# Digital Ultrasonics Signal Processing: Flaw Data Post Processing Use and Description

Final Report

September 30, 1981

Prepared by:

Victor E. Buel
Intergraph Corporation
One Madison Industrial Park
Huntsville, Alabama 35807



INTERGRAPH
CORPORATION

## Table of Contents

## List of Figures

# 1. INTRODUCTION

This report documents the work performed on the Digital Ultrasonics Signal Processing System during the previous contract year. This report contains the documentation for the software generated during the year.

The use of ultrasonics evaluation techniques in determining the existance of flaws in materials has been recognized by NASA to be a useful tool in non-destructive testing. The potential exist for using ultrasonics to not only detect flaws but to identify and classify them as well.

The hardware configuration illustrated in Figure 1-1 consists of two main units. A DEC LSI-11 processor running under the RT-11 single job, version 2C-02 operating system, controls the scanner hardware and the Ultrasonic unit. A DEC PDP-11/45 processor also running under the RT-11, version 2C-02, operating system, stores, processes and displays the flaw data.

The software developed for the Ultrasonics Evaluation System, shown in Figure 1-2, is divided into two categories: transducer characterization and flaw classification. Each category is divided further into two functional tasks: a data acquisition task and a post processor task. The transducer characterization data acquisition task (PACT) and the transducer characterization post processing task (PEARL) were developed under previous contract efforts.[1] The flaw characterization data acquisition task, also developed under a previous task,[2] collects data, compresses it, and writes it to a disk file. The data is then processed by the flaw classification post processing task which was developed under this contract.

This document will describe the use and construction of a flaw data post processor.

---

[1] PACT/PEARL Program Documentation, Report No. 80-014, February 18, 1980.

[2] Final Report, Digital Ultrasonics Signal Processing: Primary Ultrasonics Task and Transducer Characterization Use and Detailed Description, Report No. 79-070, November 1, 1979.

## HARDWARE CONFIGURATION

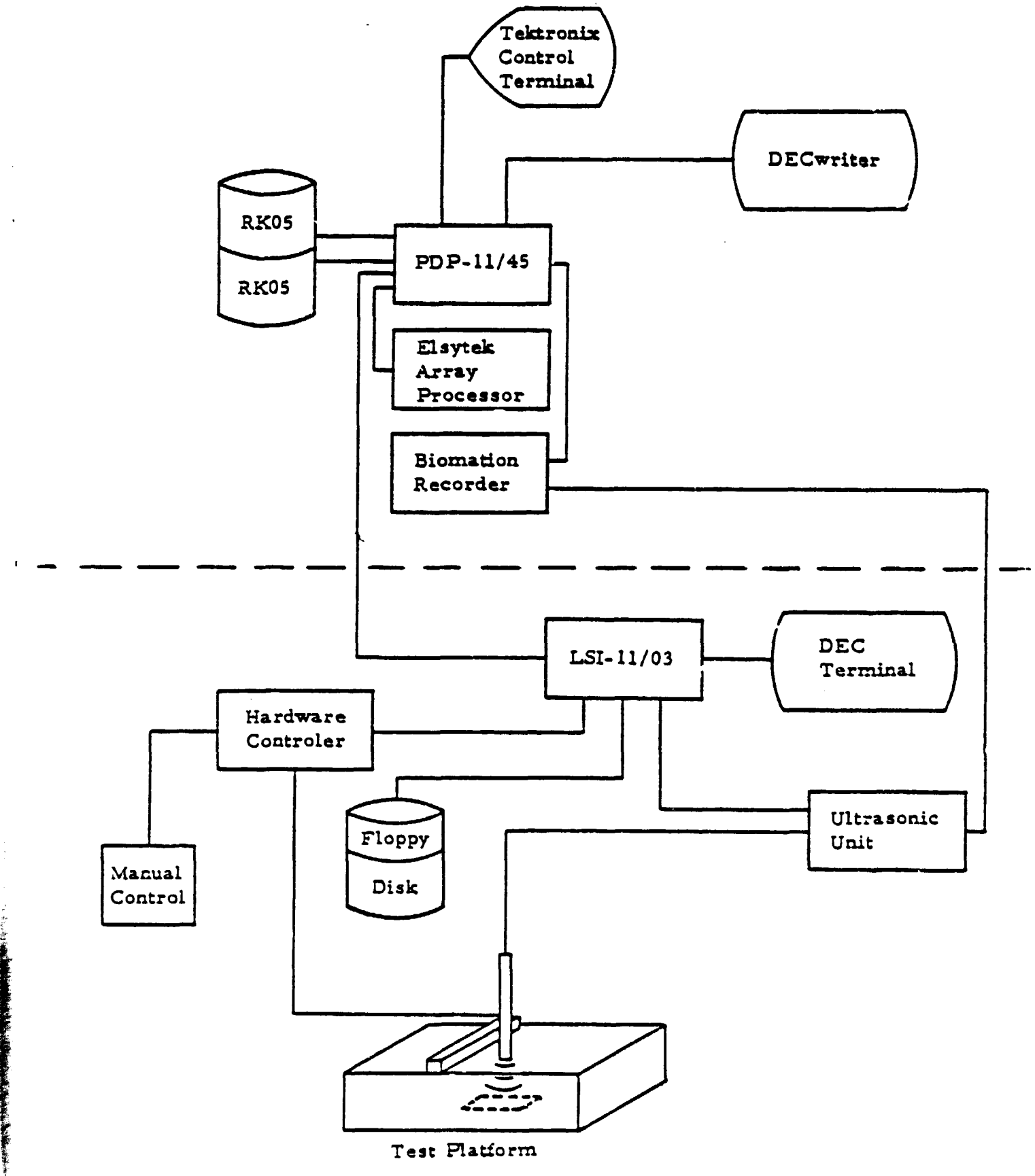


Figure 1-1

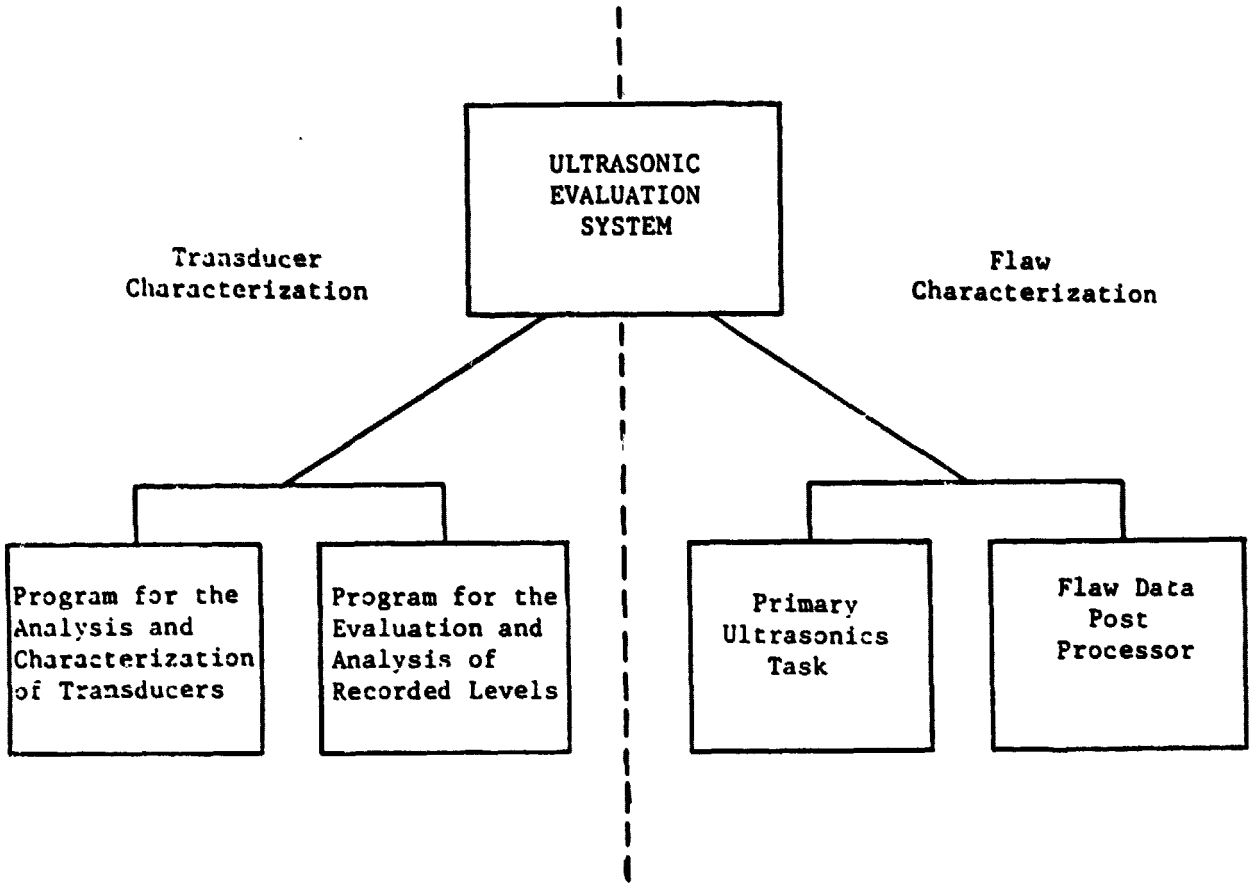


Figure 1-2 Ultrasonic Evaluation System Task Organization

## 2. OVERVIEW OF THE FLAW CHARACTERIZATION TASK

The Ultrasonics Evaluation System is a modular system composed of two sets of tasks which interprets the flaw data and allows compensation of the data due to transducer characteristics. The transducer characterization task allows the user to determine the optimal flaw scanning configuration. The flaw characterization task acquires, in a production mode, the flaw data and processes it in a post processing mode.

The real time data acquisition task for flaw data is called PUT (Primary Ultrasonics Task). PUT collects the flaw data, compresses it, and writes it to a specified disk file. The records written are of varying lengths for each flaw point detected. Figure 2-1 is a graphical representation of a flaw point recorded by the PUT task as written to a disk file. The function of the Flaw Data Post Processor (FDPP) is to process this data into meaningful information. The characterization may be accomplished by transforming the time domain data into frequency domain data. The Fourier transform of time domain flaw data may then be processed to produce information leading to flaw characterization. The problem becomes complex when one realizes that many data points compose a flaw data file; the composite of these Fourier transforms is shown in Figure 2-2.

The problem for the FDPP task is to process the composite Fourier transforms into a characteristic Fourier transform, as displayed in Figure 2-3, characterize the flaw using the principle that different flaws yield different characteristic Fourier transforms, and present a three dimensional display of the flaw.

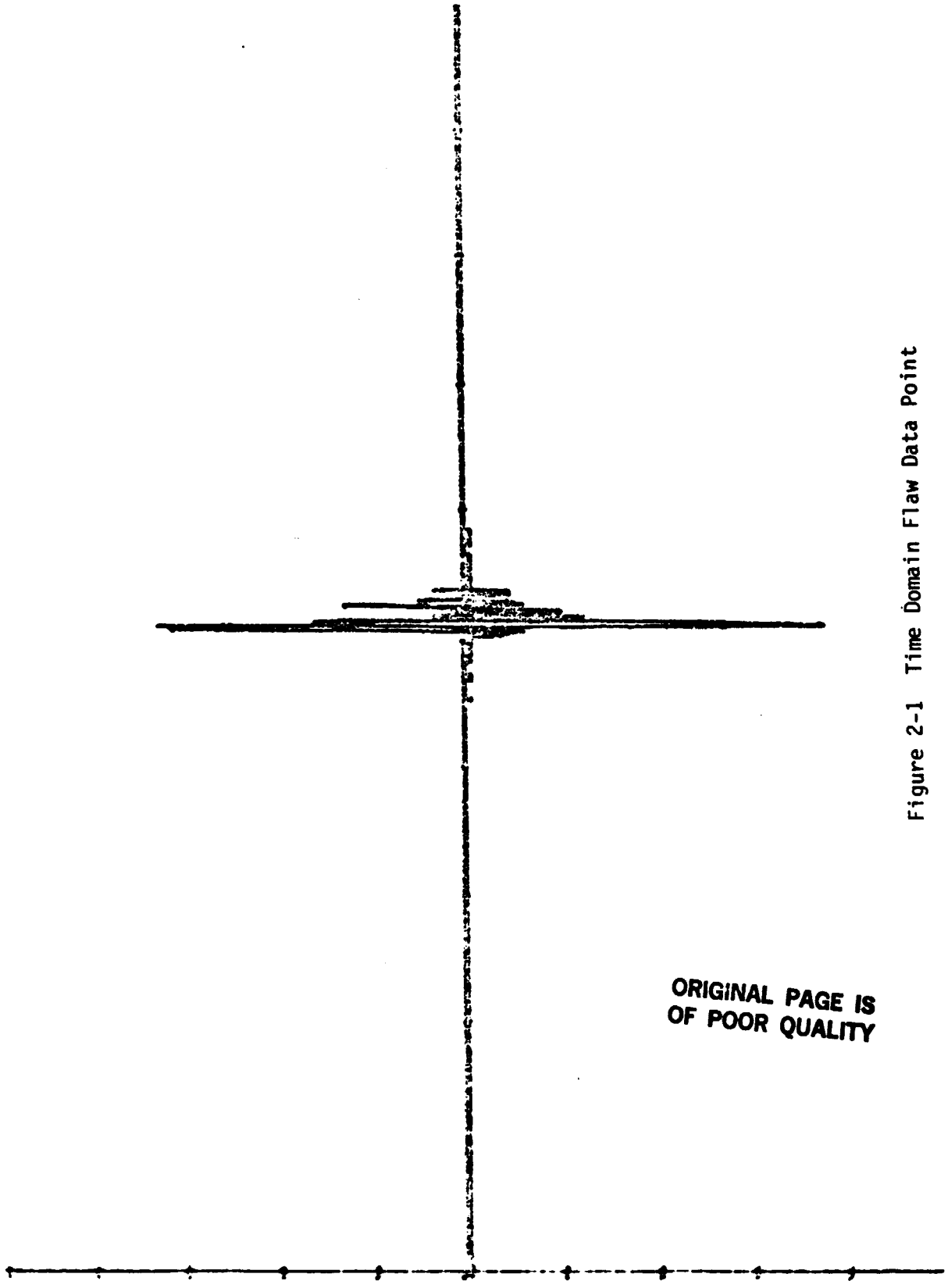

Figure 2-1 Time Domain Flaw Data Point

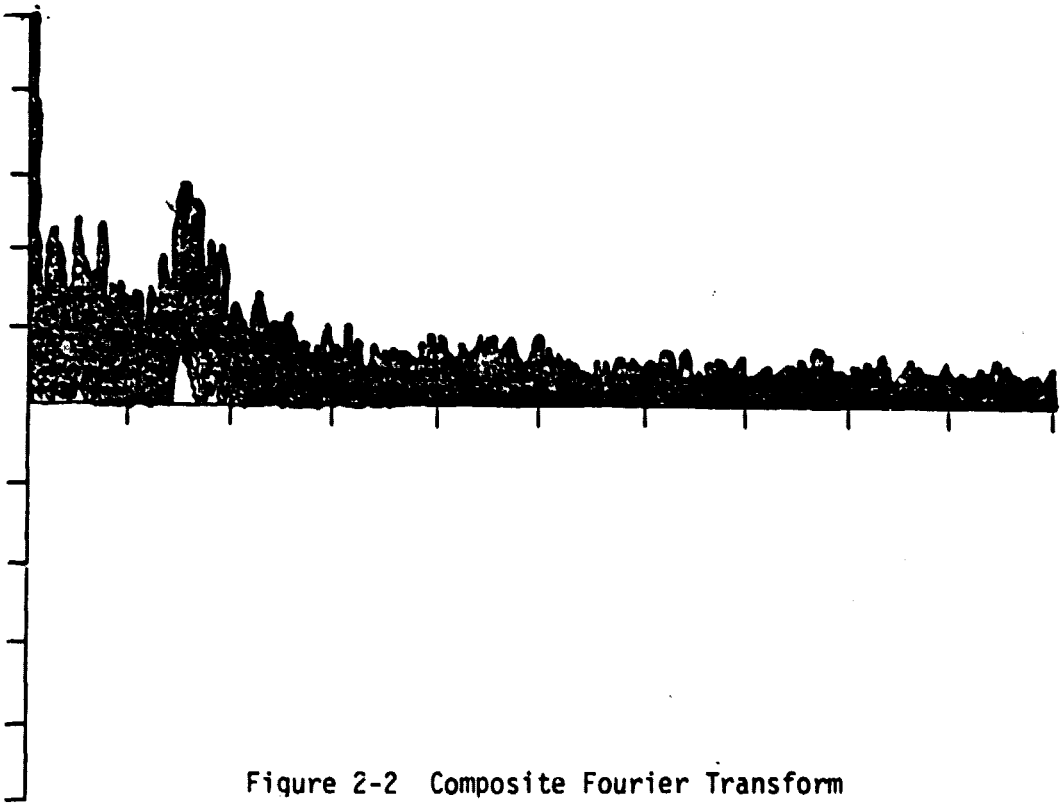

Figure 2-2 Composite Fourier Transform

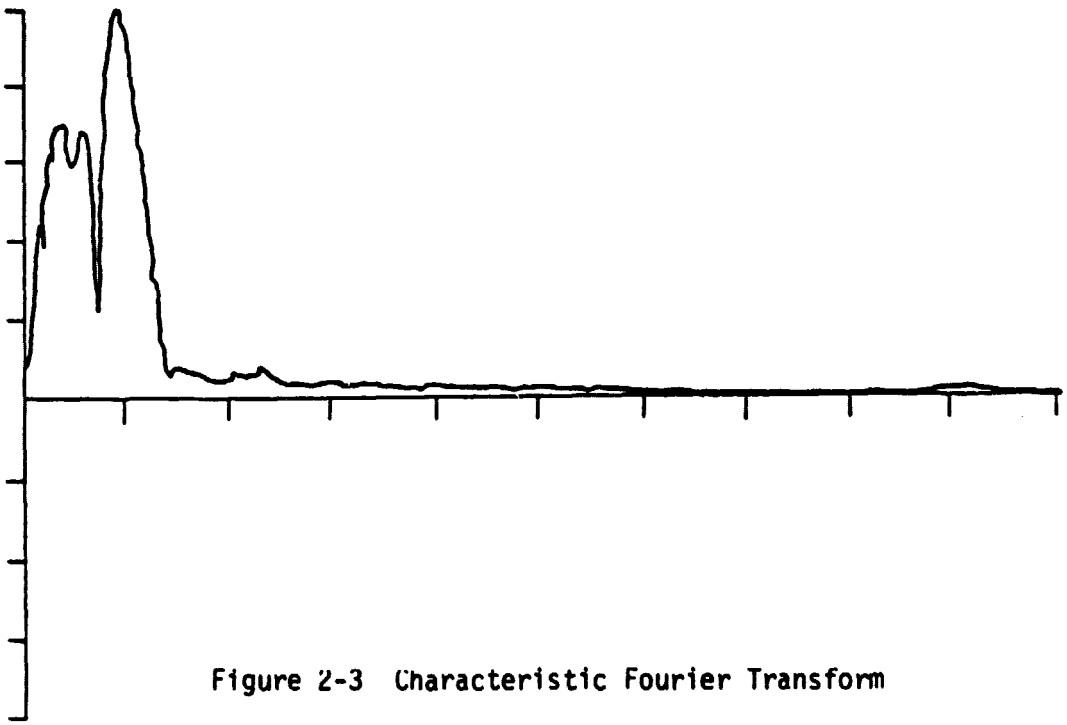

Figure 2-3 Characteristic Fourier Transform

# 3. PROCESSOR CONFIGURATION

The structure chart depicted in Figure 3-1 graphically illustrates the relationship of the modules within the FDPP task. Each of the blocks in the chart represents a module within the task. Modules called more than once are represented as rounded figures in which the module reappears, and calls to library routines are represented as blocks with double vertical lines. The structure chart is read from top to bottom and left to right; modules which appear on top of the chart are executed in sequence before those below and modules are executed in order from left to right. Curved lines which encompass a group of modules indicate that the group of modules may be called iteratively within the calling module. A functional description of each of the modules depicted in figure 3-1 is provided in Appendix A and a source listing is provided in Appendix B.

## 3.1 Processor Operations

### 3.1.1 Initial Operations

The FDPP task is initiated via the RT-11 Operating System RUN Command which is entered at the operator's console. Once initiated, the task sets up the interrupt vectors allowing unsolicited input from the Tektronix 4002A graphic display terminal. Once this function is completed, the FDPP task can receive input via the keyboard on the Tekrtonix 4002A keyboard. The task then responds by quering for a data file from the operator, then attaching and opening the specified data file for reading.

### 3.1.2 Normal Operations

Once the data file has been attached, the FDPP task reads the file and processes it for classification and characterization of the flaw data. It also presents to the operator tutorial displays and messages to aid the operator in selecting the functions to be executed by the task and displays the results of the processing of the flaw data files according to operator's selected options.

The normal execution cycle for the FDPP task is as follows:

- Clear the screen and present the user with the next display.
- Wait for the operator response.
- Collect and echo characters as they are input.

