINDOW WIDTH (SECONDS),R1)
C*
C*  READ TWIND
C*  READ TIME WINDOW LENGTH
      READ(ITX,234)NNN(1)
 234  FORMAT(S80)
      CALL INFREE(NNN,RQ,80)
      TWIND=RQ
C*
C*
C*  READ HRMIN SEC
      WRITE(ITV,222)KHR,KMN,RSEC
222   FORMAT(16H DATA STARTS AT ,I2,1H,,I2,2X,F7.2)
      WRITE(ITV,204)IQ
204   FORMAT(23H HR MIN SEC (FREE-FORM),R1,' [CR] GETS DEFAULT')
      WRITE(ITV,223)
223   FORMAT(" TYPE 'X R' FOR AN R SEC. OFFSET OF ORIGIN")
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      IF(NNN(1).EQ.IBL)GOTO 55
      IF(NNN(1).EQ.IX)GOTO 55
      THR=SQ(1)
      TMN=SQ(2)
      TSEC=SQ(3)
      GOTO 56
C   COME HERE FOR CR OR X MN RESPONSE
C
55    THR=KHR
      TMN=KMN
      TSEC=RSEC
      IF(NNN(1).EQ.IX)TSEC=RSEC+SQ(1)
C
56    CONTINUE
C COMPUTE TIME WINDOW RELATIVE TO DATA
C  TTT IS REQUESTED TIME IN SECONDS
C  RRR IS START OF DATA WINDOW IN SECONDS
      TTT=THR*3600+TMN*60+TSEC
      RRR=KHR*3600+KMN*60+RSEC
      TZARR=TTT-RRR
      IF(TZARR.LT.0.)WRITE(ITV,220)
      IF(TZARR.GT.DUR)WRITE(ITV,220)
      IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE '/JDATE/:'
     *,' IHR=',IHR,' IMN=',IMN,' TSEC=',TSEC
      IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE ' TZARR=',
     *TZARR,'TSEC=',TSEC,' RSEC=',RSEC,' TTT=',TTT,
     *' RRR=',RRR
220   FORMAT(36H REQUESTED TIME OUT OF RANGE OF DATA)
203   FORMAT(32H STARTING TIME WINDOW (HRMN,SEC),2X,2F4.0,1X,F7.2)
C*
      WRITE(ITV,203)THR,TMN,TSEC
      WRITE(ITV,221)TZARR
221   FORMAT(1H ,F7.2,15H SEC. INTO DATA)
      IHR=THR
      IMN=TMN
      OTZARR=TZARR
      RETURN
14    WRITE(ITV,205)IQ
205   FORMAT(10H 4014 TERM,R1,6HY OR N)
      READ(ITX,701)IA
701   FORMAT(A1)
      CALL TERM(1,1024)
      ITERM=1
      IF(IA.EQ.IY)CALL TERM(3,4096)
      IF(IA.EQ.IY)ITERM=4
      LWIDTH=1000
      LHY=600
      JX=10
      JY=10
      WRITE(ITV,206)IQ
206   FORMAT(12H RESET FRAME,R1)
      READ(ITX,701)IA
      IF(IA.NE.IY)GO TO 12
      WRITE(ITV,207)
207   FORMAT(33H INPUT WIDTH, HEIGHTH-(FREE FORM))
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      LWIDTH=SQ(1)
      LHY=SQ(2)
      WRITE(ITV,208)
208   FORMAT(39H INPUT SCREEN ORIGIN-IX,IY--(FREE FORM))
```

```
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      JX=SQ(1)
      JY=SQ(2)
   12 CONTINUE
      CALL SWND(JX,LWIDTH,JY,LHY)
      ISCAL=1024*ITERM
      CALL CHRSIZ(JCHAR)
C*
      RETURN
      END
      SUBROUTINE XEVPRC(IC,JC)
      OVERLAY OEVPRC
      INCLUDE 'CSGPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      DIMENSION MSG(6)
      DIMENSION NNN(41),SQ(10)
      DATA MSG/80,65,83,83,32,48/
      COMMON /ZER/IZER
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),OT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /CPARM/NSTA,ISTA(2,32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,ICHAR
      COMMON /TVTUNE/ITV,ITX
      COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
      COMMON /MISC/ZX,ZY,CX,CY
      COMMON /DECI/IDF,DECMAX,DLIM
      INTEGER CX,CY,ZX,ZY
C   EVENT REQUEST PROCESSOR
      IF(IC.EQ.6)GO TO 52
      WRITE(ITV,302)
302   FORMAT(20H SELECT EVENT NUMBER)
      READ(ITX,234)NNN(1)
 234  FORMAT(S80)
      CALL INFREE(NNN,SQ,80)
      EVENT=SQ(1)
      CALL RQEVT(LIST,EVENT)
      IF(LIST(1).NE.0)GOTO 50
C OUTPUT ERROR MESSAGE AND RETURN
      WRITE(ITV,303)
303   FORMAT(18H EVENT NOT ON DISK)
      RETURN
50    CONTINUE
C LOAD BPARM AND KDATE VARIBLES#
      KYR=LIST(2)
      KMO=LIST(3)
      KDA=LIST(4)
      KHR=LIST(5)
      KMN=LIST(6)
      RSEC=SEC
      DUR=DURAT
      DTG=DELT
```

```
C  REZERO TZARR (POINT TO START OF DATA)
      TZARR=0.
      OTZARR=TZARR
C     TYPE 'SEC,DURAT,DELT=',SEC,DURAT,DELT
      NPASS=LIST(LSZ1)
C LOAD DT ARRAY
      DO 40 J=1,32
40    DT(J)=DTG
C
C  EQUATE JDATE WITH KDATE FOR INITIAL PLOTS
      IHR=KHR
      IMN=KMN
      TSEC=RSEC
C
C  IGAP IS SIZE OF GAP(IN TEKPOINTS) BETWEEN VIEWPORTS
      IGAP=120/NPASS
      KX1=30
C  COMPUTE WIDTH OF VIEWPORTS
C
      LXX=900/NPASS-(NPASS-1)*IGAP
      JY=10
      LHY=600
      CALL ERASE
C
      DO 51 N=1,NPASS
C  COMPUTE X-ORIGIN OF NTH VIEWPORT
      JX=KX1+(N-1)*(IGAP+LXX)
      CALL SWND(JX,LXX,JY,LHY)
C  SAVE SCREEN CO-ORDINATES OF CURRENT VIEWPORT
      VSAVE(1,N)=JX
      VSAVE(2,N)=LXX
      VSAVE(3,N)=JY
      VSAVE(4,N)=LHY
C PLOT THIS PASS
C  SETUP I/O TO PLOT ALL DATA
      MUX=LIST(N+LSZ1)
      IF(MUX.EQ.0)GOTO 51
      NCHAN=MUX
      NREQ=MUX
C  EQUATE PREVIOUS TRACE STATE TO CURRENT STATE
      NTEMP=NREQ
      DO 30 J=1,NREQ
      ITEMP(J)=J
30    IREQ(J)=J
C
C  SET WINDOW TO DURATION TIME OF DATA
      TWIND=DUR
      CALL DECIM
C
      CALL RQPAZ(ISTA,EVENT,N)
C
      CALL NOSHF
C  SPEED BY-PASS
C   COMMAND EVSP MEANS GET EVENT BUT DON'T PLOT
C  COMMAND EVSP CORRESPONDS TO JC=3
      IF(JC.NE.3)GOTO 51
C
      CALL RSETD
      IF(IZER.EQ.1)CALL ZLEV
      CALL ARRAY
      CALL LBSTA(1)
      MXX=LXX/2+JX
      MYY=LHY+JY+20
      CALL MOVABS(MXX,MYY)
      MSG(6)=48+N
      CALL ANSTR(6,MSG)
51    CONTINUE
      IF(JC.NE.3)GOTO 52
C   LABEL PLOT WITH EVENT NAME
     :CALL ENAME(NNN,EVENT)
      MZZ=LHY+JY+40
      CALL MOVABS(358,MZZ)
      CALL AOUTST(40,NNN)
C
C  PASS REQUEST PROCESSOR
      CALL HOME
      CALL ANMODE
      CALL TSEND
52    CONTINUE
      WRITE(ITV,304)
304   FORMAT(19H SELECT PASS NUMBER)
      CALL NEWLIN
      CALL SEELOC(CX,CY)
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      PASS=SQ(1)
      N=PASS
      CALL RQPAZ(ISTA,EVENT,PASS)
C  - - - DEFINE TRACES FOR THIS PASS - - -
      MUX=LIST(LSZ1+N)
      IF(MUX.EQ.0) GOTO 99
      NREG=MUX
      NTEMP=MUX
      DO 1 J=1,NREG
      ITEMP(J)=J
1     IREG(J)=J
      CALL NOSHF
      CALL RSETD
C  DO THINGS USUALLY DONE BY ARRAY
      AR=AZ+AL
      CALL VWINDO(TZARR,TWIND,AZ,AR)
      CALL SWND(VSAVE(1,N),VSAVE(2,N),VSAVE(3,N),VSAVE(4,N))
      CALL SWINDO(IX,LX,IY,LY)
C - - - - - - - - - - - - - - -
99    CONTINUE
C  RESET DECIMATION FACTOR
      IOF=1
C  NULL OUT DPARM BLOCK
      CALL DEFAULT
      RETURN
      END
      SUBROUTINE XFAZPU(IS)
C     - - - P.R.STEVENSON - - -   8/24/78
      OVERLAY OFAZPU
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'CSPICK'
      INTEGER PRMK,SRMK,RNK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFNOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'CSGPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INTEGER OLDEVT,PUFLG,PUBLNK
      DATA IBLANK/2H  /,IZZ/2HZZ/,IPRMK/2H P/,ISRMK/2H S/
      DATA IP/1HP/,IY/1HY/
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
C
C
C       PUFLG=0,FIRST TIME THRU FOR THIS PHASE FILE
C
C       THIS ROUTINE PUNCHES PHASE CARDS DESTINED FOR USGS EARTHQUAKE
C       HYPOCENTER ROUTINES
C       IF IS=2, PUNCH BLANK AND ZZ DELIMITER
C       AND RETURN
C       CHECK TO SEE IF SAME EVENT AS BEFORE
      GOTO(10,20),IS
20    IF(PUFLG.EQ.0)RETURN
      OPEN 7,NFIL,ATT='A'
      WRITE(7,701)IBLANK
C*    WRITE(7,701)IZZ
      RETURN
10    CONTINUE
      WRITE(ITV,401)NFIL
401   FORMAT(' PHASE FILE=',20A2,)
      OPEN 7,NFIL,ATT='A'
C  FIRST TIME THRU, JUST PUNCH LIST
      IF(PUFLG.EQ.0)GOTO 40
C  SAME EVENT, JUST PUNCH LIST
      IF(OLDEVT.EQ.EVENT)GOTO 30
C  OTHERWISE PUNCH BLANK FIRST
      WRITE(7,701)IBLANK
701   FORMAT(A2,78X)
40    WRITE(ITV,400)
400   FORMAT("NEW PHASE FILE?(Y OR N)")
```

```
      READ(ITX,200,ERR=30)IYY
200   FORMAT(A1)
      IF(IYY.NE.IY)GOTO 30
      PUFLG=0
      CLOSE 7
      WRITE(ITY,402)
402   FORMAT("NAME OF PHASE FILE?")
      READ(ITX,234)NFIL(1)
234   FORMAT(S17)
      OPEN 7,NFIL,ATT='A'
30    CONTINUE
C  PUNCH PHASE LIST
      DO 1 J=1,32
C  CHECK FOR NULL STATIONS
      IF(PTIME(J).LE.0..AND.STIME(J).LE.0..AND.
     *SAMP(J).LE.0..AND.AMP(J).LE.0.)GO TO 1
C  CHECK FOR NULL LABELS
      IF(ISTA(1,J).EQ.IBLANK)GOTO 1
C  OUTPUT PHASE CARD FOR THIS STATION
C  THE FOLLOWING ALGORTHM CORRECTS FOR THE PROPER P(OR S) SECOND
C   AND RIPPLES THE CORRECTION THRU THE CLOCK AND CALENDAR.
C   IF ONLY S TIME IS PRESENT IT IS RIPPLED.IF BOTH P AND S
C   ARE PRESENT THEN THE P TIME IS RIPPLED.THE S TIME MUST BE
C   THE SAME MINUTE,HOUR,ETC. AS THE P TIME.
      PTA=PTIME(J)+RSEC
      ST=0.
      PT=0.
      IF(PTIME(J).EQ.0..AND.STIME(J).GT.0.)PTA=STIME(J)+RSEC
C
      XT=AMOD(PTA,60)
      JT=(PTA-XT)/60.
      IF(PTIME(J).GT.0.)PT=PTIME(J)+RSEC-JT*60.
      IF(STIME(J).GT.0.)ST=STIME(J)+RSEC-JT*60.
      JMN=KMN+JT
      JHR=KHR
      JDA=KDA
      IF(JMN.GT.60)JHR=KHR+1
      IF(JMN.GT.60)JMN=KMN-60
      IF(JHR.GT.24)JDA=KDA+1
      IF(JHR.GT.24)JHR=KHR-24
      IAMP=AMP(J)
      ISAMP=SAMP(J)
C  COMPUTE CODA FROM FTIME AND PTIME
      FMP(J)=FTIME(J)-PTIME(J)
      IF(FMP(J).LE.0.)FMP(J)=0.
C  IF STIME=0 THEN SGRADE=4 AND SRMK=BLANK
C   IF PTIME=0 THEN IGRADE=4 AND PRMK=BLANK
      IF(ST.EQ.0.)SGRADE(J)=4
      IF(ST.EQ.0.)SRMK(J)=IBLANK
      IF(PT.EQ.0.)IGRADE(J)=4
      IF(PT.EQ.0.)PRMK(J)=IBLANK
C
C
71    WRITE(7,700)ISTA(1,J),ISTA(2,J),PRMK(J),IFMOT(J),IGRADE(J),
     *KYR,KMO,JDA,JHR,JMN,PT,IAMP,ST,SRMK(J),SGRADE(J),ISAMP,
     *RMK(1,J),RMK(2,J),CLK(J),RMK(3,J),FMP(J)
700   FORMAT(2A2,A2,A1,I1,1X,5I2,F5.2,I5,2X,F5.2,A2,1X,I1,4X,I3,15X,A2,
     *A1,F5.2,A1,I4,5X)
1     CONTINUE
      PUFLG=1
```

```
        OLDEVT=EVENT
        CLOSE 7
        RETURN
        END
```

```
ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978  2:14:47 PM

                              FRAME.FR

      SUBROUTINE XFRAME(R1,RL,S1,SL,TIC,IC)
      OVERLAY OFRAME
      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,ICHAR
      DIMENSION SXB(2),SXL(2),SBB(2)
CC
CC  IF IC=0, DRAW BOX ONLY
CC  IF IC=1, PUT ON TIC MARKS AT TIC INTERVALS
CC  STARTING AT FIRST UNIT INTERVAL
CC  IF IC=2, ALSO PUT ON FINE TICS AT TIC/10
CC  INTERVALS
CC
      S2=S1+SL
      R2=R1+RL
      S=SL/LY
CC
      CALL BOX(IX,LX,IY,LY)
      IF(IC.EQ.0)GO TO 99
      TT=TIC/10
      Y=R1/TIC+1
      Z=INT(Y)*TIC
      NTIC=RL/TIC+1
CC
      SXB(1)=-15*S
      SXB(2)=15*S
      SXL(1)=-10*S
      SXL(2)=10*S
      SBB(1)=S2
      SBB(2)=S1
CC
      DO 20 L=1,2
CC
      SB=SBB(L)
      SIT=SXB(L)
      SLT=SXL(L)
      IF(IC.EQ.1)GO TO 6
      DO 5 K=1,9
      XTT=Z-K*TT
      IF(XTT.LE.R1)GO TO 6
      SLT=SXL(L)
      CALL MOVEA(XTT,SB)
      CALL DRAWR(0.,SLT)
5     CONTINUE
CC
6     CONTINUE
CC
      DO 1 J=1,NTIC
      XT=Z+(J-1)*TIC
      CALL MOVEA(XT,SB)
CC  LARGE TIC DRAWN DOUBLE
      CALL DRAWR(0.,SIT)
      CALL DRAWR(0.,-SIT)
      IF(IC.EQ.1)GO TO 1
CC
CC  DRAW SMALL TICS
      DO 2 K=1,9
      XTT=XT+TT*K
      IF(XTT.GE.R2)GO TO 1
      CALL MOVEA(XTT,SB)
      CALL DRAWR(0.,SLT)
```

```
2       CONTINUE
CC
1       CONTINUE
20      CONTINUE
CC
99      RETURN
        END
```

```
      SUBROUTINE XGPICK(ICC)
      OVERLAY OGPICK
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      DIMENSION ICARD(41),LAB(3),NTT(3)
      DATA IDOL/2H$$/
      DATA NTT/2HEV,2HEN,2HT /
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      DIMENSION IREC(45),NNN(41)
      DIMENSION NAM(2),IDONE(32),KRMK(2)
C   ICC=1:READ AUTO PICKS
C   ICC=2:READ PHASE LIST
C   ICC=3:READ HYPOUT
C
      WRITE(ITV,150)
150   FORMAT(' FILENAME?')
C  READ '5' RATHER THAN ITX BECAUSE WISH INPUT
C  FROM USER DURING PICK MODE
      READ(5,441)NNN(1)
441   FORMAT(S17)
      OPEN 15,NNN
      GOTO(10,20,30),ICC
10    CONTINUE
3     READ(15,101,END=500)ICARD(1)
101   FORMAT(S80)
C         LOOK FOR EVENT LABEL
      DECODE(ICARD,102) (LAB(J),J=1,3)
102   FORMAT(3A2)
      IF(LAB(1).EQ.NTT(1).AND.LAB(2).EQ.NTT(2).AND.LAB(3).EQ.
     *NTT(3))GOTO 4
      GOTO 3
4     DECODE(ICARD,103)JEVENT
103   FORMAT(6X,I2)
C   - - - TEST FOR CORRECT EVENT NUMBER
      IF (JEVENT.NE.LIST(1))GOTO 3
C
      TYPE 'EVENT NO.',JEVENT
C   ZERO DONE FLAGS
```

```
      DO 5 J=1,32
5      IDONE(J)=0
C  LOOP AND READ UNTIL END OF EVENT
    2 READ(15,100,END=500)NAM,TIME,IFMT,DRTN
100   FORMAT(2A2,F7.2,A2,F7.2)
C  IF NAME = "$$",IT IS THE END OF THE EVENT
      IF(NAM(1).EQ.IDOL)GOTO 500
      DO 1 J=1,32
      IF(ISTA(1,J).NE.NAM(1))GOTO 1
      IF(ISTA(2,J).NE.NAM(2))GOTO 1
C  TAKE FIRST OCCURENCE ONLY
      IF(IDONE(J).EQ.1)GOTO 1
C  LOAD UP OPARM TABLE
      PTIME(J)=TIME
      FMP(J)=DRTN
      IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
      IFMOT(J)=IFMT
      IDONE(J)=1
      GOTO 2
1     CONTINUE
      GOTO 2
C   COME HERE IF NO FILE
 500  CONTINUE
C     RETURN
C  COME HERE WHEN EOF
502   CLOSE 15
      RETURN
C
20       CONTINUE
22       READ(15,200,END=502)NAM,IPR,IFMT,IG,JYR,JMO,
     *JDA,JHR,JMN,TIME,IPA,ST,ISR,ISG,ISA,KRMK,CC,JRMK,DRTN
200   FORMAT(3A2,A1,I1,1X,5I2,F5.2,I5,2X,F5.2,A2,1X,I1,4X,I3,
     *15X,A2,A1,F5.2,A1,F4.0,5X)
      GOTO 24
30    CONTINUE
31    READ(15,300,END=502)IREC
300   FORMAT(45A2)
      DECODE(IREC,301,ERR=502)NAM,IPR,IFMT,IG,JYR,JMO,
     *JDA,JHR,JMN,TIME,PRZ,PWW,ST,ISR,ISG,SRZ,ISA,SWW,
     *PD,SD,EPIDST,ANGLE,CC,JRMK,DRTN,AZ,DMAG,AMAG,PIMP,SIMP
301   FORMAT(3A2,A1,I1,1X,5I2,F5.2,F4.2,F3.2,F5.2,A2,1X,I1,
     *F4.2,F3.0,F3.2,2F4.2,F4.1,F3.0,F5.2,A1,F4.0,
     *F3.0,2F2.1,2F4.3)
24    CONTINUE
      DO 23 J=1,32
C  COMMON SECTION FOR HYPOUT AND FZLINK
      IF(ISTA(1,J).NE.NAM(1))GOTO 23
      IF(ISTA(2,J).NE.NAM(2))GOTO 23
      PTIME(J)=TIME-RSEC
      IF(PTIME(J).LT.0)PTIME(J)=PTIME(J)+60.
      FMP(J)=DRTN
      IFMOT(J)=IFMT
      STIME(J)=ST-RSEC
      IF(STIME(J).LT.PTIME(J).AND.ST.NE.0.)
     *STIME(J)=STIME(J)+60.
      IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
      CROSS(J)=0.
      PCROSS(J)=0.
      IGRADE(J)=IG
      SGRADE(J)=ISG
      SAMP(J)=ISA
      RMK(3,J)=JRMK
      SRMK(J)=ISR
      PRMK(J)=IPR
C  IF NOT FZLINK,SKIP NEXT SECTION
      IF(ICC.NE.2)GOTO 25
      AMP(J)=IPA
      RMK(1,J)=KRMK(1)
      RMK(2,J)=KRMK(2)
C IF NOT HYPOUT,SKIP NEXT SECTION
25    IF(ICC.NE.3)GOTO 26
      PRES(J)=PRZ
      SRES(J)=SRZ
      EPD(J)=EPIDST
      CLK(J)=CC
      GOTO 26
23    CONTINUE
C  COME HERE AFTER SUCCESSFUL MATCH
26    CONTINUE
      GOTO(502,22,31),ICC
C

      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:16:56 PM

INFREE.FR

```
      SUBROUTINE XINFREE(L,R,NC)
      OVERLAY OINFREE
      INTEGER P,S,D
      DIMENSION R(1),L(1),LL(96)
      DECODE(L,440)(LL(J),J=1,96)
  440 FORMAT(96R1)
      MINUS=45
      NZER=48
      N9=57
      NPERI=46
      I=0
      J=0
    6 J=J+1
      N=0
      P=0
      S=0
      M=1
      F=1.
  5   I=I+1
      IF(I.GT.NC)RETURN
      D=LL(I)-8192
      K=4
      IF(D.EQ.MINUS)K=2
      IF(D.GE.NZER.AND.D.LE.N9)K=1
      IF(D.EQ.NPERI)K=3
      K=K+S
      GOTO(1,2,3,5,1,4,3,4)K
  1    N=N*10+D-NZER
      S=4
      GO TO 5
  2   M=-1
      S=4
      GO TO 5
  3   P=I
      S=4
      GO TO 5
  4    IF(P.NE.0)F=10.**(I-P-1)
      R(J)=N/F*M
      GO TO 6
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, OCTOBER 27, 1978 7:09:54 PM

ISDS7.FR

```
C     PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
C --WRITTEN BY PETER R. STEVENSON (1976-78)--
C --U.S.GEOLOGICAL SURVEY--
C --OFFICE OF EARTHQUAKE STUDIES--
C --MENLO PARK,CALIF.--
      COMMON /TVTUNE/ITV,ITX
      COMMON /RPLAY/VFILE
      INTEGER VFILE
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      COMMON /ZER/IZER
      COMMON/ZZZ/IZIT
      COMMON /DECI/IDF,DECMAX,DLIM
      INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$APARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'C$BPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /SCAL/ISDUM(7)
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE  'C$PICK'
      INTEGER PRNK,SRNK,RNK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SHCOM/ISH
      COMMON /IOPRM/IODUM(6)
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      INTEGER OLDEVT,PUFLG,PUBLNK
      DIMENSION NNN(20)/2HFA,2HZL,2HST,17*0/
      COMMON /MISC/MDUM(4)
C  LOAD PHASE FILE NAME WITH DEFAULT 'FAZLST'
      DO 10 J=1,20
10    NFIL(J)=NNN(J)
      IZER=0
      IZIT=0
        ITV=4
```

```
       ITX=5
      OPEN 4,'$TTO1'
      OPEN 5,'$TTI1'
      OPEN 3,'JTEK'
      VFILE=-1
      OPEN 9,'TEMP'
      CALL OVOPN("ISDS.OL",IER)
      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER
      CALL INITT(0)
      CALL SETBUF(3)
      CALL CHRSIZ(4)
      CALL TSEND
      CALL HITEK(1,0)
      CALL MCAM1
      CALL DIRC1
      CALL DEFAULT
      CALL SPLOT
      STOP
      END
```

LBSCL.FR