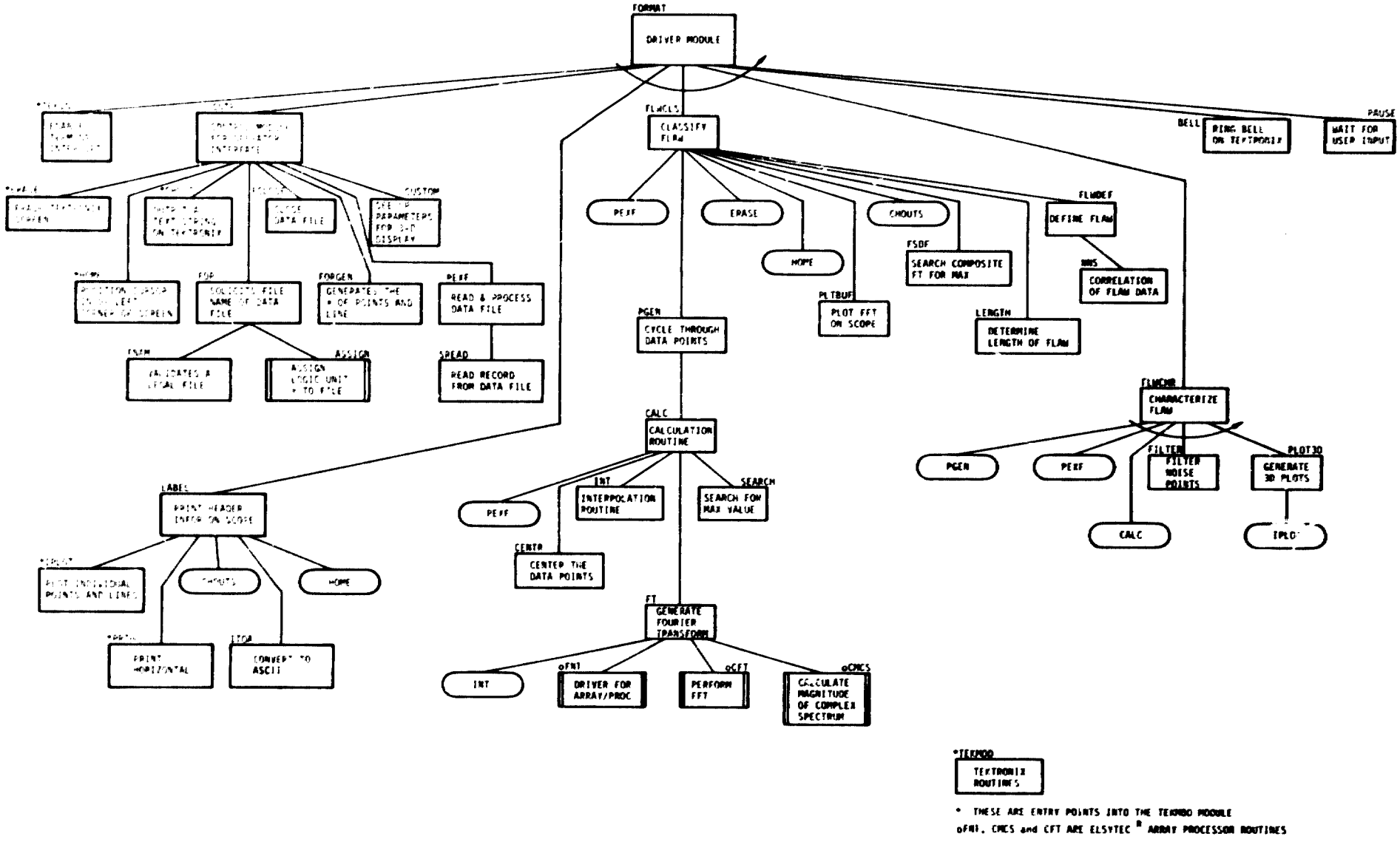


Figure 3-1. FDPP Structure Chart

- Decode and process the input commands once the operator enters them.

- Read and process the flaw data files specified by the operator.

- Present the results of the processing of the flaw data files in the form of plots and displays.

### 3.1.3 Termination Operation

The FDPP task is terminated via an operator selected option entered via the Tektronix 4002A keyboard. The termination sequence closes the data file used by the task and returns control of the console to the RT-11 Operating System and exits.

### 3.1.4 I/O Communications

Once the FDPP task is initiated, I/O Communications with the operator is provided through two mediums; the console keyboard and the Tektronix 4002A display terminal keyboard. The consoled keyboard is supported by library service routines.

The FDPP task performs all I/O communications between the operator and the Tektronix display keyboard through the use of a Tektronix I/O processing service routine (TEKMOD) which is a module of multiple entry points, each with specific I/O functions located in the FDPP task. (Refer to Appendix A for a functional description.) On initiating execution of FDPP the CSR's and interrupt vectors are set up for subsequent I/O through the graphic display terminal. Functions performed by the FDPP processor include input of ASCII characters from the Tektronix keyboard, output of messages, displays, plots and tutorial information to the Tektronix display.

The FDPP task provides the capability to control and communicate with a DEC RK05 disk drive. Files written by the PUT data acquisition task are read by the FDPP task and expanded for processing.

## 4. USER'S GUIDE

The Keyboard Monitor (KMON) is the routine within the RT-11 Operating System which provides the interface between the user and the operating system. The user initiates the Flaw Data Post Processor (FDPP) via using the RUN command as follows:

R(UN) FDPP

the characters enclosed within parenthesis are optional. Once the command is entered, the FDPP task is initiated and the initial display is presented.

### 4.1 Data File Selection

The first display the user will see once the FDPP task is initiated is a query for a flaw data file to be read and processed. The query displayed on the Tektronix screen is as follows:

ENTER DESIRED FILENAME:
DK1:XXXXXX.1ND

The user's response is a flaw data file name generated by the PUT data acquisition task. The response input by the user is a filename of up to six characters followed by a "." (period) and a three character extension identification. The "DK1:" presented by the query indicates that all data files are written to the alternate disk unit denoted by DK1: on conventional RT11 notation and does not need to be echoed in the user response. The file name can be from one to six alphabetic characters followed by a period. The extension identification is three alphabetic characters and may denote file type ('.DAT' is the usual denotation for data files in RT-11) or it may be used for an extension of the file name.

### 4.2 Plot Viewing Angles

The display depicted in Figure 4-1 illustrates the options available to the user for viewing a three dimensional representation of the flaw. The user makes the selection by keying in a letter designation which controls the orientation of the three dimensional display of the flaw. There are four standard orientations of the flaw display which the user may select or the user may choose to set up his own viewing parameters. The first four views differ only in the angle they are viewed. All other scaling parameters are the same. If the user selects A (The Orthonormal View) the view is rotated counter clockwise about the y axis 45 degrees and then rotated counter clockwise about the x axis of the resulting coordinate system 45 degrees. With the selection of B

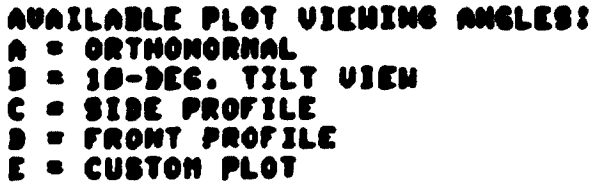


Figure 4-1 Display of Plot Viewing Options

(the 10-degree Tilt View) the rotation about the original y axis is the same as that of A but the view is only rotated 10 degrees counter clockwise about the x axis of the resulting coordinate system. If C (Side Profile) is selected, the original y axis is rotated 90 degrees only and with the selection of D (Front Profile) no rotation about the axis is performed.

## 4.3 Custom Plot Parameters

With the selection of E from the display of Available PLOT Viewing Angles depicted in Figure 4-1, the user may specify the parameters for displaying the three dimensional representation of the flaw. The user is prompted tutorially for the viewing parameters with the series of queries illustrated in Figure 4-2. The user is presented with the parameter and a format by which to enter his value. The parameters XSCALE, YSCALE and ZSCALE are the spacing in inches between the points. The parameters PHI and THETA are rotation angles in degrees which determine the orientation of the three dimensional display of the flaw. The value entered for PHI will rotate the view counter clockwise about the y axis of the original coordinate system and the value entered for THETA will rotate the coordinates about the x axis of the resulting coordinate system. The values for XREF and YREF are the coordinates in inches, relative to the plotter origin, to be used as the origin of the figure. These prompts and responses will be displayed and entered from the DECWRITER III Console.

## 4.4 FDPP Control Options

The main options of the FDPP task are depicted as follows:

OPTIONS CURRENTLY AVAILABLE:

A = CHARACTERIZE AND CLASSIFY FLAW

I = ENTER NEW FILENAME

J = EXIT THE PROGRAM

These options are displayed whenever the FDPP task completes the processing of a flaw data file and displaying the subsequent information selected by the user. The following is a description of the available options when this display is active

A = CHARACTERIZE AND CLASSIFY FLAW.

Selection of this command allows the operator to specify a different set of parameters for viewing and classifying the flaw data file. This selection will carry the user back to the options of Figure 4-1.

I = ENTER NEW FILENAME

```
ENTER XSCALE, (XX.XXXXX)

ENTER ZSCALE, (XX.XX)

ENTER YSCALE, (XX.XXXXX)

ENTER PHI, (XXX.X)

ENTER THETA, (XXX.X)

ENTER XREF, (XXX.X)

ENTER YREF, (XXX.X)
```

Figure 4-2 Prompts for Custom Plot Parameters

This option enables the user to select another flaw data file for processing. On selecting this option the user will be queried for a flaw data file name; the procedures governing the selection of the file name will be the same as described in Section 4.2.

J = EXIT THE PROGRAM

Selection of this option terminates the Flaw Data Post Processor task and returns control to the RT-11 Operating System.

## 4.5 Three Dimensional Display

Once the operator selects the data file to process, the FDPP task processes the file and generates the three dimensional representation of the flaw illustrated in Figures 4-3 and 4-4. The view presented in Figure 4-3 is a three dimensional plot where the operator has selected the A option from the display depicted in Figure 4-1. The view presented in Figure 4-4 illustrates output of the B option from the same display. The plot title information from the display is taken from the flaw data file, and operator selections made prior to generating the display. The parameters PHI and THETA are the rotations of the display in degrees described in section 4.3. The values X-LENGTH and Y-LENGTH are the lengths in milli-inches of scans in the x and y direction recorded by the data acquisition task PUT on the file. The parameters TRANSDUCER and SAMPLE are user identification symbols for the transducer used for collecting the data and the sample material respectively. Both these parameters are also recorded on the flaw data file. ALTITUDE is the height of the transducer above the sample material when the data was recorded.

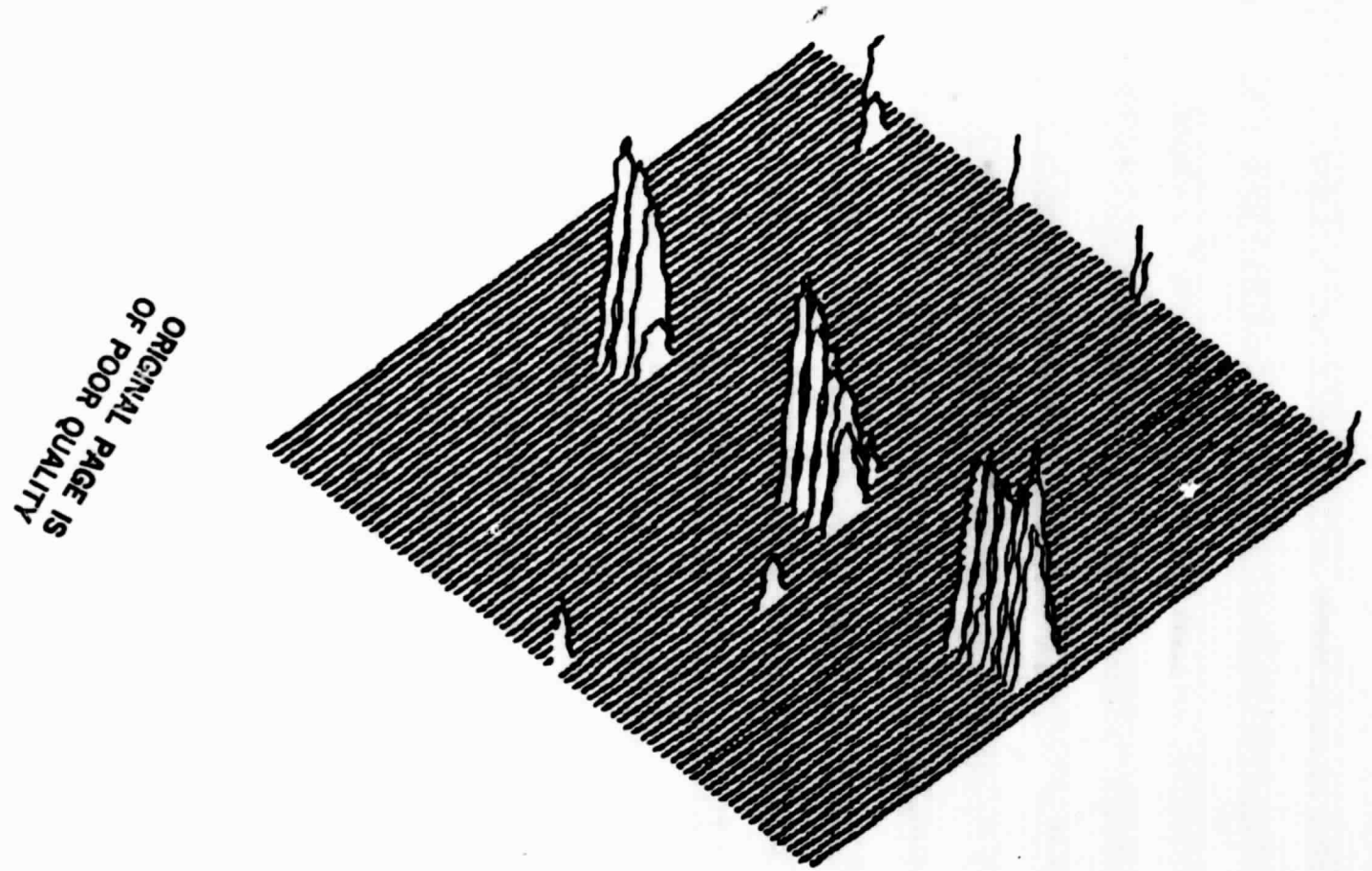


Figure 4-3 Three Dimensional Plot: Orthogonal View

TIME DOMAIN
ALTITUDE: 00000
PHI: 045
X LENGTH: 01750

TRANSDUCER: 1310
SAMPLE: .02
THETA: 010
Y LENGTH: 03549

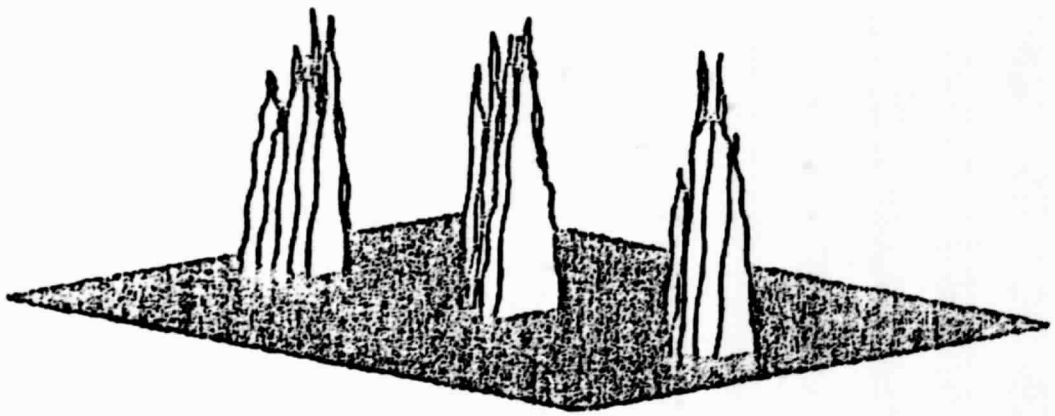

Figure 4-4 Three Dimensional Plot: 10 Degree Tilt

# APPENDIX A

# FDPP MODULE FUNCTIONAL DESCRIPTIONS

Module: CALC-Calculation routine

Language: FORTRAN
Called By: PGEN, FLWCHR
Calling Sequence: CALL CALC

External References: Common Blocks
EXDCOM
FCCOM
PEXCOM
FORIER

External References: Subroutines

| | |
|---|---|
| CENTR | Center contents of the data buffer |
| FT | Take the Fourier transform |
| INT | Interpolation routine |
| PEXF | Read data from the disc and expand it |
| SEARCH | Search the expanded data buffer for the maximum value |

Functional Description:

CALC performs the calculations on the data points in the file according to which function has been selected. The module first converts negative coordinates to positive values and validates the point (i.e., account for any deficiencies in a point location). Once a point is validated, one of two functions is performed. If flaw characterization is being performed the data buffer is searched for the maximum intensity of the signal returned from the flaw for each point.

If flaw classification is being performed, the module generates the Fourier transform of the point in the time domain. The routine cycles through all the data points and generates the Fourier transform for each and generates a composite Fourier transform for the flaw. This process amplifies trends and minimizes noise fluctuations.

Module: CENTR Shifts data in the buffer

Language: FORTRAN
Called by: CALC
Calling Sequence: CALL CENTR (IVAR, IDIFF)
IVAR - 0: additive Centering
1: Subtractive Centering
IDIFF - The number of elements to shift in the buffer

External references: Common Blocks
EXDCOM

Functional Description:

CENTR logically shifts the elements in the data buffer, representing a flaw point, in either direction.

Module: CNTRL System Control module

Language: MACRO-11
Called by: FORMAT
Calling Sequence: CALL CNTRL

External References: Common Blocks
EXDCOM
FCCOM
PEXCOM
TEKCOM

External References: Subroutines

| | |
|---|---|
| CHOUT | Output a text string on Tektronix |
| CUSTOM | Set up parameters for 3-D plots |
| ERASE | Erases Tektronix screen |
| FCLOSE | Closes data file |
| FOP | Opens data file |
| FORGEN | Generates the number of points and lines for 3-D plots |
| HOME | Homes cursor on terminal |
| PEXF | Processes the packed data file |

Functional Description:

CNTRL displays menu options, solicits and accepts input from the operator. On initial execution of the module, it initializes system parameters, reads and unpacks selected data files, and automatically classifies and characterizes the flaw. CNTRL acts as the operator interface for the FORMAT task.

Module: CUSTOM - Set customized parameters for three dimensional display.

Language: FORTRAN
Called by: CNTRL
Calling Sequence: CALL CUSTOM

External References: Common Blocks
FCCOM

Function Description

The FORMAT task list a number of viewing options for operator selection to represent the flaw in three dimensional display. If none of these views are chosen, the operator may select his own viewing angle. CUSTOM module will tutorially prompt the operator for plot parameters which the three dimensional plot is to be displayed.

Module: FCLOSE - Closes data file

Language: FORTRAN
Called by: CNTRL
Calling Sequence: CALL FCLOSE

External References:

CLOSE - Closes the specified logical unit
(FORTRAN library routine)

Functional Description

FCLOSE closes the packed data file once the task has completed its processing of the file data.

Module: FILTER

Language: FORTRAN Noise filtering routine
Called by: FLWCHR
Calling Sequence: CALL FILTER

External References: Common Blocks
FCCOM

Functional Description

FILTER reduces noise peaks to background level and brings signal nulls up to the surrounding signals level. A signal is a non-zero value two or more index points long. The routine scans the array in ascending order until it encounters a non-zero value, and looks ahead to the next value. If the next value is also non-zero, the scan assumes a "signal state" or if the next value returned to zero, the point is discarded as noise. The function operates in a complementary fashion when it is in a "signal state" and encounters a signal null (zero value).

Module: FLWCHR Flaw Characterization routine

Language: FORTRAN
Called by: FORMAT
Calling Sequence: CALL FLWCHR

External References: Common Blocks
EXDCOM
FCCOM
PEXCOM

External References: Subroutines
CALC Calculation routine
FILTER Filters out "noise" in flaw data
PEXF Read and process a record from packed data file
PGEN Generate Fourier transforms
PLOT3D Generate three dimensional plots

Functional Description

FLWCHR process the flaw data in the time domain and generates a three dimensional representation of the flaw. The routine cycles through all data points in the file and calculates the peak intensity for each point. The result is then filtered and plotted.

Module: FLWCLS

Language: FORTRAN
Called by: FORMAT
Calling Sequence: CALL FLWCLS

External References: Common Blocks
FCCOM
PEXCOM
FORIER

External References: Subroutines

| | |
|---|---|
| CHOUTS | Print normal size character strings |
| ERASE | Erase Tektronix screen |
| FLWDEF | Comparison of flaw data |
| FSDF | Search the characteristic FFT array for the maximum value |
| HOME | Homes the cursor on terminal |
| LENGTH | Determine the length of the flaw |
| PEXF | Read and process a record from the packed data file |
| PGEN | Generate Fourier transform |

Functional Description

FLWCLS processes the selected flaw data file and classifies it. It initializes the data file by positioning at the beginning of the flaw data, generates a characteristic Fourier transform of the flaw and classifies it; all processing is performed in the frequency domain. The generated result is used to correlate the flaw data with previous stored flaw data.

Module: FLWDEF - Defines the flaw

Language: FORTRAN
Called by: FLWCLS
Calling Sequence: CALL FLWDEF

External References: Common Blocks
FCCOM

External References: Subroutines
NNS - Correlation of flaw data

Functional Description

FLWDEF classifies the flaw according to the results of the correlation of the data processed. If the flaw is classified as a known flaw where its data matches a previously stored flaw, it is defined. If the data does not match, then its parameters are retained for future comparisons.

Module: FNAM - File name input routine

Language: MACRO-11
Called by: FOP
Calling Sequence: CALL FNAM

External References: Subroutines
TEKCOM
ICHBUF

External References: Subroutines
CHOUT Writes characters to Tektronix
CRLF Sends carriage return/line feed

Functional Description:

FNAM presents to the operator the following prompt:

ENTER FILENAME

The program then accepts a filename in the form NAME.EXT where standard length limitations apply. The file is looked up on the disk directly and, if it exists, FSTAT is set to 1.

Module: FOP - Data file operations routine

Language: FORTRAN
Called by: CNTRL
Calling Sequence: CALL FOP

External References: Common Blocks
FCCOM
PEXCOM

External References: Subroutines

| | |
|---|---|
| FNAM | File name input routine |
| ASSIGN | Assigns logic unit number to file name (library routine) |
| DEFINE | Opens the data file and defines the file parameters (library routine) |

Functional Description

FOP drives the modules which solicit a file name, opens it and sets it up for I/O.

Module: FORGEN - Generate the number of points and lines

Language: FORTRAN
Called by: CNTRL
Calling Sequence: CALL FORGEN

External References: Common Blocks
PEXCOM
FCCOM

Functional Description:

FORGEN generates the number of lines and the number of points in a line for the data file from the coordinates of the scan limits.