```
    SUBROUTINE XLBSCL
    OVERLAY OLBSCL
    INCLUDE 'CSBLAN'
    PARAMETER MAXX=1000
    COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
    INTEGER A
    INCLUDE 'CSAPARM'
    COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
    INCLUDE 'CSBPARM'
    COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
    COMMON/SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
    COMMON/SFL/SFLAG
    INTEGER SFLAG
    DIMENSION AA(9)
C  FIND UPPER LEFT CORNER
    PLACE=AL
    XX=TZARR+TWIND
C  VOLT IS VOLTS INTO A/D
    VOLT=5.
C  DVCMM IS DEVELOCORDER FULL-SCALE MILLIMETER EQUIVALENTS
    DVCMM=160.
C
C  DRAW TICS ON RIGHT HAND SIDE,STARTING AT UPPER RIGHT
C  HAND CORNER AND MOVING DOWN ARNG UNITS FOR EACH TIC
C
1   CALL MOVEA(XX,PLACE)
    JL=15
    CALL DRWREL(JL,0)
    CALL DRWREL(-JL,0)
    PLACE=PLACE-ARNG
    IF(PLACE.LT.AZ)GOTO 2
    GOTO 1
C
C  FIND MIDDLE OF FIRST INTERVAL
2   FIRST=AL-.5*ARNG
C  LABEL MIDDLE OF EACH INTERVAL WITH APPROPRIATE
C  SCALE FACTOR
C  SFLAG INDICATES WHICH SCALING OPTION TO USE
C  SFLAG=1 FOR DIGITAL COUNTS
C  SFLAG=2 FOR VOLTS OUT
C  SFLAG=3 FOR EQUIVALENT DEVELOCORDER MM'S OUT
C
    JW=5
    DO 3 J=1,NREQ
    JJ=IREQ(J)
C
    GOTO (10,20,30),SFLAG
10  ISCL=ARNG/S(JJ)
    ENCODE(AA,100)ISCL
    GOTO 40
C
20  TSCL=VOLT/S(JJ)
    ENCODE(AA,101)TSCL
    GOTO 40
C
30  TSCL=DVCMM/S(JJ)
    ENCODE(AA,102)TSCL
    GOTO 40
```

```
C
40    CALL MOVEA(XX,FIRST)
      CALL MOVREL(JW,0)
      CALL AOUTST(9,AA)
3     FIRST=FIRST-ARNG
C
100   FORMAT(I7,"   ")
101   FORMAT(F8.3,"V")
102   FORMAT(F7.1,"MM")
C
      RETURN
      END
```

```
C
C  PUT WINDOW START TIME ABOVE UPPER RIGHT CORNER
C
      KYY=KY+LY+20
      CALL MOVABS(KX,KYY)
      ENCODE(MTIME,100)IHR,IMN,TSEC
100   FORMAT(2I2,2X,F6.2)
      CALL AOUTST(12,MTIME)
C  PUT WINDOW END TIME ABOVE UPPER LEFT CORNER
C
      TLAST=TZARR+TWIND+RSEC
C  CONVERT TO PROPER TIME
      JJJ=TLAST/60
      TLAST=TLAST-JJJ*60
      LMN=IMN+JJJ
      KKK=LMN/60
      LMN=LMN-KKK*60
      LHR=IHR+KKK
C
      KXX=KX+LX-70
      KYY=KY+LY+20
      CALL MOVABS(KXX,KYY)
      ENCODE(MTIME,100)LHR,LMN,TLAST
      CALL AOUTST(12,MTIME)
C
C LABEL TIME AXIS
2     MXX=LX/2+KX
      MYY=KY+10
      CALL MOVABS(MXX,MYY)
      CALL ANSTR(7,MSG)
      RETURN
      END
      SUBROUTINE XLBSTA(KC)
      OVERLAY OLBSTA
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
      COMMON /CPARM/NSTA,ISTA(2,32)
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
     *D(32)
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      INCLUDE 'CSGPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      DIMENSION MTIME(9),NUM(2)
      DIMENSION MSG(7)
      DATA MSG/83,69,67,79,78,68,83/
      DO 1 K=1,NREQ
      JJ=IREQ(K)
      Y=D(K)
      X=TZARR
C*  POSITION BEAM TO APROPRIATE TRACE
      CALL MOVEA(X,Y)
      CALL MOVREL(10,5)
C*  WRITE STATION NAME
      I1=ISTA(1,JJ)
      I2=ISTA(2,JJ)
      CALL AOUTST(2,I1)
      CALL AOUTST(2,I2)
C   PUT OUT TRACE NUMBER BY TRACE START
      CALL MOVEA(X,Y)
      CALL MOVREL(-30,-4)
      ENCODE(NUM,102)JJ
102   FORMAT(I2)
      CALL AOUTST(2,NUM)
C
C  PUT OUT SCALE FACTOR
      IF(S(JJ).EQ.1)GOTO 1
      Y=D(K)
      X=TZARR+TWIND
      CALL MOVEA(X,Y)
      CALL MOVREL(-90,10)
      ENCODE(MTIME,101)S(JJ)
101   FORMAT(F8.1,'X')
      CALL AOUTST(9,MTIME)
1     CONTINUE
C  IF KC=2,SKIP TIME LABELS
      IF(KC.EQ.2)GOTO 2
C
C  LABEL BOX WITH CURRENT TIME
```

LPICK.FR

```
      SUBROUTINE XLPICK
      OVERLAY OLPICK
CC  SUBROUTINE TO DISPLAY PICKS
C  P.R.STEVENSON  8/21/78
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
     *D(32)
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,HT,DTG,
     *SG
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /MISC/ZX,ZY,CX,CY
      INTEGER ZX,ZY,CX,CY
      DIMENSION MSGP(40)
      DATA IBLNK/2H  /
      TRNG=TZARR+TWIND
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
2     CONTINUE
CC
3     CONTINUE
      REWIND 9
C  SAVE HEADER ON SCRATCH FILE (L.U. 89)
      WRITE(9,903)
903   FORMAT(33HNU.  STA  P-TIME  P-AMPL.  S-TIME,3X,7HS-AMPL.,2X,
     *3HSMP,5X,3HFMP,2X,4HP-CR,2X,3HFM ,4HPRMK,2X,4HSRMK,2X,1HR)
      CALL SETMRG(150,1023)
      DO 41 K=1,NREQ
      J=IREQ(K)
CC  DRAW MARK FOR P PICK
      IF(PTIME(J)+SHFT(J).LE.TZARR.OR.PTIME(J)+SHFT(J).GT.TRNG)GOTO 20
      CALL PMARK(PTIME(J)+SHFT(J),D(K),80)
C
C  DRAW DOTTED LINE FOR P RESIDUAL
      PRZ=PRES(J)+PTIME(J)+SHFT(J)
      IF(PRZ.LE.TZARR.OR.PRZ.GT.TRNG.OR.PRES(J).EQ.0.)GOTO 20
      CALL PMARK(PRZ,D(K),-82)
C
20    CONTINUE
CC
CC   DRAW MARK FOR S PICK
      IF(STIME(J)+SHFT(J).LE.TZARR.OR.STIME(J)+SHFT(J).GT.TRNG)GOTO 30
      CALL PMARK(STIME(J)+SHFT(J),D(K),83)
C
C   DRAW DOTTED LINE FOR S RESIDUAL
      SRZ=SRES(J)+STIME(J)+SHFT(J)
      IF(SRZ.LE.TZARR.OR.SRZ.GT.TRNG.OR.SRES(J).EQ.0.)GOTO 30
      CALL PMARK(SRZ,D(K),-114)
C
30    CONTINUE
      PCROSS(J)=CROSS(J)-PTIME(J)
      SMP(J)=STIME(J)-PTIME(J)
      FMP(J)=FTIME(J)-PTIME(J)
      IF(PCROSS(J).LT.0.)PCROSS(J)=0.
      IF(SMP(J).LT.0..OR.STIME(J).LT.0.)SMP(J)=0.
      IF(FMP(J).LT.0.)FMP(J)=0.
C  DRAW MARK FOR F PICK
      IF(FTIME(J)+SHFT(J).LE.TZARR.OR.FTIME(J)+SHFT(J).GT.TRNG)GOTO 10
      IF(FMP(J).GT.0.)CALL PMARK(FTIME(J)+SHFT(J),D(K),70)
10    CONTINUE
C PRETTY UP NULL S-TIMES
      IF(STIME(J).LT.0..AND.PTIME(J).GE.0.)STIME(J)=0.
CC
      CALL NEWLIN
      IXX=0
      IYY=750*ITERM
      CALL MOVABS(IXX,IYY)
      CALL ANMODE
      CALL TSEND
      PT=PTIME(J)+RSEC
      ST=STIME(J)+RSEC
      IF(PTIME(J).EQ.0.)PT=0.
      IF(STIME(J).EQ.0.)ST=0.
      IF(ISTA(1,J).EQ.0)ISTA(1,J)=IBLNK
      IF(ISTA(2,J).EQ.0)ISTA(2,J)=IBLNK
      IF(ST.EQ.0.)SGRADE(J)=4
      IF(ST.EQ.0.)SRMK(J)=IBLNK
      IF(PT.EQ.0.)IGRADE(J)=4
      IF(PT.EQ.0.)PRMK(J)=IBLNK
C  SAVE THE JTH ENTRY ON SCRATCH FILE
      WRITE(9,900)J,ISTA(1,J),ISTA(2,J),PT,AMP(J),ST,SAMP(J),SMP(J),
     *FMP(J),PCROSS(J),IFMOT(J),PRMK(J),IGRADE(J),SRMK(J),SGRADE(J),RMK(3,J)
900   FORMAT(I3,1X,2A2,1X,F7.2,2X,F7.2,2X,F7.2,2X,3F7.2,2X,F3.2,2X,A2,
     *2X,A2,I1,3X,A2,I1,2X,A1)
41    CONTINUE
CC
C  READ BACK INFO FROM SCRATCH FILE
      REWIND 9
      READ(9,905)MSGP
905   FORMAT(40A2)
C  OUTPUT HEADER TO TEK SCREEN
      CALL AOUTST(80,MSGP)
      DO 44 K=1,NREQ
      J=IREQ(K)
      READ(9,905)MSGP
```

```
921   CONTINUE
C  SKIP IF BOTH P AND S ARE ZERO
      IF(PTIME(J ).LE.0..AND.STIME(J ).LE.0.)GO TO 44
      CALL NEWLIN
C  OUTPUT ENTRY TO TEK SCREEN
      CALL AOUTST(80,MSGP)
44    CONTINUE
CC
C  SAVE BEAM POSITION
      CALL NEWLIN
      CALL SEELOC(ZX,ZY)
      CALL SETMRG(0,1023)
      RETURN
      END
```

MENU.FR

```
        SUBROUTINE XMENU(JSW)
        OVERLAY OMENU
C - - - P.R.STEVENSON (10/23/78) - - -
        COMMON /TVTUNE/ITV,ITX
        GOTO (1,2),JSW
1       CONTINUE
        WRITE(ITV,700)
700     FORMAT(' COMMAND  FUNCTION',/,' -------  --------')
        WRITE(ITV,701)
701     FORMAT(' EV',7X,'SELECT EVENT',/,' PA',7X,'SELECT PASS',
       */,' EVPL',5X,'SELECT AND PLOT EVENT',
       */,' TR',7X,'SELECT TRACES',/,' DE',7X,'DEFINE TIME WINDOW',
       */,' PL',7X,'PLOT TRACES',
       */,' SESC',5X,'SELECT SCALE UNITS OPTION',
       */,' PLDE',5X,'PLOT DECIMATED',/,' SC',7X,'SCALE TRACES',
       */,' PI',7X,'PICK TRACES',
       */,' SA',7X,'SAVE PICKS AS PHASE CARDS',
       */,' SH',7X,'ALIGN TRACES ON P-',
       *'TIMES',/,' SH-S',5X,'ALIGN TRACES ON S-',
       *'TIMES',/' ZERO',5X,'REMOVE DC LEVEL',/,' ME',7X,'PRINT MENU',
       */,' UNZE',5X,'DO NOT REMOVE DC LEVEL',
       */,' LDCO',5X,'LOAD COMMANDS FROM REMOTE FILE'
       */,' UNCO',5X,'DISABLE REMOTE(LDCO) OPTION'
       */,' COPY',5X,'MAKE HARD COPY',
       */,' PAUSE',4X,'TEMPORARILY RETURN TO MANUAL CONTROL',
       */,' RESU',5X,'RESUME CONTROL WITH REMOTE FILE',
       */,' ?',8X,'PRINT PICK MENU',
       */,' EX',7X,'EXIT PROGRAM')
        RETURN
2       CONTINUE
        WRITE(ITV,800)
900     FORMAT(' KEY  FUNCTION',/,' ---  --------')
        WRITE(ITV,801)
801     FORMAT(' P',4X,'PICK P-TIME',/,
       *' S',4X,'PICK S-TIME',/,' F',4X,'PICK END OF CODA',/,
       *' C',4X,'PICK FIRST ZERO-CROSSING',/,
       *' A',4X,'MEASURE P-AMPLITUDE(PEAK TO PEAK,',/,
       *'  ',4X,'USE SPACE-BAR TO MEASURE THE PEAKS)',/,
       *' Q',4X,'MEASURE S-AMPLITUDE(PEAK TO PEAK,',/,
       *'  ',4X,'USE SPACE-BAR TO MEASURE THE PEAKS)',/,
       *' X',4X,'REZERO P,S,F',/,
       *' R',4X,'ENTER 3 CHAR. REMARK',/,
       *' *',4X,'ENTER 1 CHAR. STATION REMARK(COL.71)',/,
       *' 0-4',2X,'GRADE OF P-PICK',/,' 5-9',2X,'GRADE(+5) OF S-PICK',/
       *' E,I',2X,'EMERGENT OR IMPULSIVE',
       *' P-ARRIVAL',/,' +,U',2X,'UP FIRST MOTION',/,
       *' -,D',2X,'DOWN FIRST MOTION',/,
       *' T',4X,'ENTER CLOCK CORRECTION ',/,
       *' >',4X,'ENTER FMP VALUE (CURSOR SELECTS STATION)',/,
       *' J,K',2X,'EMERGENT OR IMPULSIVE S-ARRIVAL',/,
       *' L',4X,'LIST PICKED VALUES',/,' G',4X,'GET AUTOMATIC PICKS',/,
       *' Y',4X,'GET PHASE LIST',/,
       *' H',4X,'GET HYPOUT FILE',/,
       *' ;',4X,'SPEEDY PICK OPTION:',/,
       *8X,'ENTER 4 CHAR. STRING(PRMK OR SRMK)',/,
       *8X,'CROSS-HAIR DETERMINES POSITION OF',/,
     *8X,'P OR S PICKS.',/,
     *' .',4X,'EXIT PICK ROUTINE')
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, OCTOBER 27, 1978 7:07:43 PM

MISDS7.FR

```
C      PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
C  --WRITTEN BY PETER R. STEVENSON (1976-78)--
C  --U.S.GEOLOGICAL SURVEY--
C  --OFFICE OF EARTHQUAKE STUDIES--
C  --MENLO PARK,CALIF.--
       COMMON /TVTUNE/ITV,ITX
       COMMON /RPLAY/VFILE
       INTEGER VFILE
       COMMON/SFL/SFLAG
       INTEGER SFLAG
       COMMON /ZER/IZER
       COMMON/ZZZ/IZIT
       COMMON /DECI/IDF,DECMAX,DLIM
       INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
       PARAMETER LSZ1=13
       PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
       COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
      *,UNQYR,UNQID,TCORR,FCA,LCA
       COMMON /JDATE/IHR,IMN,TSEC
       COMMON /XXX/XX(3),LIST(LSZ)
       EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
       COMMON /VEE/VSAVE(4,MAXPASS)
       INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
       COMMON /CPARM/NSTA,ISTA(2,32)
       INCLUDE 'C$APARM'
       COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
       INCLUDE 'C$BPARM'
       COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
       COMMON /SCAL/ISDUM(7)
       INCLUDE 'C$BLAN'
       PARAMETER MAXX=1000
       COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
       INTEGER A
       INCLUDE  'C$PICK'
       INTEGER PRNK,SRNK,RNK,SGRADE
       COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
      *CROSS(32),SMP(32),FMP(32),PCROSS(32),
      *IFMOT(32),IGRADE(32),SGRADE(32),
      *SAMP(32),RNK(3,32),SRNK(32),PRNK(32),
      *PRES(32),SRES(32),EPD(32),CLK(32)
       COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
       COMMON /SHCOM/ISH
       COMMON /IOPRM/IODUM(6)
       COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
       INTEGER OLDEVT,PUFLG,PUBLNK
       DIMENSION NNN(20)/2HFA,2HZL,2HST,17*0/
       COMMON /MISC/MDUM(4)
C  LOAD PHASE FILE NAME WITH DEFAULT 'FAZLST'
       DO 10 J=1,20
10     NFIL(J)=NNN(J)
       IZER=0
       IZIT=0
         ITV=4
```

```
        ITX=5
      OPEN 4,'$TTO1'
      OPEN 5,'$TTI1'
      OPEN 3,'JTEK'
      VFILE=-1
      OPEN 9,'TEMP'
      CALL OVOPN("MISDS.OL",IER)
      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER
      CALL INITT(0)
      CALL SETBUF(3)
      CALL CHRSIZ(4)
      CALL TSEND
      CALL MITEK(1,0)
      CALL MCAM1
      CALL DIRC1
      CALL DEFAULT
      CALL SPLOT
      STOP
      END
```

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

NOSHF.FR

```
      SUBROUTINE XNOSHF
      OVERLAY ONOSHF
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      DO 1 K=1,NREQ
      J=IREQ(K)
      SHFT(J)=0.
 1    CONTINUE
      DO 2 K=1,32
 2    Z(K)=0.
      RETURN
      END

```

PICK.FR

```
      SUBROUTINE XPICK
      OVERLAY OPICK
CC  SUBROUTINE TO PICK TRACES FROM ARRAY PLOT
C  - - - P.R.STEVENSON (8/22/78) - - -
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
     *D(32)
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,
     *SG
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /MISC/ZX,ZY,CX,CY
      INTEGER ZX,ZY,CX,CY
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      DIMENSION NFM(9),ITEMP(4),ASCAL(3),JBUF(11)
      DATA NFM/2H+ ,2H- ,2H  ,2HD ,2HU ,2HEP,2HIP,2HES,2HIS/
C  SCALE FACTORS FOR VARIOUS SCALE SELECT OPTIONS
      DATA ASCAL/1.,.2048,6.4/
CC
3     CONTINUE
4     CONTINUE
      CALL VCURSR(IB,TP,AP)
CC
CC  IDENTIFY TRACE FROM AP
      ALST=AL
      KK=1
      DO 40 K=1,NREQ
      ADIST=ABS(AP-D(K))
      IF(ADIST.LT.ALST)KK=K
      IF(ADIST.LT.ALST)ALST=ADIST
40    CONTINUE
C   KK IS RELATIVE INDEX
C    J IS ABSOLUTE INDEX
      J=IREQ(KK)
      DZR=ADIST/S(J)
C* R=ENTER 3 CHAR REMARK
      IF(IB.NE.82)GOTO 12
      CALL AINST(3,ITEMP)
      RMK(1,J)=ITEMP(1)
      RMK(2,J)=ITEMP(2)
12    CONTINUE
C* *=STATION REMARK IN COL 71(WARNING**NOT HYPO71 STANDARD)
      IF(IB.NE.42)GOTO 13
      CALL AINST(1,ITEMP)
      RMK(3,J)=ITEMP(1)
```

```
13    CONTINUE
C* E=EMERGENT P-ARRIVAL
C* I=IMPULSIVE P-ARRIVAL
      IF(IB.EQ.69)PRMK(J)=NFM(6)
      IF(IB.EQ.73)PRMK(J)=NFM(7)
C* J=EMERGENT S-ARRIVAL
C* K=IMPULSIVE S-ARRIVAL
      IF(IB.EQ.74)SRMK(J)=NFM(8)
      IF(IB.EQ.75)SRMK(J)=NFM(9)
C* C=CROSS
      IF(IB.EQ.67)CROSS(J)=TP-SHFT(J)
C* P=PTIME
      IF(IB.EQ.80)PTIME(J)=TP-SHFT(J)
C* A=AMPLITUDE
      IF(IB.NE.65)GOTO 10
      CALL VCURSR(IBB,TP1,AP1)
      CALL VCURSR(JBB,TP2,AP2)
C  ONLY SPACE-BAR CAN PICK PEAKS
      IF(IBB.NE.32.OR.JBB.NE.32)GOTO 10
      AMP(J)=(ABS(AP1-AP2))/(S(J)*ASCAL(SFLAG))
10    CONTINUE
C* Q=S-AMPLITUDE
      IF(IB.NE.81)GOTO 11
      CALL VCURSR(IBB,TP1,AP1)
      CALL VCURSR(JBB,TP2,AP2)
C  ONLY SPACE-BAR CAN PICK PEAKS
      IF(IBB.NE.32.OR.JBB.NE.32)GOTO 11
      SAMP(J)=(ABS(AP1-AP2))/(S(J)*ASCAL(SFLAG))
11    CONTINUE
C* S=STIME
      IF(IB.EQ.83)STIME(J)=TP-SHFT(J)
C* F=TIME OF CODA DIE OUT
      IF(IB.EQ.70)FTIME(J)=TP-SHFT(J)
C* Y=GET PICKS FROM PHASE LIST
      IF(IB.EQ.89)CALL DEFAULT
      IF(IB.EQ.89)CALL GPICK(2)
      IF(IB.EQ.89)CALL LPICK
C
C* H=GET PICKS AND RESIDUALS FROM HYPOUT FOR DISPLAY
      IF(IB.EQ.72)CALL DEFAULT
      IF(IB.EQ.72)CALL GPICK(3)
      IF(IB.EQ.72)CALL LPICK
C* G=GET AUTO PICKS FOR DISPLAY
      IF(IB.EQ.71)CALL DEFAULT
      IF(IB.EQ.71)CALL GPICK(1)
      IF(IB.EQ.71)CALL LPICK
C* NUMBER=ENTER GRADE OF PICK
C* 0-4 ENTER GRADE OF P PICK
      IF(IB.GE.48.AND.IB.LE.52)IGRADE(J)=IB-48
C* 5-9 ENTER GRADE OF S PICK(GRADE IS NUMBER-5)
      IF(IB.GE.54.AND.IB.LE.57)SGRADE(J)=IB-53
C* L=LOOK AT VALUES
      IF(IB.EQ.76)CALL LPICK
C* .=FINISHED WITH THIS ARRAY
      IF(IB.EQ.46)GO TO 7
C
C* ;=SPEEDY PICK MODE
C   FOLLOWING ";",ENTER PRMK OR SRMK (4 CHAR. STRING).
C   THE POSITION OF THE CROSS-HAIR DETERMINES THE
C   PICK. OTHER ELEMENTS ARE DECODED AND STORED IN
```

```
C   THE APPROPRIATE VARIBLES IN DPARM.
C   E.G. ";",<CR>,IPU1 OR ES 2,<CR>.
C
      IF(IB.NE.59)GOTO 14
C  PRELOAD TEMP TO TAKE CARE OF DEFAULT CASE:
C  IE. 3 BLANKS AND AN ASCII ZERO
C
      ITEMP(1)=32
      ITEMP(2)=32
      ITEMP(3)=32
      ITEMP(4)=48
      CALL TINSTR(4,ITEMP)
      IF(ITEMP(2).NE.80)GOTO 15
      PTIME(J)=TP-SHFT(J)
C  ASSIGN P-REMARK AND P 1ST MOTION
      IF(ITEMP(1).EQ.69)PRMK(J)=NFM(6)
      IF(ITEMP(1).EQ.73)PRMK(J)=NFM(7)
      IF(ITEMP(3).EQ.85)IFMOT(J)=NFM(5)
      IF(ITEMP(3).EQ.68)IFMOT(J)=NFM(4)
      IGRADE(J)=ITEMP(4)-48
15    IF(ITEMP(2).NE.83)GOTO 14
C  ASSIGN S-REMARK
      STIME(J)=TP-SHFT(J)
      IF(ITEMP(1).EQ.69)SRMK(J)=NFM(8)
      IF(ITEMP(1).EQ.73)SRMK(J)=NFM(9)
      SGRADE(J)=ITEMP(4)-48
14    CONTINUE
C* >=ENTER FMP MANUALY
C     HORIZONTAL CURSOR SELECTS STATION
      IF(IB.NE.62)GOTO 16
      CALL AINST(20,JBUF)
      CALL INFREE(JBUF,F,20)
      FMP(J)=F
      FTIME(J)=PTIME(J)+FMP(J)
      F=0.
16    CONTINUE
C
C* T=ENTER CLOCK CORRECTION MANUALLY
C  HORIZONTAL CURSOR SELECTS STATION
      IF (IB.NE.84)GOTO 17
      CALL AINST(20,JBUF)
      CALL INFREE(JBUF,T,20)
C  CLOCK CORRECTION IS A GLOBAL FUNCTION
      DO 5 JJ=1,32
5     CLK(JJ)=T
      T=0.
17    CONTINUE
C
C* U,D,+,- = FIRST MOTION
C
      IF(IB.EQ.43)IFMOT(J)=NFM(1)
      IF(IB.EQ.45)IFMOT(J)=NFM(2)
      IF(IB.EQ.85)IFMOT(J)=NFM(5)
      IF(IB.EQ.68)IFMOT(J)=NFM(4)
CC
C   X=REZERO PICKS
      IF(IB.EQ.88)PTIME(J)=0.
      IF(IB.EQ.88)STIME(J)=0.
      IF(IB.EQ.88)FTIME(J)=0.
      GO TO 4
```

```
    7     CONTINUE
          RETURN
          END
```