Module: FORMAT - main driver

Language: FORTRAN
External References: Common Blocks
FCCOM

External References: Subroutines

| | |
|---|---|
| BELL | Ring bell on terminal |
| CNTRL | User interface module |
| FLWCHR | Generate three dimensional plots |
| FLWCLS | Generate characteristics Fourier transforms of flaw plots |
| LABEL | Outputs the header information of flaw characterization results |
| PAUSE | Suspend execution until user input |
| TEKIN | Enable terminal interrupt |

Functional Description:

FORMAT drives all of the subroutines in the flaw data post processor task. Upon entry, it solicits information, initializes parameters, and opens data files, essential to the execution of the task. On subsequent task it calls other modules based on operator input.

Module: FSDF - Searches characteristic fourier transform
for the peak value

Language: FORTRAN
Called by: FLWCLS
Calling Sequence: CALL FSDF

External References: Common Blocks
FORIER
FCCOM

Functional Description

FSDF searches the array DAT which has the values of the characteristic Fourier transform for the maximum value and and records it and its location.

Module: FT - calculations routine

Language: MACRO-11
Called by: MNS
Calling Sequence: CALL FT

External References: Commons
EXDCOM
FORIER

External References: Global Variables

| | |
|---|---|
| ACUMHI | MFFT hardware resistor |
| ACUMLO | MFFT hardware resistor |
| MWDCT | Number of elements to process |
| CTABL | Control table for Elsytec |
| TAD | Temporary area address |
| LTMP | Length of temporary area in bytes |
| INVFG | Inverse FFT flag |

External References: Subroutines

| | |
|---|---|
| FN1 | Calls Elsytec hardware functions |
| CFT | Performs forward and inverse FFTs |
| CTO | Converts for all-real data |
| CMCS | Calculates magnitudes of complex arrays |

Functional Description

FT performs the FFT calculations on the raw data.

Module: INT - Interpolation routine

Language: FORTRAN
Called by: CALC
Calling Sequence: CALL INT

External References: Common Blocks
EXDCOM
FORIER

Functional Description:

The flaw data points consist of records of varying lengths from one case to another. INT takes the data points and expands the data into a buffer of uniform length interpolating for the missing elements.

Module: ITOA - Converts to ASCII

Language: FORTRAN
Called by: LABEL
Calling Sequence: CALL ITOA (I, IA, N)
I - the value to be converted
IA - the ASCII representation
N - the number of characters to convert

Functional Description

ITOA converts up to 5 digit integer values to ASCII characters.

Module: LABEL - Print header information on Tektronix

Language: FORTRAN
Called by: FORMAT
Calling Sequence: CALL LABEL (PHI, THETA)
PHI - Rotation in degrees about the y-axis of the original coordinate system for the three dimensional plot.
THETA - Rotation in degrees about the x-axis of the resulting coordinate system of the PHI rotation for the three dimensional plot.

External References: Subroutines
CHOUTS Print normal sized character strings
HOME Homes cursor on terminal
IPLOT Plot individual points and lines
ITOA Convert register format numbers to ASCII
PRTH Print horizontal label

Functional Description:

LABEL displays titles, header information and processed information on the Tektronix screen based on file information, user input, and results of the flaw characterization processing of the task.

Module: LENTGI - Return the length of the flaw

Language: FORTRAN
Called by: FLWCLS
Calling Sequence: CALL LENGTH

External References: Common Blocks
FORIER
FCCOM

Functional Description

LENGTH determines the length of the flaw. The length is defined as three decibels above the average of the last ten values of the array containing the characteristic Fourier transform.

Module: NNS - Correlation Routine

Language: FORTRAN
Called by: FLWDEF
Calling Sequence: Call NNS (IREF, IC, IFLAW, NUM)

IREF - Will contain one of two values. The peak amplitude of the characteristic Forier transform or the frequency shift of the peak amplitude of the characteristics Forier transform.

IC - Refers to which value is IREF.
1. The frequency shift of the peak amplitude.
2. The peak amplitude.

IFLAW - The flaws which most closely match the characteristics of the flaw.

NUM - The number of flaws which closely match the characteristics of the given flaw.

Functional Description:

NNS performs a nearest neighbor approximation of the current flaw and the flaw data table.

Module: PAUSE - Pause for operator input

Language: MACRO-11
Called by: FORMAT
Calling Sequence: CALL PAUSE

External References: Common Blocks
TEKCOM

Functional Description
PAUSE suspends execution of the task until the operator keys in an entry from the Tektronix terminal.

Module: PEXF - Expand the next data point.

Language: MACRO-11
Called by: CNTRL
Calling Sequence: CALL PEXF

External References: Common Blocks
FCCOM
PEXCOM
EXDCOM

External References:
SREAD - Read compressed data file

Functional Description:
When PEXF has been called a data file has been opened. After compressed data has been read into the data buffer DBUF, PEXF expands the points into the buffer IBUF in byte packed form. The values found in point headers and scan headers will be transferred to the appropriate common locations.

Module: PGEN - Driver for the calculation routines

Language: FORTRAN
Called by: FLWCLS
Calling Sequence: CALL PGEN

External References: Common Blocks
FCCOM
EXDCOM
PEXCOM

External References: Subroutines
CALC - Calculation routine

Functional Description

PGEN acts as a driver for the calculation routines. It cycles through each point of each line to generate the characteristic Fourier transform.

Module: PLOT3D - Displays a three dimensional representation of the flaw.

Language: FORTRAN
Called by: FLWCHR
Calling Sequence: CALL PLOT3D (IVXYZ, XDATA, ZDATA, XSCALE, YSCALE, ZSCALE, NLINE, NPNTS, PHI, THETA, XREF, YREF, XLENGTH)
(A description of the calling arguments is carried in the functional description.)

External References: Common Blocks
FCCOM
FORIER

External References: Subroutines
IPLOT - Plots, points and lines

Functional Description:

Masked 3-D Dimensional Plot Program with Rotations

This routine will accept 3-dimensional data in various forms as input, rotate it in 3-space to any angle, and plot the projection of the resulting figure on to the xy plane. Linear interoplation is used between data points. Those lines of a figure which should be hidden by a previous line are masked.

The masking technique used by this routine is based on two premises--

Lines in the foreground (positive Z direction) are plotted before lines in the background.
A line or portion of a line is masked (hidden) if it lies within the region bounded by previously plotted lines.

Each call to PLOT3D causes one line of a figure to be plotted. Two parameters of the plotter are set on the initial call for each figure--

(PIPI) is the number of plotter increments per inch.
(NYPI) is the number of increments available across the width of the paper (Y-direction).

When a new figure is initiated, the plotter origin is set at the bottom of the paper by PLOT3D and should not be moved until the figure is completed.

Input parameters -

(IVXYZ) is a four digit decimal integer which is used to select various input/output options. These digits, in decreasing order of magnitude, will be referred to as V, X, Y, and Z.

In terms of rotations of axes, the initial system of axes, xyz, is rotated by an angle (PHI) counterclockwise about the y-axis, and the resultant system is labeled the tuv axes. The uv axes are then rotated by an angle (THETA) counterclockwise about the t-axis, and this final system is labeled the pqr axes. The plotted figure is the projection of the original figure onto the pg-plane. In terms of rotations of coordinates, the figure is first rotated by an angle (THETA) clockwise about the x-axis. The resultant figure is then rotated by an angle (PHI) clockwise about its y-axis. The plotted figure is the warning. Some rotations will alter the foreground/background relationships between the lines, and thus the order in which they should be plotted. (XREF) and (YREF) are the coordinates, in inches, relative to the plotter origin, to be used as the origin of the figure.

(XLENTH) is the length, in inches, to which the plot is restricted. Any point which exceeds this limit, or the limits of the paper in the y direction (NYPI), will be set to that limit.

Module: PLTBUF

Language: FORTRAN
Called by: FLWCLS
Calling Sequence: CALL PLTBUF (IZE)
IZE - Number of points to be plotted.

External References: Common Blocks
FORIER

IPLOT Plots, points and lines on graphics terminal

Functional Description:

PLTBUF plots the characteristic Fourier transform on the terminal along with the labels and axes.

Module: SEARCH - Return the maximum value in the buffer

Language: FORTRAN
Called by: CALC
Calling Sequence: CALL SEARCH

External References: Common Blocks
EXDCOM
FCCOM

Functional Description:

SEARCH cycles through the data buffer IBUF which contains the data for each point and returns the maximum value for each.

Module: SREAD - Reads the data file

Language: FORTRAN
Called by: PEXF
Calling Sequence: CALL SREAD

External References: Common Blocks
PEXCOM

Functional Description:

SREAD reads the data file in 256 word blocks, four blocks at a time. It sets a flag word when the end of the file is encountered.

Module: TEKMOD - Tektronix module

Language: MACRO-11

External References: I/O Registers

| | | |
|---|---|---|
| 175610 | r/w | CSR for terminal interface |
| 175612 | ro | Read register for terminal interface |
| 175614 | r/w | |
| 175616 | wo | Write register for terminal interface |
| 370 | wo | Terminal interrupt vector |

Functional Description:

This is a group of utility routines which allows the user to interface with and control the Tektronix 4002A terminal. The following is a brief description of each function within the package along with the software interface to the function.

Entry Point: ALPHA - Puts terminal in alpha mode
Calling Sequence: Call ALPHA

Entry Point: ALPHA2 - Puts terminal in large character mode
Calling Sequence: CALL ALPHA2

Entry Point: BELL - Rings bell on Tektronix terminal
Calling Sequence: CALL BELL

Entry Point: CHOUT - Outputs character strings to the terminal
Calling Sequence: CALL CHOUT (MBUF)
Where: MBUF is an array, the first element of which is a word containing the character count.

Entry Point: CR - Sends a carriage return to the terminal.
Calling Sequence: CALL CR

Entry Point: CRLF - Sends a carriage return/line feed to the terminal
Calling Sequence: CALL CRLF

Entry Point: ERASE - Erases screen on Tektronix terminal.
Calling Sequence: CALL ERASE

Entry Point: GRIN - Graphic Input Routine
Calling Sequence: CALL GRIN (IALPHA, IX, IY)
Where: IALPHA is key pressed to enter coordinate
IX is X coordinate of graphic crosshair
IY is Y coordinate of graphic crosshair

Entry Point: HOME - Homes cursor to upper left
Calling Sequence: Call HOME

Entry Point: IPLOT - Positions graphic cursor
Calling Sequence: CALL IPLOT (IX, IY, IPEN)
Where: IX is X coordinate to which cursor is to be moved
IY is Y coordinate to which cursor is to be moved
IPEN is pen control 0/1 for Up/Down

Entry Point: LF - Sends line feed character to terminal
Calling Sequence: CALL LF

Entry Point: LFCR - Sends line feed/carriage return to terminal
Calling Sequence: CALL LFCR

Entry Point: PRTH - Prints horizontally on terminal
Calling Sequence: CALL PRTH

Entry Point: PRTV - Prints vertically on terminal
Calling Sequence: CALL PRTV

Entry Point: TEKIN - Enable interrupts from Tektronix terminal
Calling Sequence: CALL TEKIN

After TEKIN has been called, unsolicited characters input on the terminal are placed in ICHBUF of the common TEKCOM.

Exceptions are:

| | | |
|---|---|---|
| ESCAPE | (ASCII 33) | - Aborts current operation and returns system to primary option display. |
| CNTRL Q | (ASCII 21) | - Sends 'Stop Scanner' command to LSI-11 |
| CNTRL R | (ASCII 22) | - Sends 'reset' to LSI-11 to reset scanner system. |

# APPENDIX B

# FDPP MODULE SOURCE LISTINGS

```
0001          SUBROUTINE CALC  !CALCULATION ROUTINE
0002          COMMON/EXDCOM/ IHI, IDSTAT,IXS,IYS,IE(3),ISTRT
             $,ISTP,IH(7),IBUF(2048)
0003          COMMON /FCCOM/IFDFLG, IFFLG, IA(2), VAL, ITHRSH,ID(9),
             $IST,IPLOC,ITOTF,ITOTE,NUMF,NUME,IDA(8),ICNT
             $,IK(6),OUTBUF(3072)
0004          COMMON /PEXCOM/ IB(2), IXPNT, IYPNT, IJ(35), INCR
0005          COMMON /FURIER/ MD, IFFT,IG(513),MDATA(2050)
0006          DATA IVAR/2/
0007          DIMENSION C(4),IVAR1(11)
0008          LOGICAL*1 IBUF                !IBUF IS A BYTE ARRAY
          C
          C   LOOK FOR POINT
          C
0009          VAL = FLOAT(ITHRSH)
          C
          C   CONVERT NEGATIVE COORDINATES TO POSITIVE
          C
0010    21    C(1) = FLOAT( IXPNT )
0011          C(2) = FLOAT( IYPNT )
0012          C(3) = FLOAT( IXS )
0013          C(4) = FLOAT( IYS )
0014          DO 22 I = 1,4
0015          IF(C(I) .GE. 0) GO TO 22
0017          C(I) = 65535. + C(I)                    !CORRECTION FACTOR
0018    22    CONTINUE
0019          A1 = FLOAT(INCR ) / 2                   !MARGIN OF ERROR
0020          IF(IDSTAT .NE. 0) GO TO 10  !IS THE POINT THERE
          C
          C   HAS Y COORD. BEEN PASSED?
          C
0022          IF(C(4) .LT. (C(2)-A1)) GO TO 10
0024          IF(C(4) .GT. (C(2)+A1)) GO TO 20
0026          IF(IST .EQ. 0) GO TO 11
          C
          C   HAS X COORD. BEEN PASSED?
          C
0028          IF(C(3) .LT. (C(1)-A1)) GO TO 20
0030          GO TO 19
0031    20    CALL PEXF             !READ IN NEW POINT
0032          GO TO 21
0033     1    IF(C(3) .GT. (C(1)+A1)) GO TO 20

          C   ARE X,Y COORD.'S WITIHIN SPECIFIED LIMITS
          C
0035    19    AA = C(1) - A1
0036           B = C(1) + A1
0037          IF((C(3) .GE. AA) .AND. (C(3) .LE. AB)) GO TO 12
0039          GO TO 10
0040    12    AA = C(2) - A1
0041          AB = C(2) + A1
0042          IF((C(4) .GE. AA) .AND. (C(4) .LE. AB)) GO TO 1
0044          GO TO 10
0045    1     IF(IFDFLG .EQ. 0) GO TO 18
```

```
      C
      C     USE FORRIER ANALYSIS TO CLASSIFY FLAW
      C
0047        IFFT = 0     !CLEAR INVERSE FFT FLAG
0048        MD = 0               !SET DATA SCALE FACTOR
      C
      C     CENTER DATA IN MIDDLE OF IBUF
      C
0049  16    MDPT = ISTRT + (ISTP - ISTRT)/2            !CALCULATE THE MIDPOINT
0050        IDIFF = IABS(MDPT - 1024)
0051        IF(MDPT .GT. 1025) CALL CENTR(1,IDIFF)             !SUBTRACTIVE CENTERING
0053        IF(MDPT .LT. 1023) CALL CENTR(0,IDIFF)             !ADDITIVE CENTERING
0055        IF((MDPT .GE. 1023) .AND. (MDPT .LE. 1025)) GO TO 17
0057        GO TO 16
      C
      C     IS THE DATA POINT A PART OF THE FLAW?
      C
0058  17    IDSTX = IABS(IXPNT - IXOLD)
0059        IDSTY = IABS(IYPNT - IYOLD)
0060        IERR = (2 * INCR) + IFIX( A1 )
0061        IF((IDSTX .GT. IERR) .OR. (IDSTY .GT. IERR)) GO TO 2
      C
      C     PERFORM FORRIER ANALYSIS
      C
0063        ISTRT = MDPT - 256
0064        ISTP = MDPT + 256
0065        CALL FT              !TAKE THE FFT
0066        CALL LENGTH !CALCULATE ENVELOPE LENGTH
0067        WRITE(7,101)IPLOC
0068  101   FORMAT(3X,'IPLOC LENGTH = ',I5)
0069        ITOTE = ITOTE + IPLOC
0070        NUME = NUME + 1
0071        ISTRT = MDPT - 128
0072        ISTP = MDPT + 128
0073        CALL FT              !TAKE THE FFT
0074        CALL FSDF   !FIND THE FREQUENCY SHIFT DUE TO THE FLAW
0075        WRITE(7,100)IPLOC
0076  100   FORMAT(3X,'IPLOC FSDF = ',I5)
0077        ITOTF = ITOTF + IPLOC
0078        NUMF = NUMF + 1
0079        GO TO 2
      C
      C     PERFORM FLAW CHARACTERIZATION
      C
0080  18    CALL SEARCH          !FIND DATA POINT PEAK
      C
      C     RECORD LOCATION OF PREVIOUS POINT
      C
0081  2     IXOLD = IXPNT
0082        IYOLD = IYPNT
0083        CALL PEXF            !READ NEW DATA POINT
      C
      C     RETURN TO CALLING PROGRAM
      C
```

```
0084  10     RETURN
0085           END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

C       000014  REAL*4     ARRAY (4)
IVAR1   000034  INTEGER*2  ARRAY (11)
IVAR    000062  INTEGER*2  VARIABLE
FLOAT   000000  REAL*4     PROCEDURE
I       000150  INTEGER*2  VARIABLE
A1      000152  REAL*4     VARIABLE
PEXF    000000  REAL*4     PROCEDURE
AA      000156  REAL*4     VARIABLE
AB      000162  REAL*4     VARIABLE
MDPT    000166  INTEGER*2  VARIABLE
IDIFF   000170  INTEGER*2  VARIABLE
IABS    000000  INTEGER*2  PROCEDURE
CENTR   000000  REAL*4     PROCEDURE
IDSTX   000172  INTEGER*2  VARIABLE
IXOLD   000174  INTEGER*2  VARIABLE
IDSTY   000176  INTEGER*2  VARIABLE
IYOLD   000200  INTEGER*2  VARIABLE
IERR    000202  INTEGER*2  VARIABLE
IFIX    000000  INTEGER*2  PROCEDURE
FT      000000  REAL*4     PROCEDURE
LENGTH  000000  INTEGER*2  PROCEDURE
FSDF    000000  REAL*4     PROCEDURE
SEARCH  000000  REAL*4     PROCEDURE

COMMON BLOCK /EXDCOM/   LENGTH 004040

IHI     000000  INTEGER*2  VARIABLE
IDSTAT  000002  INTEGER*2  VARIABLE
IXS     000004  INTEGER*2  VARIABLE
IYS     000006  INTEGER*2  VARIABLE
IE      000010  INTEGER*2  ARRAY (3)
ISTRT   000016  INTEGER*2  VARIABLE
ISTP    000020  INTEGER*2  VARIABLE
IH      000022  INTEGER*2  ARRAY (7)
IBUF    000040  LOGICAL*1  ARRAY (2048)

COMMON BLOCK /FCCOM/    LENGTH 030112

IFDFLG  000000  INTEGER*2  VARIABLE
IPFLG   000002  INTEGER*2  VARIABLE
IA      000004  INTEGER*2  ARRAY (2)
VAL     000010  REAL*4     VARIABLE
ITHRSH  000014  INTEGER*2  VARIABLE
ID      000016  INTEGER*2  ARRAY (9)
IST     000040  INTEGER*2  VARIABLE
IPLOC   000042  INTEGER*2  VARIABLE
ITOTF   000044  INTEGER*2  VARIABLE
ITOTE   000046  INTEGER*2  VARIABLE
NUMF    000050  INTEGER*2  VARIABLE
NUME    000052  INTEGER*2  VARIABLE
IDA     000054  INTEGER*2  ARRAY (8)
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

ICNT     000074   INTEGER*2 VARIABLE
IK       000076   INTEGER*2 ARRAY (6)
OUTBUF   000112   REAL*4    ARRAY (3072)

COMMON BLOCK /PEXCOM/   LENGTH 000120

IB       000000   INTEGER*2 ARRAY (2)
IXPNT    000004   INTEGER*2 VARIABLE
IYPNT    000006   INTEGER*2 VARIABLE
IJ       000010   INTEGER*2 ARRAY (35)
INCR     000116   INTEGER*2 VARIABLE

COMMON BLOCK /FORIER/   LENGTH 012012

MD       000000   INTEGER*2 VARIABLE
IFFT     000002   INTEGER*2 VARIABLE
IG       000004   INTEGER*2 ARRAY (513)
MDATA    002006   INTEGER*2 ARRAY (2050)
*
```

```
0001          SUBROUTINE CENTR(IVAR, IDIFF)  !CENTERS FLAW RETURN
0002          COMMON /EXDCOM/ IA(7), ISTRT, ISTP, IB(7), IBUF(2048)
0003          LOGICAL * 1 IBUF
0004          IF(IVAR .EQ. 0) GO TO 1              !CHOOSE ACTIVE CENTERING
      C
      C       SUBTRACTIVE CENTERING
      C
0006          ISTRT = ISTRT - IDIFF                !RESET ISTRT & ISTP
0007          ISTP = ISTP - IDIFF
0008          DO 2 I = 1,(2047 - IDIFF)
0009          IBUF( I ) = IBUF( I+IDIFF )          !SHIFT ENTIRE CURVE TO LEFT
0010    2     CONTINUE
      C
      C       ZERO NEW ELEMENTS
      C
0011          DO 5 I = (2048 - IDIFF),2048
0012          IBUF(I) = 0
0013    5     CONTINUE
0014          GO TO 3
      C
      C       ADDITIVE CENTERING
      C
0015    1     ISTRT = ISTRT + IDIFF                !RESET ISTRT & ISTP
0016          ISTP = ISTP + IDIFF
0017          DO 4 I = 1,(2047-IDIFF)
0018          IBUF(2048-I) = IBUF(2048 - IDIFF - I)       !SHIFT ENTIRE CURVE TO RIGHT
0019    4     CONTINUE
      C
      C       ZERO NEW ELEMENTS
      C
0020          DO 6 I = 1,(IDIFF + 1)
0021          IBUF(I) = 0
0022    6     CONTINUE
0023    3     RETURN
0024          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IVAR    000014  INTEGER*2 PARAMETER VARIABLE
IDIFF   000016  INTEGER*2 PARAMETER VARIABLE
I       000020  INTEGER*2 VARIABLE

COMMON BLOCK /EXDCOM/   LENGTH 004040

IA      000000  INTEGER*2 ARRAY (7)
ISTRT   000016  INTEGER*2 VARIABLE
ISTP    000020  INTEGER*2 VARIABLE
IB      000022  INTEGER*2 ARRAY (7)
IBUF    000040  LOGICAL*1 ARRAY (2048)
$
```

```
                                                .TITLE  CNTRL   ;CONTROL MODULE
                                                .GLOBL  CNTRL,CHOUT,PEXF,CALC,FORGEN
                                                .GLOBL  FOP,ERASE,HOME,FCLOSE,HEIGHT
                                                .GLOBL  CUSTOM,CNTCAL,PAUSE,RITE
                                                .MCALL  SAVE,UNSAVE,CHSTUP,CHTXT,.REGDEF
                                                .MCALL  ..V2...,.EXIT
 000000                                         ..V2...
 000000                                         .REGDEF
 000000                                 CNTRL:
 000000  005767  000070'                 C12:    TST     FTT             ;;TEST FOR THE FIRST TIME
 000004  001020                                 BNE     C2
 000006  004767  000000G                        JSR     PC,ERASE
 000012  004767  000000G                        JSR     PC,HOME
 000016  012705  000500'                        MOV     #FMSG,R5
 000022  004767  000000G                        JSR     PC,CHOUT
 000026  012767  000001  000070'               MOV     #1,FTT          ;SET FIRST TIME THROUGH FLAG
 000034  004767  000000G                        JSR     PC,FOP
 000040  004767  000000G                        JSR     PC,ERASE
 000044  000456                                 BR      C1              ;CLASSIFY AND CHARACTERIZE FLAW
                                         ;
                                         ;      DISPLAY OPTIONS
                                         ;
 000046                                 CNTCAL:
 000046  005767  000002'                 C2:     TST     PTYP            ;FLAW CLASSIFICATION?
 000052  001457                                 BEQ     C3
 000054  004767  000000G                        JSR     PC,ERASE        ;ERASE THE PAGE
 000060  004767  000000G                        JSR     PC,HOME         ;HOME THE CURSOR
 000064  012705  000574'                        MOV     #MENU,R5
 000070  004767  000000G                        JSR     PC,CHOUT        ;PRINT MENU
 000074  005067  000000'                        CLR     ICHBUF
 000100  005767  000000'                 1$:     TST     ICHBUF
 000104  001775                                 BEQ     1$      ;WAIT FOR CHARACTER
                                         ;
                                         ;      TAKE ACTION BASED UPON INPUT
                                         ;
                                         ;
                                         ;      FLAW CLASSIFICATION OPTION
                                         ;
 000106  026727  000000' 000101                 CMP     ICHBUF,#101
 000114  001003                                 BNE     C10
 000116  005067  000002'                        CLR     PTYP
 000122  000427                                 BR      C1
                                         ;
                                         ;      NEW FILENAME OPTION
                                         ;
 000124  026727  000000' 000111  C10:    CMP     ICHBUF,#111
 000132  001005                                 BNE     C11
 000134  005067  000070'                        CLR     FTT
 000140  004767  000000G                        JSR     PC,FCLOSE       ;FORTRAN FILE CLOSE ROUTINE
 000144  000715                                 BR      C12
                                         ;
                                         ;      EXIT OPTION
                                         ;
 000146  026727  000000' 000112  C11:    CMP     ICHBUF,#112
 000154  001334                                 BNE     C2
 000156  004767  000000G                        JSR     PC,ERASE        ;ERASE THE PAGE
 000162  004767  000000G                        JSR     PC,HOME         ;HOME THE CURSOR
```

```
                                        MOV     #TERM,R5        ;PRINT TERMINATION MESSAGE
                                        JSR     PC,CHOUT
                                        .EXIT
                                        BR      C5
                                ;
                                ;       CLASSIFY FLAW
                                ;
000202  012767  000001  000000' C1:     MOV     #1,FDFLG
000210  000405                          BR      C5
                                ;
                                ;       CHARACTERIZE FLAW
                                ;
000212  005067  000000'         C3:     CLR     FDFLG
000216  012767  000001  000002'         MOV     #1,PTYP
                                ;
                                ;       GET DATA FILE INFORMATION
                                ;
000224  005067  000072'         C5:     CLR     CNT
000230  012767  177777  000012'         MOV     #-1,CDB
000236  012767  000001  000016'         MOV     #1,SBLK
000244  012705  001426'                 MOV     #BUG,R5
000250  004767  000000G                 JSR     PC,CHOUT
000254  004767  000000G                 JSR     PC,PEXF
                                ;;
                                ;       GENERATE FORMULA FOR POINT SEQUENCE
                                ;
000260  004767  000000G                 JSR     PC,FORGEN
000264  004767  000000G                 JSR     PC,PEXF
                                ;       JSR     PC,HEIGHT       ;CALCULATE HEIGHT OF XDUCER ABOVE METAL
                                ;
                                ;       SELECT VIEWING ANGLE
                                ;
000270  005767  000000'         V6:     TST     FDFLG           ;IF CLASSIFY, THEN SKIP VIEWING ANGLE
000274  001474                          BEQ     V2
000276  012705  001426'                 MOV     #BUG,R5
000302  004767  000000G                 JSR     PC,CHOUT
                                ;       JSR     PC,HOME         ;HOME THE CURSOR
000306  012705  001152'                 MOV     #PVA,R5
000312  004767  000000G                 JSR     PC,CHOUT
000316  005067  000000'                 CLR     ICHBUF
000322  005767  000000'         1$:     TST     ICHBUF
000326  001775                          BEQ     1$              ;WAIT FOR INPUT
                                ;
                                ;       ORTHONORMAL VIEW
                                ;
000330  026727  000000' 000101         CMP     ICHBUF,#101
000336  001004                          BNE     V1
000340  012767  000001  000016'         MOV     #1,PLT
000346  000447                          BR      V2
                                ;
                                ;       10-DEG TILT
                                ;
000350  026727  000000' 000102  V1:     CMP     ICHBUF,#102
000356  001004                          BNE     V3
000360  012767  000002  000016'         MOV     #2,PLT
000366  000437                          BR      V2
                                ;
```

```
                                        ;       SIDE PROFILE
                                        ;
000370  026727  000000'  000103  V3:     CMP     ICHBUF,#103
000376  001004                         BNE     V4
000400  012767  000003   000016'        MOV     #3,PLT
000406  000427                         BR      V2
                                        ;
                                        ;       FRONT PROFILE
                                        ;
000410  026727  000000'  000104  V4:     CMP     ICHBUF,#104
000416  001004                         BNE     V5
000420  012767  000004   000016'        MOV     #4,PLT
000426  000417                         BR      V2
                                        ;
                                        ;       CUSTOM PLOT OPTION
                                        ;
000430  026727  000000'  000105  V5:     CMP     ICHBUF,#105
000436  001006                         BNE     V8
000440  012767  000005   000016'        MOV     #5,PLT
000446  004767  0000006                 JSR     PC,CUSTOM       ;ENTERS PLOT PARAMETERS
000452  000405                         BR      V2
                                        ;
                                        ;       NO CHAR. AVAILABLE MESSAGE
                                        ;
000454  012705  001076'          V8:    MOV     #NSG,R5
000460  004767  0000006                 JSR     PC,CHOUT
000464  000701                         BR      V6
000466  004767  0000006         V2:     JSR     PC,ERASE        ;ERASE THE PAGE
000472  004767  0000006                 JSR     PC,HOME         ;HOME THE CURSOR
000476  000207                         RTS     PC
000500                                  CHSTUP  FMSG,MENU
000506                                  CHTXT   <ENTER DESIRED FILENAME:>
000546                                  CHTXT   <DK1:XXXXXX.IND>
000574                                  CHSTUP  MENU,TERM
000602                                  CHTXT   <OPTIONS CURRENTLY AVAILABLE:>
000646                                  CHTXT   < >
000660                                  CHTXT   <A = CHARACTERIZE AND CLASSIFY FLAW>
000732                                  CHTXT   <I = ENTER NEW FILENAME>
000770                                  CHTXT   <J = EXIT THE PROGRAM>
001024                                  CHSTUP  TERM,NSO
001032                                  CHTXT   <PROGRAM EXECUTION TERMINATED>
001076                                  CHSTUP  NSO,PVA
001104                                  CHTXT   <OPTION NOT CURRENTLY AVAILABLE>
001152                                  CHSTUP  PVA,BUG
001160                                  CHTXT   <AVAILABLE PLOT VIEWING ANGLES:>
001226                                  CHTXT   <A = ORTHONORMAL>
001256                                  CHTXT   <B = 10-DEG. TILT VIEW>
001314                                  CHTXT   <C = SIDE PROFILE>
001344                                  CHTXT   <D = FRONT PROFILE>
001376                                  CHTXT   <E = CUSTOM PLOT>
001426                                  CHSTUP  BUG,V7
001434                                  CHTXT   < DEBUG SIGNAL>
001462                          V7:
         000000'                        .CSECT  FCCOM
000000  000000                  FDFLG:  .WORD   0
000002  000000                  PTYP:   .WORD   0
000004  000000                  NPNTS:  .WORD   0
```

```
  172 000006  000000                NLINES: .WORD   0
  173 000010                        VAL:    .BLKW   2
  174 000014  000000                THRSH:  .WORD   0
  175 000016  000000                PLT:    .WORD   0
  176                                       .BLKW   20.
  177 000070  000000                FTT:    .WORD   0
  178 000072  000000                CNT:    .WORD   0
  179         000000'                       .CSECT  PEXCOM
  180                                       .BLKW   5.
  181 000012  000000                CDB:    .WORD   0
  182 000014  000000                        .WORD   0
  183 000016  000000                SBLK:   .WORD   0
  184                                       .BLKW   23.
  185 000076  000000                XLIM1:  .WORD   0
  186 000100  000000                YLIM1:  .WORD   0
  187 000102  000000                XLIM2:  .WORD   0
  188 000104  000000                YLIM2:  .WORD   0
  189                                       .BLKW   4
  190 000116  000000                INCR:   .WORD   0
  191         000000'                       .CSECT  TEKCOM
  192 000000  000000                ICHBUF: .WORD   0
  193         000000'                       .CSECT  EXDCOM
  194                                       .BLKW   2
  195 000004  000000                IXS:    .WORD   0
  196 000006  000000                IYS:    .WORD   0
  197                                       .BLKW   2
  198 000014  000000                SINC:   .WORD   0
  199 000016  000000                ISTRT:  .WORD   0
  200 000020  000000                ISTP:   .WORD   0
  201 000022  000000                HXS:    .WORD   0
  202 000024  000000                HYS:    .WORD   0
  203         000001'                       .END
```

```
CNTRL   ;CONTROL MODULE RT-11 MACRO VM02-12   15-SEP-80 PAGE 1+
SYMBOL TABLE

BUG     001426R         CALC  = ****** G        CDB     000012R    003  CHOUT = ****** G        CNT     000072R    002
CNTCAL  000046RG        CNTRL   000000RG        CUSTOM= ****** G        C1      000202R         C10     000124R
C11     000146R         C12     000000R         C2      000046R         C3      000212R         C5      000224R
ERASE = ****** G        FCLOSE= ****** G        FDFLG   000000R    002  FMSG    000500R         FOP   = ****** G
FORGEN= ****** G        FTT     000070R    002  HEIGHT= ****** G        HOME  = ****** G        HXS     000022R    005
HYS     000024R    005  ICHBUF  000000R    004  INCR    000116R    003  ISTP    000020R    005  ISTRT   000016R    005
IXS     000004R    005  IYS     000006R    005  MENU    000574R         NLINES  000006R    002  NPNTS   000004R    002
NSO     001076R         NUM   = 000001          NUM1  = 000015          PAUSE = ****** G        PC    =Z000007
PEXF  = ****** G        PLT     000016R    002  PTYP    000002R    002  PVA     001152R         RITE  = ****** G
R0    =Z000000          R1    =Z000001          R2    =Z000002          R3    =Z000003          R4    =Z000004
R5    =Z000005          SBLK    000016R    003  SINC    000014R    005  SP    =Z000006          TERM    001024R
THRSH   000014R    002  VAL     000010R    002  V1      000350R         V2      000466R         V3      000370R
V4      000410R         V5      000430R         V6      000270R         V7      001462R         V8      000454R
XLIM1   000076R    003  XLIM2   000102R    003  YLIM1   000100R    003  YLIM2   000104R    003  ...V2 = 000001

. ABS.  000000      000
        001462      001
FCCOM   000074      002
PEXCOM  000120      003
TEKCOM  000002      004
EXDCOM  000026      005
ERRORS DETECTED: 0
FREE CORE: 17482. WORDS

,TT:/N:TTM=CNTRLF

ERRORS DETECTED: 0
FREE CORE: 17482. WORDS

$
```

```
0001          SUBROUTINE CUSTOM  !CUSTOM PLOT PARAMETERS
0002          COMMON /FCCOM/ IA(5156),XSCALE,YSCALE,ZSCALE
             $,PHI,THETA,XREF,YREF
        C
        C     RING DECWRITER BELL
        C
0003          I = 7
0004          WRITE(7,11)I
0005    11    FORMAT(3X,A1)
        C
        C     ENTER XSCALE
        C
0006          WRITE(7,1)
0007    1     FORMAT(3X,'ENTER XSCALE, (XX.XXXXX)')
0008          READ(5,2)XSCALE
0009    2     FORMAT(F8.5)
        C
        C     ENTER YSCALE
        C
0010          WRITE(7,3)
0011    3     FORMAT(3X,'ENTER ZSCALE, (XX.XX)')
0012          READ(5,4)YSCALE
0013    4     FORMAT(F5.2)
        C
        C     ENTER ZSCALE
        C
0014          WRITE(7,5)
0015    5     FORMAT(3X,'ENTER YSCALE, (XX.XXXXX)')
0016          READ(5,2)ZSCALE
        C
        C     ENTER ANGLE PHI
        C
0017          WRITE(7,6)
0018    6     FORMAT(3X,'ENTER PHI, (XXX.X)')
0019          READ(5,7)PHI
0020    7     FORMAT(F5.1)
        C
        C     ENTER ANGLE THETA
        C
0021          WRITE(7,8)
0022    8     FORMAT(3X,'ENTER THETA, (XXX.X)')
0023          READ(5,7)THETA
        C
        C     ENTER XREF
        C
0024          WRITE(7,9)
0025    9     FORMAT(3X,'ENTER XREF, (XXX.X)')
0026          READ(5,7)XREF
        C
        C     ENTER YREF
        C
0027          WRITE(7,10)
0028    10    FORMAT(3X,'ENTER YREF, (XXX.X)')
0029          READ(5,7)YREF
```

```
      C
      C     RING BELL ON TEKTRONIX
      C
0030        CALL BELL
0031        RETURN
0032        END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

I        000326   INTEGER*2 VARIABLE
BELL     000000   REAL*4    PROCEDURE


COMMON BLOCK /FCCOM/     LENGTH 024144

IA       000000   INTEGER*2 ARRAY (5156)
XSCALE   024110   REAL*4    VARIABLE
YSCALE   024114   REAL*4    VARIABLE
ZSCALE   024120   REAL*4    VARIABLE
PHI      024124   REAL*4    VARIABLE
THETA    024130   REAL*4    VARIABLE
XREF     024134   REAL*4    VARIABLE
YREF     024140   REAL*4    VARIABLE
*
```

```
0001          SUBROUTINE FCLOSE  !CLOSES FORTRAN FILES
0002          CALL CLOSE(2)       !CLOSE THE FILE
0003          RETURN
0004          END
```

```
FORTRAN IV      STORAGE MAP
NAME    OFFSET  ATTRIBUTES
CLOSE   000000  REAL*4    PROCEDURE
$
```

```
0001          SUBROUTINE FILTER
0002          COMMON /FCCOM/ IA(2), NPNTS, IC(33) ,P(2048)
0003          I = 1                    !INITIALIZE OFFSET COUNTER
0004    2     IF(ABS(P(I)) .GT. 0.) GO TO 1
0006    3     I = I + 1
0007          IF(I .GE. (NPNTS+1)) GO TO 5
0009          GO TO 2
0010    1     I = I + 1
              C
              C     CHECK FOR NOISE PEAKS
              C
0011          IF(I .GE. (NPNTS+1)) GO TO 5
0013          IF(ABS(P(I)) .GT. 0.) GO TO 4
              C
              C     FILTER OUT NOISE PEAKS
              C
0015          P(I - 1) = 0.
0016          GO TO 3
0017    4     I = I + 1
              C
              C     CHECK FOR SIGNAL NULLS
              C
0018          IF(I .GE. (NPNTS+1)) GO TO 5
0020          IF(ABS(P(I)) .GT. 0.) GO TO 4
0022          I = I + 1
0023          IF(I .GE. (NPNTS+1)) GO TO 5
0025          IF(ABS(P(I)) .LE. 0.) GO TO 3
              C
              C     FILTER OUT SIGANL NULL
              C
0027          P(I) = P(I - 1)
0028          GO TO 4
              C
              C     RETURN TO CALLING PROGRAM
              C
0029    5     RETURN
0030          END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

I        000014   INTEGER*2 VARIABLE
ABS      000000   REAL*4    PROCEDURE


COMMON BLOCK /FCCOM/    LENGTH 020110

IA       000000   INTEGER*2 ARRAY (2)
NPNTS    000004   INTEGER*2 VARIABLE
IC       000006   INTEGER*2 ARRAY (33)
P        000110   REAL*4    ARRAY (2048)
$
```

```
0001          SUBROUTINE FLWCHR
0002          COMMON /EXDCOM/ IS,IDSTAT,IXS,IYS,IB(3),ISTRT,ISTP
0003          COMMON /FCCOM/IK(2),NPNTS,NLINES,VAL,IJ(10),IST,IPLOC,
             $IL(8),MAX,IH(2),ICNT,II(6),OUTBUF(3072),XSCALE,YSCALE,ZSCALE,
             $PHI,THETA,XREF,YREF
0004          COMMON /PEXCOM/IC(5),ICDB,IG,ISBLK,
             $IM(23),IXL1,IYL1,IN(6),INCR
0005          CALL PGEN
0006          ICDB = -1
0007          ISBLK = 1
0008          ICNT = 0
0009          CALL PEXF
0010          CALL PEXF
0011          CALL HEIGHT
0012          IST = 0
0013          DO 40 IY = 1,NLINES
0014          DO 25 IX = 1,NPNTS
0015          IF(IST .EQ. 1)GO TO 10
0017          IXS = (IX-1) * INCR + IXL1
0018          IYS = (IY-1) * INCR + IYL1
0019          GO TO 15
0020    10    IXS = (((NPNTS+1) - IX) * INCR) - (INCR/2) + IXL1
0021          IYS = (IY - 1) * INCR + IYL1
0022          GO TO 20
0023    15    CALL CALC
0024          OUTBUF(IX) = VAL / FLOAT(MAX)
0025          GO TO 25
0026    20    CALL CALC
0027          OUTBUF((NPNTS+2)-IX) = VAL / FLOAT(MAX)
0028    25    CONTINUE
0029          IF(IST .EQ. 0) GO TO 30
0031          IST = 0
0032          GOT O 35
0033    30    IST = 1
0034    35    CONTINUE
0035          OUTBUF(IX) = 0
0036          CALL FILTER
0037          CALL PLOT3D(0010,0.,0.,XSCALE,YSCALE,ZSCALE,IY,NPNTS,PHI,THETA,
             $XREF,YREF,16.)
0038    40    CONTINUE
0039          RETURN
0040          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

PGEN    000000  REAL*4     PROCEDURE
PEXF    000000  REAL*4     PROCEDURE
HEIGHT  000000  REAL*4     PROCEDURE
IY      000026  INTEGER*2  VARIABLE
IX      000030  INTEGER*2  VARIABLE
CALC    000000  REAL*4     PROCEDURE
FLOAT   000000  REAL*4     PROCEDURE
FILTER  000000  REAL*4     PROCEDURE
PLOT3D  000C00  REAL*4     PROCEDURE


COMMON BLOCK /EXDCOM/   LENGTH 000022

IS      000000  INTEGER*2  VARIABLE
IDSTAT  000002  INTEGER*2  VARIABLE
IXS     000004  INTEGER*2  VARIABLE
IYS     000006  INTEGER*2  VARIABLE
IB      000010  INTEGER*2  ARRAY (3)
ISTRT   000016  INTEGER*2  VARIABLE
ISTP    000020  INTEGER*2  VARIABLE

COMMON BLOCK /FCCOM/    LENGTH 030144

IK      000000  INTEGER*2  ARRAY (2)
NPNTS   000004  INTEGER*2  VARIABLE
NLINES  000006  INTEGER*2  VARIABLE
VAL     000010  REAL*4     VARIABLE
IJ      000014  INTEGER*2  ARRAY (10)
IST     000040  INTEGER*2  VARIABLE
IPLOC   000042  INTEGER*2  VARIABLE
IL      000044  INTEGER*2  ARRAY (8)
MAX     000064  INTEGER*2  VARIABLE
IH      000066  INTEGER*2  ARRAY (2)
ICNT    000072  INTEGER*2  VARIABLE
II      000074  INTEGER*2  ARRAY (6)
OUTBUF  000110  REAL*4     ARRAY (3072)
XSCALE  030110  REAL*4     VARIABLE
YSCALE  030114  REAL*4     VARIABLE
ZSCALE  030120  REAL*4     VARIABLE
PHI     030124  REAL*4     VARIABLE
THETA   030130  REAL*4     VARIABLE
XREF    030134  REAL*4     VARIABLE
YREF    030140  REAL*4     VARIABLE

COMMON BLOCK /PEXCOM/   LENGTH 000120

IC      000000  INTEGER*2  ARRAY (5)
ICDB    000012  INTEGER*2  VARIABLE
IG      000014  INTEGER*2  VARIABLE
ISBLK   000016  INTEGER*2  VARIABLE
IM      000020  INTEGER*2  ARRAY (23)
IXL1    000076  INTEGER*2  VARIABLE
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IYL1    000100  INTEGER*2 VARIABLE
IN      000102  INTEGER*2 ARRAY (6)
INCR    000116  INTEGER*2 VARIABLE
*
```

```
0001          SUBROUTINE FLWCLS
0002          COMMON /FCCOM/IA(17),IPLOC,IB,NUM1,NUMF,IC,IFSAV,IEAV,ID(5),
             $ICNT,IE(6),OUTBUF(3072)
0003          COMMON /PEXCOM/IG(5),ICDB,IH,ISBLK
0004          COMMON /FORIER/DAT(2050)
0005          NUM1 = 256
0006          IBR = 0
0007    4     DO 1 I = 1,3072
0008          OUTBUF(I) = 0
0009    1     CONTINUE
0010          NUMF = 0
0011          IF(IBR .EQ. 0) GO TO 6
0013          ICDB = -1
0014          ISBLK = 1
0015          ICNT = 0
0016          CALL PEXF
0017          CALL PEXF
0018    6     CALL PGEN
0019          AMAX = 0.0
0020          IPLOC = 0
0021          CALL ERASE
0022          CALL HOME
0023          IF(IBR .EQ. 1) GO TO 3
0025          DO 8 I=1,1024
0026          DAT(I) = OUTBUF(I+2048)
0027    8     CONTINUE
0028          CALL PLTBUF(0)
0029          CALL CHOUTS(23,1)
0030          CALL FSDF
0031          CALL RITE
0032          IFSAV = IPLOC
0033          IBR = 1
0034          NUM1 = 512
0035          GO TO 4
0036          IPLOC = 0
0037    3     CONTINUE
0038          CALL RITE
0039          DO 9 I = 1,1024
0040          DAT(I) = OUTBUF(I+2048)
0041    9     CONTINUE
0042          CALL PLTBUF(0)
0043          CALL CHOUTS(23,1)
0044          CALL LENGTH
0045          IEAV = IPLOC
0046          CALL FLWDEF
0047          RETURN
0048          END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

IBR      000022   INTEGER*2 VARIABLE
I        000024   INTEGER*2 VARIABLE
PEXF     000000   REAL*4    PROCEDURE
PGEN     000000   REAL*4    PROCEDURE
AMAX     000026   REAL*4    VARIABLE
ERASE    000000   REAL*4    PROCEDURE
HOME     000000   REAL*4    PROCEDURE
PLTBUF   000000   REAL*4    PROCEDURE
CHOUTS   000000   REAL*4    PROCEDURE
FSDF     000000   REAL*4    PROCEDURE
RITE     000000   REAL*4    PROCEDURE
LENGTH   000000   INTEGER*2 PROCEDURE
FLWDEF   000000   REAL*4    PROCEDURE


COMMON BLOCK /FCCOM/    LENGTH 030110

IA       000000   INTEGER*2 ARRAY (17)
IPLOC    000042   INTEGER*2 VARIABLE
IB       000044   INTEGER*2 VARIABLE
NUM1     000046   INTEGER*2 VARIABLE
NUMF     000050   INTEGER*2 VARIABLE
IC       000052   INTEGER*2 VARIABLE
IFSAV    000054   INTEGER*2 VARIABLE
IEAV     000056   INTEGER*2 VARIABLE
ID       000060   INTEGER*2 ARRAY (5)
ICNT     000072   INTEGER*2 VARIABLE
IE       000074   INTEGER*2 ARRAY (6)
OUTBUF   000110   REAL*4    ARRAY (3072)

COMMON BLOCK /PEXCOM/   LENGTH 000020

IG       000000   INTEGER*2 ARRAY (5)
ICDB     000012   INTEGER*2 VARIABLE
IH       000014   INTEGER*2 VARIABLE
ISBLK    000016   INTEGER*2 VARIABLE

COMMON BLOCK /FORIER/   LENGTH 020010

DAT      000000   REAL*4    ARRAY (2050)
$
```

```
0001          SUBROUTINE FLWDEF
0002          COMMON /FCCOM/IA(17), IPLOC, IR(4),IFSAV,
             $IEAV, ITP, IDP, MAX, IBTM
0003          DIMENSION IFLAW(10), IFLAW1(10)
0004          ITP = 5                !SET FLAW TYPE TO UNKNOWN
        C
        C     FIND ALL FLAWS DISPLAYING SAME FREQ. SHIFT CHARS.
        C
0005          CALL NNS(IFSAV, 1, IFLAW, NUM)
0006          NUM1 = NUM
0007          DO 29 I = 1,NUM1
0008          IFLAW1(I) = IFLAW(I)
0009    29    CONTINUE
        C
        C     FIND ALL FLAWS DISPLAYING SAME LENGTH CHARACTERISTICS
        C
0010          CALL NNS(IEAV, 2, IFLAW, NUM)