PLPRC.FR

```
      SUBROUTINE XPLPRC(JC)
      OVERLAY OPLPRC
      COMMON/SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /DECI/IDF,DECMAX,DLIM
      COMMON /MISC/ZX,ZY,CX,CY
      INCLUDE 'CSGPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      DIMENSION NAM(40),MSG(22)
      INTEGER ZX,ZY,CX,CY
      COMMON /ZER/IZER
      COMMON/SFL/SFLAG
      INTEGER SFLAG
C   RESET SHIFTS TO ZERO
      CALL NOSHF
C  RESET DISTANCE
      CALL RSETD
      CALL ERASE
C  DECIMATE IF REQUESTED
      IF(JC.EQ.1)CALL DECIM
C  CHECK FOR ZERO-LEVEL FLAG
      IF(IZER.EQ.1)CALL ZLEV
C  PLOT TRACES
      CALL ARRAY
C  LABEL TRACES
      CALL LBSTA(1)
C  PUT EVENT NAME ON PLOT
      CALL ENAME(NAM,EVENT)
      LHY=600
      MZZ=LHY+KY+40
      CALL MOVABS(358,MZZ)
      CALL AOUTST(40,NAM)
C
C   PUT EVENT # AND PASS # ON THE PLOT
      ENCODE(MSG,100)EVENT,PASS
100   FORMAT(' EVENT',I3,' PASS',I3)
      CALL AOUTST(17,MSG)
C   PUT DECIMATION FACTOR ON SCREEN IF NOT 1
      ENCODE(MSG,101)IDF
101   FORMAT(' DECIMATION FACTOR=',I3)
      CALL MOVREL(35,0)
      IF(IDF.NE.1)CALL AOUTST(22,MSG)
C
C  PUT SCALE UNITS ON RIGHT SIDE OF PLOT
      CALL LBSCL
C
      CALL HOME
C  UPDATE BEAM POINTERS
      CALL SEELOC(CX,CY)
      ZX=0
```

```
        ZY=750*ITERM
  C  RESTORE DECIMATION FACTOR
        IDF=1
        RETURN
        END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, AUGUST 21, 1978 11:54:21 AM

PMARK.FR

```
      SUBROUTINE XPMARK(X,Y,IA)
      OVERLAY OPMARK
CC
C   P.R.STEVENSON 8/21/78
C
CC  PLOT VERTICAL LINE AT (X,Y) AND
CC  LABEL WITH ADE CHARACTER IA
CC
      CALL MOVEA(X,Y)
      CALL MOVREL(0,-20)
C  IF IA > 0 , DRAW VERTICAL SOLID LINE
      IF(IA.LT.0)GOTO 1
      CALL DRWREL(0,40)
      CALL MOVREL(-5,0)
      CALL ANCHO(IA)
      GOTO 2
C  ELSE DRAW VERTICAL DOTTED LINE
1     CALL DSHREL(0,40,1)
      CALL MOVREL(-5,0)
      CALL ANCHO(-IA)
2     CALL MOVEA(X,Y)
      RETURN
      END
```

PSHIFT.FR

```
      SUBROUTINE XPSHIFT(ISH)
      OVERLAY OPSHIFT
CC  SHIFT PROCESSOR
C - - - P.R.STEVENSON (8/22/78) - - -
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
C  ZERO-LEVEL CONTROL SWITCH
      COMMON /ZER/IZER
C
      INCLUDE 'CSPICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /IOPRM/NS,NSCAN,IDUM(4)
      DIMENSION KEY(32),X(32)
      DIMENSION TIME(32),JPNAM(25),JWNAM(19)
      DATA JPNAM/84,62,65,67,69,83,32,83,72,73,70,
     *84,69,68,32,66,89,32,80,45,84,73,77,69,83/
      DATA JWNAM/87,65,68,65,84,73,32,83,69,67,84,
     *73,79,78,32,80,76,79,84/
C   COMPUTE FIDUCIAL SHIFT MARK TO BE 1/10 WINDOW LENGTH
      FIDUC=TWIND/10
C ROUND DOWN TO NEXT SECOND
      IFID=INT(FIDUC)
C  IF IT'S ZERO,MAKE IT ONE
      IF(IFID.EQ.0)IFID=1
C   IF THE WINDOW IS MORE THAN SPCSE SECONDS DO IT
C   ELSE JUST TAKE 10%
      SPCSE=2.0
       IF(TWIND.GT.SPCSE)FIDUC=IFID
C
C
C  ISH=83 CORRESPONDS TO ALIGN ON S-TIMES
      IF(ISH.EQ.83)GO TO 9
      DO 8 J=1,32
8     TIME(J)=PTIME(J)
      GO TO 11
9     CONTINUE
      DO 10 J=1,32
10    TIME(J)=STIME(J)
11    CONTINUE
C
C    LOAD X WITH TIMES,KEY POINTS TO FORMER ORDER
      DO 1 J=1,NREQ
      KK=IREQ(J)
      X(J)=TIME(KK)
      IF(TIME(KK).EQ.0.)X(J)=10000.
1     CONTINUE
C
C  SORT TIMES, KEY POINTS TO FORMER ORDER
      CALL SORT(X,KEY,NREQ)
CC  ELIMINATE NULL TIMES(IE.TIME=10000.)
      KNT=0
      DO 2 J=1,NREQ
      IF(X(J).GE.10000.)GO TO 3
2     KNT=KNT+1
3     CONTINUE
C  REORDER TRACES ACCORDING TO SORT KEYS
      DO 6 J=1,KNT
      JJ=KEY(J)
      KK=ITEMP(JJ)
      IREQ(J)=KK
6     CONTINUE
      NREQ=KNT
CC
C  SET UP WINDOW TO BE CLOSE TO START OF EARLIEST
C  SHIFT TIME (AVOID NEEDLESS CLIPPING TIME)
      JR=IREQ(1)
C  SAVE OLD WINDOW TIME-ORIGIN
      OTZARR=TZARR
C
      IF(ISH.EQ.83)GOTO 5
C  COME HERE TO SHIFT ON P-TIMES
      TZARR=PTIME(JR)-FIDUC
      IF(TWIND.GT.SPCSE)TZARR=AINT(TZARR)
      IF(TZARR.LE.0.)TZARR=0.
      DO 7 J=1,NREQ
      KK=IREQ(J)
7     SHFT(KK)=TZARR+FIDUC-PTIME(KK)
      GOTO 4
C  HERE TO SHIFT ON S-TIMES
5     TZARR=STIME(JR)-FIDUC
      IF(TWIND.GT.SPCSE)TZARR=AINT(TZARR)
      IF(TZARR.LE.0)TZARR=0.
      DO 12 J=1,NREQ
      KK=IREQ(J)
12    SHFT(KK)=TZARR+FIDUC-STIME(KK)
C
4     CALL ERASE
C
C  RECALCULATE TRACE "DISTANCE" FOR P AND S ALIGNMENT
      IF(ISH.EQ.80.OR.ISH.EQ.83)CALL RSETO
C  CALL SPECIAL ROUTINE FOR WADATI PLOT
C*    IF(ISH.EQ.87)CALL DWAD(X,NREQ)
C
C  REMOVE DC SHIFT IF ZERO FLAG IS ON
      IF(IZER.EQ.1)CALL ZLEV
C  PLOT DATA
      CALL ARRAY
C  LABEL OPTION 2 ,NO START AND STOP TIME LABELS
C  (SINCE MEANINGLESS ON A SHIFTED TRACE)
      CALL LBSTA(2)
C  PUT AMPLITUDE SCALE FACTORS ON PLOT
      CALL LBSCL
C
C   DRAW FIDUCIAL TIC
      LXX=(LX*FIDUC)/TWIND
      IXX=KX+LXX
      IYY=KY+LY
      CALL MOVABS(IXX,IYY)
      CALL DSHABS(IXX,KY,1)
C
CC    PLOT TITLE AT TOP OF WINDOW
      IXCNT=KX+(2*LX)/5
      IYCNT=KY+LY+LY/20
      CALL MOVABS(IXCNT,IYCNT)
      JPNAM(19)=ISH
C   ALIGN ON P-TIMES
      IF(ISH.EQ.80)CALL ANSTR(25,JPNAM)
C   ALIGN ON S-TIMES
      IF(ISH.EQ.83)CALL ANSTR(25,JPNAM)
C   WADATI PLOT
C*    IF(ISH.EQ.87)CALL ANSTR(19,JWNAM)
CC
C  RESTORE OLD WINDOW TIME-ORIGIN
C
      RETURN
      END
```

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

RSETD.FR

```
       SUBROUTINE XRSETD
       OVERLAY ORSETD
       INCLUDE 'CSBLAN'
       PARAMETER MAXX=1000
       COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
       INTEGER A
       COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
      *D(32)
       COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,
      *DTG,SG
       AL=AZ+NREQ*ARNG
       DO 1 J=1,NREQ
       D(J)=AL-(J-0.5)*ARNG
 1     CONTINUE
       RETURN
       END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:31:43 PM

RSTAT.FR

```
      SUBROUTINE XRSTAT
      OVERLAY ORSTAT
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
C     THIS ROUTINE RESTORES OLD STATE
C     INTO NREQ AND IREQ
C
      NREQ=NTEMP
      DO 1 J=1,NTEMP
1     IREQ(J)=ITEMP(J)
C  ALSO RESETS SCREEN WINDOW
      CALL SWND(30,900,10,600)
C   RESTORE OLD WINDOW TIME-ORIGIN
      TZARR=OTZARR
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, OCTOBER 20, 1978 4:25:26 PM

SCALER.FR

```
      SUBROUTINE XSCALER
      OVERLAY OSCALER
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSCPARM'
      COMMON /CPARM/NSTA,ISTA(2,32)
C
      COMMON /TVTUNE/ITV,ITX
      INTEGER X,SEMI,PTR,VEC,HI,BLNK,VCNT,DOT,SP,DSH,EOR
      DIMENSION VEC(32),LIST(96),ITEMP(12),IN(50)
      PARAMETER X=20130K,HI=20071K,LO=20060K,BLNK=20040K
      PARAMETER DOT=20056K,SEMI=20073K,DSH=20055K,EOR=20000K
      PARAMETER LAST=96
C
C  PROMPT USER
      WRITE(ITV,100)
100   FORMAT(' ENTER SCALE FACTOR AND TRACE LIST',/,
     *'EXAMPLE   5.56X 7 5-2 ABCD')
C
C   INPUT CHAR STRING
      READ(ITX,200)IN(1)
200   FORMAT(S96)
      DECODE(IN,201,ERR=500)LIST
201   FORMAT(96R1)
500   CONTINUE
C
C
      IP=0
      LPTR=LAST
      SCL=1.
C  SEARCH FOR X
      DO 1 J=1,LAST
      IF(LIST(J).EQ.EOR.OR.LIST(J).EQ.SEMI)GOTO 600
      IF(LIST(J).NE.X)GOTO 1
      PTR=J+1
      IP=J-1
1     CONTINUE
600   CONTINUE
C  ERROR IF NO PRECEDING NUMBER
      IF(IP.LT.1)GOTO 550
C
C  EXTRACT SCALE FACTOR
      DO 2 K=1,IP
      JK=IP-K+1
C
C  CHECK FOR NUMBER,.,OR -
      IF(LIST(JK).GE.LO.AND.LIST(JK).LE.HI)GOTO 2
      IF(LIST(JK).EQ.DOT.OR.LIST(JK).EQ.DSH)GOTO 2
C  IF COME HERE AND NOT BLANK, THEN ERROR
      IF(LIST(JK).NE.BLNK)GOTO 550
C  ALL DONE, NOW DECODE
      SP=JK+1
      GOTO 3
2     CONTINUE
C
C  COME HERE TO DECODE SCALE FACTOR
C
3     KNT=IP-SP+1
C  MOVE ASCII CHARACNTERS
C
C  DECODE SCALE FACTOR
      CALL INFREE(IN,SCL,96)
C
C
      VCNT=32
      CALL TPARS(LIST,PTR,LPTR,ISTA,VEC,VCNT,SEMI,IER)
C  ASSIGN SCALE FACTOR
      DO 30 K=1,VCNT
      II=VEC(K)
30    S(II)=SCL
      RETURN
550   WRITE(ITV,510)
510   FORMAT(' SCALE SYNTAX ERROR')
      RETURN
      END
```

```
ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978  2:33:36 PM

                               SORT.FR

      SUBROUTINE XSORT(X,KEY,NO)
      OVERLAY OSORT
CC  SORT ARRANGES THE ELEMENTS OF X IN ASCENDING ORDER AND
CC  CONSTRUCTS AN ARRAY  KEY" OF SUBSCRIPTS OF X
CC
CC   ARGUMENTS
CC      X      THE ARRAY TO BE ORDERED, REAL
CC      KEY    THE ARRAY OF SUBSCRIPTS OF X FORMED BY SORT
CC      NO     THE NUMBER OF ELEMENTS IN THE X ARRAY
CC
      DIMENSION X(NO),KEY(NO)
      DO 1 I=1,NO
      KEY(I)=I
1     CONTINUE
      MO=NO
2     IF(MO-15)21,21,23
21    IF(MO-1)9,9,22
22    MO=2*(MO/4)+1
      GO TO 24
23    MO=2*(MO/8)+1
24    KO=NO-MO
      JO=1
25    I=JO
26    IF(X(I)-X(I+MO))28,28,27
27    TEMP=X(I)
2701  X(I)=X(I+MO)
2702  X(I+MO)=TEMP
2703  KEMP=KEY(I)
2704  KEY(I)=KEY(I+MO)
2705  KEY(I+MO)=KEMP
      I=I-MO
      IF(I-1)28,26,26
28    JO=JO+1
      IF(JO-KO)25,25,2
9     RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, OCTOBER 27, 1978 5:06:28 PM

SPLOT.FR

```
      SUBROUTINE SPLOT
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$APARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'C$BPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      COMMON /ZER/IZER
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SHCOM/ISH
      COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
      INCLUDE 'C$GPARM'
C  MODIFIED 5/12/78 FOR 3-PASSES
      PARAMETER LSZ1=13
      PARAMETER MAXPASS=3,MAXVP=3,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INTEGER OLDEVT,PUFLG,PUBLNK,CHAIN
      COMMON /MISC/ZX,ZY,CX,CY
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      INTEGER ZX,ZY,CX,CY
      DIMENSION MSG1(2),MSG2(3),MSG4(4)
      DIMENSION NNN(41),SQ(40)
      DATA IQ/63/
      DATA IY/1HY/
C    DEFAULT SCALE UNITS OPTION IS 1
      IPAUS=0
      SFLAG=1
      CHAIN=0
      MAXCH=32
      IFILE=1
      MUX=10
      PUBLNK=0
      PUFLG=0
      OLDEVT=0
      EVENT=0
      CX=0
      CY=780*ITERM
      ZX=0
      ZY=750*ITERM
C  GO IMMEDIATELY TO EVENT PROCESSOR
      IC=14
      GO TO 39
C
C  COME HERE TO START ANOTHER COMMAND
20    CONTINUE
      CALL COMMAND(IC,JC)
      GO TO (1,2,3,4,5,44,7,8,9,10,31,32,33,39,35,4,20,90,20,20,88,20,
     *20,38,20,20,20,40,42,20,41,43,20,20,2,777),IC
777   CONTINUE
      IF(IPAUS.EQ.1)GOTO 45
      GO TO 20
C*
C*  DEFINE THE WINDOW
1     CONTINUE
      CALL DEPRC(IC,JC)
C*
C*
      GO TO 20
2     CONTINUE
      CALL TRPRC(IC,JC)
C*
      GO TO 20
3     CONTINUE
      IF(JC.EQ.15)GO TO 36
      IF(JC.EQ.9.OR.JC.EQ.12)GO TO 10
C*  PLOT PROCESSOR
C  RESTORE STATE BEFORE PLOTTING
      CALL RSTAT
      CALL PLPRC(JC)
900   CONTINUE
      CALL MOVABS(CX,CY)
      CALL ANMODE
      CALL TSEND
      GO TO 20
4     CONTINUE
C***SHIFT PROCESSOR
      CALL RSTAT
C
C   ALIGN P-PHASES
      ISH=80
C   ALIGN S-PHASES
      IF(JC.EQ.26)ISH=83
      IF(IC.EQ.14)ISH=83
C   WADATI PLOT
      IF(IC.EQ.16)ISH=87
C
      CALL PSHIFT(ISH)
      GO TO 900
5     CONTINUE
C* PICK PROCESSOR
      CALL PICK
      GO TO 900
6     CONTINUE
C*
      GO TO 20
C*  RESET PLOT PARMS
7     CONTINUE
C  IF 'RESU',RESUME REMOTE FILE CONTROL
      IF(JC.EQ.34)GOTO 45
      IF(JC.EQ.27)GOTO 90
      CALL DEFAULT
      IZER=0
      GO TO 20
C*  AMPLITUDE SCALING PROCESSOR
8     CONTINUE
      CALL SCALER
      GO TO 20
C
C  SELECT SCALING OPTION
88    IF(JC.NE.8)GOTO 20
      WRITE(ITV,800)
800   FORMAT(" SELECT INPUT SCALE OPTION",/,
    *"1=DIGITAL COUNTS",/,"2=VOLTS OUT",/,
    *"3=EQUIVALENT DEVELOCORDER MILLIMETERS")
      READ(ITX,234)NNN(1)
234   FORMAT(S81)
      CALL INFREE(NNN,RQ,96)
      SFLAG=RQ
      WRITE(ITV,801)SFLAG
801   FORMAT(" OPTION ",I1," SELECTED")
      GOTO 20
C
C*  CROSS ? AUTO-CORRELATION PROCESSOR
9     CONTINUE
C     CALL SCRSS
      GO TO 20
C*  PLOT CROSS & AUTO-CORRELOGRAMS
10    CONTINUE
C     CALL PCRSS
C     GO TO 900
35    CONTINUE
C*  SPECTRA PROCESSOR
C      CALL SFAST
      GO TO 20
36    CONTINUE
C     CALL PFAST
C     GO TO 900
C  PUNCH PHASE CARDS
31    CONTINUE
      CALL FAZPU(1)
      GO TO 20
33    CONTINUE
C  USER ROUTINE
C  ROUTINE SETS KC IN ORDER TO CONTROL ITS RETURN
C     CALL USER(JC,KC)
C     GO TO (20,900)KC
C
C   MENU
32    CONTINUE
      CALL MENU(1)
      GO TO 20
40    CONTINUE
      CALL MENU(2)
      GOTO 20
38    CONTINUE
      IZER=1
      GO TO 20
C   COME HERE FOR 'UN' COMMANDS
42    IF(JC.EQ.24)IZER=0
      IF(JC.EQ.32)ITX=5
      IF(JC.EQ.32)CLOSE 13
      IF(JC.EQ.32)CHAIN=0
      GOTO 20
C  COME HERE FOR 'PA'
44    IF(JC.NE.13)GOTO 39
      ITX=5
      WRITE(ITV,544)
544   FORMAT('REMOTE PAUSE')
      IPAUS=1
      GOTO 20
C  COME HERE TO RESUME REMOTE CONTROL
45    ITX=13
      IPAUS=0
      GOTO 20
C  HARDCOPY HERE
43    IF(JC.NE.33)GOTO 20
      CALL HDCOPY
      CALL TSEND
C  10 SEC DELAY FOR HARDCOPY UNIT
      CALL FDELY(100)
      GOTO 20
C   COME HERE FOR COMMANDS FROM REMOTE FILE
41    WRITE(ITV,440)
440   FORMAT(" INPUT REMOTE FILE NAME(10 CHARS. MAX)")
      READ(ITX,441)NNN(1)
441   FORMAT(S10)
      IF(CHAIN.EQ.1)CLOSE 13
C   NNN IS ARRAY CONTAINING ASCII NAME OF FILE
      OPEN 13,NNN
      CHAIN=1
      ITX=13
      GOTO 20
C
39    CONTINUE
C  EVENT/PASS PROCESSOR
C  CHECK ZERO-LEVEL FLAG
      IF(IC.EQ.14)IZER=0
      IF(JC.EQ.24)IZER=1
      CALL EVPRC(IC,JC)
      GO TO 20
90    CONTINUE
C  OUTPUT ZZ CARD AT END OF PHASE LIST
      CALL FAZPU(2)
      WRITE(ITV,990)
990   FORMAT(10H EXIT ISDS)
      RETURN
      END
```

SWND.FR

```
      SUBROUTINE SWND(NX,NLX,NY,NLY)
CC  RESET PHYSICAL SCREEN BOUNDS
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      KX=NX*ITERM
      KY=NY*ITERM
      LX=NLX*ITERM
      LY=NLY*ITERM
      RETURN
      END

```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, OCTOBER 20, 1978 3:06:31 PM