        C
        C     DO ANY FLAWS MATCH
        C
0011          IFL = 100
0012          DO 30 I1 = 1,NUM1
0013          DO 31 I = 1,NUM
0014          IF(IFLAW1(I1) .NE. IFLAW(I)) GO TO 31
0016          IFL = IFLAW(I)
0017          WRITE(7,100)IFL
0018    100   FORMAT(3X,'MATCHED FLAW = ',I2)
0019    31    CONTINUE
0020    30    CONTINUE
0021          IF(IFL .EQ. 100) GO TO 28
0023          GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,
             $18, 19, 20, 21, 22, 23, 24, 25, 26, 27) IFL
0024    1     ITP = 1
0025          IDP = 100
0026          IRD = 100
0027          IBTM = 2
0028          GO TO 28
0029    2     ITP = 1
0030          IDP = 100
0031          IRD = 80
0032          IBTM = 2
0033          GO TO 28
0034    3     ITP = 1
0035          IDP = 100
0036          IRD = 60
0037          IBTM = 2
0038          GO TO 28
0039    4     ITP = 1
0040          IDP = 100
0041          IRD = 40
0042          IBTM = 2
0043          GO TO 28
0044    5     ITP = 1
0045          IDP = 100
```

```
0046          IRD = 20
0047          IBTM = 2
0048          GO TO 28
0049    6     ITP = 1
0050          IDP = 60
0051          IRD = 100
0052          IBTM = 2
0053          GO TO 28
0054    7     ITP = 1
0055          IDP = 60
0056          IRD = 80
0057          IBTM = 2
0058    8     ITP = 1
0059          IDP = 60
0060          IRD = 60
0061          IBTM = 2
0062          GO TO 28
0063    9     ITP = 1
0064          IDP = 60
0065          IRD = 40
0066          IBTM = 2
0067          GO TO 28
0068    10    ITP = 1
0069          IDP = 60
0070          IRD = 20
0071          IBTM = 2
0072          GO TO 28
0073    11    CONTINUE
0074    12    CONTINUE
0075    13    CONTINUE
0076    14    CONTINUE
0077    15    CONTINUE
0078    16    CONTINUE
0079    17    CONTINUE
0080    18    CONTINUE
0081    19    CONTINUE
0082    20    CONTINUE
0083    21    CONTINUE
0084    22    CONTINUE
0085    23    CONTINUE
0086    24    CONTINUE
0087    25    CONTINUE
0088    26    CONTINUE
0089    27    CONTINUE
0090    28    RETURN
0091          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IFLAW   000014  INTEGER*2 ARRAY (10)
IFLAW1  000040  INTEGER*2 ARRAY (10)
NNS     000000  INTEGER*2 PROCEDURE
NUM     000120  INTEGER*2 VARIABLE
NUM1    000122  INTEGER*2 VARIABLE
I       000124  INTEGER*2 VARIABLE
IFL     000126  INTEGER*2 VARIABLE
I1      000130  INTEGER*2 VARIABLE
IRD     000132  INTEGER*2 VARIABLE

COMMON BLOCK /FCCOM/    LENGTH 000070

IA      000000  INTEGER*2 ARRAY (17)
IPLOC   000042  INTEGER*2 VARIABLE
IB      000044  INTEGER*2 ARRAY (4)
IFSAV   000054  INTEGER*2 VARIABLE
IEAV    000056  INTEGER*2 VARIABLE
ITP     000060  INTEGER*2 VARIABLE
IDP     000062  INTEGER*2 VARIABLE
MAX     000064  INTEGER*2 VARIABLE
IBTM    000066  INTEGER*2 VARIABLE
$
```

```
                                                .TITLE  FNAM
                                                .GLOBL  FNAM,CRLF,CHOUT
                                                .MCALL  SAVE,UNSAVE,.REGDEF,..V2...,CHSTUP,CHTXT
                                                ..V2...
                                                .REGDEF
000000                                  FNAM:   SAVE    012345
000014  004767  000000G                F3:     JSR     PC,CRLF
000020  005001                                  CLR     R1              ;R1 IS BUFFER POINTER
000022  005002                                  CLR     R2              ;R2 IS EXTENSION COUNTER
000024  005003                                  CLR     R3              ;R3 IS "IN EXTENSION" FLAG
000026  012705  000316'                         MOV     #ECHO,R5
000032  005067  000000'                1$:     CLR     ICHBUF
000036  005767  000000'                2$:     TST     ICHBUF
000042  001775                                  BEQ     2$              ;WAIT FOR INPUT
000044  026727  000000' 000056                 CMP     ICHBUF,#56      ;CHECK FOR "."
000052  001007                                  BNE     3$
000054  005703                                  TST     R3              ;CHECK FOR "." PREVIOUSLY RECEIVED
000056  001365                                  BNE     1$
000060  005701                                  TST     R1
000062  001763                                  BEQ     1$
000064                                  21$:
000064  012703  000001                         MOV     #1,R3           ;SET "IN EXTENSION FLAG"
000070  000415                                  BR      4$
000072  020127  000006                3$:     CMP     R1,#6           ;IF   6TH CHAR. NOT "."
000076  001002                                  BNE     31$             ;AND EXTENSION NOT BEGUN
000100  005703                                  TST     R3              ;START OVER
000102  001744                                  BEQ     F3
000104                                  31$:
000104  026727  000000' 000101                 CMP     ICHBUF,#101     ;CHECK FOR "A"
000112  002747                                  BLT     1$
000114  026727  000000' 000132                 CMP     ICHBUF,#132     ;CHECK FOR "Z"
000122  003343                                  BGT     1$
000124  116761  000000' 000024' 4$:     MOVB    ICHBUF,FBUF(R1)
000132  005201                                  INC     R1
000134  016767  000000' 000162                 MOV     ICHBUF,KCHBUF
000142  004767  000000G                        JSR     PC,CHOUT
000146  060302                                  ADD     R3,R2
000150  020227  000004                         CMP     R2,#4           ;CHECK FOR END O EXTENSION
000154  002726                                  BLT     1$
000156  010167  000036'                         MOV     R1,NUML         ;LENGTH OF FILENAME
000162  004767  000000G                        JSR     PC,CRLF
000166  012702  000024'                         MOV     #FBUF,R2        ;R2 IS ASCII
000172  012703  000330'                         MOV     #RBUF+2,R3      ;R3 IS RAD50
000176  004767  000040                         JSR     PC,A2R50        ;CONVERT FIRST THREE CHARS.
000202  012703  000332'                         MOV     #RBUF+4,R3
000206  004767  000030                         JSR     PC,A2R50
000212  005202                                  INC     R2              ;SKIP OVER THE "."
000214  012703  000334'                         MOV     #RBUF+6,R3
000220  004767  000016                         JSR     PC,A2R50        ;CONVERT EXTENSION
000224                                  UNSAVE  543210
000240  000207                                  RTS     PC              ;RETURN TO CALLING PROGRAM
000242                                  A2R50:  SAVE    540
000250  005005                                  CLR     R5
000252  012704  000003                         MOV     #3,R4
000256  070527  000050                1$:     MUL     #50,R5
000262  122227  000040                         CMPB    (R2)+,#40
000266  001405                                  BEQ     2$
```

```
      58 000270  005302                          DEC     R2
      59 000272  112200                          MOVB    (R2)+,R0
      60 000274  060005                          ADD     R0,R5
      61 000276  162705  000100                  SUB     #100,R5
      62 000302  077413                  2$:     SOB     R4,1$
      63 000304  010513                          MOV     R5,(R3)
      64 000306                                  UNSAVE  045
      65 000314  000207                          RTS     PC
      66 000316  000001  000322' 000001  ECHO:   .WORD   1,ECHO+4,1
      67 000324  000000                  KCHBUF: .WORD   0
      68 000326  015327                  RBUF:   .RAD50  /DK1/
      69                                         .BLKW   3
      70         000000'                         .CSECT  FCCOM
      71                                         .BLKW   8.
      72 000020     104      113                 .ASCII  /DK/
      73 000022     061      072                 .ASCII  /1:/
      74 000024                          FBUF:   .BLKB   10.
      75 000036  000000                  NUML:   .WORD   0
      76         000000'                         .CSECT  TEKCOM
      77 000000  000000                  ICHBUF: .WORD   0
      78         000001'                         .END
```

```
FNAM    RT-11 MACRO VM02-12   15-SEP-80 PAGE 1+
SYMBOL TABLE

A2R50   C00242R          CHOUT = ****** G        CRLF  = ****** G        ECHO    000316R          FBUF    000024R    002
FNAM    000000RG         F3      000014R         ICHBUF  000000R    003 KCHBUF  000324R          NUML    000036R    002
PC    =%000007           RBUF    000326R         R0    =%000000          R1    =%000001          R2    =%000002
R3    =%000003           R4    =%000004          R5    =%000005          SP    =%000006          ...V2 = 000001

. ABS.  000000     000
        000336     001
FCCOM   000040     002
TEKCOM  000002     003
ERRORS DETECTED: 0
FREE CORE: 17796. WORDS

,TT:/N:TTM=FNAM

ERRORS DETECTED: 0
FREE CORE: 17796. WORDS

.
```

```
0001          SUBROUTINE FOP
0002          COMMON /FCCOM/ IA(8), IFBUF(7),NUML
0003          COMMON /PEXCOM/ IB(22), IH(256)
0004          CALL FNAM              !GET FILENAME
0005          ICNT = 4 + NUML
0006          CALL ASSIGN(2, IFBUF, ICNT, 'KDO', 'NC', 1)          !ASSIGN LUN TO FILE
0007          DEFINE FILE 2(2882, 256, U, IVAL )            !OPEN FILE
0008          RETURN
0009          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

FNAM    000000  REAL*4     PROCEDURE
ICNT    000034  INTEGER*2  VARIABLE
ASSIGN  000000  REAL*4     PROCEDURE
IVAL    000036  INTEGER*2  VARIABLE


COMMON BLOCK /FCCOM/    LENGTH 000040

IA      000000  INTEGER*2  ARRAY (8)
IFBUF   000020  INTEGER*2  ARRAY (7)
NUML    000036  INTEGER*2  VARIABLE

COMMON BLOCK /PEXCOM/   LENGTH 001054

IB      000000  INTEGER*2  ARRAY (22)
IH      000054  INTEGER*2  ARRAY (256)
*
```

```
0001          SUBROUTINE FORGEN  !FORMULA GENERATION ROUTINE
0002          COMMON /PEXCOM/ IA(31), IXL1, IYL1, IXL2, IYL2, IB(4), INCR
0003          COMMON /FCCOM/ IC(2), NPNTS, NLINES
0004          DIMENSION C(4)
          C
          C     COMPENSATE FOR NEGATIVE COORD.
          C
0005          C(1) = FLOAT( IXL1 )
0006          C(2) = FLOAT( IXL2 )
0007          C(3) = FLOAT(IYL1 )
0008          C(4) = FLOAT( IYL2 )
0009          DO 1 I = 1,4
0010          IF(C(I) .GE. 0) GO TO 1
0012          C(I) = 65535. + C(I)                    !CORRECTION FACTOR
0013     1    CONTINUE
          C
          C     CALCULATE THE NUMBER OF POINTS & LINES
          C
0014          NPNTS = IFIX((C(2) - C(1)) / FLOAT(INCR))
0015          NLINES = IFIX((C(4) - C(3)) / FLOAT(INCR))
          C
          C     RETURN TO CALLING ROUTINE
          C
0016          RETURN
0017          END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

C        000014   REAL*4     ARRAY (4)
FLOAT    000000   REAL*4     PROCEDURE
I        000040   INTEGER*2  VARIABLE
IFIX     000000   INTEGER*2  PROCEDURE


COMMON BLOCK /PEXCOM/   LENGTH 000120

IA       000000   INTEGER*2  ARRAY (31)
IXL1     000076   INTEGER*2  VARIABLE
IYL1     000100   INTEGER*2  VARIABLE
IXL2     000102   INTEGER*2  VARIABLE
IYL2     000104   INTEGER*2  VARIABLE
IB       000106   INTEGER*2  ARRAY (4)
INCR     000116   INTEGER*2  VARIABLE

COMMON BLOCK /FCCOM/    LENGTH 000010

IC       000000   INTEGER*2  ARRAY (2)
NPNTS    000004   INTEGER*2  VARIABLE
NLINES   000006   INTEGER*2  VARIABLE
*
```

```
0001          PROGRAM FORMAT
0002          COMMON /FCCOM/IFDFLG,IPROC,NPNTS,NLINES,IA(3),IPLT,IB(20),
             $IFTT,IC(7),OUTBUF(3072),XSCALE,YSCALE,ZSCALE,PHI,THETA,
             $XREF,YREF
      C
      C       INITIALIZE PROGRAM
      C
0003          CALL TEKIN
0004          IFTT = 0
0005          IPROC = 0
      C
      C       ENTER THE COMMAND ROUTINE
      C
0006    2     CALL CNTRL
0007          IF(IPLT .EQ. 1) GO TO 18      !ORTHONORMAL
0009          IF(IPLT .EQ. 2) GO TO 19      !10-DEG. TILT
0011          IF(IPLT .EQ. 3) GO TO 20      !SIDE
0013          IF(IPLT .EQ. 4) GO TO 21      !FRONT
0015          IF(IPLT .EQ. 5) GO TO 23
      C
      C       ORTHONORMAL PLOT
      C
0017   18     XSCALE = .1
0018          YSCALE = 3.
0019          ZSCALE = .1
0020          PHI = 45.
0021          THETA = 45.
0022          XREF = 8.5
0023          YREF = .5
0024          GO TO 23
      C
      C     10-DEG. TILT
      C
0025   19     XSCALE = .1
0026          YSCALE = 3.
0027          ZSCALE = .1
0028          PHI = 45.
0029          THETA = 10.
0030          XREF = 7.5
0031          YREF = 2.5
0032          GO TO 23
      C
      C       SIDE VIEW
      C
0033   20     XSCALE = .1
0034          YSCALE = 3.
0035          ZSCALE = .1
0036          PHI = 90.
0037          THETA = 0.
0038          XREF = 10.5
0039          YREF = 4.0
0040          GO TO 23
      C
      C       FRONT VIEW
```

```
      C
0041  21    XSCALE = .1
0042        YSCALE = 3.
0043        ZSCALE = .1
0044        PHI = 0.
0045        THETA = 0.
0046        XREF = 2.
0047        YREF = 4.0
0048  23    IF(IFDFLG .EQ. 1) GO TO 40
0050        CALL LABEL(PHI, THETA)                !LABEL THE PLOT
0051  41    IF(IPLT .EQ. 5) GO TO 42    !IF CUSTOM PLOT, DO NOT ADJUST PARAMETERS
      C
      C     ADJUST PLOT PARAMETERS TO COMPENSATE FOR PLOT SIZE
      C
0053        IF(NPNTS .GT. 100) GO TO 37
0055        XSCALE = XSCALE + .000022 * (FLOAT( 100-NPNTS )) ** 2
0056        GO TO 38
0057  37    XSCALE = XSCALE - .006 * SQRT( FLOAT(NPNTS - 100))
0058  38    IF(NLINES .GT. 100)GO TO 39
0060        ZSCALE = ZSCALE + .000022 * (FLOAT(100 - NLINES)) ** 2
0061        GO TO 42
0062  39    ZSCALE = ZSCALE - .006 * SQRT( FLOAT(NLINES - 100))
0063        IF(IFDFLG .EQ. 0)GO TO 42
0065  40    CALL FLWCLS
0066        GO TO 2
0067  42    CALL FLWCHR
0068        CALL BELL
0069        CALL PAUSE
0070        GO TO 2
0071        STOP
0072        END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET  ATTRIBUTES

TEKIN    000000  REAL*4     PROCEDURE
CNTRL    000000  REAL*4     PROCEDURE
LABEL    000000  INTEGER*2  PROCEDURE
FLOAT    000000  REAL*4     PROCEDURE
SQRT     000000  REAL*4     PROCEDURE
FLWCLS   000000  REAL*4     PROCEDURE
FLWCHR   000000  REAL*4     PROCEDURE
BELL     000000  REAL*4     PROCEDURE
PAUSE    000000  REAL*4     PROCEDURE

COMMON BLOCK /FCCOM/    LENGTH 030144

IFDFLG   000000  INTEGER*2 VARIABLE
IPROC    000002  INTEGER*2 VARIABLE
NPNTS    000004  INTEGER*2 VARIABLE
NLINES   000006  INTEGER*2 VARIABLE
IA       000010  INTEGER*2 ARRAY (3)
IPLT     000016  INTEGER*2 VARIABLE
IB       000020  INTEGER*2 ARRAY (20)
IFTT     000070  INTEGER*2 VARIABLE
IC       000072  INTEGER*2 ARRAY (7)
OUTBUF   000110  REAL*4    ARRAY (3072)
XSCALE   030110  REAL*4    VARIABLE
YSCALE   030114  REAL*4    VARIABLE
ZSCALE   030120  REAL*4    VARIABLE
PHI      030124  REAL*4    VARIABLE
THETA    030130  REAL*4    VARIABLE
XREF     030134  REAL*4    VARIABLE
YREF     030140  REAL*4    VARIABLE
*
```

```
0001          SUBROUTINE FSDF
0002          COMMON / FORIER / DAT(2048)
0003          COMMON / FCCOM / IA(17), IPLOC
0004          AMAX = 0.0
0005          DO 5 I = 30,1024
0006          IF (DAT(I) .LE. AMAX) GO TO 5
0008          IPLOC = I
0009          AMAX = DAT(I)
0010    5     CONTINUE
0011          RETURN
0012          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

AMAX    000014  REAL*4     VARIABLE
I       000020  INTEGER*2  VARIABLE

COMMON BLOCK /FORIER/   LENGTH 020000

DAT     000000  REAL*4     ARRAY (2048)

COMMON BLOCK /FCCOM/    LENGTH 000044

IA      000006  INTEGER*2  ARRAY (17)
IPLOC   000042  INTEGER*2  VARIABLE
*
```

```
                                                        .TITLE  FT  ;PERFORMS FOURRIER OPERATIONS
                                                        .GLOBL  FT
                                                        .GLOBL  ACUMHI,ACUMLO
                                                        .GLOBL  FN1,MWDCT,CFT,CTO,CMCS
                                                        .GLOBL  CTABL,TAD,LTMP,INVFG
                                                        .MCALL  SAVE,UNSAVE,.REGDEF,..V2...
                                                        ..V2...
                                                        .REGDEF
                                        .OBF=0
                                        .NLN=2
                                        .SCL=4
                                        .FLG=6
000000                                  FT:     SAVE    012345
000014  005067  001102                        CLR     E0
000020  000403                               BR      FT2
000022  012767  000001  001072          FT1:    MOV     #1,E0
000030  005067  000000'                  FT2:    CLR     MD
                                        ;
                                        ;       COPY REQUESTED DATA TO MDATA
                                        ;
000034  005367  000020'                       DEC     STOPX
000040  016703  000020'                       MOV     STOPX,R3
000044  166703  000016'                       SUB     STARTX,R3
000050  005203                               INC     R3
000052  012701  004000                       MOV     #2048.,R1
000056  005000                               CLR     R0
000060  071003                               DIV     R3,R0
000062  012702  000040'                       MOV     #IBUF,R2
000066  012704  002006'                       MOV     #MDATA,R4
000072  066702  000016'                       ADD     STARTX,R2
000076  006300                               ASL     R0
000100  112214                          MV1:    MOVB    (R2)+,(R4)              ;SPREADS OUT THE DATA
000102  060004                               ADD     R0,R4
000104  077303                               SOB     R3,MV1
000106  112214                               MOVB    (R2)+,(R4)
                                        ;
                                        ;       INTERPOLATE BETWEEN VALUES
                                        ;
000110  020027  000002                  MV2:    CMP     R0,#2
000114  003417                               BLE     MV4
000116  012702  002006'                       MOV     #MDATA,R2
000122  010003                               MOV     R0,R3
000124  006203                               ASR     R3
000126  060203                               ADD     R2,R3
000130  011201                          MV3:    MOV     (R2),R1         ;INTERPOLATE
000132  060002                               ADD     R0,R2
000134  061201                               ADD     (R2),R1
000136  006201                               ASR     R1
000140  010113                               MOV     R1,(R3)
000142  060003                               ADD     R0,R3
000144  020204                               CMP     R2,R4
000146  001370                               BNE     MV3
000150  006200                               ASR     R0
000152  000756                               BR      MV2
                                        ;
                                        ;       REMOVE DC COMPONENT
                                        ;
```

```
      58 000154  012767  174000  000000G MV4:    MOV     #-2048.,MWDCT
      59 000162  012767  002006' 000736         MOV     #MDATA,EACC+2
      60 000170  012767  002006' 000752         MOV     #MDATA,ESUB+6
      61 000176  012767  002006' 000750         MOV     #MDATA,ESUB+10.
      62 000204  012705  001124'                MOV     #EACC,R5
      63 000210  004767  000000G                JSR     PC,FN1
      64 000214  016700  000000G                MOV     ACUMHI,R0
      65 000220  016701  000000G                MOV     ACUMLO,R1
      66 000224  073027  000005                 ASHC    #5,R0
      67 000230  010067  000724                 MOV     R0,AVG
      68 000234  012705  001142'                MOV     #ESUB,R5
      69 000240  004767  000000G                JSR     PC,FN1
      70                                    ;
      71                                    ;       SET UP FOR FFT
      72                                    ;
      73 000244  012767  001164' 000000G        MOV     #TEMP,TAD
      74 000252  012767  004004  000000G        MOV     #2052.,LTMP
      75 000260  012767  002006' 000000G        MOV     #MDATA,CTABL+.OBF
      76 000266  012767  002000  000002G        MOV     #1024.,CTABL+.NLN
      77 000274  012767  177770  000004G        MOV     #-8.,CTABL+.SCL
      78 000302  012767  177770  000006G        MOV     #-8.,CTABL+.FLG
      79 000310  005067  000000G                CLR     INVFG
      80                                    ;
      81                                    ;       PERFORM FFT
      82                                    ;
      83 000314  004767  000000G                JSR     PC,CFT
      84 000320  005702                         TST     R2
      85 000322  001401                         BEQ     RCONV
      86 000324  000000                         HALT
      87 000326  016700  000002G        RCONV:  MOV     CTABL+.NLN,R0   ;SET UP FOR REAL COORDINATE CONVERSION
      88 000332  006300                         ASL     R0
      89 000334  006300                         ASL     R0
      90 000336  016701  000000G                MOV     CTABL+.OBF,R1
      91 000342  060100                         ADD     R1,R0
      92 000344  012120                         MOV     (R1)+,(R0)+
      93 000346  012110                         MOV     (R1)+,(R0)
      94 000350  004767  000000G                JSR     PC,CTO          ;CONVERT ARRAY
      95 000354  005702                         TST     R2
      96 000356  001401                         BEQ     2$
      97 000360  000000                         HALT
      98 000362  005267  000004G        2$:     INC     CTABL+.SCL      ;SUPPOSED TO BE DONE BY CTO, BUT IS NOT
      99 000366  005767  000002'        29$:    TST     IFFT            ;CHECK INVERSE FFT REQUESTED
     100 000372  001447                         BEQ     CM
     101 000374  012767  177777  000000G        MOV     #-1,INVFG
     102 000402  016700  000002G                MOV     CTABL+.NLN,R0   ;SET UP FOR INVERSE FFT
     103 000406  006300                         ASL     R0
     104 000110  016701  000000G                MOV     CTABL+.OBF,R1
     105 000414  060001                         ADD     R0,R1           ;ADRS CENTER
     106 000416  060100                         ADD     R1,R0           ;ADRS   END
     107 000420  012120                         MOV     (R1)+,(R0)+
     108 000422  011110                         MOV     (R1),(R0)
     109 000424  004767  000000G                JSR     PC,CTO
     110 000430  005702                         TST     R2
     111 000432  001401                         BEQ     3$
     112 000434  000000                         HALT
     113 000436  005267  000004G        3$:     INC     CTABL+.SCL
     114 000442  004767  000000G                JSR     PC,CFT          ;PERFORM INVERSE FFT
```

```
115 000446  005702                          TST     R2
116 000450  001401                          BEQ     4$
117 000452  000000                          HALT
118 000454                          4$:     ;SHIFT RESULTS TO CONFORM TO ORIGINAL SCALE FACTOR
119 000454  016701  000004G                 MOV     CTABL+,SCL,R1
120 000460  012702  002006'                 MOV     #MDATA,R2
121 000464  012703  004000                  MOV     #2048.,R3
122 000470  005401                          NEG     R1
123 000472  162701  000010                  SUB     #8.,R1
124 000476  011200                  5$:     MOV     (R2),R0
125 000500  072001                          ASH     R1,R0
126 000502  010022                          MOV     R0,(R2)+
127 000504  077304                          SOB     R3,5$
128 000506  000167  000360                  JMP     OUT
129 000512  012701  002006'          CM:     MOV     #MDATA,R1       ;SAVE REAL AND IMAGINARY VALUES
130 000516  012702  005170'                 MOV     #TMP2,R2
131 000522  012703  002000                  MOV     #1024.,R3
132 000526  012112                  21$:    MOV     (R1)+,(R2)
133 000530  012162  004000                  MOV     (R1)+,2048.(R2)
134 000534  005722                          TST     (R2)+
135 000536  077305                          SOB     R3,21$
136 000540  004767  000000G                 JSR     PC,CMCS         ;CALCULATE MAGNITUDE COMPLEX SPECTRUM
137 000544  005702                          TST     R2
138 000546  001401                          BEQ     3$
139 000550  000000                          HALT
140                                 ;       CALCULATE PHASE ANGLE
141 000552  012767  176000  000000G 3$:     MOV     #-1024.,MWBCT   ;SET UP TO CALCULATE COSINES
142 000560  012767  002006' 014404          MOV     #MDATA,EDIV+2
143 000566  012767  006006' 014406          MOV     #MDATA+2048.,EDIV+10.
144 000574  012705  015170'                 MOV     #EDIV,R5
145 000600  004767  000000G                 JSR     PC,FN1
146 000604  012703  002000                  MOV     #1024.,R
147 000610  012700  006006'                 MOV     #MDATA+2048.,R0
148 000614  011004                  NXV:    MOV     (R0),R4
149 000616  012702  001000                  MOV     #512.,R2        ;FIND ARC-COSINES
150 000622  010201                          MOV     R2,R1
151 000624  005704                          TST     R4
152 000626  002001                          BGE     NXT
153 000630  005404                          NEG     R4
154 000632  006201                  NXT:    ASR     R1
155 000634  026204  000004'                 CMP     KOSINE(R2),R4
156 000640  002410                          BLT     2$
157 000642  003002                          BGT     1$
158 000644  000167  000024                  JMP     FND
159 000650  032701  000001          1$:     BIT     #1,R1
160 000654  001007                          BNE     FND
161 000656  060102                          ADD     R1,R2
162 000660  000764                          BR      NXT
163 000662  032701  000001          2$:     BIT     #1,R1
164 000666  001002                          BNE     FND
165 000670  160102                          SUB     R1,R2
166 000672  000757                          BR      NXT
167 000674  006202                  FND:    ASR     R2
168 000676  005402                          NEG     R2
169 000700  062702  001000                  ADD     #512.,R2
170 000704  010220                          MOV     R2,(R0)+
171 000706  077336                          SOB     R3,NXV
```

```
172 000710  012704  005170'                 MOV     #TMP2,R4
173 000714  012702  006006'                 MOV     #MDATA+2048.,R2
174 000720  012703  002000                  MOV     #1024.,R3
175 000724  011200                  LP:     MOV     (R2),R0         ;CONVERT INDEX TO DEGREES
176 000726  070027  026400                  MUL     #26400,R0       ;R0=R1*90/512
177 000732  005714                          TST     (R4)
178 000734  002003                          BGE     1$
179 000736  005400                          NEG     R0
180 000740  062700  000264                  ADD     #180.,R0
181 000744  005764  004000          1$:     TST     2048.(R4)
182 000750  002001                          BGE     2$
183 000752  005400                          NEG     R0
184 000754  010022                  2$:     MOV     R0,(R2)+
185 000756  005724                          TST     (R4)+
186 000760  077317                          SOB     R3,LP
187                                         ;SHIFT  MCS DATA TO CONFORM TO ORIGINAL SCALE FACTOR
188 000762  016704  000046G                 MOV     CTABL+.SCL,R4   ;GET SCALE
189 000766  012702  002006'                 MOV     #MDATA,R2
190 000772  012703  002000                  MOV     #1024.,R3
191 000776  005404                          NEG     R4
192 001000  162704  000010                  SUB     #8.,R4
193 001004  016701  000066G                 MOV     CTABL+.FLG,R1           ;GET MAGNITUDE FLAG
194 001010  060401                          ADD     R4,R1
195 001012  020127  177766                  CMP     R1,#-10.        ;CHECK RESULTANT MAGNITUDE GE 16
196 001016  003002                          BGT     5$
197 001020  005267  000000'                 INC     MD
198 001024  011201                  5$:     MOV     (R2),R1
199 001026  006700                          SXT     R0
200 001030  005767  000000'                 TST     MD              ;#100 IF VALUES .LT. 16
201 001034  001403                          BEQ     55$
202 001036  010100                          MOV     R1,R0
203 001040  070027  000144                  MUL     #100.,R0
204 001044  073004                  55$:    ASHC    R4,R0           ;MCS DATA SHIFTED
205 001046  020127  000002                  CMP     R1,#2
206 001052  003005                          BGT     6$
207 001054  005062  004000                  CLR     2048.(R2)       ;CLEAR PHASE VALUE IF MAG LE 2
208 001060  003002                          BGT     6$
209 001062  005062  004000                  CLR     2048.(R2)
210 001066  010122                  6$:     MOV     R1,(R2)+
211 001070  077323                          SOB     R3,5$
212 001072  005767  000024          OUT:    TST     EQ
213 001076  001002                          BNE     FT3
214 001100  000167  176716                  JMP     FT1
215 001104                          FT3:    UNSAVE  543210
216 001120  000207                          RTS     PC
217 001122  000000                  EQ:     .WORD   0
218 001124  002260  000000  000002  EACC:   .WORD   2260,0,2,0,0,0,0
    001132  000000  000000  000000
    001140  000000
219 001142  001402  001160' 000000  ESUB:   .WORD   1402,AVG,0,0,2,0,2
    001150  000000  000002  000000
    001156  000002
220 001160                          AVG:    .BLKW   2
221 001164                          TEMP:   .BLKW   1026.
222 005170                          TMP2:   .BLKW   2048.
223 015170  002246  000000  000000  EDIV:   .WORD   2246,0,0,TMP2,0,0,0
    015176  005170' 000000  000000
```

```
    015204  000000
224         000000'                         .CSECT  EXDCOM
225                                         .BLKW   7.
226 000016  000000              STARTX: .WORD   0
227 000020  000000              STOPX:  .WORD   0
228                                         .BLKW   7.
229 000040                      IBUF:   .BLKB   2048.
230         000000'                         .CSECT  FORIER
231 000000  000000              MD:     .WORD   0
232 000002  000000              YFFT:   .WORD   0
233 000004                      KOSINE: .BLKW   513.
234 002C06                      MDATA:  .BLKW   2050.
235         000001'                         .END
```

```
FT  ;PERFORMS FOURRIER OPERATIO RT-11 MACRO VM02-12   15-SEP-80 PAGE 1+
SYMBOL TABLE

ACUMHI= ****** G        ACUMLO= ****** G        AVG     001160R        CFT   = ****** G        CM      000512R
CMCS  = ****** G        CTABL = ****** G        CTO   = ****** G       EACC    001124R        EDIV    015170R
ED      001122R         ESUB    001142R         FND     000674R        FN1   = ****** G        FT      000000RG
FT1     000022R         FT2     000030R         FT3     001104R        IBUF    000040R    002 IFFT    000002R    003
INVFG = ****** G        KOSINE  000004R    003  LP      000724R        LTMP  = ****** G        MD      000000R    003
MDATA   002006R    003  MV1     000100R         MV2     000110R        MV3     000130R        MV4     000154R
MWDCT = ****** G        NXT     000632R         NXV     000614R        OUT     001072R        PC    =%000007
RCONV   000326R         R0    =%000000          R1    =%000001         R2    =%000002         R3    =%000003
R4    =%000004          R5    =%000005          SP    =%000006         STARTX  000016R    002 STOPX   000C20R    002
TAD   = ****** G        TEMP    001164R         TMP2    005170R        .FLG  = 000006         .NLN  = 000002
.OBF  = 000000          .SCL  = 000004          ...V2 = 000001
. ABS.  000000     000
        015206     001
EXDCOM  004040     002
FORIER  012012     003
ERRORS DETECTED: 0
FREE CORE: 17728. WORDS

,TT:/N:TTM=FT

ERRORS DETECTED: 0
FREE CORE: 17728. WORDS

*
```

```
                                        .TITLE HEIGHT  ;HEIGHT CALCULATION
                                        .GLOBL  HEIGHT
                                        .MCALL  SAVE,UNSAVE,.REGDEF,..V2...
                                        .  V2...
                                        .REGDEF
000000                          HEIGHT: SAVE    012345
                                ;
                                ;       FIND SAMPLE RATE
                                ;
000014  016700  000114'                 MOV     IRNG,R0
000020  006300                          ASL     R0
000022  016001  000070'                 MOV     RTBL(R0),R1
                                ;
                                ;       CALCULATE DIST. TO PLATE
                                ;
000026  070167  000030'                 MUL     TWOFFS,R1
000032  005000                          CLR     R0
000034  071027  000310                  DIV     #200.,R0
000040  010001                          MOV     R0,R1
000042  070127  000073                  MUL     #59.,R1
                                ;
                                ;       PLACE HEIGHT IN COMMON
                                ;
000046  010167  000106'                 MOV     R1,ZPNT
000052                                  UNSAVE  543210
000066  000207                          RTS     PC
000070  000001  000002  000005  RTBL:   .WORD   1,2,5.,10.,50.,100.,200.,500.,1000.
000076  000012  000062  000144
000104  000310  000764  001750
        000000'                         .CSECT  PEXCOM
                                        .BLKW   12.
000030  000000                  TWOFFS: .WORD   0
                                        .BLKW   22.
000106  000000                  ZPNT:   .WORD   0
                                        .BLKW   2
000114  000000                  IRNG:   .WORD   0
        000001'                         .END
```

```
HEIGHT  HEIGHT CALCULATION     RT-11 MACRO VM02-12   15-SEP-80 PAGE 1

SYMBOL TABLE

HEIGHT  000000RG        IRNG    000114R    002  PC    =%000007          RTBL    000070R          R0    =%000000
R1    =%000001          R2    =%000002          R3    =%000003          R4    =%000004          R5    =%000005
SP    =%000006          TWOFFS  000030R    002  ZPNT    000106R    002  ...V2 = 000001
. ABS.  000000      000
        000112      001
PEXCOM  000116      002
ERRORS DETECTED: 0
FREE CORE: 17937. WORDS

,TT:/N:TTM=HEIGHT

ERRORS DETECTED: 0
FREE CORE: 17937. WORDS

$
```

```
0001          SUBROUTINE INT
0002          COMMON / EXDCOM / IA(7), ISTRT, ISTP, IB(7), IBUF(2048)
0003          COMMON / FORIER /  DAT(2050)
0004          LOGICAL*1 IBUF
0005          LIM = ISTP - ISTRT
0006          INC = 2048 / LIM
0007          DAT(1) = IBUF(ISTRT)
0008          DO 1 I = 1,LIM
0009          DAT(I*INC+1) = IBUF(I + ISTRT)
0010    1     CONTINUE
0011    3     IF (INC .EQ. 1) GO TO 4
0013          I1 = 1
0014          DO 2 I = 1,((2048/INC)-1)
0015          DAT(I1 + (INC/2)) = (DAT(I1) + DAT(I1+INC))/2
0016          I1 = I1 + INC
0017    2     CONTINUE
0018          INC = INC / 2
0019          GO TO 3
0020    4     RETURN
0021          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

LIM     000014  INTEGER*2 VARIABLE
INC     000016  INTEGER*2 VARIABLE
I       000020  INTEGER*2 VARIABLE
I1      000022  INTEGER*2 VARIABLE


COMMON BLOCK /EXDCOM/   LENGTH 004040

IA      000000  INTEGER*2 ARRAY (7)
ISTRT   000016  INTEGER*2 VARIABLE
ISTP    000020  INTEGER*2 VARIABLE
IB      000022  INTEGER*2 ARRAY (7)
IBUF    000040  LOGICAL*1 ARRAY (2048)

COMMON BLOCK /FORIER/   LENGTH 020010

DAT     000000  REAL*4    ARRAY (2050)
*
```

```
0001          SUBROUTINE ITOA(I,IA,N)
0002          LOGICAL*1 IA
0003          DIMENSION IA(10)
0004          A = FLOAT( I ) / 10.          !NORMALIZE THE NUMBER TO BE CONVERTED
0005          DO 2 L = 1,(N - 1)
0006          A = A / 10.
0007    2     CONTINUE
0008          DO 1 L = 1,N
0009          I1 = IFIX( A*10. )            !GET ONE DIGIT
0010          A = (A * 10.) - I1            !RENORMALIZE THE ORIGINAL NUMBER
0011          IA(L) = 48 + I1               !ADD 48 TO THE DIGIT TO GET ASCII
0012    1     CONTINUE
0013          RETURN
0014          END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

IA       000016   LOGICAL*1 PARAMETER ARRAY (10)
I        00001A   INTEGER*2 PARAMETER VARIABLE
N        000020   INTEGER*2 PARAMETER VARIABLE
A        000022   REAL*4    VARIABLE
FLOAT    000000   REAL*4    PROCEDURE
L        000026   INTEGER*2 VARIABLE
I1       000030   INTEGER*2 VARIABLE
IFIX     000000   INTEGER*2 PROCEDURE
*
```

```
0001          SUBROUTINE LABEL(PHI, THETA)
0002          COMMON /FCCOM/ IFD, IPTYP, NPNTS, NLINES
0003          COMMON /PEXCOM/ IC(35), IZPNT, ID(3), INC, IH,
             $IXD(3), ISMP(3)
0004          DIMENSION IE(5), IHDR(54)
0005          DATA IHDR /
             $'TI', 'ME', ' D', 'OM', 'AI', 'N ', 'FR', 'EQ', 'UE', 'NC',
             $'Y ', 'AL', 'TI', 'TU', 'DE', ': ', 'PH', 'I ', 'TH', 'ET',
             $'A ', 'TR', 'AN', 'SD', 'UC', 'ER', 'SA', 'MP', 'LE', ' S',
             $'PE', 'CT', 'RU', 'M ', 'EN', 'V.', ' P', 'EA', 'K ', 'X ',
             $'Y ', 'LE', 'NG', 'TH', ' S', 'PE', 'CT', 'RU', 'M ', 'EN',
             $'V.', ' P', 'EA', 'K '/
      C
      C       PRINT PLOT HEADING
      C
0006          CALL IPLOT(0, 710, 0)                    !MOVE THE CURSOR
0007          CALL PRTH(IHDR(45),20)
      C
      C       PRINT DOMAIN
      C
0008    7     CALL IPLOT(0, 650, 0)                    !MOVE THE CURSOR
0009          IF(IFD .EQ. 1) GO TO 8
0011          CALL CHOUTS(IHDR(1), 11)      !TIME DOMAIN
0012          GO TO 9
0013    8     CALL CHOUTS(IHDR(7),9)                   !FREQUENCY DOMAIN
0014          CALL CHOUTS(IHDR(3), 7)
      C
      C       PRINT ALTITUDE
      C
0015    9     CALL IPLOT(0, 625, 0)                    !MOVE THE CURSOR
0016          CALL CHOUTS(IHDR(12), 10)     !PRINT SUBHEADER
0017          CALL ITOA(IZPNT, IE, 5)       !CONVERT ALT. TO ASCII
0018          CALL CHOUTS(IE(1), 5)                    !PRINT ALTITUDE
      C
      C       PRINT ANGLE PHI
      C
0019          CALL IPLOT(0, 600, 0)         !MOVE THE CURSOR
0020          CALL CHOUTS(IHDR(17), 3)      !PRINT SUBHEADER
0021          CALL CHOUTS(IHDR(16), 2)
0022          IG = IFIX( PHI )
0023          CALL ITOA( IG, IE, 3)                    !CONVERT TO ASCII
0024          CALL CHOUTS(IE(1), 3)                    !PRINT ALTITUDE
      C
      C       PRINT TRANSDUCER ID
      C
0025          CALL IPLOT(750, 650, 0)
0026          CALL CHOUTS(IHDR(22), 10)     !PRINT SUBHEADING
0027          CALL CHOUTS(IHDR(16), 2)
0028          CALL CHOUTS(IXD(1), 6)                   !PRINT THE TRANSDUCER ID
      C
      C       PRINT THE SAMPLE ID
      C
0029          CALL IPLOT(750, 625, 0)                  !MOVE THE CURSOR
0030          CALL CHOUTS(IHDR(27), 6)      !PRINT SUBHEADING
```

```
0031          CALL CHOUTS(IHDR(16), 2)
0032          CALL CHOUTS(ISMP(1), 6)              !PRINT SAMPLE ID
      C
      C       PRINT ANGLE THETA
      C
0033          CALL IPLOT(750, 600, 0)
0034          CALL CHOUTS(IHDR(19), 5)             !PRINT SUBHEADER
0035          CALL CHOUTS(IHDR(16), 2)
0036          IG = IFIX( THETA )
0037          CALL ITOA(IG, IE, 3)                 !CONVERT TO ASCII
0038          CALL CHOUTS(IE(1), 3)                !PRINT THETA
      C
      C       PRINT X LENGTH
      C
0039          CALL IPLOT(0,575,0)
0040          CALL CHOUTS(IHDR(40),2)
0041          CALL CHOUTS(IHDR(42),6)
0042          CALL CHOUTS(IHDR(16),2)
0043          IG = NPNTS * INC
0044          CALL ITOA(IG,IE,5)
0045          CALL CHOUTS(IE(1),5)
      C
      C       PRINT Y LENGTH
      C
0046          CALL IPLOT(750,575,0)
0047          CALL CHOUTS(IHDR(41),2)
0048          CALL CHOUTS(IHDR(42),6)
0049          CALL CHOUTS(IHDR(16),2)
0050          IG = NLINES * INC
0051          CALL ITOA(IG,IE,5)
0052          CALL CHOUTS(IE(1),5)
      C
      C       RETURN TO CALLING PROGRAM
      C
0053          CALL HOME
0054          RETURN
0055          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IE      000020  INTEGER*2 ARRAY (5)
IHDR    000032  INTEGER*2 ARRAY (54)
PHI     000014  REAL*4    PARAMETER VARIABLE
THETA   000016  REAL*4    PARAMETER VARIABLE
IPLOT   000000  INTEGER*2 PROCEDURE
PRTH    000000  REAL*4    PROCEDURE
CHOUTS  000000  REAL*4    PROCEDURE
ITOA    000000  INTEGER*2 PROCEDURE
IG      000246  INTEGER*2 VARIABLE
IFIX    000000  INTEGER*2 PROCEDURE
HOME    000000  REAL*4    PROCEDURE


COMMON BLOCK /FCCOM/    LENGTH 000010

IFD     000000  INTEGER*2 VARIABLE
IPTYP   000002  INTEGER*2 VARIABLE
NPNTS   000004  INTEGER*2 VARIABLE
NLINES  000006  INTEGER*2 VARIABLE

COMMON BLOCK /PEXCOM/   LENGTH 000136

IC      000000  INTEGER*2 ARRAY (35)
IZPNT   000106  INTEGER*2 VARIABLE
ID      000110  INTEGER*2 ARRAY (3)
INC     000116  INTEGER*2 VARIABLE
IH      000120  INTEGER*2 VARIABLE
IXD     000122  INTEGER*2 ARRAY (3)
ISMP    000130  INTEGER*2 ARRAY (3)
$
```

```
0001          SUBROUTINE LENGTH
0002          COMMON   / FORIER / DAT(2048)
0003          COMMON   / FCCOM  / IA(17), IPLOC
0004          AVG = 0
0005          DO 5 I = 1015,1024
0006          AVG = DAT(I) + AVG     ! AVERAGE LAST 10 PNTS
0007    5     CONTINUE
0008          AVG = AVG / 10
0009          DB3 = 1.414 * AVG      ! 3 DECIBELS ABOVE AVG
0010          DO 10 I = 1,1024
0011          IF (DAT(I) .LT. DB3) IPLOC = I
0013    10    CONTINUE
0014          RETURN
0015          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

AVG     000020  REAL*4     VARIABLE
I       000024  INTEGER*2  VARIABLE
DB3     000026  REAL*4     VARIABLE

COMMON BLOCK /FORIER/   LENGTH 020000

DAT     000000  REAL*4     ARRAY (2048)

COMMON BLOCK /FCCOM/    LENGTH 000044

IA      000000  INTEGER*2  ARRAY (17)
IPLOC   000042  INTEGER*2  VARIABLE
$
```

```
0001          SUBROUTINE NNS(IREF, IC, IFLAW, NUM)
0002          DIMENSION IFLAW( 10 ), IPMTR(2, 24)
          C
          C     STORE FLAW DATA IN TABLE
          C
0003          DATA IPMTR/
             $550, 480, 636, 630, 591, 510, 497, 430, 425, 520,
             $522, 560, 441, 570, 410, 500, 544, 480, 491, 510,
             $563, 560, 531, 570, 376, 430, 530, 390,
             $312, 620, 307, 570, 397, 540, 419, 530, 388, 780,
             $390, 660, 660, 660, 571, 660, 606, 660, 616, 890/
0004          NUM = 0
0005          MIN = 1000
          C
          C     PERFORM NEAREST NEIGHBOR APPROX.
          C
0006          DO 1 I = 1,24
0007          IDIFF = IABS(IREF - IPMTR(IC, I))   !FIND AMOUNT OF ERROR
0008          IF(IDIFF .GT. MIN) GO TO 1  !MINIMIZE ERROR
0010          MIN = IDIFF
0011          I1 = I
0012        1 CONTINUE
          C
          C     STORE ALL FLAWS DISPLAYING SMALLEST ERROR
          C
0013          DO 2 I = 1,24
0014          IF(IPMTR(IC,I) .NE. IPMTR(IC,I1)) GO TO 2
0016          NUM = NUM + 1
0017          IFLAW( NUM ) = I
0018        2 CONTINUE
          C
          C     RETURN TO CALLING PROGRAM
          C
0019          RETURN
0020          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IFLAW   000020  INTEGER*2 PARAMETER ARRAY (10)
IPMTR   000024  INTEGER*2 ARRAY (2,24)
IREF    000014  INTEGER*2 PARAMETER VARIABLE
IC      000016  INTEGER*2 PARAMETER VARIABLE
NUM     000022  INTEGER*2 PARAMETER VARIABLE
MIN     000164  INTEGER*2 VARIABLE
I       000166  INTEGER*2 VARIABLE
IDIFF   000170  INTEGER*2 VARIABLE
IABS    000000  INTEGER*2 PROCEDURE
I1      000172  INTEGER*2 VARIABLE
$
```

```
                                                .TITLE  PAUSE  !WAITS FOR CHAR.
                                                .GLOBL  PAUSE
                                                .MCALL  SAVE,UNSAVE,..V2...,.REGDEF
000000                                          ..V2...
000000                                          .REGDEF
000000                                  PAUSE:  SAVE    012345
000014  005067  000000'                         CLR     ICHBUF
                                        ;
                                        ;       WAIT FOR CHARACTER
                                        ;
000020  005767  000000'                 1$:     TST     ICHBUF
000024  001775                                  BEQ     1$
000026                                          UNSAVE  543210
000042  000207                                  RTS     PC
        000000'                                 .CSECT  TEKCOM
000000  000000                          ICHBUF: .WORD   0
        000001'                                 .END
```

```
PAUSE  !WAITS FOR CHAR. RT-11 MACRO VM02-12   15-SEP-80 PAGE 1+
SYMBOL TABLE

ICHBUF  000000R     002  PAUSE   000000RG        PC     =%000007          R0     =%000000          R1     =%000001
R2     =%000002          R3     =%000003          R4     =%000004          R5     =%000005          SP     =%000006
...V2 = 000001
. ABS.  000000      000
        000044      001
TEKCOM  000002      002
ERRORS DETECTED: 0
FREE CORE: 17973. WORDS

,TT:/N:TTM=PAUSE

ERRORS DETECTED: 0
FREE CORE: 17973. WORDS

$
```

```
                                                .TITLE  PEXF
                                                .GLOBL  PEXF,$READ,CHOUT
                                                .MCALL  .CLOSE,.RCTRLO,.PRINT,.REGDEF,..V2...
                                                .MCALL  CHTXT,SUPTXT,SAVE,CHSTUP,UNSAVE
000000                                          ..V2...
000000                                          .REGDEF
000000                                  PEXF:   SAVE    012345
000014  016702  000072'                        MOV     CNT,R2
000020  005067  000002'                        CLR     DSTAT           ;INITIALIZE STATUS
000024  026767  000012' 000016'                CMP     CDB,SBLK
000032  001402                                 BEQ     S1C
000034  004767  000462                         JSR     PC,NEW1         ;GET NEW DATA FOR FIRST BLOCK IN FILE
000040  004767  000464                 S1C:    JSR     PC,NEWDAT       ;GET NEW DATA (IF NEEDED)
000044  016203  000152'                        MOV     DBUF(R2),R3     ;GET RECORD     ID
000050  005202                                 INC     R2
000052  005202                                 INC     R2
000054  020327  177777                         CMP     R3,#-1
000060  003073                                 BGT     S3C             ;GO TO PLOT MODE
000062  002434                                 BLT     EOF             ;END OF FILE MODE
000064  001400                                 BEQ     SHDR            ;SCAN HEADER MODE
                                        ;
                                        ;       TRANSFER HEADER BLOCK
000066  012703  000674'                 SHDR:   MOV     #T0,R3          ;ADDRESS OF BUFFER FOR HEADER
000072  012704  000020                         MOV     #16.,R4
000076  004767  000426                 1$:     JSR     PC,NEWDAT
000102  016233  000152'                        MOV     DBUF(R2),@(R3)+ ;MOVE A WORD
000106  005202                                 INC     R2
000110  005202                                 INC     R2
000112  005304                                 DEC     R4
000114  005704                                 TST     R4
000116  003367                                 BGT     1$
000120  012703  000011                         MOV     #9.,R3          ;# OF SPARE WORDS
000124  004767  000400                 2$:     JSR     PC,NEWDAT       ;GET MORE DATA IF NECESSARY
000130  005202                                 INC     R2      ;INCREMENT WORD COUNTER
000132  005202                                 INC     R2
000134  077305                                 SOB     R3,2$           ;DO FOR ALL SPARE WORDS
000136  012767  177777  000002'                MOV     #-1,DSTAT               ;SET SCAN HEADER FLAG
000144  016767  000116' 000014'                MOV     INCR,SINCR
000152  000403                                 BR      OUT
000154  012767  177776  000002'         EOF:   MOV     #-2,DSTAT       ;FLAG THE END OF FILE
000162  005000                         OUT:    CLR     R0
000164  005001                                 CLR     R1
000166  010201                                 MOV     R2,R1
000170  071027  000002                         DIV     #2.,R0
000174  005701                                 TST     R1
000176  001401                                 BEQ     2$
000200  005302                                 DEC     R2
000202  010267  000072'                 2$:     MOV     R2,CNT
000206  012704  003777                         MOV     #2047.,R4       ;CLEAR IBUF TO END OF BUFFER
000212  166704  000020'                        SUB     STOPX,R4
000216  012705  000040'                        MOV     #IBUF,R5
000222  066705  000020'                        ADD     STOPX,R5
000226  105025                         3$:     CLRB    (R5)+
000230  077402                                 SOB     R4,3$
000232                                         UNSAVE  543210
000246  000207                                 RTS     PC              ;RETURN TO CALLING PROGRAM
000250  012701  000006                 S3C:    MOV     #6,R1           ;SET UP TO MOVE 6 VALUES
```

```
                                                MOV     #THERE,R0
                                2$:             JSR     PC,NEWDAT       ;GET NEW DATA IF NEEDED
                                                MOV     DBUF(R2),@(R0)+
                                                INC     R2
                                                INC     R2
                                                SOB     R1,2$           ;GO AGAIN
                                                MOV     #IBUF,R4
                                                MOV     STARTX,R0               ;CLEAR BUFFER UP TO STARTX
                                                BEQ     3$
                                                CLC
                                                ROR     R0
                                                BCC     1$
                                                CLRB    IBUF-1(R0)
                                1$:             CLR     (R4)+
                                                SOB     R0,1$
                                3$:             MOV     SVAL,R3         ;GET START VALUE
                                                MOVB    R3,(R4)+        ;INSERT START VALUE
                                SC:             JSR     PC,NEWDAT               ;GET NEW DATA IF NEEDED
                                                MOVB    DBUF(R2),R1             ;GET COMMAND
                                                INC     R2
                                                ASHC    #8.,R0
                                                CLR     R0
                                                ASHC    #3.,R0                  ;SEPERATE COMMAND & COUNT
                                                ASL     R0
                                                SWAB    R1
                                                ASH     #-3.,R1
                                                BNE     1$
                                                JSR     PC,NEWDAT
                                                MOVB    DBUF(R2),R1             ;GET COUNT
                                                INC     R2
                                                BIC     #177400,R1              ;REMOVE SIGN EXTENSION
                                1$:             JMP     @SCJ(R0)                ;JUMP TO PROPER ROUTINE
000414' 000470' 000436' 000434' SCJ:            .WORD   SC0,SC1,SC2,SC3,SC4,SC5,SC6,SC7
000422' 000444' 000454' 000462'
000430' 000470' 000470'
                                SC2:            INC     R3                      ;UP ONE
                                SC1:            MOVB    R3,(R4)+                ;INSERT VALUE
                                                SOB     R1,SC1                  ;GO AGAIN
                                                BR      SC
                                SC3:            DEC     R3              ;DOWN ONE
                                1$:             MOVB    R3,(R4)+        ;INSERT VALUE
                                                SOB     R1,1$           ;GO AGAIN
                                                BR      SC
                                SC4:            ADD     R1,R3           ;UP COUNT
                                                MOVB    R3,(R4)+        ;INSERT VALUE
                                                BR      SC
                                SC5:            SUB     R1,R3           ;DOWN COUNT
                                                MOVB    R3,(R4)+        ;INSERT VALUE
                                                BR      SC
                                SC7:
                                SC6:
004040' 000470' 012700          SC0:            MOV     #IBUF+2048.,R0  ;GET END OF BUFFER
                                                SUB     R4,R0           ;COMPUTE NUMBER OF VALUES
                                                BLE     3$
                                                CLC
                                                ROR     R0
                                                BLE     3$
```

```
113 000506  103001                          BCC     1$
114 000510  105024                          CLRB    (R4)+
115 000512  005024                  1$:     CLR     (R4)+           ;CLEAR WORD
116 000514  077002                          SOB     R0,1$           ;GO AGAIN
117 000516  000167  177440          3$:     JMP      OUT
118 000522                          NEW1:   SAVE    05
119 000526  000405                          BR      ND1
120 000530  020227  004000          NEWDAT: CMP     R2,#2048.       ;CHECK FOR END OF BUFFER
121 000534  002450                          BLT     ND5             ;IF NOT , RETURN
122 000536                                  SAVE    05
123 000542  016767  000016' 000012' ND1:    MOV     SBLK,CDB
124 000550  010067  000074'                 MOV     R0,Z0
125 000554  010167  000076'                 MOV     R1,Z1
126 000560  010267  000100'                 MOV     R2,Z2
127 000564  010367  000102'                 MOV     R3,Z3
128 000570  010467  000104'                 MOV     R4,Z4
129 000574  010567  000106'                 MOV     R5,Z5
130 000600  004767  0000000                 JSR     PC,SREAD        ;READ DATA FILE
131 000604  016700  000074'                 MOV     Z0,R0
132 000610  016701  000076'                 MOV     Z1,R1
133 000614  016702  000100'                 MOV     Z2,R2
134 000620  016703  000102'                 MOV     Z3,R3
135 000624  016704  000104'                 MOV     Z4,R4
136 000630  016705  000106'                 MOV     Z5,R5
137 000634  005002                  ND4:    CLR     R2
138 000636  062767  000004  000016'        ADD     #4,SBLK         ;INCREMENT BLOCK POINTER
139 000644  016767  000016' 000012'        MOV     SBLK,CDB        ;UPDATE CURRENT DATA BLOCK
140 000652                                  UNSAVE  50
141 000656  000207                  ND5:    RTS     PC              ;RETURN TO SCAND
142 000660  000004' 000006' 000016' THERE:  .WORD   XPNT,YPNT,STARTX,STOPX,TWOFFS,SVAL
    000666  000020' 000030' 000734'
143 000674  000076' 000100' 000102' TO:     .WORD   XLIM1,YLIM1,XLIM2,YLIM2
    000702  000104'
144 000704  000106' 000110' 000112'        .WORD   ZPNT,TDLY,IUTS,IRNG,INCR
    000712  000114' 000116'
145 000716  000122' 000124' 000126'        .WORD   XDCRID,XDCRID+2,XDCRID+4,SMPLID
    000724  000130'
146 000726  000132' 000134' 000120'        .WORD   SMPLID+2,SMPLID+4,OF
147 000734  000000                  SVAL:   .WORD   0
148 000736  000000                  STAX:   .WORD   0
149         000000'                         .CSECT  FCCOM
150                                         .BLKW   28.
151 000070  000000                  IFTT:   .WORD   0
152 000072  000000                  CNT:    .WORD   0
153 000074  000000                  Z0:     .WORD   0
154 000076  000000                  Z1:     .WORD   0
155 000100  000000                  Z2:     .WORD   0
156 000102  000000                  Z3:     .WORD   0
157 000104  000000                  Z4:     .WORD   0
158 000106  000000                  Z5:     .WORD   0
159         000000'                         .CSECT  PEXCOM
160 000000  000000                  XSC:    .WORD   0       ;X SCAN CONVERSION
161 000002  000000                  YSC:    .WORD   0       ;Y SCAN CONVERSION
162 000004  000000                  XPNT:   .WORD   0       ;X COORDINATE RELATIVE TO SCANNER ORIGIN
163 000006  000000                  YPNT:   .WORD   0       ;Y COORDINATE RELATIVE TO SCANN ORIGIN
164 000010  000000                  DFBC:   .WORD   0       ;DATA FILE BLOCK COUNT
165 000012  000000                  CDB:    .WORD   0       ;CURRENT DATA BLOCK #
```

```
                                DBPNTR: .WORD   0       ;DATA BLOCK POINTER
                                SBLK:   .WORD   0       ;REQUESTED DATA BLOCK
                                SOFFS:  .WORD   0       ;OFFSET OF START OF REQUESTED POINT
                                DCHNUM: .WORD   0       ;DATA FILE CHANNEL #
                                CIB:    .WORD   0       ;CURRENT INDEX BLOCK ( IN IRS )
                                INDCH:  .WORD   0       ;INDEX FILE CHANNEL #
                                TWOFFS: .WORD   0       ;TIME WINDOW OFFSET
                                SPARES: .BLKW   9.
                                IH:                     ;INDEX HEADER
                                FB:     .BLKW   4       ;FILEBLOCK
                                NORP:   .WORD   0       ;# OF RECORDED POINTS
                                NIB:    .WORD   0       ;# OF INDEX BLOCKS
                                        .BLKW   3
                                XLIM1:  .WORD   0       ;LOW X LIMIT OF SCAN
                                YLIM1:  .WORD   0       ;LOW Y LIMIT OF SCAN
                                XLIM2:  .WORD   0       ;HIGH X LIMIT OF SCAN
                                YLIM2:  .WORD   0       ;HIGH Y LIMIT OF SCAN
                                ZPNT:   .WORD   0       ;Z VALUE OF SCAN
                                TDLY:   .WORD   0       ;TOTAL TRIGGER DELAY
                                IUTS:   .WORD   0       ;SAMPLE RATE UNITS
                                IRNG:   .WORD   0       ;SAMPLE RATE RANGE
                                INCR:   .WORD   0       ;SCAN INCREMENT
                                OF:     .WORD   0       ;OFFSET FLAG
                                XDCRID: .BLKB   6
                                SMPLID: .BLKB   18.
                                DBUF:   .BLKB   2048.           ;COMPRESSED DATA BUFFER
                        000000' .CSECT  EXDCOM
000000  000000                 STYP:   .WORD   0       ;SCAN TYPE
000002  000000                 DSTAT:  .WORD   0       ;DATA STATUS
000004  000000                 XS:     .WORD   0       ;REQUESTED X COORDINATE RELATIVE TO SCAN ORIGIN
000006  000000                 YS:     .WORD   0       ;REQUESTED Y COORDINATE RELATIVE TO SCAN ORIGIN
000010  000000                 TWDLY:  .WORD   0       ;TOTAL WINDOWED DELAY
000012  000000                 FUPS:   .WORD   0       ;FUNDAMENTAL UNITS (OF TIME) PER SAMPLE
000014  000000                 SINCR:  .WORD   0       ;SCAN INCREMENT
000016  000000                 STARTX: .WORD   0       ;START OF DATA TO BE PROCESSED
000020  000000                 STOPX:  .WORD   0       ;END OF DATA TO BE PROCESSED (RELATIVE TO IBUF)
000022  000000                 HXS:    .WORD   0       ;HIGH X OF SCAN
000024  000000                 HYS:    .WORD   0       ;HIGH Y OF SCAN
000026                          SPARE:  .BLKW   5       ;SPARE COMMON LOCATIONS
000040                          IBUF:   .BLKB   2048.   ;DATA BUFFER
                        000001' .END
```

```
PEXF    RT-11 MACRO VM02-12   15-SEP-80 PAGE 14
SYMBOL TABLE

CDB      000012R     003  CHOUT = ****** G        CIB      000024R     003  CNT      000072R     002  DBPNTR   000014R     003
DBUF     000152R     003  DCHNUM   000022R     003  DFBC     000010R     003  DSTAT    000002R     004  EOF      000154R
FB       000054R     003  FUPS     000012R     004  HXS      000022R     004  HYS      000024R     004  IBUF     000040R     004
IFTT     000070R     002  IH       000054R     003  INCR     000116R     003  INDCH    000026R     003  IRNG     000114R     003
IUTS     000112R     003  ND1      000542R          ND4      000634R          ND5      000656R          NEWDAT   000530R
NEW1     000522R          NIB      000066R     003  NORP     000064R     003  OF       000120R     003  OUT      000162R
PC     =%000007           PEXF     000000RG         R0     =%000000           R1     =%000001           R2     =%000002
R3     =%000003           R4     =%000004           R5     =%000005           SBLK     000016R     003  SC       000334R
SCJ      000414R          SC0      000470R          SC1      000436R          SC2      000434R          SC3      000444R
SC4      000454R          SC5      000462R          SC6      000470R          SC7      000470R          SHDR     000066R
SINCR    000014R     004  SMPLID   000130R     003  SOFFS    000020R     003  SP     =%000006           SPARE    000026R     004
SPARES   000032R     003  SREAD = ****** G         STARTX   000016R     004  STAX     000736R          STOPX    000020R     004
STYP     000000R     004  SVAL     000734R          SIC      000040R          S3C      000250R          TDLY     000110R     003
THERE    000660R          TO       000674R          TWDLY    000010R     004  TWOFFS   000030R     003  XDCRID   000122R     003
XLIM1    000076R     003  XLIM2    000102R     003  XPNT     000004R     003  XS       000004R     004  XSC      000000R     003
YLIM1    000100R     003  YLIM2    000104R     003  YPNT     000006R     003  YS       000006R     004  YSC      000002R     003
ZPNT     000106R     003  Z0       000074R     002  Z1       000076R     002  Z2       000100R     002  Z3       000102R     002
Z4       000104R     002  Z5       000106R     002  ...V2 = 000001
. ABS.   000000      000
         000740      001
FCCOM    000110      002
PEXCOM   004152      003
EXDCOM   004040      004
ERRORS DETECTED: 0
FREE CORE: 17351. WORDS

,TT:/N:TTM=PEXF

ERRORS DETECTED: 0
FREE CORE: 17351. WORDS