TPARS.FR

```
          SUBROUTINE XTPARS(LIST,PTR,LPTR,SLIST,VEC,VCNT,EOL,IER)
          OVERLAY OTPARS
      C
      C  P.R.STEVENSON 9-20-78
      C
      C  LIST IS AN ASCII STRING CONTAINING ONLY 1 OR 2 CHAR.
      C  NUMBERS SEPERATED BY EITHER BLANKS OR -,
      C  WHERE N-M MEANS THE RANGE OF NUMBERRS FROM N
      C  TO M, AND A SINGLE NUMBER MEANS JUST THAT
      C  NUMBER
      C
      C  PTR IS INDEX WITHIN LIST WHERE SCAN IS TO BEGIN
      C  THE LIST ENDS WITH THE EOL CHARACHTER
      C
      C  LPTR IS INDEX WITHIN LIST CHERE SCAN IS TO END IF
      C  EOL CHARACHTER IS NOT ENCOUNTERED
      C
      C  VEC IS THE VECTOR CONTAINING THE LIST OF
      C  NUMBERS GENERATED FROM LIST
      C
      C  VCNT IS THE COUNT OF ELEMENTS IN VEC
      C
      C  EOL IS A VARIBLE CONTAINING ASCII CODE FOR THE
      C  END OF THE CHARACHTER STRING
      C
          PARAMETER LASTK=96
          PARAMETER DSH=20055K,BLNK=20040K,OFF=20060K
          PARAMETER COLON=20072K,COMMA=20054K,NINE=20071K
      C
          DIMENSION NAM(2)
          INTEGER SMAX,SLIST
          PARAMETER SMAX=32,MASK=177K
          DIMENSION SLIST(2,SMAX)
          LOGICAL NEW,PREV
          INTEGER DSH,BLNK,OFF,COLON,COMMA,P1,P2
          INTEGER PTR,VEC,VCNT,EOL
          DIMENSION P1(50),P2(50),LIST(1),VEC(1)
      C
      C  PARSE LINE
      C
          IER=0
          PREV=.FALSE.
          NEW=.FALSE.
          KNT=0
          INDEX=PTR-1
          P1(1)=0
          P2(1)=-1
      C
      C
      C   GET A CHARACHTER
      C
      1   INDEX=INDEX+1
      C
      C  IS IT THE END OF THE LINE?
          IF(LIST(INDEX-1).EQ.EOL)GOTO 9
          IF(INDEX.EQ.LPTR)GOTO 9
```

```
            IF(INDEX.GT.LASTX)GOTO 98
            PREV=NEW
      C
      C  IS IT BLANK(OR COMMA OR EOL) OR NON-BLANK?
      C
            NEW=.TRUE.
            IF(LIST(INDEX).EQ.BLNK)NEW=.FALSE.
            IF(LIST(INDEX).EQ.COMMA.OR.LIST(INDEX).EQ.EOL)NEW=.FALSE.
            IF(INDEX.EQ.LPTR)NEW=.FALSE.
      C
      C  IF PREVIOUS IS FALSE AND NEW IS TRUE, THEN
      C  START NEW STRING
            IF((.NOT.PREV).AND.NEW)GOTO 3
      C
      C  IF PREVIOUS IS TRUE AND NEW IS FALSE, THEN
      C  END OF STRING
            IF(PREV.AND.(.NOT.NEW))GOTO 2
      C
      C  ELSE JUST GET ANOTHER CHARACHTER
            GOTO 1
      C
      C
      C  COME HERE FOR END OF STRING
      C
      2      P2(KNT)=INDEX-1
            GOTO 1
      C
      C   COME HERE FOR START OF STRING
      C
      3     KNT=KNT+1
            P1(KNT)=INDEX
            GOTO 1
      C
      C
      C- - - - - - - - - - - - - - - - - - - - -
      C
      C   END OF LINE
      C
      9     CONTINUE
      C
      C   NOW EXPAND AND EVALUATE EXPRESSIONS
      C
            LAST=KNT
            JV=0
            DO 10 J=1,LAST
      C  COMPUTE STRING LENGTH
            NCHAR=P2(J)-P1(J)+1
      C  VALIDITY CHECK
      C
            IF(NCHAR.LT.1.OR.NCHAR.GT.5)GOTO 98
      C
      C  IF LENGTH GREATER THAN 2, IT IS A
      C  RANGE EXPRESSION
            N1=P1(J)
            N2=P2(J)
            IF(NCHAR.GT.2)GOTO 16
      C  ELSE IT IS A SINGLE NUMBER
            JV=JV+1
      C  COMPUTE NUMBER FROM ASCII CHARACHTER
            DO 15 K=N1,N2
```

```
C  CHECK FOR ILLEGAL NUMERALS
        IF(LIST(K).LT.OFF.OR.LIST(K).GT.NINE)GOTO 98
        VEC(JV)=LIST(N2)-OFF
        IF(NCHAR.EQ.2)VEC(JV)=VEC(JV)+(LIST(N1)-OFF)*10
15      CONTINUE
        GOTO 10
C
C  COME HERE TO EVALUATE RANGE EXPRESSIONS
16       N3=0
        DO 17 K=N1,N2
        IF(LIST(K).EQ.DSH)N3=K
17      CONTINUE
C  IF NOT DASH, NOT A RANGE EXPRESSION
        IF(N3.EQ.0)GOTO 20
C  CHECK FOR SYNTAX
        IF((K-N1).GT.5.OR.(K-N1).LT.1)GOTO 98
        N4=N3-1
        N5=N3+1
C  CHECK FOR ILLEGAL NUMERALS IN RANGE EXPRESSION
        DO 18 K=N1,N2
        IF(K.EQ.N3)GOTO 18
        IF(LIST(K).LT.OFF.OR.LIST(K).GT.NINE)GOTO 98
18      CONTINUE
        JSTART=LIST(N4)-OFF
        IF(N1.LT.N4)JSTART=JSTART+(LIST(N1)-OFF)*10
        JLAST=LIST(N2)-OFF
        IF(N5.LT.N2)JLAST=JLAST+(LIST(N5)-OFF)*10
C
C  IS IT 'UP TO' OR 'DOWN TO' ?
C
        IUD=1
        IF(JLAST.LT.JSTART)IUD=-1
        IRNG=IABS(JLAST-JSTART)+1
C
C  EXPAND RANGE INTO LIST OF NUMBERS
        DO 19 K=1,IRNG
        JV=JV+1
19      VEC(JV)=JSTART+IUD*(K-1)
        GOTO 10
20       CONTINUE
C  PACK UP THE CHARACNTER STRING FOR COMPARE
        DO 23 K=1,2
        K1=N1+(K-1)*2
        ITEMP=IAND(MASK,LIST(K1))
        ITEMP=ITEMP*256
        JTEMP=IAND(MASK,LIST(K1+1))
        NAM(K)=IOR(ITEMP,JTEMP)
23      CONTINUE
         NZ=N1+3
C  SEARCH SLIST FOR 4-CHAR NAME
        DO 21 K=1,SMAX
        KK=K
        IF(NAM(1).EQ.SLIST(1,K).AND.NAM(2).EQ.SLIST(2,K))GOTO 22
21      CONTINUE
        GOTO 98
C  COME HERE TO ADD SEQ. NUMBER TO STATION TO VEC
22      JV=JV+1
        VEC(JV)=KK
C
10      CONTINUE
```

```
      VCNT=JV
      RETURN
   90 IER=1
      VCNT=JV
      RETURN
      END
```

TRPRC.FR

```
      SUBROUTINE XTRPRC(IC,JC)
      OVERLAY OTRPRC
C  ROUTINE TO SELECT,MANIPULATE AND LIST TRACES
C
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'C$CPARM'
      COMMON /CPARM/NSTA,ISTA(2,32)
C
      COMMON /IOPRM/NS,NSCAN,NREN,NUX,IFILE,IUNIT
      COMMON /TVTUNE/ITV,ITX
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      LOGICAL EOC
      INTEGER DUM(2,32),VEC(32),VCNT,IN(50),LIST(96)
      INTEGER SEMI,COLON,BLANK,EX,SE,DE,AP,PTR,CC
      PARAMETER LAST=96,SEMI=20073K,COLON=20072K,
     *BLANK=20040K,EX=20105K,SE=20123K,DE=20104K,
     *LI=20114K,AP=20101K,MA=20116K,MAXCH=32
C
C  TRACE MANIPULATION COMMANDS MUST BE OF THE FORM:
C        <COMMAND>: I J-K ABCD  ;<...>: L M ;  . . . ETC.
C
C   WHERE ';' SEPERATES MORE THAN ONE COMMAND ON THE SAME LINE.
C   EITHER SINGLE TRACES MAY BE SPECIFIED, OR RANGES(BOTH
C   INCREASING OR DECREASING) OF TRACES, OR THE TRACES MAY
C   BE REFERENCED BY STATION NAME(ABCD).
C
C   PROMPT
C
9     WRITE(ITV,101)
101   FORMAT(' <SE,AP,DE,LI,MA,EX>: I J-K ABCD ;ETC...')
C
C   READ LINE
C
      ISEMI=0
      LPTR=0
      EOC=.FALSE.
      K1=1
      READ(ITX,200)IN(1)
200   FORMAT(S96)
      DECODE(IN,201,ERR=88)LIST
201   FORMAT(96R1)
88    CONTINUE
C
C    EXTRACT COMMAND
C
14    JP=0
C
C  SEARCH FOR ':'
C
      DO 70 K=K1,LAST
      IF(LIST(K).EQ.COLON)GOTO 71
      IF(LIST(K).EQ.BLANK)GOTO 70
      IF(LIST(K).EQ.SEMI)GOTO 72
      IF(JP.EQ.1)GOTO 70
C  FIRST TIME HERE, GET THE COMMAND!
      CC=LIST(K)
      JP=1
      GOTO 70
C  GOT THE COLON
71    PTR=K+1
70    IP=K
C  IF COME HERE, NO ';'
      LPTR=LAST
      EOC=.FALSE.
      GOTO 73
C
C
C  FOUND A SEMICOLON
72    ISEMI=K
C  RESET COMMAND SCAN POINTER
      K1=K+1
      EOC=.TRUE.
C  SET END OF ARG FIELD
      LPTR=SEMI
C
73    IER=0
C  CALL SCANNER/PARSER FOR TRACE LIST
      IF(CC.EQ.SE.OR.CC.EQ.AP.OR.CC.EQ.DE)CALL TPARS(LIST,
     *PTR,LPTR,ISTA,VEC,VCNT,SEMI,IER)
C
      IF(IER.NE.0)GOTO 8
C  WHAT COMMAND IS CC?
C
C  RETURN IF EXIT
      IF(CC.EQ.EX)RETURN
C  SELECT?
      IF(CC.EQ.SE)GOTO 10
C  APPEND?
      IF(CC.EQ.AP)GOTO 20
C  LIST?
      IF(CC.EQ.LI)GOTO 30
C LIST BY STATION NAME?
      IF(CC.EQ.NA)GOTO 35
C  DELETE?
      IF(CC.EQ.DE)GOTO 40
C  TRY AGAIN!
      GOTO 9
C
C
C  LOAD TRACE LIST
10    J1=1
      JN=VCNT
      IF(JN.GT.NUX)WRITE(ITV,150)NUX
      IF(JN.GT.NUX)GOTO 9
      GOTO 15
C  APPEND TO EXISTING TRACE LIST
20    J1=NREQ+1
      JN=NREQ+VCNT
      IF(JN.GT.MAXCN)WRITE(ITV,150)MAXCN
      IF(JN.GT.MAXCN)GOTO 9
C  COMMON PATH
15    JJ=0
150   FORMAT(' MORE THAN ',I3,' TRACES REQUESTED')
```

```
      IF(JN.GT.32)JN=32
      DO 16 J=J1,JN
      JJ=JJ+1
16    IREQ(J)=VEC(JJ)
      NREQ=JN
      GOTO 19
8     WRITE(ITV,800)
800   FORMAT('*** SYNTAX ERROR *** ')
      GOTO 9
C
C   WRITE TRACE LIST
30    WRITE(ITV,102) (IREQ(J),J=1,NREQ)
102   FORMAT(' TRACES:',32I3)
      GOTO 19
C
C   WRITE TRACE LIST BY STATION NAME
35    CONTINUE
      DO 36  J=1,NREQ
      JJ=IREQ(J)
      DUM(1,J)=ISTA(1,JJ)
36    DUM(2,J)=ISTA(2,JJ)
C
      WRITE(ITV,103) (DUM(1,J),DUM(2,J),J=1,NREQ)
103   FORMAT(' TRACES:',16(2A2,1X))
      GOTO 19
C
C   DELETE TRACES
40    CONTINUE
C   LOOP ON NUMBER OF ITEMS TO BE DELETED
      DO 3 I=1,VCNT
      ITOK=VEC(I)
      KILL=-1
C  LOOK FOR FIRST OCCURRENCE OF ITEM TO BE DELETED
C
      DO 31 J=1,NREQ
      IF(ITOK.EQ.IREQ(J))KILL=J
      IF(ITOK.EQ.IREQ(J))GOTO 34
31    CONTINUE
      GOTO 3
34    CONTINUE
C  SQUEEZE ITEM FROM LIST
C
      IF(NREQ.EQ.1)GOTO 8
      NREQ=NREQ-1
      DO 33 J=KILL,NREQ
33    IREQ(J)=IREQ(J+1)
3     CONTINUE
      GOTO 19
C  BACKUP TRACES
19    NTEMP=NREQ
      DO 190 J=1,NREQ
190   ITEMP(J)=IREQ(J)
C
C    DO WE HAVE A ';'?
C  IF YES, THEN GET ANOTHER COMMAND
      IF(EOC)GOTO 14
C  ELSE GO GIVE ANOTHER PROMPT
      GOTO 9
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, JULY 17, 1978 2:36:44 PM

ZLEV.FR

```
      SUBROUTINE XZLEV
      OVERLAY OZLEV
      INTEGER START
      DIMENSION AV(32)
      PARAMETER NZ=100
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
C
C  COMPUTE START
      START=TZARR/DTG+1
C  CLEAR ACCUMULATORS
      DO 6 I=1,NREQ
6     AV(I)=0.
C  COMPUTE SEGMENT LENGTH
      IB=MAXX/NREQ+1
      NBUF=NZ/IB+1
      MSCAN=NZ/NBUF
      CALL FRAMD(MSCAN,IREQ,NREQ,START,IERR)
      IF(IERR.NE.0)TYPE 'ZERO-FRAMD ERROR:IERR=',IERR
      DO 1 K=1,NBUF
      CALL GETFM(A,NSCAN,IEND)
      IF(IEND.EQ.1.AND.IEND.EQ.0)TYPE 'EOF IN ZLEV'
      IF(IEND.EQ.1.AND.IEND.EQ.0)RETURN
      DO 2 I=1,NREQ
      DO 3 J=1,NSCAN
      JJ=MSCAN*(I-1)+J
      R=A(JJ)
3     AV(I)=R+AV(I)
   2  CONTINUE
1     CONTINUE
C  LOAD Z ARRAY
      DO 5 I=1,NREQ
      II=IREQ(I)
      Z(II)=AV(I)/NZ
5     CONTINUE
      RETURN
      END
```

APPENDIX O: LISTINGS OF SOME SUPPORT SOFTWARE

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

                                    DIRC1

      SUBROUTINE DIRC1
C
C  INTERPROCESS COMMUNICATIONS DIRECTORY INITIALIZER
C
C  P.A. MARSHALL
C  SEPTEMBER 1976
C----------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C
C *** DIRECTORY RECORD SIZE
      PARAMETER RECSIZ = 160
C
C *** RECORD 0:
C
      PARAMETER ENX = 1            ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1        ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1        ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1        ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1        ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1        ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+5        ; MASTER POINTER LIST
      PARAMETER EPL = MPL+95       ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1            ; EVENT NO.
      PARAMETER ENA = ENO+1        ; EVENT NAME
      PARAMETER EIB = ENA+20       ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1        ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1        ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2        ; DURATION (SECS)
      PARAMETER EYR = DRA+2        ; YEAR
      PARAMETER EMO = EYR+1        ; MONTH
      PARAMETER EDA = EMO+1        ; DAY
      PARAMETER EHR = EDA+1        ; HOUR
      PARAMETER EMI = EHR+1        ; MINUTE
      PARAMETER ESE = EMI+1        ; SECONDS
      PARAMETER EP1 = ESE+2        ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1        ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1        ; EVENT, PASS 3
C
      PARAMETER ETM = EYR          ; EVENT TIME
      PARAMETER EPP = EP1-1        ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0            ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1        ; PASS NO.
      PARAMETER PIB = PNO+1        ; INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1        ; NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1        ; FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1        ; LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1        ; POINTER LIST
      PARAMETER PSW = PLS+32       ; TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1        ; FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1        ; A/D ERROR STATUS
C****
```

```
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1              ; EVENT DIRECTORY
      PARAMETER IML = 2              ; MASTERLIST DIRECTORY (CARD IMAGES)
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
C
C
C --- OPEN DIRECTORY FILES
      OPEN IED,'DIRECTORY',LEN=2*RECSIZ
      OPEN IML,'MASTERLIST',LEN=80
C
C --- READ DIRECTORY RECORD ZERO
      READ(IED,REC=1) REC
C
C --- INITIALIZE DATABASE PARAMETERS
      NEVENT = REC(ENX)-1
      OFFSET = REC(DOF)
C
      RETURN
      END
```

```
        SUBROUTINE XENAME(EVTNAME,EVTNUM)
      OVERLAY OENAME
C
C   EDITED 10/3/77 BY P.R.STEVENSON TO SAVE STACK SPACE
C
C *********************************************************************
C
C     HERE IS A ROUTINE TO READ AN EVENT NAME FROM A FILE CALLED "EVDIR"
C     WHICH RESIDES ON REMOVABLE DISK DP0. THIS IS NOT A STAND-ALONE
C     ROUTINE. SUBROUTINE ENAME SHOULD BE USED IN CONJUNCTION WITH 2
C     OTHER SUBROUTINES, ONE OF WHICH MUST BE CALLED PRIOR TO CALLING
C     SUBROUTINE ENAME. SUBROUTINE ENAME READS  RECORDS FROM
C     THE EVDIR FILE, BUT ASSUMES THIS FILE HAS BEEN PREVIOUSLY OPENED
C     WITH A CALL TO SUBROUTINE "DIRC1".
C
C     NOTE: IN ORDER TO USE SUBROUTINE DIRC1, THE USER MUST HAVE CRE-
C           ATED A LINK TO FILE DP0:EVDIR USING THE LINK NAME
C           "DIRECTORY".
C
C     FORMAT OF CALL:  CALL ENAME(NAME,K)
C
C     WHERE K IS (UPON CALL) THE INTEGER NUMBER OF THE EVENT FOR WHICH
C     THE NAME (IN ASCII) IS REQUESTED,
C
C     AND NAME IS A 20 WORD INTEGER ARRAY WHICH WILL BE STUFFED WITH
C     THE 40 CHARACTER EVENT NAME, CORRESPONDING TO THE EVENT NUMBER.
C
C     IF THE EVENT NUMBER PASSED IN ARGUMENT "K" IS LESS THAN 1 OR
C     GREATER THAN THE NUMBER OF EVENTS CONTAINED WITHIN THE EVDIR
C     FILE, SUBROUTINE ENAME RETURNS TO THE CALLING ROUTINE WITH
C     ASTERISKS IN THE 20 WORD "NAME" ARRAY.
C
C     IN MOST INSTANCES THE USER WILL ALSO BE USING THE DATA FILE
C     ASSOCIATED WITH A PARTICULAR EVENT. THIS REQUIRES A CALL
C     TO SUBROUTINE "MCAM1", WHICH OPENS FILE "EVDAT" (ALSO ON DP0).
C
C     NOTE: IN NON-TASKING ENVIRONMENT, USER MUST LINK DP0:EVDAT
C           USING LINK NAME "WHOLE1".
C
C     THE RECOMMENDED CALLING SEQUENCE IS:
C
C          CALL MCAM1
C          CALL DIRC1
C              .
C              .
C              .
C              .
C              .
C          CALL ENAME(NAME,K)
C              .
C              .
C
C     NOTE: SUBROUTINE ENAME MAY BE CALLED SEVERAL TIMES DURING EXECU-
C           TION OF A PROGRAM (OR ANOTHER SUBROUTINE), HOWEVER, SUB-
C           ROUTINE MCAM1 AND DIRC1 SHOULD BE CALLED ONCE AND ONLY
C           ONCE DURING THE EXECUTION OF A PROGRAM.
C
C *********************************************************************
C
```

```
CC ** MODIFIED FOR 3 PASSES BY P.STEVENSON (5-12-78)
   PARAMETER NPASS=3
    PARAMETER REKSIZE = 160
    PARAMETER IED = 1
    PARAMETER ENO = 1
    PARAMETER ENA = ENO+1
    IMPLICIT INTEGER (A-Z)
    DIMENSION EVTNAME(1)
    COMMON /IOBDA/ NCHAN,SEGSZ,REMDR,LTBUF,FIRST
   COMMON /EDIRC/NEVENT,PASSN(NPASS),REC(REKSIZE)
C
C --- INITIALIZE THE EVENT NAME ARRAY. PUT ASTERISKS IN ALL
C      LOCATIONS.
C
    DO 5 K=1,20
  5 EVTNAME(K) = 2H**
C
C --- FIND OUT HOW MANY EVENTS ARE ALREADY WITHIN THE DIRECTORY
C
    MAXEVTS = NEVENT
C
C --- CHECK THE PASSED PARAMETER "EVTNUM" TO SEE IF IT IS WITHIN
C      REASONABLE BOUNDS. IF NOT, RETURN.
C
    IF (EVTNUM .LT. 1 .OR. EVTNUM .GT. MAXEVTS) GO TO 20
C
C
C --- READ THE RECORD CORRESPONDING TO THE PARTICULAR EVENT NUMBER
C
    READ(IED,REC=EVTNUM+1) REC
C
C
C --- NOW FILL THE EVENT NAME ARRAY WITH THE 40 ASCII CHARACTERS
C      STORED IN FILE "EVDIR".
C
    DO 10 K=1,20
 10 EVTNAME(K) = REC( ENO + K )
C
 20 RETURN
C
    END
```

```
      SUBROUTINE XRQEVT(LIST, PARM1)
      OVERLAY ORQEVT
C
C  EVENT REQUEST PROCESSOR
C
C  P.A. MARSHALL
C  SEPTEMBER 1976
C------------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C
C *** DIRECTORY RECORD SIZE
      PARAMETER RECSIZ = 160
C
C *** RECORD 0:
C
      PARAMETER ENX = 1             ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1         ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1         ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1         ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1         ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1         ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+5         ; MASTER POINTER LIST
      PARAMETER EPL = MPL+95        ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1             ; EVENT NO.
      PARAMETER ENA = ENO+1         ; EVENT NAME
      PARAMETER EIB = ENA+20        ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1         ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1         ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2         ; DURATION (SECS)
      PARAMETER EYR = DRA+2         ; YEAR
      PARAMETER EMO = EYR+1         ; MONTH
      PARAMETER EDA = EMO+1         ; DAY
      PARAMETER EHR = EDA+1         ; HOUR
      PARAMETER EMI = EHR+1         ; MINUTE
      PARAMETER ESE = EMI+1         ; SECONDS
      PARAMETER EP1 = ESE+2         ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1         ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1         ; EVENT, PASS 3
C
      PARAMETER ETM = EYR           ; EVENT TIME
      PARAMETER EPP = EP1-1         ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0             ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1         ; PASS NO.
      PARAMETER PIB = PNO+1         ; INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1         ; NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1         ; FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1         ; LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1         ; POINTER LIST
      PARAMETER PSW = PLS+32        ; TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1         ; FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1         ; A/D ERROR STATUS
```

```
C****
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1             ! EVENT DIRECTORY
      PARAMETER IML = 2             ! MASTERLIST DIRECTORY (CARD IMAGES)
C****
C**** REQUEST LIST OFFSETS
C
      PARAMETER LEN = 1             ! EVENT NO.
      PARAMETER LET = LEN+1         ! EVENT TIME
      PARAMETER LED = LET+7         ! DURATION
      PARAMETER LDT = LED+2         ! DELTA-T
      PARAMETER LNP = LDT+2         ! NUMBER OF PASSES
      PARAMETER LM1 = LNP+1         ! MUX, PASS 1
      PARAMETER LM2 = LM1+1         ! MUX, PASS 2
      PARAMETER LM3 = LM2+1         ! MUX, PASS 3
C
      PARAMETER LMP = LM1-1         ! MUX FINDER
      PARAMETER LSZ = LM3-LEN+1     ! LIST SIZE
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      DIMENSION LIST(LSZ)
C
C --- CLEAR LIST
      DO 1 J=1,LSZ
    1 LIST(J) = 0
C
C --- SET EVENT, CHECK FOR VALIDITY
      EVENT = PARM1
      IF (EVENT .LE. 0) RETURN
C
C --- READ IN DIRECTORY RECORD, CHECK VALIDITY
      READ(IED,REC=EVENT+1) REC
      IF (REC(ENO) .EQ. 0) RETURN
C
C --- STUFF PARAMETERS FROM DIRECTORY RECORD INTO RQEQUEST LIST
      LIST(LEN) = REC(ENO)
      LIST(LET) = REC(EYR)
      LIST(LET+1) = REC(EMO)
      LIST(LET+2) = REC(EDA)
      LIST(LET+3) = REC(EHR)
      LIST(LET+4) = REC(EMI)
      LIST(LET+5) = REC(ESE)
      LIST(LET+6) = REC(ESE+1)
      LIST(LED) = REC(DRA)
      LIST(LED+1) = REC(DRA+1)
      LIST(LDT) = REC(DLT)
      LIST(LDT+1) = REC(DLT+1)
C
C --- FIND WHICH PASSES EXIST AND INTERPRET AS A PASS COUNT
      K = 0
      DO 2 J=1,3
C --- CLEAR PASS MEMORY
      PASSN(J) = 0
C --- POINT TO PASS, SEE IF IT EXISTS
      PNT = REC(EPP+J)
      NCHAN = REC(PNT+PLC)-REC(PNT+PFC)+1
      IF (NCHAN .EQ. 1 .AND. REC(PNT+PFC) .EQ. 0) GO TO 2
C
C --- HAVE PASS. COUNT PASS, SAVE PASS NO., STUFF MUX NO.
      K = K+1
      PASSN(K) = J
      LIST(LMP+K) = NCHAN
    2 CONTINUE
C
C --- STUFF PASS COUNT
      LIST(LNP) = K
      RETURN
      END
      SUBROUTINE XRQPAZ(LIST, PARM1, PARM2)
      OVERLAY ORQPAZ
C
C  EVENT/PASS REQUEST PROCESSOR
C
C  P.A. MARSHALL
C  SEPTEMBER 1976
C---------------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C
C *** DIRECTORY RECORD SIZE
      PARAMETER RECSIZ = 160
C
C *** RECORD 0:
C
      PARAMETER ENX = 1             ! NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1         ! NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1         ! DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1         ! FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1         ! NEXT BLOCK NO.
      PARAMETER CTE = DNX+1         ! CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+5         ! MASTER POINTER LIST
      PARAMETER EPL = MPL+95        ! END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1             ! EVENT NO.
      PARAMETER ENA = ENO+1         ! EVENT NAME
      PARAMETER EIB = ENA+20        ! INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1         ! TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1         ! DELTA-T (SECS)
      PARAMETER DRA = DLT+2         ! DURATION (SECS)
      PARAMETER EYR = DRA+2         ! YEAR
      PARAMETER EMO = EYR+1         ! MONTH
      PARAMETER EDA = EMO+1         ! DAY
      PARAMETER EHR = EDA+1         ! HOUR
      PARAMETER EMI = EHR+1         ! MINUTE
      PARAMETER ESE = EMI+1         ! SECONDS
      PARAMETER EP1 = ESE+2         ! EVENT, PASS 1
      PARAMETER EP2 = EP1+1         ! EVENT, PASS 2
      PARAMETER EP3 = EP2+1         ! EVENT, PASS 3
C
      PARAMETER ETM = EYR           ! EVENT TIME
      PARAMETER EPP = EP1-1         ! EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0             ! ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1         ! PASS NO.
      PARAMETER PIB = PNO+1         ! INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1         ! NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1         ! FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1         ! LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1         ! POINTER LIST
      PARAMETER PSW = PLS+32        ! TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1         ! FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1         ! A/D ERROR STATUS
```

```
C****
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1              ; EVENT DIRECTORY
      PARAMETER IML = 2              ; MASTERLIST DIRECTORY (CARD IMAGES)
C****
C**** REQUEST LIST PARAMETERS
      PARAMETER NSTAR = 2            ; WIDTH OF LIST ENTRY (IN WORDS)
      PARAMETER LSZ = 32*NSTAR       ; LIST SIZE
      PARAMETER LST = 1              ; START OF LIST
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /EDIRC/ NEVENT, PASSN(3), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
      COMMON /IOBDA/ NCHAN, IDUM3(4)
      DIMENSION LIST(LSZ), STAREC(40)
C
C --- CLEAR LIST
      DO 1 J=1,LSZ
    1 LIST(J) = 0
C
C --- CHECK EVENT AND PASS FOR VALIDITY
      IF (PARM1 .NE. EVENT) RETURN
      PASS = PARM2
      IF (PASS .LT. 1 .OR. PASS .GT. 3) RETURN
      IF (PASSN(PASS) .EQ. 0) RETURN
C --- POINT TO PASS
      PNT = REC(EPP+PASSN(PASS))
C
C --- SET "MCAM" PARAMETERS
      EPISEC = REC(PNT+PIB)-OFFSET
      LSEC = EPISEC+REC(PNT+PNB)-1
      FCN = REC(PNT+PFC)
      LCN = REC(PNT+PLC)
      NCHAN = LCN-FCN+1
C
C --- SET POINTER LIST, FETCH STATION LIST DATA
      M = PNT+PLS-1
      K = LST
      DO 2 J=FCN,LCN
      STAREC(1) = 0
C --- GET STATION RECORD FROM MASTERLIST
      MP = REC(M+J)
      IF (MP .LE. 0) GO TO 21
      READ(IML,REC=MP) STAREC
   21 CONTINUE
C  -*-*  MOD (4/22/77) BY P.R.S.; IJK FORMERLY 2
C  .IJK=1 FOR FORMAT 2, SET IJK=2 FOR FORMAT 1
      IJK=1
C --- OR SHOW AS 'IRIG' CHANNEL
      IF (MP .EQ. -1) CALL FBLM(STAREC(IJK), 'IRIG', 2)
C --- OR 'WWVB' CHANNEL
      IF (MP .EQ. -2) CALL FBLM(STAREC(IJK), 'WWVB', 2)
C --- OR CLEAR RECORD IF "UNDEFINED" CHANNEL
      IF (MP .EQ. 0) CALL FBLM(STAREC(2), STAREC(1), 39)
C
C --- MOVE DATA INTO REQUEST LIST
      CALL FBLM(LIST(K), STAREC(IJK), 2)
    2 K = K+NSTAR
C
      RETURN
      END
      SUBROUTINE MCAM1
C
C  M C A M  INITIALIZER MODULE
C
C  P.A. MARSHALL
C  MARCH 1976
C-----------------------------------------------------------------------
C
C --- "MULTIPLEXED CHANNEL ACCESS METHOD" (MCAM) IS AN I/O SUBSYSTEM
C  PROVIDING INPUT IN SEGMENTED SERIAL FORM OF MULTIPLEXED CHANNEL
C  DATA STORED ON DISK.
C
C --- THIS ROUTINE IS THE INITIALIZER FOR THE SUITE OF MCAM MODULES
C  AND MUST BE CALLED ONCE DURING THE EXECUTION OF A PROGRAM BEFORE
C  ANY OF THE OTHER MCAM MODULES ARE USED.
C
C
C --- USE
C
C        CALL MCAM1
C
C  THE CALL IS MADE BEFORE ANY OTHER MCAM MODULES ARE USED.  ONLY ONE
C  SUCH CALL SHOULD BE MADE DURING THE EXECUTION OF A PROGRAM.
C
C  MCAM MODULES:
C
C     MCAM1  - "MCAM" INITIALIZER
C     SEGMO  - SEGMENT READ-IN
C     SEGIN  - SEGMENT BUFFER QUEUING
C     DISKI  - SEGMENT INPUT TASK
C
C  USER INTERFACE ROUTINES:
C
C     FRAMD  - INPUT FRAME DEFINITION
C     GETFM  - DATA FRAME INPUT
C
C  OTHER SUBPROGRAMS REQUIRED:
C
C     DEMUX  - CHANNEL DEMULTIPLEXING
C     BLM    - BLOCK MOVE, DATA IN CORE
C
C-----------------------------------------------------------------------
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072          ;NO. OF WORDS PER TRACK
      PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256           ;NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2          ;SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5          ;BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32         ;MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2           ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
```

```
      INCLUDE 'MCAMSB'
C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
      PARAMETER BUFAD = 0           ;BUFFER ADDRESS (INDEX)
      PARAMETER BUFLEN = 1          ;BUFFER LENGTH
      PARAMETER BUFLAG = 2          ;BUFFER STATUS FLAG
      PARAMETER SECNO = 3           ;RELATIVE SECTOR NO.
      PARAMETER BUFPNT = 4          ;POINTER TO NEXT BUFFER
C****
      INCLUDE 'MCAMSI'
C**** PARAMETERS - MCAM INITIALIZER
      PARAMETER IDAFNO = 0         ;FORTRAN FILE NO. FOR DATABASE FILE
      PARAMETER EVNDA = 1          ;EVENT FOR 'DISKI' TASK
      PARAMETER SECEND = 479     ;LAST SECTOR NO. IN DATABASE FILE
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
      COMMON /IOBDA/ NCHAN, SEGSZ, MEMDR, LTBUF, FIRST
      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP), BUFFER(BUFFRP)
C
C
C --- INITIALIZE BUFFER CONTROL BLOCKS
C
      JBUFAD = MAXCHN
      SEGSZ = NWS*SEGMNT
C
C --- LOOP ONCE FOR EACH BLOCK
      BUFP = -BUFCBL+1
      DO 1 J=1,NBUFS
      BUFP = BUFP+BUFCBL
      BUFCB(BUFP+BUFAD) = JBUFAD
      BUFCB(BUFP+BUFLEN) = SEGSZ
      BUFCB(BUFP+BUFLAG) = -1
      BUFCB(BUFP+SECNO) = -1
      BUFCB(BUFP+BUFPNT) = BUFP+BUFCBL
    1 JBUFAD = JBUFAD+SEGSZ+MAXCHN-1
      BUFCB(BUFP+BUFPNT) = 1
C
      IFNO = IDAFNO
      EVENT = EVNDA
      NSEC = SEGMNT
C
C --- OPEN THE DATABASE FILE
      OPEN IFNO,'WHOLE1',ATT='C',REC=SECEND+1
C
C --- SETUP SOME DEFAULTS
      EPISEC = 0
      OFFSET = 0
      LSEC = SECEND
C
      RETURN
      END
      SUBROUTINE FRAMD(WIDTH, KSEL, K, ISCAN, IERR)
C
C  FRAME DEFINITION
C
C  P.A. MARSHALL
C  MARCH 1976
C----------------------------------------------------------------------
C
C --- THIS ROUTINE INITIALIZES THE FILE POINTERS AND DEFINES THE
C  FRAME FOR ACCESSING CHANNELS FROM THE EVENT FILE.  SUBSEQUENT CALLS
C  TO SUBROUTINE 'GETFM' WILL ACCESS THE FILE SEQUENTIALLY, BRINGING
C  IN "FRAMES" OF DATA AS DEFINED BELOW, AND STARTING WITH THE INITIAL
C  SCAN SPECIFIED BY 'ISCAN'.
C
C      A FRAME IS DEFINED TO BE AN ORDERED SET OF CHANNEL SEGMENTS
C  OF EQUAL SIZE.  THE SIZE OF A SEGMENT IS CALLED THE FRAME 'WIDTH'.
C  DATA IN EACH SEGMENT OCCURS SERIALLY AND HAS BEEN DEMULTIPLEXED IN
C  PARALLEL WITH EVERY OTHER SEGMENT.  THERE ARE 'K' SEGMENTS IN A
C  FRAME, EACH SEGMENT (I) BEING IDENTIFIED WITH THE CHANNEL NUMBER
C  GIVEN BY THE ASSOCIATED ELEMENT 'KSEL(I)' FROM THE ARRAY OF
C  CHANNEL SELECT ORDINALS 'KSEL'.  ANY CHANNEL NUMBER IDENTIFIED IN
C  'KSEL' NEED NOT BE UNIQUE.
C
C --- STRUCTURE OF A FRAME CONSISTING OF K CHANNELS:
C
C        ------------   -
C        !  CHAN. # !   !_ FRAME WIDTH
C        !  KSEL(1) !   !
C        ------------   -
C        !  CHAN. # !
C        !  KSEL(2) !
C        ------------
C        !     :    !
C        !     :    !
C        ------------
C        !  CHAN. # !
C        !  KSEL(K) !
C        ------------
C
C  -- THE FILE IS INITIALLY POSITIONED BY A CALL TO 'FRAMD' AT THE
C  SCAN SPECIFIED BY 'ISCAN' WITHIN THE CURRENT EVENT.  THIS SCAN
C  WILL BE MADE AVAILABLE AS THE FIRST SCAN OF THE FIRST FRAME
C  ACCESSED BY THE NEXT CALL TO 'GETFM'.  SUBSEQUENT CALLS TO 'GETFM'
C  WILL RETRIEVE FRAMES SEQUENTIALLY FROM THE INITIAL ONE.
C
C      THE CALL TO 'FRAMD' ALSO EFFECTS INITIALIZATION OF THE MCAM
C  I/O MODULES.
C
C
C --- USE
C
C        CALL FRAMD(WIDTH, KSEL, K, ISCAN, IERR)
C
C  WHERE, AT CALL
C
C      WIDTH  - FRAME WIDTH IN NO. OF SCANS PER CHANNEL
C
C      KSEL   - THE ARRAY OF CHANNEL SELECT ORDINALS (CHANNEL NOS.)
C
```

```
C     K      - THE NO. OF CHANNELS SELECTED (DIMENSION OF 'KSEL')
C
C     ISCAN  - THE INITIAL SCAN NO. (FIRST SCAN IN FILE = 1)
C
C  AND UPON RETURN
C
C     IERR   - FILE STATUS RETURN. THE FOLLOWING VALUES ARE DEFINED:
C
C                IERR = 0  - NORMAL RETURN (FRAME DEF. & FILE OK)
C                IERR = -1 - INVALID ARGUMENTS IN 'FRAMD' CALL
C                IERR = 1  - INVALID FILE POSITION (ISCAN NOT THERE)
C
C-----------------------------------------------------------------------
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072          !NO. OF WORDS PER TRACK
      PARAMETER NST = 12            !NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256           !NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2          !SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5          !BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32         !MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2           !NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
      INCLUDE 'MCAMSB'
C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
      PARAMETER BUFAD = 0           !BUFFER ADDRESS (INDEX)
      PARAMETER BUFLEN = 1          !BUFFER LENGTH
      PARAMETER BUFLAG = 2          !BUFFER STATUS FLAG
      PARAMETER SECNO = 3           !RELATIVE SECTOR NO.
      PARAMETER BUFPNT = 4          !POINTER TO NEXT BUFFER
C****
      IMPLICIT INTEGER (A-Z)
      LOGICAL FIRST, INIT
      DIMENSION KSEL(1)
      COMMON /IOBDA/ NCHAN, SEGSZ, REMDR, LTBUF, FIRST
      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC
      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP)
      COMMON /UFRAM/ WID, KCHAN, KEY(MAXCHN)
C
C
C --- CHECK ALL ARGUMENTS FOR VALIDITY
      IERR = 0
      IF (MINO(WIDTH, K, ISCAN) .LE. 0) GO TO 51
      IF (K .GT. MAXCHN) GO TO 51
      DO 11 J=1,K
      IF (KSEL(J) .LE. 0 .OR. KSEL(J) .GT. NCHAN) GO TO 51
   11 CONTINUE
C
C --- STORE ARGUMENTS IN FRAME DEFINITION TABLE
      DO 12 J=1,K
   12 KEY(J) = KSEL(J)
      KCHAN = K
      WID = WIDTH
C
      FIRST = .TRUE.
      INIT = .TRUE.
```

```
C
C --- ALGORITHM TO COMPUTE THE INITIAL SECTOR NO. AND REMAINDER
C  FOR A SPECIFIED INITIAL SCAN.
C
C --- THE FOLLOWING VARIABLES ARE DEFINED:
C         NCHAN   - NO. OF CHANNELS THIS EVENT
C         SEGSZ   - SEGMENT SIZE (IN WORDS)
C         NSPT    - NO. OF SCANS PER TRACK
C         NWL     - LOGICAL TRACK LENGTH (IN WORDS)
C         ISCAN   - INITIAL SCAN NO. (FIRST = 1)
C         ITRK    - INITIAL, RELATIVE TRACK NO. (FIRST = 0)
C         IPTS    - WORD COUNT OFFSET IN INITIAL TRACK (INCLUDES 1ST SCAN)
C         LSEG    - SEGMENT NO. CONTAINING LAST POINT IN INITIAL SCAN
C         ISEG    - SEGMENT NO. CONTAINING FIRST POINT IN INITIAL SCAN
C         EPISEC  - EVENT/PASS FILE OFFSET (RELATIVE SECTOR)
C         ISEC    - INITIAL, RELATIVE SECTOR NO. (FILE POSITION)
C
      NSPT = NWT/NCHAN
      NWL = NSPT*NCHAN
      ITRK = (ISCAN-1)/NSPT
      IPTS = NCHAN*ISCAN-ITRK*NWL
      LSEG = (IPTS-1)/SEGSZ
      ISEG = (IPTS-NCHAN)/SEGSZ
C
      ISEC = ITRK*NST+ISEG*NSEC+EPISEC
C --- CHECK FOR VALID FILE POSITION
      IF (ISEC .GT. LSEC) GO TO 52
C
C --- INITIALIZE MCAM MODULES
C
C --- FILL THE FIRST 'NBUFS-1' BUFFERS
      BUFP = 1
      N = NBUFS-1
      DO 1 J=1,N
      CALL SEGIN
    1 BUFP = BUFCB(BUFP+BUFPNT)
C
C --- COMPUTE NEGATIVE REMAINDER TO OFFSET OF INITIAL SCAN
      REMDR = -MOD(IPTS-NCHAN, SEGSZ)
      LTBUF = BUFP
      BUFP = 1
C
C --- IF FIRST SCAN SPANS A SEGMENT, INPUT NEW BUFFER AND ACCEPT
C  NEW REMAINDER
      IF (LSEG .NE. ISEG) CALL SEGMO($52, IDUM, IDUM)
      RETURN
C
C --- ERRORS
C
   51 IERR = -1
      RETURN
C
   52 IERR = 1
      RETURN
      END
      SUBROUTINE GETFM(FRAME, M, IEND)
C
C  FRAME INPUT MODULE
C
C  P.A. MARSHALL
C  MARCH 1976
C-----------------------------------------------------------------------
C
C --- THIS ROUTINE FILLS A FRAME WITH DEMULTIPLEXED DATA FROM THE
C  EVENT FILE.  THE FRAME MUST HAVE BEEN PREVIOUSLY DEFINED BY A CALL
C  TO 'FRAMD'.  DATA IS LOADED INTO THE FRAME BEGINNING WITH THE NEXT
C  SEQUENTIAL SCAN IN THE FILE.
C
C     THE NUMBER OF SCANS PER CHANNEL LOADED INTO THE FRAME IS
C  RETURNED AS THE ARGUMENT 'M'.  THE VALUE OF 'M' IS NORMALLY FIXED
C  AND EQUAL TO THE FRAME "WIDTH," HOWEVER WHEN AN END-OF-FILE
C  CONDITION OCCURS WHILE LOADING THE FRAME THE VALUE RETURNED FOR
C  'M' WILL BE LESS THAN THE WIDTH AND WILL REFLECT THE ACTUAL NUMBER
C  OF SCANS LOADED INTO THE FRAME.  (IT IS TO BE NOTED THAT THE FRAME
C  WILL NOT BE COMPRESSED IN THIS LATTER CASE, SO THAT THOSE PORTIONS
C  OF THE FRAME NOT FILLED WILL CONTAIN INVALID DATA.)
C
C
C --- USE
C
C       CALL GETFM(FRAME, M, IEND)
C
C  WHERE, AT CALL
C
C     FRAME  - (INTEGER ARRAY) IS THE DATA AREA WHICH WILL RECEIVE
C              THE FRAME
C
C  AND UPON RETURN
C
C     M       - RECEIVES A VALUE EQUAL TO THE NUMBER OF SCANS PER
C               CHANNEL ACTUALLY LOADED INTO THE FRAME
C
C     IEND    - END OF FILE INDICATOR, AS FOLLOWS:
C
C                 IEND = 0  - NORMAL RETURN (M = "WIDTH")
C                 IEND = 1  - END-OF-FILE (0 >= M < "WIDTH")
C                  ("WIDTH" HAVING BEEN DEFINED BY A CALL TO 'FRAMD')
C
C-----------------------------------------------------------------------
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072          ;NO. OF WORDS PER TRACK
      PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256           ;NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2          ;SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5          ;BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32         ;MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2           ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
      IMPLICIT INTEGER (A-Z)
      LOGICAL FIRST
      COMMON /UFRAM/ WIDTH, KCHAN, KSEL(MAXCHN)
      COMMON /IOBDA/ NCHAN, DUM(3), FIRST
      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP), BUFFER(BUFFRP)
      DIMENSION FRAME(1)
      STATIC NSCAN, BUFPT
C
C
C --- INITIALIZE FRAME INDEX AND COUNT
      M = 0
C
C --- ON FIRST CALL INITIALIZE SEGMENT BUFFER SCAN COUNT
      IF (FIRST) NSCAN = 0
      FIRST = .FALSE.
C
C --- LOOP TO LOAD THE FRAME
C
C --- BRING IN NEW SEGMENT AS REQUIRED
   10 IF (NSCAN .EQ. 0) CALL SEGMO($50, BUFPT, NSCAN)
C
C --- FAN OUT TO FRAME AS MANY SCANS AS REQUIRED OR AVAILABLE
      NMUX = MIN0(NSCAN, WIDTH-M)
      CALL DEMUX(BUFFER(BUFPT), NCHAN, NMUX, WIDTH-NMUX, KSEL, KCHAN,
     * FRAME(M+1))
C
C --- UPDATE BUFFER POINTER AND COUNTS
      BUFPT = BUFPT+(NMUX*NCHAN)
      NSCAN = NSCAN-NMUX
      M = M+NMUX
C
C --- SEE IF WE HAVE FULLY LOADED FRAME
      IF (M .LT. WIDTH) GO TO 10
      IEND = 0
      RETURN
C
C --- END OF EVENT FILE
   50 IEND = 1
      RETURN
      END
```

SEGIN

```
      SUBROUTINE SEGIN
C
C   M C A M  SUBSYSTEM - SEGMENT BUFFER QUEUING FOR INPUT
C
C  P.A. MARSHALL
C  FEBRUARY 1976
C-----------------------------------------------------------------------
C
C --- THE ROUTINE REMEMBERS THE INITIAL SECTOR NO. OF THE LAST
C     SEGMENT INPUT.
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072         ;NO. OF WORDS PER TRACK
      PARAMETER NST = 12           ;NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256          ;NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2         ;SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5         ;BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32        ;MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2          ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
      INCLUDE 'MCAMSB'
C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
      PARAMETER BUFAD = 0          ;BUFFER ADDRESS (INDEX)
      PARAMETER BUFLEN = 1         ;BUFFER LENGTH
      PARAMETER BUFLAG = 2         ;BUFFER STATUS FLAG
      PARAMETER SECNO = 3          ;RELATIVE SECTOR NO.
      PARAMETER BUFPNT = 4         ;POINTER TO NEXT BUFFER
C****
      IMPLICIT INTEGER (A-Z)
      LOGICAL INIT
      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
      COMMON /IOBDA/ DUM(2), REMDR
      COMMON /BUFDA/ BUFPRM, BUFCB(BUFCBP)
      STATIC SECTOR
C
C
      IF (INIT) SECTOR = ISEC
C
C --- LOG SECTOR NO. FOR SEGMENT
      BUFCB(BUFPRM+SECNO) = SECTOR
C
C --- CLEAR BUFFER FLAG AND START I/O
      BUFCB(BUFPRM+BUFLAG) = 0
C**** TASK WAKUP
      CALL DISKI
C
      SECTOR = SECTOR+NSEC
      INIT = .FALSE.
      RETURN
      END
```

SEGMO

```
      SUBROUTINE SEGMO(S, BSTART, NSCAN)
C
C   M C A M  SUBSYSTEM - SEGMENT READ-IN
C
C  P.A. MARSHALL
C  FEBRUARY 1976
C------------------------------------------------------------------------
C
C --- READ IN NEXT SEGMENT, PREFIX REMAINDER, IDENTIFY START OF
C      BUFFER (BSTART), RETURN NO. OF SCANS (NSCAN) IN THIS BUFFER
C      SEGMENT.
C
C --- DEFINITION: A "SEGMENT" IS THE BUFFERED UNIT OF DATA READ IN
C      FROM DISK, CONSISTING OF "NSEC" CONTIGUOUS SECTORS. THERE ARE
C      AN INTEGRAL NUMBER OF SUCH SEGMENTS PER TRACK.
C
C------------------------------------------------------------------------
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072          ;NO. OF WORDS PER TRACK
      PARAMETER NST = 12            ;NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256           ;NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2          ;SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5          ;BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32         ;MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2           ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
      INCLUDE 'MCAMSB'
C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
      PARAMETER BUFAD = 0           ;BUFFER ADDRESS (INDEX)
      PARAMETER BUFLEN = 1          ;BUFFER LENGTH
      PARAMETER BUFLAG = 2          ;BUFFER STATUS FLAG
      PARAMETER SECNO = 3           ;RELATIVE SECTOR NO.
      PARAMETER BUFPNT = 4          ;POINTER TO NEXT BUFFER
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /IOBDA/ NCHAN, SEGSZ, REMDR, LTBUF
      COMMON /BUFDA/ BUFPRM, BUFCB(BUFCBP), BUFF(BUFFRP)
C
C
C --- CHECK STATUS OF LAST READ
C**** TASK CALL REC
      STAT = BUFCB(BUFPRM+BUFLAG)
      IF (STAT .EQ. 9) RETURN 1
      CALL CHECK(STAT)
C
C --- COMPUTE STARTING ADDRESS OF BUFFER; I.E, SEGMENT ADDRESS
C  MINUS PREFIX
C --- REMAINDER IS ZEROED AT START OF A NEW TRACK
      IF (REMDR .GT. 0 .AND.
     * MOD(BUFCB(BUFPRM+SECNO), NST) .EQ. 0) REMDR = 0
      BSTART = BUFCB(BUFPRM+BUFAD)-REMDR
```

```
C
C --- IF REMAINDER FROM PRIOR READ IS POSITIVE THEN MOVE IT TO PREFIX
      IF (REMDR .GT. 0) CALL FBLM(BUFF(BSTART),
     * BUFF(BUFCB(LTBUF+BUFAD)+SEGSZ-REMDR), REMDR)
C
C --- COMPUTE CURRENT BUFFER SIZE (INCLUDES PREFIX) AND NO. OF SCANS
      BUFSZ = SEGSZ+REMDR
      NSCAN = BUFSZ/NCHAN
C
C --- COMPUTE NEW REMAINDER
      REMDR = MOD(BUFSZ, NCHAN)
C
C --- SAVE CURRENT BUFFER, START INPUT ON LAST BUFFER (NOW AVAILABLE)
      BUFSV = BUFPRM
      BUFPRM = LTBUF
      CALL SEGIN
C
C --- CHAIN TO NEW SEGMENT BUFFERS
      LTBUF = BUFSV
      BUFPRM = BUFCB(BUFSV+BUFPNT)
      RETURN
      END
      SUBROUTINE DISKI
C
C   M C A M  SUBSYSTEM - SEGMENT INPUT TASK
C
C  P.A. MARSHALL
C  FEBRUARY 1976
C----------------------------------------------------------------------
C
C *** SINGLE TASK VERSION - THIS IS NOT A TASK.
C
C --- THIS IS THE INPUT TASK FOR DISK FILE SEGMENTS.  BUFFERS ARE
C     CHAINED VIA POINTERS IN THE BUFFER CONTROL TABLES.
C----------------------------------------------------------------------
C
      INCLUDE 'MCAMSD'
C**** PARAMETERS - DISK ORGANIZATION
      PARAMETER NWT = 3072         ;NO. OF WORDS PER TRACK
      PARAMETER NST = 12           ;NO. OF SECTORS PER TRACK
      PARAMETER NWS = 256          ;NO. OF WORDS PER SECTOR
C****
      INCLUDE 'MCAMSP'
C**** PARAMETERS -  CONTROLLING BUFFER AND TABLE SIZES AT COMPILATION
      PARAMETER SEGMNT = 2         ;SEGMENT BUFFER SIZE IN SECTORS
      PARAMETER BUFCBL = 5         ;BUFFER CONTROL BLOCK LENGTH
      PARAMETER MAXCHN = 32        ;MAXIMUM MULTIPLEX FACTOR
      PARAMETER NBUFS = 2          ;NO. OF SEGMENT BUFFERS (2 MINIMUM)
C *** ALLOCATION FOR BUFFER POOL AND BUFFER CONTROL BLOCK POOL:
      PARAMETER BUFFRP = NBUFS*(NWS*SEGMNT+MAXCHN-1)
      PARAMETER BUFCBP = NBUFS*BUFCBL
C****
      INCLUDE 'MCAMSB'
C**** PARAMETERS - BUFFER CONTROL BLOCK OFFSETS
      PARAMETER BUFAD = 0          ;BUFFER ADDRESS (INDEX)
      PARAMETER BUFLEN = 1         ;BUFFER LENGTH
      PARAMETER BUFLAG = 2         ;BUFFER STATUS FLAG
      PARAMETER SECNO = 3          ;RELATIVE SECTOR NO.
      PARAMETER BUFPNT = 4         ;POINTER TO NEXT BUFFER
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /IOPDA/ INIT, NSEC, ISEC, LSEC, EPISEC, OFFSET, IFNO, EVENT
      COMMON /BUFDA/ BUFP, BUFCB(BUFCBP), BUFF(BUFFRP)
C
C
C --- I/O IS INITIATED HERE
C**** 9 WAIT EVNDA
    9 CONTINUE
      CBUF = BUFP
C
C --- I/O CONTINUES HERE
C**** CHANGE 'RETURN' TO 'GO TO 9'
   10 IF (BUFCB(CBUF+BUFLAG) .NE. 0) RETURN
C
C --- CHECK FOR EOF; ELSE DO INPUT
      IERR = 9
      IF (BUFCB(CBUF+SECNO) .LE. LSEC)
     * CALL RDBLK(IFNO, BUFCB(CBUF+SECNO)+OFFSET,
     * BUFF(BUFCB(CBUF+BUFAD)), NSEC, IERR)
C
C --- LOG STATUS
```