*
```

```
0001          SUBROUTINE PGEN
0002          COMMON /FCCOM/IA(2),NPNTS,NLINES,VAL,IB(10),IST,IC(9),MAX
0003          COMMON /EXCOM/ID(2),IXS,IYS
0004          COMMON /PEXCOM/IE(31),IXL1,IYL1,IG(6),INCR
0005          IST = 0
0006          DO 1 IY = 1,(NLINES + 1)
0007          DO 2 IX = 1,(NPNTS + 1)
0008          IF(IST .EQ  1) GO TO 3
0010          IXS = (IX - 1) * INCR + IXL1
0011          IYS = (IY - 1) * INCR + IYL1
0012          GO TO 4
0013    3     IXS = (((NPNTS + 1) - IX) * INCR) - (INCR/2) + IXL1
0014          IYS = (IY - 1) * INCR + IYL1
0015    4     CALL CALC
0016          IF(ABS(VAL) .GT. FLOAT(MAX)) MAX = IFIX(VAL)
0018    2     CONTINUE
0019          IF(IST .EQ. 0) GO TO 5
0021          IST = 0
0022          GO TO 1
0023    5     IST = 1
0024    1     CONTINUE
0025          RETURN
0026          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IY      000014  INTEGER*2 VARIABLE
IX      000016  INTEGER*2 VARIABLE
CALC    000000  REAL*4    PROCEDURE
ABS     000000  REAL*4    PROCEDURE
FLOAT   000000  REAL*4    PROCEDURE
IFIX    000000  INTEGER*2 PROCEDURE


COMMON BLOCK /FCCOM/    LENGTH 000066

IA      000000  INTEGER*2 ARRAY (2)
NPNTS   000004  INTEGER*2 VARIABLE
NLINES  000006  INTEGER*2 VARIABLE
VAL     000010  REAL*4    VARIABLE
IB      000014  INTEGER*2 ARRAY (10)
IST     000040  INTEGER*2 VARIABLE
IC      000042  INTEGER*2 ARRAY (9)
MAX     000064  INTEGER*2 VARIABLE

COMMON BLOCK /EXDCOM/   LENGTH 000010

ID      000000  INTEGER*2 ARRAY (2)
IXS     000004  INTEGER*2 VARIABLE
IYS     000006  INTEGER*2 VARIABLE

COMMON BLOCK /PEXCOM/   LENGTH 000120

IE      000000  INTEGER*2 ARRAY (31)
IXL1    000076  INTEGER*2 VARIABLE
IYL1    000100  INTEGER*2 VARIABLE
IG      000102  INTEGER*2 ARRAY (6)
INCR    000116  INTEGER*2 VARIABLE
*
```

```
0001          SUBROUTINE PLOT3D(IVXYZ, XDATA,  ZDATA, XSCALE,
             .YSCALE,ZSCALE, NLINE, NPNTS, PHI, THETA, XREF,
             .YREF, XLENTH)
0002          COMMON /FCCOM/ IZED(36), YDATA(3072)
0003          COMMON / FORIER / MASK(2017)
0004          INTEGER     HIGH,OLDHI, OLDLOW
0005          DATA INIT, JVXYZ, SPHI, STHETA/-1, -1, -1.0E38,
             .-1.0E38/
0006          IF( NLINE .EQ. 0 ) GO TO 550
0008          IF( NLINE .NE. 1 ) GO TO 70
0010          PIPI = 63.0
0011          MYPI = 1090
0012          CALL IPLOT( 0, 0, 0 )
0013          I = IFIX(2. * (XLENTH * PIPI) + 1.)
0014          DO 10 K = 1,I
0015          MASK( K ) = INIT
0016    10    CONTINUE
0017          INIT = -1
0018          INCI = -1
0019          I = 0
0020    70    IF ( PHI .EQ. SPHI .AND. THETA .EQ. STHETA) GO TO 80
0022          SPHI = SIN( .0174533 * PHI )
0023          CPHI = COS( .0174533 * PHI )
0024          STHETA = SIN( .0174533 * THETA )
0025          CTHETA = COS( .0174533 * THETA )
0026          A11 = CPHI
0027          A13 = -SPHI
0028          A21 = STHETA * SPHI
0029          A22 = CTHETA
0030          A23 = STHETA * CPHI
0031          SPHI = PHI
0032          STHETA = THETA
0033    80    INCI = -INCI
0034          IF ( I .NE. 0 ) I = NPNTS + 1
0036          WRITE(7,8000) (YDATA(M),M=1,8)
0037     8000 FORMAT(8(1X,F10.5))
0038          DO 530 K = 1,NPNTS
0039          I = I + INCI
0040          X = (I-1) * XSCALE
0041    130   Y = YDATA(I) * YSCALE
0042    150   Z = (NLINE-1) * ZSCALE
0043    170   XXX = A11 * X + A13 * Z + XREF
0044          IF( XXX) 171, 172, 172
0045    171   IX = IFIX( XXX * PIPI - .5 )
0046          GO TO 173
0047    172   IX = IFIX( XXX * PIPI + .5 )
0048    173   CONTINUE
0049          YYY = A21 * X + A23 *Z + YREF
0050          YY = YYY + A22 * Y
0051          IF( YY ) 174, 175, 175
0052    174   IY = IFIX( YY * PIPI - .5 )
0053          GO TO 176
0054    175   IY = IFIX(YY * PIPI + .5 )
0055    176   CONTINUE
```

```
0056          IF( K .NE. 1 ) GO TO 250
0058          LOW = IX + IX
0059          HIGH = LOW - 1
0060          MLOW = MASK( LOW )
0061          MHIGH = MASK( HIGH)
0062          IF ( MHIGH - IY ) 200, 210, 180
0063   180    IF( MLOW - IY ) 190, 230, 220
0064   190    LOCOLD = 0
0065          GO TO 240
0066   200    MASK( HIGH ) = IY
0067          IF( MLOW .EQ. -1 ) MASK( LOW ) = IY
0069   210    LOCOLD = +1
0070          GO TO 240
0071   220    MASK( LOW ) = IY
0072   230    LOCOLD = -1
0073   240    IF(IY .LT. 0) GO TO 720
0075          IF(IY .GT. 724)GO TO 720
0077          CALL IPLOT( IX, IY, 0 )
0078   720    JX = IX
0079          JY = IY
0080          IYREF = IY
0081          GO TO 530
0082   250    IF ( IX .NE. JX ) GO TO 260
0084          JY = IY
0085          GO TO 280
0086   260    YINC = FLOAT(IY-JY)/ABS(FLOAT(IX - JX ))
0087          INCX = (IX - JX) / IABS( IX - JX )
0088          YJ = JY
0089   270    JX = JX + INCX
0090          YJ = YJ + YINC
0091   272    JY = IFIX(YJ + .5)
0092   273    CONTINUE
0093          LOW = JX + JX
0094          HIGH = LOW - 1
0095          MLOW = MASK(LOW)
0096          MHIGH = MASK(HIGH)
0097   280    IF( MHIGH - JY ) 300, 300, 290
0098   290    IF ( MLOW - JY ) 310, 320, 320
0099   300    LOC = + 1
0100          IF( LOCOLD ) 360, 370, 430
0101   310    LOC = 0
0102          IF( LOCOLD) 340, 350, 330
0103   320    LOC = -1
0104          IF( LOCOLD ) 510, 450, 440
0105   330    IF ( MHIGH .LE. IYREF ) GO TO 700
0107          GO TO 350
0108   700    IF(MHIGH .LT. 0) GOT O 530
0110          IF(MHIGH .GT. 724) GO TO 530
0112          CALL IPLOT(JX,MHIGH,1)
0113          GO TO 350
0114   340    IF( MLOW .GE. IYREF) GO TO 710
0116          GO TO 350
0117   710    IF(MLOW .LT. 0)GO TO 530
0119          IF(MLOW .GT. 724)GO TO 530
```

```
0121          CALL IPLOT(JX, MLOW, 1)
0122    350   IF(JY .LT. 0) GO TO 530
0124          IF(JY .GT. 724) GO TO 530
0126          CALL IPLOT( JX, JY, 0)
0127          GO TO 520
0128    360   IF( MLOW - IYREF ) 370, 380,380
0129    370   IF( MHIGH - IYREF) 400, 390, 390
0130    380   IF(MLOW .LT. 1) GO TO 530
0132          IF(MLOW .GT. 724) GO TO 530
0134          CALL IPLOT( JX, MLOW, 1 )
0135    390   IF(MHIGH .LT. 0) GO TO 530
0137          IF(MHIGH .GT. 724) GO TO 530
0139          CALL IPLOT( JX, MHIGH, 0 )
0140          GO TO 430
0141    400   IF ( MHIGH .EQ. -1 ) GO TO 430
0143          OLDHI = HIGH - 2 * INCX
0144          IF( MASK(OLDHI ) - JY ) 420, 420, 410
0145    410   IF(JY .LT. 0) GOT O 530
0147          IF(JY .GT. 724) GO TO 530
0149          CALL IPLOT( JX, JY, 0 )
0150          GO TO 430
0151    420   IF(MASK(OLDHI) .LT. 0)GO TO 530
0153          IF(MASK(OLDHI).GT.724) GO TO 530
0155          CALL IPLOT((JX -INCX ), MASK(OLDHI), 0 )
0156    430   IF(JY .LT. 0) GO TO 530
0158          IF(JY .GT. 724) GO TO 530
0160          IF((JX .LT. 0) .OR. (JX .GT. 1024)) GO TO 520
0162          MASK( HIGH ) = JY
0163          IF( MLOW .EQ. -1 ) MASK( LOW ) = JY
0165          CALL IPLOT( JX, JY, 1 )
0166          GO TO 520
0167    440   IF( MHIGH - IYREF ) 460, 460, 450
0168    450   IF( MLOW - IYREF ) 470, 470, 480
0169    460   IF(MHIGH .LT. 0) GO TO 530
0171          IF(MHIGH .GT. 724) GO TO 530
0173          CALL IPLOT( JX, MHIGH, 1 )
0174    470   IF(MLOW .LT. 0) GO TO 530
0176          IF(MLOW .GT. 724) GO TO 530
0178          CALL IPLOT( JX, MLOW, 0 )
0179          GO TO 510
0180    480   OLDLOW = LOW - 2 * INCX
0181          IF( MASK( OLDLOW) - JY ) 490, 500, 500
0182    490   IF(JY .LT. 0) GO TO 530
0184          IF(JY. GT. 724) GO TO 530
0186          CALL IPLOT( JX, JY, 0 )
0187          GO TO 510
0188    500   IF(MASK(OLDLOW) .LT. 0)GO TO 530
0190          IF(MASK(OLDLOW) .GT. 724) GO TO 530
0192          CALL IPLOT((JX - INCX ),MASK(OLDLOW), 0 )
0193    510   IF(JY .LT. 0)GO TO 530
0195          IF(JY .GT. 724) GO TO 530
0197          IF((JX .LT. 0) .OR. (JX .GT. 1024)) GO TO 520
0199          MASK( LOW ) = JY
0200          CALL IPLOT( JX, JY, 1 )
```

```
0201   520    IYREF = JY
0202          LOCOLD = LOC
0203          IF( JX .NE. IX ) GO TO 270
0205   530    CONTINUE
0206          IF(JY .LT. 0)GO TO 730
0208          IF(JY .GT. 724)GO TO 730
0210          CALL IPLOT(JX, JY, 0 )
0211   730    IF(INDV .EQ. 1 ) GO TO 540
0213          INDEX = -INCI + 6
0214   540    I = I - 1
0215          RETURN
0216   550    INIT = 0
0217          RETURN
0218          END
```