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

```
      SUBROUTINE DEMUX(MUX, NCHAN, NSCAN, ISPAN, KEY, K, AREA)
C
C  DIGITAL DATA DEMULTIPLEXING
C
C  P.A. MARSHALL
C  NOVEMBER 1975
C-----------------------------------------------------------------------
C
C --- THIS ROUTINE DEMULTIPLEXES DIGITAL CHANNELS FROM THE INPUT "MUX"
C     ARRAY TO AN OUTPUT "AREA".  INPUT DATA IS A FULLY MULTIPLEXED
C     ARRAY OF INTEGER VALUES, CONSISTING OF "NSCAN" SETS OF "NCHAN"
C     CHANNELS EACH.  CHANNELS ARE NUMBERED FROM ONE TO "NCHAN"
C     CONTIGUOUSLY WITHIN EACH SET (SCAN).
C
C     A SUBSET OF CHANNELS WILL BE DEMULTIPLEXED INTO THE OUTPUT AREA
C     ACCORDING TO AN ARRAY ("KEY") OF CHANNEL SELECT ORDINALS.  FOR
C     EACH CHANNEL NAMED (BY CHANNEL NO.) IN THE KEY ARRAY, "NSCAN"
C     POINTS FOR THAT CHANNEL WILL BE DEMULTIPLEXED AND TRANSFERRED
C     SEQUENTIALLY INTO THE OUTPUT AREA.  THE MULTIPLEXED CHANNELS MAY
C     BE SELECTED IN ANY ORDER FOR OUTPUT BY THIS PROCESS AND INDEED,
C     ANY CHANNEL MAY BE SELECTED AND TRANSFERRED TO THE OUTPUT AREA
C     MORE THAN ONCE.
C
C     CHANNELS MAY BE SEPARATED FROM EACH OTHER IN THE OUTPUT AREA
C     BY SOME FIXED AMOUNT SPECIFIED BY 'ISCAN'.  THIS QUANTITY
C     SPECIFIES THE SEPARATION IN WORDS BETWEEN THE LAST DATA POINT
C     OF ONE CHANNEL AND THE FIRST DATA POINT OF THE SECCEEDING
C     CHANNEL IN THE OUTPUT AREA.
C
C     UPON RETURN FROM THIS ROUTINE, IF THERE WERE "K" CHANNELS
C     SELECTED FOR OUTPUT BY THE KEY ARRAY THEN THE OUTPUT AREA WILL
C     CONTAIN "K" SETS OF "NSCAN" SAMPLES EACH, SEPARATED BY AN
C     AMOUNT 'ISPAN', WHERIN EACH SET LIE CONTIGUOUSLY ALL THE DATA
C     SAMPLES FOR A PARTICULAR CHANNEL.  THE ORDERING OF THE SETS
C     WITHIN THE OUTPUT AREA WILL BE THE SAME AS THE ORDERING OF THE
C     CHANNELS WITHIN THE KEY ARRAY.
C
C  USE
C
C     CALL DEMUX(MUX, NCHAN, NSCAN, ISPAN, KEY, K, AREA)
C
C  WHERE
C
C    MUX    - IS THE INPUT ARRAY OF MULTIPLEXED CHANNELS
C
C    NCHAN  - IS THE NO. OF MULTIPLEXED CHANNELS CONTAINED IN "MUX"
C
C    NSCAN  - IS THE NO. OF DATA POINTS PER CHANNEL IN "MUX"
C
C    ISPAN  - IS THE SEPARATION (IN WORDS) BETWEEN CHANNELS IN THE
C             OUTPUT AREA. (SET TO ZERO FOR NO SEPARATION.)
C
C    KEY    - IS THE ARRAY OF CHANNEL SELECT ORDINALS
C
C    K      - IS THE NO. OF CHANNELS SELECTED (DIMENSION OF "KEY")
C
C    AREA   - (INTEGER) IS THE OUTPUT ARRAY FOR DEMULTIPLEXED DATA
C
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(ICHAN)
        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

        .NREL
FBLM:   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 5, VERSION 5.10 -- FRIDAY, NOVEMBER 10, 1978  11:50:20 AM

                                    DEPRC.FR

      SUBROUTINE XDEPRC(IC,JC)
      OVERLAY ODEPRC
      COMMON /TVTUNE/ITV,ITX
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      INCLUDE 'C$BLAH'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'C$CPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      DIMENSION NNN(80),SQ(40)
      DATA IQ/63/
      DATA IY/1HY/
      DATA IX/1HX/
      DATA IBL/1H /
      INTEGER THR,TMN
C*  DEFINE THE WINDOW
1     CONTINUE
      IF(JC.EQ.19)GO TO 13
      IF(JC.EQ.20)GO TO 14
C*  DEFINE TIME WINDOW
13    WRITE(ITV,210)
210   FORMAT(19H DEFINE TIME WINDOW)
      WRITE(ITV,211)IQ
211   FORMAT(23H WINDOW WIDTH (SECONDS),R1)
C*
C*  READ TWIND
C*  READ TIME WINDOW LENGTH
      READ(ITX,234)NNN(1)
 234  FORMAT(S80)
      CALL INFREE(NNN,RQ,80)
      TWIND=RQ
C*
C*
C*  READ HRMIN SEC
      WRITE(ITV,222)KHR,KMN,RSEC
222   FORMAT(16H DATA STARTS AT ,I2,1H,,I2,2X,F7.2)
      WRITE(ITV,204)IQ
204   FORMAT(23H HR MIN SEC (FREE-FORM),R1,' [CR] GETS DEFAULT')
      WRITE(ITV,223)
223   FORMAT(" TYPE 'X R' FOR AN R SEC. OFFSET OF ORIGIN")
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      IF(NNN(1).EQ.IBL)GOTO 55
      IF(NNN(1).EQ.IX)GOTO 55
      THR=SQ(1)
      TMN=SQ(2)
      TSEC=SQ(3)
      GOTO 56
C   COME HERE FOR CR OR X MN RESPONSE
C
55    THR=KHR
      TMN=KMN
      TSEC=RSEC
      IF(NNN(1).EQ.IX)TSEC=RSEC+SQ(1)
C
56    CONTINUE
C COMPUTE TIME WINDOW RELATIVE TO DATA
C  TTT IS REQUESTED TIME IN SECONDS
C  RRR IS START OF DATA WINDOW IN SECONDS
      TTT=THR*3600+TMN*60+TSEC
      RRR=KHR*3600+KMN*60+RSEC
      TZARR=TTT-RRR
      IF(TZARR.LT.0.)WRITE(ITV,220)
      IF(TZARR.GT.DUR)WRITE(ITV,220)
      IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE '/JDATE/:'
     *,' IHR=',IHR,' IMN=',IMN,' TSEC=',TSEC
      IF(TZARR.LT.0..OR.TZARR.GT.DUR)TYPE ' TZARR=',
     *TZARR,'TSEC=',TSEC,' RSEC=',RSEC,' TTT=',TTT,
     *' RRR=',RRR
220   FORMAT(36H REGUESTED TIME OUT OF RANGE OF DATA)
203   FORMAT(32H STARTING TIME WINDOW (HRMN.SEC),2X,2F4.0,1X,F7.2)
C*
      WRITE(ITV,203)THR,TMN,TSEC
      WRITE(ITV,221)TZARR
221   FORMAT(1H ,F7.2,15H SEC. INTO DATA)
      IHR=THR
      IMN=TMN
      OTZARR=TZARR
      RETURN
14    WRITE(ITV,205)IQ
205   FORMAT(10H 4014 TERM,R1,6HY OR N)
      READ(ITX,701)IA
701   FORMAT(A1)
      CALL TERM(1,1024)
      ITERM=1
      IF(IA.EQ.IY)CALL TERM(3,4096)
      IF(IA.EQ.IY)ITERM=4
      LWIDTH=1000
      LHY=600
      JX=10
      JY=10
      WRITE(ITV,206)IQ
206   FORMAT(12H RESET FRAME,R1)
      READ(ITX,701)IA
      IF(IA.NE.IY)GO TO 12
      WRITE(ITV,207)
207   FORMAT(33H INPUT WIDTH, HEIGHTH-(FREE FORM))
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      LWIDTH=SQ(1)
      LHY=SQ(2)
      WRITE(ITV,208)
```

```
209    FORMAT(39H INPUT SCREEN ORIGIN-IX,IY--(FREE FORM))
       READ(ITX,234)NNN(1)
       CALL INFREE(NNN,SQ,80)
       JX=SQ(1)
       JY=SQ(2)
12     CONTINUE
       CALL SWND(JX,LWIDTH,JY,LHY)
       ISCAL=1024*ITERM
       CALL CHRSIZ(JCHAR)
C*
       RETURN
       END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, JUNE 30, 1978 4:22:34 PM

EVPRC.FR

```
      SUBROUTINE XEVPRC(IC,JC)
      OVERLAY OEVPRC
      INCLUDE 'CSGPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      DIMENSION MSG(7)
      DIMENSION NNN(41),SQ(10)
      DATA MSG/80,65,83,83,32,32,48/
      COMMON /ZER/IZER
      INCLUDE 'CSAPARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'CSBPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /CPARM/NSTA,ISTA(2,32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SCAL/IX,LX,IY,LY,ITERM,ISCAL,ICHAR
      COMMON /TVTUNE/ITV,ITX
      COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
      COMMON /MISC/ZX,ZY,CX,CY
      COMMON /DECI/IDF,DECMAX,DLIM
      INTEGER CX,CY,ZX,ZY
C   EVENT REQUEST PROCESSOR
      IF(IC.EQ.6)GO TO 52
20    WRITE(ITV,302)
302   FORMAT('**ISDS** - SELECT EVENT NUMBER')
      READ(ITX,234)NNN(1)
 234  FORMAT(S80)
      CALL INFREE(NNN,SQ,80)
      EVENT=SQ(1)
      CALL RQEVT(LIST,EVENT)
      IF(LIST(1).NE.0)GOTO 50
C OUTPUT ERROR MESSAGE AND RETURN
      WRITE(ITV,303)
303   FORMAT(18H EVENT NOT ON DISK)
      RETURN
50    CONTINUE
C LOAD BPARM AND KDATE VARIBLES#
      KYR=LIST(2)
      KMO=LIST(3)
      KDA=LIST(4)
      KHR=LIST(5)
      KMN=LIST(6)
      RSEC=SEC
      DUR=DURAT
      INTEGER MUX(1), AREA(1), KEY(K)
C
      L = -ISPAN
C
      DO 10 I=1,K
      L = L+ISPAN
      ICHAN = KEY(I)
C
      DO 10 J=1,NSCAN
      L = L+1
      AREA(L) = MUX(ICHAN)
   10 ICHAN = ICHAN+NCHAN
C
      RETURN
      END
      DTG=DELT
      TCORR=TIMCOR
      UNQYR=LIST(15)
      UNQID=LIST(16)
C  PRINT OUT UNIQUE ID NUMBER
      WRITE(ITV,306)UNQYR,UNQID
306   FORMAT(' UNIQUE ID:',I3,'-',I5)
      WRITE(ITV,307)
307   FORMAT(' OK?(Y OR N)')
      CALL TINPUT(IA)
      IF(IA.NE.89)GOTO 20
C  REZERO TZARR (POINT TO START OF DATA)
      TZARR=0.
C     TYPE 'SEC,DURAT,DELT=',SEC,DURAT,DELT
      NPASS=LIST(LSZ1)
C LOAD DT ARRAY
      DO 40 J=1,32
40    DT(J)=DTG
C
C  EQUATE JDATE WITH KDATE FOR INITIAL PLOTS
      IHR=KHR
      IMN=KMN
      TSEC=RSEC
C
C  IGAP IS SIZE OF GAP(IN TEKPOINTS) BETWEEN VIEWPORTS
C   MAXVP IS PARAMETER=MAXIMUM NUMBER OF VIEWPORTS ON
C   MVP IS MIN OF MAXVP AND NUMBER OF PASSES
C
      MVP=MIN0(MAXVP,NPASS)
C    SCREEN AT ANY GIVEN TIME
      IGAP=160/MVP
      KX1=30
C  COMPUTE WIDTH OF VIEWPORTS
      LXX=990/MVP-IGAP
      JY=10
      LHY=600
      CALL ERASE
C
      DO 51 N=1,NPASS
      NCYCLE=MOD(N,MVP)
      IF(NCYCLE.EQ.0)NCYCLE=MVP
C  COMPUTE X-ORIGIN OF NTH VIEWPORT
      JX=KX1+(NCYCLE-1)*(IGAP+LXX)
      CALL SWND(JX,LXX,JY,LHY)
C  SAVE SCREEN CO-ORDINATES OF CURRENT VIEWPORT
      VSAVE(1,N)=JX
      VSAVE(2,N)=LXX
      VSAVE(3,N)=JY
      VSAVE(4,N)=LHY
C PLOT THIS PASS
C  SETUP I/O TO PLOT ALL DATA
      MUX=LIST(N+LSZ1)
      IF(MUX.EQ.0)GOTO 51
      NCHAN=MUX
      NREQ=MUX
C  EQUATE PREVIOUS TRACE STATE TO CURRENT STATE
      NTEMP=NREQ
      DO 30 J=1,NREQ
      ITEMP(J)=J
30    IREQ(J)=J
C
C  SET WINDOW TO DURATION TIME OF DATA
      TWIND=DUR
      CALL DECIM
C
      CALL RQPAZ(ISTA,FCA,LCA,EVENT,N)
C
      CALL NOSHF
C  SPEED BY-PASS
C  JC=3 MEANS EV (NOT EVPL)
      IF(JC.NE.3)GOTO 51
C     :
      CALL RSETD
      IF(IZER.EQ.1)CALL ZLFV
      CALL ARRAY
C  LABEL PLOT ,OPTION 1 MEANS PUT START AND STOP TIMES
C   IN ADDITION TO THE OTHER LABELS
      CALL LHSTA(1)
C   PUT PASS LABEL ON PLOT
      MXX=LXX/2+JX
      MYY=LHY+JY+40
      CALL MOVABS(MXX,MYY)
      MSG(7)=48+N
      MSG(6)=32
      IF(N.GT.9)MSG(6)=49
      IF(N.GT.9)MSG(7)=N-10+48
      CALL ANSTR(7,MSG)
      IF(NCYCLE.NE.MVP)GOTO 51
C   LABEL PLOT WITH EVENT NAME AND ID-NUMBER
      CALL ENAME(NNN,EVENT)
      MZZ=LHY+JY+60
      CALL MOVABS(358,MZZ)
      CALL AOUTST(40,NNN)
      ENCODE(NNN,100)UNQYR,UNQID
100   FORMAT('UNIQUE ID:',I2,'-',I5)
      CALL MOVABS(800,MZZ)
      CALL AOUTST(18,NNN)
C   AFTER MVP WINDOWS ARE PLOTTED,PAUSE
C   GIVING OPERATOR CHANCE TO MAKE COPY.
C   TO CONTINUE,STRIKE ANY KEY,THEN SCREEN IS ERASED
      IF(N.EQ.NPASS)GOTO 51
      CALL TINPUT(IA)
C  IF YOU INPUT A PERIOD(.),PLOTTING WILL BE TERMINATED
      IF(IA.EQ.46)JC=15
      CALL NEWPAG
51    CONTINUE
      IF(JC.NE.3)GOTO 52
C
C  PASS REQUEST PROCESSOR
      CALL HOME
      CALL ANMODE
      CALL TSEND
52    CONTINUE
      WRITE(ITV,304)
304   FORMAT(19H SELECT PASS NUMBER)
      CALL NEWLIN
      CALL SEELOC(CX,CY)
      READ(ITX,234)NNN(1)
      CALL INFREE(NNN,SQ,80)
      PASS=SQ(1)
      N=PASS
      CALL RQPAZ(ISTA,FCA,LCA,EVENT,PASS)
C  - - - DEFINE TRACES FOR THIS PASS - - -
      MUX=LIST(LSZ1+N)
      IF(MUX.EQ.0)GOTO 99
      NREQ=MUX
      NTEMP=MUX
      DO 1 J=1,NREQ
      ITEMP(J)=J
 1    IREQ(J)=J
      CALL NOSHF
      CALL RSETD
C  DO THINGS USUALLY DONE BY ARRAY
      AR=AZ+AL
      CALL VWINDO(TZARR,TWIND,AZ,AR)
      CALL SWND(VSAVE(1,N),VSAVE(2,N),VSAVE(3,N),VSAVE(4,N))
      CALL SWINDO(IX,LX,IY,LY)
C  - - - - - - - - - - - - -
 99   CONTINUE
C  RESET DECIMATION FACTOR
      IDF=1
C  NULL OUT DPARM BLOCK
      CALL DEFAULT
      RETURN
      END
```

FAZPU.FR

```
      SUBROUTINE XFAZPU(IS)
      OVERLAY OFAZPU
      COMMON /TVTUNE/ITV,ITK
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INTEGER OLDEVT,PUFLG,PUBLNK,FZ,CC
      INTEGER CLIST(32),DFN(7)
      DIMENSION INAME(5)
      DATA FZ/2HFZ/,CC/2HCC/
      DATA IBLANK/2H  /,IZZ/2HZZ/,IPRMK/2H P/,ISRMK/2H S/
      DATA IP/1HP/
      DATA DFN/2HDP,2HO:,5*2H  /
      EQUIVALENCE (INAME(1),DFN(3))
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      PARAMETER ZPICK=4
C GET AT EVDIR RECORD P.R.S. 3/29/78 RECSIZ=628 FOR 12 PASSES
      INTEGER REC,PASSN,RECSIZ
      PARAMETER RECSIZ=628
      COMMON/EDIRC/NEVENT,PASSN(MAXPASS),REC(RECSIZ)
C
C        THIS ROUTINE PUNCHES PHASE CARDS DESTINED FOR USGS EARTHQUAKE
C        HYPOCENTER ROUTINES
C        IF IS=2, PUNCH BLANK AND ZZ DELIMITER
C        AND RETURN
C        CHECK TO SEE IF SAME EVENT AS BEFORE
      GOTO(10,20),IS
20    IF(PUFLG.EQ.0)RETURN
      WRITE(7,701)IBLANK
C*    WRITE(7,701)IZZ
701   FORMAT(A2,78X)
      CLOSE 7
      CLOSE 14
      RETURN
10    CONTINUE
C  IF NOT NEW EVENT, JUST PUNCH LIST
      IF(OLDEVT.EQ.EVENT)GOTO 30
C  FIRST TIME THRU?
```

```
      IF(PUFLG.EQ.0)GOTO 31
C  IF NOT, CLOSE OUT OLD FILES
      WRITE(7,701)IBLANK
C*    WRITE(7,701)IZZ
      CLOSE 7
      CLOSE 14
31    CONTINUE
C  OPEN NEW FZ AND CC FILES
      CALL MAKNAM(FZ,UNQID,INAME)
      OPEN 7,DFN
      CALL MAKNAM(CC,UNQIB,INAME)
      OPEN 14,DFN
C FIRST TIME THIS EVENT:WRITE EVENT INFO ON CC FILE
C AND UPDATE EPLOG FILE
      ISCAN=1
      WRITE(14,800)UNQYR,UNQID,TCORR,ISCAN,EVENT,MAXPASS
900   FORMAT(I2,I5,1X,F5.3,I5,2I2)
C  UPDATE EPLOG FOR THIS EVENT
      OPEN 15,'EPLOG',LEN=512
      CALL UTEP(UNQID,ZPICK,0,IER)
      CLOSE 15
30    CONTINUE
C  PUNCH PHASE LIST
      KNT=0
      DO 1 J=1,32
C  CHECK FOR NULL STATIONS
      IF(PTIME(J).LE.0..AND.STIME(J).LE.0..AND.
     *SAMP(J).LE.0..AND.AMP(J).LE.0.)GO TO 1
C  CHECK FOR NULL LABELS
      IF(ISTA(1,J).EQ.IBLANK)GOTO 1
C  OUTPUT PHASE CARD FOR THIS STATION
C  THE FOLLOWING ALGORTHM CORRECTS FOR THE PROPER P(OR S) SECOND
C   AND RIPPLES THE CORRECTION THRU THE CLOCK AND CALENDAR.
C   IF ONLY S TIME IS PRESENT IT IS RIPPLED.IF BOTH P AND S
C   ARE PRESENT THEN THE P TIME IS RIPPLED.THE S TIME MUST BE
C   THE SAME MINUTE,HOUR,ETC. AS THE P TIME.
      PTA=PTIME(J)+RSEC
      ST=0.
      PT=0.
      IF(PTIME(J).EQ.0..AND.STIME(J).GT.0.)PTA=STIME(J)+RSEC
C
      XT=AMOD(PTA,60)
      JT=(PTA-XT)/60.
      IF(PTIME(J).GT.0.)PT=PTIME(J)+RSEC-JT*60.
      IF(STIME(J).GT.0.)ST=STIME(J)+RSEC-JT*60.
      JMN=KMN+JT
      JHR=KHR
      JDA=KDA
      IF(JMN.GT.60)JHR=KHR+1
      IF(JMN.GT.60)JMN=KMN-60
      IF(JHR.GT.24)JDA=KDA+1
      IF(JHR.GT.24)JHR=KHR-24
      IAMP=AMP(J)
      ISAMP=SAMP(J)
C  COMPUTE CODA FROM FTIME AND PTIME
      FMP(J)=FTIME(J)-PTIME(J)
      IF(FMP(J).LE.0.)FMP(J)=0.
C  IF STIME=0 THEN SGRADE=4 AND SRMK=BLANK
C   IF PTIME=0 THEN IGRADE=4 AND PRMK=BLANK
      IF(ST.EQ.0.)SGRADE(J)=4
      IF(ST.EQ.0.)SRMK(J)=IBLANK
      IF(PT.EQ.0.)IGRADE(J)=4
      IF(PT.EQ.0.)PRMK(J)=IBLANK
C
C
71    WRITE(7,700)ISTA(1,J),ISTA(2,J),PRMK(J),IFMOT(J),IGRADE(J),
     *KYR,KMO,JDA,JHR,JMN,PT,IAMP,ST,SRMK(J),SGRADE(J),ISAMP,
     *RMK(1,J),RMK(2,J),CLK(J),RMK(3,J),FMP(J)
700   FORMAT(2A2,A2,A1,I1,1X,5I2,F5.2,I5,2X,F5.2,A2,1X,I1,4X,I3,15X,A2,
     *A1,F5.2,A1,I4,5X)
      KNT=KNT+1
      CLIST(KNT)=J
1     CONTINUE
C DETERMINE ABSOLUTE PASS FROM RELATIVE PASS
C APASS IS ABSOLUTE PASS NUMBER(REL. TO ANALOG TAPE)
      APASS=PASSN(PASS)
      WRITE(14,801)PASS,APASS,FCA,LCA,KNT,(CLIST(J),J=1,KNT)
801   FORMAT(36I2)
      PUFLG=1
      OLDEVT=EVENT
      RETURN
      END
```

GPICK.FR

```
      SUBROUTINE XGPICK(ICC)
      OVERLAY OGPICK
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      DIMENSION ICARD(41),LAB(3),NTT(3)
      DATA IDOL/2H$$/
      DATA NTT/2HEV,2HEN,2HT /
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      DIMENSION IREC(45),NNN(41)
      DIMENSION NAM(2),IDONE(32),KRMK(2)
C    ICC=1:READ AUTO PICKS
C    ICC=2:READ PHASE LIST
C    ICC=3:READ HYPOUT
C
      WRITE(ITV,150)
150   FORMAT(' FILENAME?')
C  READ '5' RATHER THAN ITX BECAUSE WISH INPUT
C  FROM USER DURING PICK MODE
      READ(5,441)NNN(1)
441   FORMAT(S17)
      OPEN 15,NNN
      GOTO(10,20,30),ICC
10    CONTINUE
3     READ(15,101,END=500)ICARD(1)
101   FORMAT(S80)
C         LOOK FOR EVENT LABEL
      DECODE(ICARD,102) (LAB(J),J=1,3)
102   FORMAT(3A2)
      IF(LAB(1).EQ.NTT(1).AND.LAB(2).EQ.NTT(2).AND.LAB(3).EQ.
     *NTT(3))GOTO 4
      GOTO 3
4     DECODE(ICARD,103)JEVENT
103   FORMAT(6X,I2)
C     - - - TEST FOR CORRECT EVENT NUMBER
      IF (JEVENT.NE.LIST(1))GOTO 3
C
      TYPE 'EVENT NO.',JEVENT
```

```
C   ZERO DONE FLAGS
      DO 5 J=1,32
5      IDONE(J)=0
C  LOOP AND READ UNTIL END OF EVENT
    2 READ(15,100,END=500)NAM,TIME,IFMT,DRTN
100   FORMAT(2A2,F7.2,A2,F7.2)
C  IF NAME = "$$",IT IS THE END OF THE EVENT
      IF(NAM(1).EQ.IDOL)GOTO 500
      DO 1 J=1,32
      IF(ISTA(1,J).NE.NAM(1))GOTO 1
      IF(ISTA(2,J).NE.NAM(2))GOTO 1
C  TAKE FIRST OCCURENCE ONLY
      IF(IDONE(J).EQ.1)GOTO 1
C  LOAD UP DPARM TABLE
      PTIME(J)=TIME
      FMP(J)=DRTN
      IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
      IFMOT(J)=IFMT
      IDONE(J)=1
      GOTO 2
1     CONTINUE
      GOTO 2
C   COME HERE IF NO FILE
 500  CONTINUE
C     RETURN
C  COME HERE WHEN EOF
502   CLOSE 15
      RETURN
C
20      CONTINUE
22      READ(15,200,END=502)NAM,IPR,IFMT,IG,JYR,JMO,
     *JDA,JHR,JMN,TIME,IPA,ST,ISR,ISG,ISA,KRMK,CC,JRMK,DRTN
200   FORMAT(3A2,A1,I1,1X,5I2,F5.2,I5,2X,F5.2,A2,1X,I1,4X,I3,
     *15X,A2,A1,F5.2,A1,F4.0,5X)
      GOTO 24
30    CONTINUE
31    READ(15,300,END=502)IREC
300   FORMAT(45A2)
      DECODE(IREC,301,ERR=502)NAM,IPR,IFMT,IG,JYR,JMO,
     *JDA,JHR,JMN,TIME,PRZ,PWW,ST,ISR,ISG,SRZ,ISA,SWW,
     *PD,SD,EPIDST,ANGLE,CC,JRMK,DRTN,AZ,DMAG,AMAG,PIMP,SIMP
301   FORMAT(3A2,A1,I1,1X,5I2,F5.2,F4.2,F3.2,F5.2,A2,1X,I1,
     *F4.2,F3.0,F3.2,2F4.2,F4.1,F3.0,F5.2,A1,F4.0,
     *F3.0,2F2.1,2F4.3)
24    CONTINUE
      DO 23 J=1,32
C  COMMON SECTION FOR HYPOUT AND FZLINK
      IF(ISTA(1,J).NE.NAM(1))GOTO 23
      IF(ISTA(2,J).NE.NAM(2))GOTO 23
      PTIME(J)=TIME-RSEC
      IF(PTIME(J).LT.0)PTIME(J)=PTIME(J)+60.
      FMP(J)=DRTN
      IFMOT(J)=IFMT
      STIME(J)=ST-RSEC
      IF(STIME(J).LT.PTIME(J).AND.ST.NE.0.)
     *STIME(J)=STIME(J)+60.
      IF(FMP(J).NE.0.)FTIME(J)=PTIME(J)+FMP(J)
      CROSS(J)=0.
      PCROSS(J)=0.
      IGRADE(J)=IG
      SGRADE(J)=ISG
      SAMP(J)=ISA
      RMK(3,J)=JRMK
      SRMK(J)=ISR
      PRMK(J)=IPR
C  IF NOT FZLINK,SKIP NEXT SECTION
      IF(ICC.NE.2)GOTO 25
      AMP(J)=IPA
      RMK(1,J)=KRMK(1)
      RMK(2,J)=KRMK(2)
C IF NOT HYPOUT,SKIP NEXT SECTION
25    IF(ICC.NE.3)GOTO 26
      PRES(J)=PRZ
      SRES(J)=SRZ
      EPD(J)=EPIDST
      CLK(J)=CC
      GOTO 26
23    CONTINUE
C  COME HERE AFTER SUCCESSFUL MATCH
26    CONTINUE
      GOTO(502,22,31),ICC
C

      END
```

LBSTA.FR

```
      SUBROUTINE XLBSTA(KC)
      OVERLAY OLBSTA
      INCLUDE 'CSBLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
      COMMON /CPARM/NSTA,ISTA(2,32)
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
     *D(32)
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      INCLUDE 'CSGPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      DIMENSION MTIME(9),NUM(2)
      DIMENSION MSG(7)
      DATA MSG/83,69,67,79,78,68,83/
      DO 1 K=1,NREQ
      JJ=IREQ(K)
      Y=D(K)
      X=TZARR
C*  POSITION BEAM TO APROPRIATE TRACE
      CALL MOVEA(X,Y)
      CALL MOVREL(10,5)
C*  WRITE STATION NAME
      I1=ISTA(1,JJ)
      I2=ISTA(2,JJ)
      CALL AOUTST(2,I1)
      CALL AOUTST(2,I2)
C   PUT OUT TRACE NUMBER BY TRACE START
      CALL MOVEA(X,Y)
      CALL MOVREL(-30,-4)
      ENCODE(NUM,102)JJ
102   FORMAT(I2)
      CALL AOUTST(2,NUM)
C
C  PUT OUT SCALE FACTOR
      IF(S(JJ).EQ.1)GOTO 1
      Y=D(K)
      X=TZARR+TWIND
      CALL MOVEA(X,Y)
      CALL MOVREL(-90,10)
      ENCODE(MTIME,101)S(JJ)
101   FORMAT(F8.1,'X')
      CALL AOUTST(9,MTIME)
1     CONTINUE
C  IF KC=2,SKIP TIME LABELS
      IF(KC.EQ.2)GOTO 2
C
```

```
C  LABEL BOX WITH CURRENT TIME
C
C  PUT WINDOW START TIME ABOVE UPPER RIGHT CORNER
C
      KYY=KY+LY+20
      CALL MOVABS(KX,KYY)
      ENCODE(MTIME,100)IHR,IMN,TSEC
100   FORMAT(2I2,2X,F6.2)
      CALL AOUTST(12,MTIME)
C  PUT WINDOW END TIME ABOVE UPPER LEFT CORNER
C
      TLAST=TZARR+TWIND+RSEC
C  CONVERT TO PROPER TIME
      JJJ=TLAST/60
      TLAST=TLAST-JJJ*60
      LMN=IMN+JJJ
      KKK=LMN/60
      LMN=LMN-KKK*60
      LHR=IHR+KKK
C
      KXX=KX+LX-90
      KYY=KY+LY+20
      CALL MOVABS(KXX,KYY)
      ENCODE(MTIME,100)LHR,LMN,TLAST
      CALL AOUTST(12,MTIME)
C
C LABEL TIME AXIS
2     MXX=LX/2+KX
      MYY=KY+10
      CALL MOVABS(MXX,MYY)
      CALL ANSTR(7,MSG)
      RETURN
      END
```

```
C     PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
C  --WRITTEN BY PETER R. STEVENSON (1976-78)--
C  --U.S.GEOLOGICAL SURVEY--
C  --OFFICE OF EARTHQUAKE STUDIES--
C  --MENLO PARK,CALIF.--
      COMMON /TVTUNE/ITV,ITX
      COMMON /RPLAY/VFILE
      INTEGER VFILE
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      COMMON /ZER/IZER
      COMMON/ZZZ/IZIT
      COMMON /DECI/IDF,DECMAX,DLIM
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$APARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'C$BPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /SCAL/ISDUM(7)
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE  'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SHCOM/ISH
      COMMON /IOPRM/IODUM(6)
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      INTEGER OLDEVT,PUFLG,PUBLNK
      COMMON /MISC/MDUM(4)
      IZER=0
      IZIT=0
        ITV=4
        ITX=5
      OPEN 4,'$TTO1'
      OPEN 5,'$TTI1'
      OPEN 3,'JTEK'
      VFILE=-1
      OPEN 9,'TEMP'
      CALL OVOPN("ISDS.OL",IER)
      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER
      CALL INITT(0)
      CALL SETBUF(3)
      CALL CHRSIZ(4)
      CALL TSEND
      CALL HITEK(1,0)
      CALL MCAM1
      CALL DIRC1
      CALL DEFAULT
      CALL SPLOT
      STOP
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, NOVEMBER 10, 1978 1:44:34 PM

LPICK.FR

```
      SUBROUTINE XLPICK
      OVERLAY OLPICK
CC  SUBROUTINE TO DISPLAY PICKS
C  P.R.STEVENSON  8/21/78
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE 'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),
     *D(32)
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,
     *SG
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /MISC/ZX,ZY,CX,CY
      INTEGER ZX,ZY,CX,CY
      DIMENSION MSGP(40)
      DATA IBLNK/2H  /
      TRNG=TZARR+TWIND
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
2     CONTINUE
CC
3     CONTINUE
      REWIND 9
C  SAVE HEADER ON SCRATCH FILE (L.U. #9)
      WRITE(9,903)
903   FORMAT(33HNU.  STA  P-TIME  P-AMPL.  S-TIME,3X,7HS-AMPL.,2X,
     *3HSMP,5X,3HFMP,2X,4HP-CR,2X,3HFM ,4HPRMK,2X,4HSRMK,2X,1HR)
      CALL SETMRG(150,1023)
      DO 41 K=1,NREQ
      J=IREQ(K)
CC  DRAW MARK FOR P PICK
      IF(PTIME(J)+SHFT(J).LE.TZARR.OR.PTIME(J)+SHFT(J).GT.TRNG)GOTO 20
      CALL PMARK(PTIME(J)+SHFT(J),D(K),80)
C
C  DRAW DOTTED LINE FOR P RESIDUAL
```

```
      PRZ=PRES(J)+PTIME(J)+SHFT(J)
      IF(PRZ.LE.TZARR.OR.PRZ.GT.TRNG.OR.PRES(J).EQ.0.)GOTO 20
      CALL PMARK(PRZ,D(K),-82)
C
20    CONTINUE
CC
CC   DRAW MARK FOR S PICK
      IF(STIME(J)+SHFT(J).LE.TZARR.OR.STIME(J)+SHFT(J).GT.TRNG)GOTO 30
      CALL PMARK(STIME(J)+SHFT(J),D(K),83)
C
C   DRAW DOTTED LINE FOR S RESIDUAL
      SRZ=SRES(J)+STIME(J)+SHFT(J)
      IF(SRZ.LE.TZARR.OR.SRZ.GT.TRNG.OR.SRES(J).EQ.0.)GOTO 30
      CALL PMARK(SRZ,D(K),-114)
C
30    CONTINUE
      PCROSS(J)=CROSS(J)-PTIME(J)
      SMP(J)=STIME(J)-PTIME(J)
      FMP(J)=FTIME(J)-PTIME(J)
      IF(PCROSS(J).LT.0.)PCROSS(J)=0.
      IF(SMP(J).LT.0..OR.STIME(J).LT.0.)SMP(J)=0.
      IF(FMP(J).LT.0.)FMP(J)=0.
C  DRAW MARK FOR F PICK
      IF(FTIME(J)+SHFT(J).LE.TZARR.OR.FTIME(J)+SHFT(J).GT.TRNG)GOTO 10
      IF(FMP(J).GT.0.)CALL PMARK(FTIME(J)+SHFT(J),D(K),70)
10    CONTINUE
C PRETTY UP NULL S-TIMES
      IF(STIME(J).LT.0..AND.PTIME(J).GE.0.)STIME(J)=0.
CC
      CALL NEWLIN
      IXX=0
      IYY=750*ITERM
      CALL MOVABS(IXX,IYY)
      CALL ANMODE
      CALL TSEND
      PT=PTIME(J)+RSEC
      ST=STIME(J)+RSEC
      IF(PTIME(J).EQ.0.)PT=0.
      IF(STIME(J).EQ.0.)ST=0.
      IF(ISTA(1,J).EQ.0)ISTA(1,J)=IBLNK
      IF(ISTA(2,J).EQ.0)ISTA(2,J)=IBLNK
      IF(ST.EQ.0.)SGRADE(J)=4
      IF(ST.EQ.0.)SRMK(J)=IBLNK
      IF(PT.EQ.0.)IGRADE(J)=4
      IF(PT.EQ.0.)PRMK(J)=IBLNK
C  SAVE THE JTH ENTRY ON SCRATCH FILE
      WRITE(9,900)J,ISTA(1,J),ISTA(2,J),PT,AMP(J),ST,SAMP(J),SMP(J),
     *FMP(J),PCROSS(J),IFMOT(J),PRMK(J),IGRADE(J),SRMK(J),SGRADE(J),RMK(3,J)
900   FORMAT(I3,1X,2A2,1X,F7.2,2X,F7.2,2X,F7.2,2X,3F7.2,2X,F3.2,2X,A2,
     *2X,A2,I1,3X,A2,I1,2X,A1)
41    CONTINUE
CC
C  READ BACK INFO FROM SCRATCH FILE
      REWIND 9
      READ(9,905)MSGP
905   FORMAT(40A2)
C  OUTPUT HEADER TO TEK SCREEN
      CALL AOUTST(80,MSGP)
      DO 44 K=1,NREQ
      J=IREQ(K)
      READ(9,905)MSGP
921   CONTINUE
C  SKIP IF BOTH P AND S ARE ZERO
      IF(PTIME(J).LE.0..AND.STIME(J).LE.0.)GO TO 44
      CALL NEWLIN
C  OUTPUT ENTRY TO TEK SCREEN
      CALL AOUTST(80,MSGP)
44    CONTINUE
CC
C  SAVE BEAM POSITION
      CALL NEWLIN
      CALL SEELOC(ZX,ZY)
      CALL SETMRG(0,1023)
      RETURN
      END
```

MISDS7.FR

```
C     PROGRAM ISDS(INTERACTIVE SEISMIC DISPLAY SYSTEM)
C  --WRITTEN BY PETER R. STEVENSON (1976-78)--
C  --U.S.GEOLOGICAL SURVEY--
C  --OFFICE OF EARTHQUAKE STUDIES--
C  --MENLO PARK,CALIF.--
      COMMON /TVTUNE/ITV,ITX
      COMMON /RPLAY/VFILE
      INTEGER VFILE
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      COMMON /ZER/IZER
      COMMON/ZZZ/IZIT
      COMMON /DECI/IDF,DECMAX,DLIM
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      COMMON /CPARM/NSTA,ISTA(2,32)
      INCLUDE 'C$APARM'
      COMMON /APARM/S(32),Z(32),DT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'C$BPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON /SCAL/ISDUM(7)
      INCLUDE 'C$BLAN'
      PARAMETER MAXX=1000
      COMMON A(MAXX),MAXPTS,NREQ,IREQ(32)
      INTEGER A
      INCLUDE  'C$PICK'
      INTEGER PRMK,SRMK,RMK,SGRADE
      COMMON /DPARM/ PTIME(32),AMP(32),STIME(32),FTIME(32),
     *CROSS(32),SMP(32),FMP(32),PCROSS(32),
     *IFMOT(32),IGRADE(32),SGRADE(32),
     *SAMP(32),RMK(3,32),SRMK(32),PRMK(32),
     *PRES(32),SRES(32),EPD(32),CLK(32)
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SHCOM/ISH
      COMMON /IOPRM/IODUM(6)
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      INTEGER OLDEVT,PUFLG,PUBLNK
      COMMON /MISC/MDUM(4)
      IZER=0
      IZIT=0
        ITV=4
        ITX=5
      OPEN 4,'$TTO1'
      OPEN 5,'$TTI1'
      OPEN 3,'JTEK'
      VFILE=-1
      OPEN 9,'TEMP'
      CALL OVOPN("MISDS.OL",IER)
      IF(IER.NE.1)TYPE ' OVERLAY: IER=',IER
      CALL INITT(0)
      CALL SETBUF(3)
      CALL CHRSIZ(4)
      CALL TSEND
      CALL HITEK(1,0)
      CALL MCAM1
      CALL DIRC1
      CALL DEFAULT
      CALL SPLOT
      STOP
      END
```

SPLOT.FR

```
      SUBROUTINE SPLOT
      COMMON /TVTUNE/ITV,ITX
      INCLUDE 'C$APARM'
      COMMON /APARM/S(32),Z(32),BT(32),TZ(32),SHFT(32),D(32)
      INCLUDE 'C$BPARM'
      COMMON /BPARM/TZARR,TWIND,THR,TMN,AZ,ARNG,AL,NT,DTG,SG
C
      COMMON/SFL/SFLAG
      INTEGER SFLAG
      COMMON /ZER/IZER
      COMMON /SCAL/KX,LX,KY,LY,ITERM,ISCAL,JCHAR
      COMMON /EPARM/NTEMP,ITEMP(32),OTZARR,OTWIND
      COMMON /SHCOM/ISH
      COMMON /IOPRM/NS,NSCAN,NREM,MUX,IFILE,IUNIT
      INCLUDE 'C$GPARM'
C  MODIFIED 1/12/78 FOR 12-PASSES
      PARAMETER LSZ1=17
      PARAMETER MAXPASS=12,MAXVP=4,LSZ=LSZ1+MAXPASS
      COMMON /KDATE/KYR,KMO,KDA,KHR,KMN,RSEC,DUR,EVENT,PASS
     *,UNQYR,UNQID,TCORR,FCA,LCA
      COMMON /JDATE/IHR,IMN,TSEC
      COMMON /XXX/XX(3),LIST(LSZ)
      EQUIVALENCE (LIST(9),DURAT),(LIST(11),DELT),(LIST(7),SEC)
     *,(LIST(13),TIMCOR)
      COMMON /VEE/VSAVE(4,MAXPASS)
      INTEGER EVENT,PASS,UNQYR,UNQID,FCA,LCA,VSAVE
C
      INTEGER OLDEVT,PUFLG,PUBLNK,CHAIN
      COMMON /MISC/ZX,ZY,CX,CY
      COMMON /OLD/OLDEVT,PUFLG,PUBLNK,NFIL(20)
      INTEGER ZX,ZY,CX,CY
      DIMENSION MSG1(2),MSG2(3),MSG4(4)
      DIMENSION NNN(41),SQ(40)
      DATA IQ/63/
      DATA IY/1HY/
C   DEFAULT SCALE UNITS OPTION IS 1
      IPAUS=0
      SFLAG=1
      CHAIN=0
      MAXCH=32
      IFILE=1
      MUX=10
      PUBLNK=0
      PUFLG=0
      OLDEVT=0
      EVENT=0
      CX=0
      CY=780*ITERM
      ZX=0
      ZY=750*ITERM
C  GO IMMEDIATELY TO EVENT PROCESSOR
      IC=14
      GO TO 39
C
C  COME HERE TO START ANOTHER COMMAND
20    CONTINUE
```

```
        CALL COMMAND(IC,JC)
        GO TO (1,2,3,4,5,44,7,8,9,10,31,32,33,39,35,4,20,90,20,20,88,20,
       *20,38,20,20,20,40,42,20,41,43,20,20,2,777),IC
  777   CONTINUE
        IF(IPAUS.EQ.1)GOTO 45
        GO TO 20
  C*
  C*  DEFINE THE WINDOW
  1     CONTINUE
        CALL DEPRC(IC,JC)
  C*
  C*
        GO TO 20
  2     CONTINUE
        CALL TRPRC(IC,JC)
  C*
        GO TO 20
  3     CONTINUE
        IF(JC.EQ.15)GO TO 36
        IF(JC.EQ.9.OR.JC.EQ.12)GO TO 10
  C*  PLOT PROCESSOR
  C  RESTORE STATE BEFORE PLOTTING
        CALL RSTAT
        CALL PLPRC(JC)
  900   CONTINUE
        CALL MOVABS(CX,CY)
        CALL ANMODE
        CALL TSEND
        GO TO 20
  4     CONTINUE
  C***SHIFT PROCESSOR
        CALL RSTAT
  C
  C   ALIGN P-PHASES
        ISH=80
  C   ALIGN S-PHASES
        IF(JC.EQ.26)ISH=83
        IF(IC.EQ.14)ISH=83
  C   WADATI PLOT
        IF(IC.EQ.16)ISH=87
  C
        CALL PSHIFT(ISH)
        GO TO 900
  5     CONTINUE
  C* PICK PROCESSOR
        CALL PICK
        GO TO 900
  6     CONTINUE
  C*
        GO TO 20
  C*  RESET PLOT PARMS
  7     CONTINUE
  C  IF 'RESU',RESUME REMOTE FILE CONTROL
        IF(JC.EQ.34)GOTO 45
        IF(JC.EQ.27)GOTO 90
        CALL DEFAULT
        IZER=0
        GO TO 20
  C*  AMPLITUDE SCALING PROCESSOR
  8     CONTINUE
```

```
         CALL SCALER
         GO TO 20
C
C  SELECT SCALING OPTION
88       IF(JC.NE.8)GOTO 20
         WRITE(ITV,800)
800      FORMAT(" SELECT INPUT SCALE OPTION",/,
        *"1=DIGITAL COUNTS",/,"2=VOLTS OUT",/,
        *"3=EQUIVALENT DEVELOCORDER MILLIMETERS")
         READ(ITX,234)NNN(1)
234      FORMAT(S81)
         CALL INFREE(NNN,RQ,96)
         SFLAG=RQ
         WRITE(ITV,801)SFLAG
801      FORMAT(" OPTION ",I1," SELECTED")
         GOTO 20
C
C*   CROSS ? AUTO-CORRELATION PROCESSOR
9        CONTINUE
C        CALL SCRSS
         GO TO 20
C*   PLOT CROSS & AUTO-CORRELOGRAMS
10       CONTINUE
C        CALL PCRSS
C        GO TO 900
35       CONTINUE
C*   SPECTRA PROCESSOR
C         CALL SFAST
         GO TO 20
36       CONTINUE
C        CALL PFAST
C        GO TO 900
C  PUNCH PHASE CARDS
31       CONTINUE
         CALL FAZPU(1)
         GO TO 20
33       CONTINUE
C  USER ROUTINE
C  ROUTINE SETS KC IN ORDER TO CONTROL ITS RETURN
C        CALL USER(JC,KC)
C        GO TO (20,900)KC
C
C   MENU
32       CONTINUE
         CALL MENU(1)
         GO TO 20
40       CONTINUE
         CALL MENU(2)
         GOTO 20
38       CONTINUE
         IZER=1
         GO TO 20
C   COME HERE FOR 'UN' COMMANDS
42       IF(JC.EQ.24)IZER=0
         IF(JC.EQ.32)ITX=5
         IF(JC.EQ.32)CLOSE 13
         IF(JC.EQ.32)CHAIN=0
         GOTO 20
C  COME HERE FOR 'PA'
44       IF(JC.NE.13)GOTO 39
```

```
      ITX=5
      WRITE(ITV,544)
544   FORMAT('REMOTE PAUSE')
      IPAUS=1
      GOTO 20
C  COME HERE TO RESUME REMOTE CONTROL
45    ITX=13
      IPAUS=0
      GOTO 20
C  HARDCOPY HERE
43    IF(JC.NE.33)GOTO 20
      CALL HDCOPY
      CALL TSEND
C  10 SEC DELAY FOR HARDCOPY UNIT
      CALL FDELY(100)
      GOTO 20
C   COME HERE FOR COMMANDS FROM REMOTE FILE
41    WRITE(ITV,440)
440   FORMAT(" INPUT REMOTE FILE NAME(10 CHARS. MAX)")
      READ(ITX,441)NNN(1)
441   FORMAT(S10)
      IF(CHAIN.EQ.1)CLOSE 13
C   NNN IS ARRAY CONTAINING ASCII NAME OF FILE
      OPEN 13,NNN
      CHAIN=1
      ITX=13
      GOTO 20
C
39    CONTINUE
C  EVENT/PASS PROCESSOR
C  CHECK ZERO-LEVEL FLAG
      IF(IC.EQ.14)IZER=0
      IF(JC.EQ.24)IZER=1
      CALL EVPRC(IC,JC)
      GO TO 20
90    CONTINUE
C  OUTPUT ZZ CARD AT END OF PHASE LIST
      CALL FAZPU(2)
      WRITE(ITV,990)
990   FORMAT(10H EXIT ISDS)
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- MONDAY, SEPTEMBER 4, 1978 2:59:06 PM

MAKNAM.FR

```
      SUBROUTINE XMAKNAM(PREFIX,POSTFIX,NAME)
      OVERLAY OMAKNAM
      DIMENSION NAME(5),II(6)
      INTEGER PREFIX,POSTFIX
C  PREFIX CONTAINS A2 NAME PREFIX
C  POSTFIX IS AN INTEGER NUMBER TO BE APPENDED
C  IN ASCII FORM TO PREFIX
C  NAME IS THE RESULTS OF THE CONCATENATION
      N=POSTFIX
      NAME(1)=PREFIX
      DO 1 J=1,5
      NN=N/10
      II(J)=N-NN*10
      N=NN
1     CONTINUE
      II(6)=-48
      DO 2 J=1,3
      JJ=J+1
      JL=5-2*(J-1)
      JR=JL-1
      IF(JR.EQ.0)JR=6
      NAME(JJ)=((II(JL)+60K)*256)+(II(JR)+60K)
2     CONTINUE
      NAME(5)=0
      RETURN
      END
```

ECLIPSE FORTRAN 5, VERSION 5.10 -- FRIDAY, JUNE 30, 1978 2:03:52 PM

UTEP.FR

```
      SUBROUTINE XUTEP(IDNUM,NCP,NRDP,IERC)
      OVERLAY OUTEP
      DIMENSION IH(256),IB(256),LAWG(4,64)
C  MODIFIED 6-30-78 BY P.STEVENSON (L.U.24-L.U.15)
      EQUIVALENCE (IB(1),LAWG(1,1))
      DATA NEVPB/64/,IDIDGE/3/,MAXER/4/
      DATA KOPY/7/,IDROP/8/,INQ/9/
      DATA MINLCP/0/,MAXLCP/31/
      IERC=MAXER
      IF (NCP .LT. MINLCP .OR. NCP .GT. MAXLCP) RETURN
      IF (IDNUM .LT. 0) RETURN
C ERROR RETURN CODES
C    4=> NEW PROCESS ID IS OUT OF BOUNDS
C    3=> ID IS NEGATIVE
C    2=> ID IS NOT IN LOG
C    1=> TRIED TO READ A DISK RECORD OUTSIDE OF EPLOG
C    0=> OK
C
      IERC=IERC-1
      NREK=1
      READ(15,REC=NREK) IH
      IFURST=IH(1)
      ILAST=IH(2)
      IF (IDNUM .LT. IFURST .OR. IDNUM .GT. ILAST) RETURN
      IERC=IERC-1
      IOFF=IDNUM-IFURST
      NREK=(IOFF/NEVPB)+2
      IF (NREK .GT. IH(5)) RETURN
      IERC=IERC-1
      READ(15,REC=NREK) IB
      NK=NREK-2
      INDEKS=IOFF-(NK*NEVPB)+1
      J=INDEKS
      ILBACT=16
      IRBACT=16
      INACT=1
      ILBLCP=7
      IRBLCP=11
      ILBDID=12
      IRBDID=15
      IF(NCP .EQ. IDROP .OR. NCP .EQ. KOPY .OR. NCP .EQ. INQ) INACT=0
      IF (NCP .EQ. IDIDGE) FLD(LAWG(3,J),ILBDID,IRBDID)=NRDP
      FLD(LAWG(3,J),ILBLCP,IRBLCP)=NCP
      FLD(LAWG(3,J),ILBACT,IRBACT)=INACT
      WRITE(15,REC=NREK) IB
      IH(3)=IH(3)+INACT
      INACT=INACT-1
      IH(4)=IH(4)+INACT
      IH(225+NCP)=IH(225+NCP)+1
      NREK=1
      WRITE(15,REC=NREK) IH
      IERC=0
      RETURN
      END
      SUBROUTINE DIRC1
C
C  INTERPROCESS COMMUNICATIONS DIRECTORY INITIALIZER
C
C  P.A. MARSHALL
C  SEPTEMBER 1976
C----------------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C   MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
C
C *** NUMBER OF POSSIBLE PASSES
      PARAMETER NPASS=12
CC *** DIRECTORY RECORD SIZE
CC ** RECSIZ=52+NPASS+47*NPASS
      PARAMETER RECSIZ = 628
C
C *** RECORD 0:
C
      PARAMETER ENX = 1           ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1       ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1       ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1       ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1       ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1       ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+19       ; MASTER POINTER LIST
CC ** EPL=MPL+NPASS*32-1
      PARAMETER EPL = MPL+383   ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1           ; EVENT NO.
      PARAMETER ENA = ENO+1       ; EVENT NAME
      PARAMETER EIB = ENA+20      ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1       ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1       ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2       ; DURATION (SECS)
      PARAMETER EYR = DRA+2       ; YEAR
      PARAMETER EMO = EYR+1       ; MONTH
      PARAMETER EDA = EMO+1       ; DAY
      PARAMETER EHR = EDA+1       ; HOUR
      PARAMETER EMI = EHR+1       ; MINUTE
      PARAMETER ESE = EMI+1       ; SECONDS
      PARAMETER TCR = ESE+2       ; TIME CORRECTION (SECS)
      PARAMETER UYR = ESE+4       ; UNIQUE ID:YEAR
      PARAMETER UID = ESE+5       ; UNIQUE ID:SERIAL NUMBER
      PARAMETER EP1 = ESE+21      ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1       ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1       ; EVENT, PASS 3
      PARAMETER EP4 = EP3+1       ; EVENT, PASS 4
C
      PARAMETER ETM = EYR         ; EVENT TIME
      PARAMETER EPP = EP1-1       ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0           ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1       ; PASS NO.
      PARAMETER PIB = PNO+1         ! INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1         ! NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1         ! FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1         ! LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1         ! POINTER LIST
      PARAMETER PSW = PLS+32        ! TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1         ! FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1         ! A/D ERROR STATUS
      PARAMETER HEX = PAD+1         ! HEX COUNTER FOR THIS PASS
C****
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1             ! EVENT DIRECTORY
      PARAMETER IML = 2             ! MASTERLIST DIRECTORY (CARD IMAGES)
C****
      IMPLICIT INTEGER (A-Z)
C  MODIFIED 8-5-77 P.STEVENSON
      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
C
C
C --- OPEN DIRECTORY FILES
      OPEN IED,'DIRECTORY',LEN=2*RECSIZ
      OPEN IML,'MASTERLIST',LEN=80
C
C --- READ DIRECTORY RECORD ZERO
      READ(IED,REC=1) REC
C
C --- INITIALIZE DATABASE PARAMETERS
      NEVENT = REC(ENX)-1
      OFFSET = REC(DOF)
C
      RETURN
      END
```

```
        SUBROUTINE XENAME(EVTNAME,EVTNUM)
       OVERLAY OENAME
C
C   EDITED 10/3/77 BY P.R.STEVENSON TO SAVE STACK SPACE
C
C  *******************************************************************
C
C      HERE IS A ROUTINE TO READ AN EVENT NAME FROM A FILE CALLED "EVDIR"
C      WHICH RESIDES ON REMOVABLE DISK DP0. THIS IS NOT A STAND-ALONE
C      ROUTINE. SUBROUTINE ENAME SHOULD BE USED IN CONJUNCTION WITH 2
C      OTHER SUBROUTINES, ONE OF WHICH MUST BE CALLED PRIOR TO CALLING
C      SUBROUTINE ENAME. SUBROUTINE ENAME READS  RECORDS FROM
C      THE EVDIR FILE, BUT ASSUMES THIS FILE HAS BEEN PREVIOUSLY OPENED
C      WITH A CALL TO SUBROUTINE "DIRC1".
C
C      NOTE: IN ORDER TO USE SUBROUTINE DIRC1, THE USER MUST HAVE CRE-
C            ATED A LINK TO FILE DP0:EVDIR USING THE LINK NAME
C            "DIRECTORY".
C
C      FORMAT OF CALL:  CALL ENAME(NAME,K)
C
C      WHERE K IS (UPON CALL) THE INTEGER NUMBER OF THE EVENT FOR WHICH
C      THE NAME (IN ASCII) IS REQUESTED,
C
C      AND NAME IS A 20 WORD INTEGER ARRAY WHICH WILL BE STUFFED WITH
C      THE 40 CHARACTER EVENT NAME, CORRESPONDING TO THE EVENT NUMBER.
C
C      IF THE EVENT NUMBER PASSED IN ARGUMENT "K" IS LESS THAN 1 OR
C      GREATER THAN THE NUMBER OF EVENTS CONTAINED WITHIN THE EVDIR
C      FILE, SUBROUTINE ENAME RETURNS TO THE CALLING ROUTINE WITH
C      ASTERISKS IN THE 20 WORD "NAME" ARRAY.
C
C      IN MOST INSTANCES THE USER WILL ALSO BE USING THE DATA FILE
C      ASSOCIATED WITH A PARTICULAR EVENT. THIS REQUIRES A CALL
C      TO SUBROUTINE "MCAM1", WHICH OPENS FILE "EVDAT" (ALSO ON DP0).
C
C      NOTE: IN NON-TASKING ENVIRONMENT, USER MUST LINK DP0:EVDAT
C            USING LINK NAME "WHOLE1".
C
C      THE RECOMMENDED CALLING SEQUENCE IS:
C
C           CALL MCAM1
C           CALL DIRC1
C               .
C               .
C               .
C               .
C               .
C           CALL ENAME(NAME,K)
C               .
C               .
C
C      NOTE: SUBROUTINE ENAME MAY BE CALLED SEVERAL TIMES DURING EXECU-
C            TION OF A PROGRAM (OR ANOTHER SUBROUTINE). HOWEVER, SUB-
C            ROUTINE MCAM1 AND DIRC1 SHOULD BE CALLED ONCE AND ONLY
C            ONCE DURING THE EXECUTION OF A PROGRAM.
C
C  *******************************************************************
C
```

```
CC ** MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
      PARAMETER NPASS=12
        PARAMETER REKSIZE = 52+NPASS+47*NPASS
        PARAMETER IED = 1
        PARAMETER ENO = 1
        PARAMETER ENA = ENO+1
        IMPLICIT INTEGER (A-Z)
        DIMENSION EVTNAME(1)
        COMMON /IOBDA/ NCHAN,SEGSZ,REMOR,LTBUF,FIRST
      COMMON /EDIRC/NEVENT,PASSN(NPASS),REC(REKSIZE)
C
C --- INITIALIZE THE EVENT NAME ARRAY. PUT ASTERISKS IN ALL
C     LOCATIONS.
C
        DO 5 K=1,20
    5   EVTNAME(K) = 2H**
C
C --- FIND OUT HOW MANY EVENTS ARE ALREADY WITHIN THE DIRECTORY
C
        MAXEVTS = NEVENT
C
C --- CHECK THE PASSED PARAMETER "EVTNUM" TO SEE IF IT IS WITHIN
C     REASONABLE BOUNDS. IF NOT, RETURN.
C
        IF (EVTNUM .LT. 1 .OR. EVTNUM .GT. MAXEVTS) GO TO 20
C
C
C --- READ THE RECORD CORRESPONDING TO THE PARTICULAR EVENT NUMBER
C
        READ(IED,REC=EVTNUM+1) REC
C
C
C --- NOW FILL THE EVENT NAME ARRAY WITH THE 40 ASCII CHARACTERS
C     STORED IN FILE "EVDIR".
C
        DO 10 K=1,20
   10   EVTNAME(K) = REC( ENO + K )
C
   20   RETURN
C
        END
      SUBROUTINE XRQEVT(LIST, PARM1)
       OVERLAY ORQEVT
C
C  EVENT REQUEST PROCESSOR
C
C  P.A. MARSHALL
C  SEPTEMBER 1976
C---------------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C   MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
C
C *** NUMBER OF POSSIBLE PASSES
      PARAMETER NPASS=12
CC *** DIRECTORY RECORD SIZE
CC ** RECSIZ=52+NPASS+47*NPASS
      PARAMETER RECSIZ = 628
C
C *** RECORD 0:
C
      PARAMETER ENX = 1            ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1        ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1        ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1        ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1        ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1        ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+19        ; MASTER POINTER LIST
CC ** EPL=MPL+NPASS*32-1
      PARAMETER EPL = MPL+383  ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1            ; EVENT NO.
      PARAMETER ENA = ENO+1        ; EVENT NAME
      PARAMETER EIB = ENA+20       ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1        ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1        ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2        ; DURATION (SECS)
      PARAMETER EYR = DRA+2        ; YEAR
      PARAMETER EMO = EYR+1        ; MONTH
      PARAMETER EDA = EMO+1        ; DAY
      PARAMETER EHR = EDA+1        ; HOUR
      PARAMETER EMI = EHR+1        ; MINUTE
      PARAMETER ESE = EMI+1        ; SECONDS
      PARAMETER TCR = ESE+2        ; TIME CORRECTION (SECS)
      PARAMETER UYR = ESE+4        ; UNIQUE ID:YEAR
      PARAMETER UID = ESE+5        ; UNIQUE ID:SERIAL NUMBER
      PARAMETER EP1 = ESE+21       ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1        ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1        ; EVENT, PASS 3
      PARAMETER EP4 = EP3+1        ; EVENT, PASS 4
C
      PARAMETER ETM = EYR          ; EVENT TIME
      PARAMETER EPP = EP1-1        ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0            ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1          ! PASS NO.
      PARAMETER PIB = PNO+1          ! INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1          ! NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1          ! FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1          ! LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1          ! POINTER LIST
      PARAMETER PSW = PLS+32         ! TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1          ! FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1          ! A/D ERROR STATUS
      PARAMETER HEX = PAD+1          ! HEX COUNTER FOR THIS PASS
C****
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1              ! EVENT DIRECTORY
      PARAMETER IML = 2              ! MASTERLIST DIRECTORY (CARD IMAGES)
C****
C  MODIFIED 9-24-77 P.STEVENSON
C**** REQUEST LIST OFFSETS
C
      PARAMETER LEN = 1              ! EVENT NO.
      PARAMETER LET = LEN+1          ! EVENT TIME
      PARAMETER LED = LET+7          ! DURATION
      PARAMETER LDT = LED+2          ! DELTA-T
      PARAMETER LTC = LDT+2          ! TIME CORRECTION
      PARAMETER LUY = LTC+2          ! UNIQUE ID:YEAR
      PARAMETER LUI = LUY+1          ! UNIQUE ID:SERIAL NO.
      PARAMETER LNP = LUI+1          ! NUMBER OF PASSES
      PARAMETER LM1 = LNP+1          ! MUX, PASS 1
      PARAMETER LMN = LM1+NPASS-1    ! MUX, PASS N
C
      PARAMETER LMP = LM1-1          ! MUX FINDER
      PARAMETER LSZ = LMN-LEN+1      ! LIST SIZE
C****
      IMPLICIT INTEGER (A-Z)
      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      DIMENSION LIST(LSZ)
C
C --- CLEAR LIST
      DO 1 J=1,LSZ
    1 LIST(J) = 0
C
C --- SET EVENT, CHECK FOR VALIDITY
      EVENT = PARM1
      IF (EVENT .LE. 0) RETURN
C
C --- READ IN DIRECTORY RECORD, CHECK VALIDITY
      READ(IED,REC=EVENT+1,END=99) REC
      IF (REC(ENO) .EQ. 0) RETURN
C
C --- STUFF PARAMETERS FROM DIRECTORY RECORD INTO ROEQUEST LIST
      LIST(LEN) = REC(ENO)
      LIST(LET) = REC(EYR)
      LIST(LET+1) = REC(EMO)
      LIST(LET+2) = REC(EDA)
      LIST(LET+3) = REC(EHR)
      LIST(LET+4) = REC(EMI)
      LIST(LET+5) = REC(ESE)
      LIST(LET+6) = REC(ESE+1)
      LIST(LED) = REC(DRA)
      LIST(LED+1) = REC(DRA+1)
      LIST(LDT) = REC(DLT)
      LIST(LDT+1) = REC(DLT+1)
      LIST(LTC) = REC(TCR)
      LIST(LTC+1)= REC(TCR+1)
      LIST(LUY) = REC(UYR)
      LIST(LUI) = REC(UID)
C
C --- FIND WHICH PASSES EXIST AND INTERPRET AS A PASS COUNT
      K = 0
      DO 2 J=1,NPASS
C --- CLEAR PASS MEMORY
      PASSN(J) = 0
C --- POINT TO PASS. SEE IF IT EXISTS
      PNT = REC(EPP+J)
      NCHAN = REC(PNT+PLC)-REC(PNT+PFC)+1
      IF (NCHAN .EQ. 1 .AND. REC(PNT+PFC) .EQ. 0) GO TO 2
C
C --- HAVE PASS. COUNT PASS, SAVE PASS NO., STUFF MUX NO.
      K = K+1
      PASSN(K) = J
      LIST(LMP+K) = NCHAN
    2 CONTINUE
C
C --- STUFF PASS COUNT
      LIST(LNP) = K
99    RETURN
      END
      SUBROUTINE XRQPAZ(LIST,FCA,LCA,PARM1,PARM2)
       OVERLAY ORQPAZ
C
C  EVENT/PASS REQUEST PROCESSOR
C
C  P.A. MARSHALL
C  SEPTSMHER 1976
C------------------------------------------------------------------------
C
      INCLUDE 'PSDIR'
C**** EVENT DIRECTORY OFFSETS
C   MODIFIED FOR 12 PASSES BY P.STEVENSON (1-13-78)
C
C *** NUMBER OF POSSIBLE PASSES
      PARAMETER NPASS=12
CC *** DIRECTORY RECORD SIZE
CC ** RECSIZ=52+NPASS+47*NPASS
      PARAMETER RECSIZ = 628
C
C *** RECORD 0:
C
      PARAMETER ENX = 1           ; NEXT EVENT (LAST+1)
      PARAMETER DML = ENX+1       ; NO. OF MASTERLIST ENTRIES
      PARAMETER DOF = DML+1       ; DATABASE FILE OFFSET
      PARAMETER DSZ = DOF+1       ; FILE SIZE IN SECTORS
      PARAMETER DNX = DSZ+1       ; NEXT BLOCK NO.
      PARAMETER CTE = DNX+1       ; CURRENT EVENT SAVE AREA
      PARAMETER MPL = CTE+19       ; MASTER POINTER LIST
CC ** EPL=MPL+NPASS*32-1
      PARAMETER EPL = MPL+383  ; END OF POINTER LIST
C
C *** RECORD N:
C
      PARAMETER ENO = 1           ; EVENT NO.
      PARAMETER ENA = ENO+1       ; EVENT NAME
      PARAMETER EIB = ENA+20      ; INITIAL BLOCK NO.
      PARAMETER ENB = EIB+1       ; TOTAL NUMBER OF BLOCKS
      PARAMETER DLT = ENB+1       ; DELTA-T (SECS)
      PARAMETER DRA = DLT+2       ; DURATION (SECS)
      PARAMETER EYR = DRA+2       ; YEAR
      PARAMETER EMO = EYR+1       ; MONTH
      PARAMETER EDA = EMO+1       ; DAY
      PARAMETER EHR = EDA+1       ; HOUR
      PARAMETER EMI = EHR+1       ; MINUTE
      PARAMETER ESE = EMI+1       ; SECONDS
      PARAMETER TCR = ESE+2       ; TIME CORRECTION (SECS)
      PARAMETER UYR = ESE+4       ; UNIQUE ID:YEAR
      PARAMETER UID = ESE+5       ; UNIQUE ID:SERIAL NUMBER
      PARAMETER EP1 = ESE+21      ; EVENT, PASS 1
      PARAMETER EP2 = EP1+1       ; EVENT, PASS 2
      PARAMETER EP3 = EP2+1       ; EVENT, PASS 3
      PARAMETER EP4 = EP3+1       ; EVENT, PASS 4
C
      PARAMETER ETM = EYR         ; EVENT TIME
      PARAMETER EPP = EP1-1       ; EVENT, PASS POINTER
C
C *** PASS N:
C
      PARAMETER ENP = 0           ; ASSOCIATED EVENT NO.
      PARAMETER PNO = ENP+1          ! PASS NO.
      PARAMETER PIB = PNO+1          ! INITIAL BLOCK NO.
      PARAMETER PNB = PIB+1          ! NUMBER OF BLOCKS
      PARAMETER PFC = PNB+1          ! FCN  (FIRST CHANNEL NO.)
      PARAMETER PLC = PFC+1          ! LCN (LAST CHANNEL NO.)
      PARAMETER PLS = PLC+1          ! POINTER LIST
      PARAMETER PSW = PLS+32         ! TRACK SELECT SWITCH
      PARAMETER PFS = PSW+1          ! FILTER SELECT SWITCH
      PARAMETER PAD = PFS+1          ! A/D ERROR STATUS
      PARAMETER HEX = PAD+1          ! HEX COUNTER FOR THIS PASS
C****
      INCLUDE 'PSFFNDIR'
C**** FORTRAN FILE NOS. FOR DIRECTORIES
      PARAMETER IED = 1              ! EVENT DIRECTORY
      PARAMETER IML = 2              ! MASTERLIST DIRECTORY (CARD IMAGES)
C****
C**** REQUEST LIST PARAMETERS
      PARAMETER NSTAR = 2            ! WIDTH OF LIST ENTRY (IN WORDS)
      PARAMETER LSZ = 32*NSTAR       ! LIST SIZE
      PARAMETER LST = 1              ! START OF LIST
C****
      IMPLICIT INTEGER (A-Z)
C  MODIFIED 8-5-77 P.STEVENSON
      COMMON /EDIRC/ NEVENT, PASSN(NPASS), REC(RECSIZ)
      COMMON /EVPAZ/ EVENT, PASS, CHANNL
      COMMON /IOPDA/ IDUM1(3), LSEC, EPISEC, OFFSET, IDUM2(2)
      COMMON /IOBDA/ NCHAN, IDUM3(4)
      DIMENSION LIST(LSZ), STAREC(40)
C
C --- CLEAR LIST
      DO 1 J=1,LSZ
    1 LIST(J) = 0
C
C --- CHECK EVENT AND PASS FOR VALIUITY
      IF (PARM1 .NE. EVENT) RETURN
      PASS = PARM2
      IF (PASS .LT. 1 .OR. PASS .GT. NPASS) RETURN
      IF (PASSN(PASS) .EQ. 0) RETURN
C --- POINT TO PASS
      PNT = REC(EPP+PASSN(PASS))
C
C --- SET "MCAM" PARAMETERS
      EPISEC = REC(PNT+PIB)-OFFSET
      LSEC = EPISEC+REC(PNT+PNB)-1
      FCN = REC(PNT+PFC)
      LCN = REC(PNT+PLC)
      NCHAN = LCN-FCN+1
C
C   LOAD FCA AND LCA
      FCA=FCN
      LCA=LCN
C
C --- SET POINTER LIST, FETCH STATION LIST DATA
      M = PNT+PLS-1
      K = LST
      DO 2 J=FCN,LCN
      STAREC(1) = 0
C --- GET STATION RECORD FROM MASTERLIST
      MP = REC(M+J)
      IF (MP .LE. 0) GO TO 21
      READ(IML,REC=MP) STAREC
   21 CONTINUE
C  -** MOD (4/22/77) BY P.R.S. !IJK FORMERLY 2
```

```
C   IJK=1 FOR FORMAT 2, SET IJK=2 FOR FORMAT 1
      IJK=1
C --- OR SHOW AS 'IRIG' CHANNEL
      IF (MP .EQ. -1) CALL FBLM(STAREC(IJK), 'IRIG', 2)
C --- OR 'WWVB' CHANNEL
      IF (MP .EQ. -2) CALL FBLM(STAREC(IJK), 'WWVB', 2)
C --- OR CLEAR RECORD IF "UNDEFINED" CHANNEL
      IF (MP .EQ. 0) CALL FBLM(STAREC(2), STAREC(1), 39)
C     :
C --- MOVE DATA INTO REQUEST LIST
      CALL FBLM(LIST(K), STAREC(IJK), 2)
    2 K = K+NSTAR
C
      RETURN
      END
```