FORTRAN IV STORAGE MAP

| NAME | OFFSET | ATTRIBUTES |
|---|---|---|
| IVXYZ | 000014 | INTEGER*2 PARAMETER VARIABLE |
| XDATA | 000016 | REAL*4 PARAMETER VARIABLE |
| ZDATA | 000020 | REAL*4 PARAMETER VARIABLE |
| XSCALE | 000022 | REAL*4 PARAMETER VARIABLE |
| YSCALE | 000024 | REAL*4 PARAMETER VARIABLE |
| ZSCALE | 000026 | REAL*4 PARAMETER VARIABLE |
| NLINE | 000030 | INTEGER*2 PARAMETER VARIABLE |
| NPNTS | 000032 | INTEGER*2 PARAMETER VARIABLE |
| PHI | 000034 | REAL*4 PARAMETER VARIABLE |
| THETA | 000036 | REAL*4 PARAMETER VARIABLE |
| XREF | 000040 | REAL*4 PARAMETER VARIABLE |
| YREF | 000042 | REAL*4 PARAMETER VARIABLE |
| XLENTH | 000044 | REAL*4 PARAMETER VARIABLE |
| HIGH | 000106 | INTEGER*2 VARIABLE |
| OLDHI | 000110 | INTEGER*2 VARIABLE |
| OLDLOW | 000112 | INTEGER*2 VARIABLE |
| INIT | 000046 | INTEGER*2 VARIABLE |
| JVXYZ | 000050 | INTEGER*2 VARIABLE |
| SPHI | 000052 | REAL*4 VARIABLE |
| STHETA | 000056 | REAL*4 VARIABLE |
| PIPI | 000114 | REAL*4 VARIABLE |
| NYPI | 000120 | INTEGER*2 VARIABLE |
| IPLOT | 000000 | INTEGER*2 PROCEDURE |
| I | 000122 | INTEGER*2 VARIABLE |
| IFIX | 000000 | INTEGER*2 PROCEDURE |
| K | 000124 | INTEGER*2 VARIABLE |
| INCI | 000126 | INTEGER*2 VARIABLE |
| SIN | 000000 | REAL*4 PROCEDURE |
| CPHI | 000130 | REAL*4 VARIABLE |
| COS | 000000 | REAL*4 PROCEDURE |
| CTHETA | 000134 | REAL*4 VARIABLE |
| A11 | 000140 | REAL*4 VARIABLE |
| A13 | 000144 | REAL*4 VARIABLE |
| A21 | 000150 | REAL*4 VARIABLE |
| A22 | 000154 | REAL*4 VARIABLE |
| A23 | 000160 | REAL*4 VARIABLE |
| M | 000164 | INTEGER*2 VARIABLE |
| X | 000166 | REAL*4 VARIABLE |
| Y | 000172 | REAL*4 VARIABLE |
| Z | 000176 | REAL*4 VARIABLE |
| XXX | 000202 | REAL*4 VARIABLE |
| IX | 000206 | INTEGER*2 VARIABLE |
| YYY | 000210 | REAL*4 VARIABLE |
| YY | 000214 | REAL*4 VARIABLE |
| IY | 000220 | INTEGER*2 VARIABLE |
| LOW | 000222 | INTEGER*2 VARIABLE |
| MLOW | 000224 | INTEGER*2 VARIABLE |
| MHIGH | 000226 | INTEGER*2 VARIABLE |
| LOCOLD | 000230 | INTEGER*2 VARIABLE |
| JX | 000232 | INTEGER*2 VARIABLE |
| JY | 000234 | INTEGER*2 VARIABLE |
| IYREF | 000236 | INTEGER*2 VARIABLE |

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

YINC     000240   REAL*4     VARIABLE
FLOAT    000000   REAL*4     PROCEDURE
ABS      000000   REAL*4     PROCEDURE
INCX     000244   INTEGER*2  VARIABLE
IABS     000000   INTEGER*2  PROCEDURE
YJ       000246   REAL*4     VARIABLE
LOC      000252   INTEGER*2  VARIABLE
INDV     000254   INTEGER*2  VARIABLE
INDEX    000256   INTEGER*2  VARIABLE


COMMON BLOCK /FCCOM/    LENGTH 030110

IZED     000000   INTEGER*2  ARRAY (36)
YDATA    000110   REAL*4     ARRAY (3072)

COMMON BLOCK /FURIER/   LENGTH 007702

MASK     000000   INTEGER*2  ARRAY (2017)
*
```

```
0001          SUBROUTINE PLTBUF(IZE)
0002          COMMON /FORIER/ DAT(2050)
0003          MAX = 0
0004          NUM = 2048
0005          IF(IZE .EQ. 0)NUM = 1024
0007          CALL IPLOT(5,0,0)
0008          CALL IPLOT(5,362,0)
0009          CALL IPLOT(1023,362,1)
0010          DO 1 I = 1,10
0011          INC = 72 * I
0012          CALL IPLOT(0,INC,0)
0013          CALL IPLOT(10,INC,1)
0014    1     CONTINUE
0015          DO 2 I = 1,10
0016          INC = 102 * I
0017          CALL IPLOT(INC,360,0)
0018          CALL IPLOT(INC,364,1)
0019    2     CONTINUE
0020          AMAX = 0.
0021          DO 3 I = 1,NUM
0022          B = DAT(I)
0023          IF(B .LT. 0.) B = -B
0025          IF(B .GT. AMAX) AMAX = B
0027    3     CONTINUE
0028          IF(AMAX .EQ. 0.) GOT O 5
0030          B = DAT(1)
0031          P = B / AMAX
0032          IX = 0
0033          IY = IFIX(362. * P) + 362
0034          CALL IPLOT(IX,IY,0)
0035          F = 1024./ FLOAT(NUM)
0036          I = 2
0037          DO 4 I1 = 1,(NUM-1)
0038          B = DAT(I1)
0039          P = B / AMAX
0040          IX = IFIX(F * FLOAT(I1))
0041          IY = IFIX(362.* P) +362
0042          CALL IPLOT(IX,IY,1)
0043    4     CONTINUE
0044    5     RETURN
0045          END
```

```
FORTRAN IV      STORAGE MAP

NAME     OFFSET   ATTRIBUTES

IZE      000014   INTEGER*2 PARAMETER VARIABLE
MAX      000036   INTEGER*2 VARIABLE
NUM      000040   INTEGER*2 VARIABLE
IPLOT    000000   INTEGER*2 PROCEDURE
I        000042   INTEGER*2 VARIABLE
INC      000044   INTEGER*2 VARIABLE
AMAX     000046   REAL*4     VARIABLE
B        000052   REAL*4     VARIABLE
P        000056   REAL*4     VARIABLE
IX       000062   INTEGER*2 VARIABLE
IY       000064   INTEGER*2 VARIABLE
IFIX     000000   INTEGER*2 PROCEDURE
F        000066   REAL*4     VARIABLE
FLOAT    000000   REAL*4     PROCEDURE
I1       000072   INTEGER*2 VARIABLE

COMMON BLOCK /FURIER/   LENGTH 020010

DAT      000000   REAL*4     ARRAY (2050)
*
```

```
0001          SUBROUTINE SEARCH  !FIND ENV. PEAK
0002          COMMON /EXDCOM/ IB(7), ISTRT, ISTP, IA(7), IBUF(2048)
0003          COMMON /FCCOM/ IC(4), VAL
0004          LOGICAL * 1 IBUF
0005          VAL = 0.              !SET INITIAL MAXIMUM LOW
        C
        C     FIND MAX. ENVELOPE PEAK
        C
0006          DO 1 IVAR = ISTRT,ISTP      !DO FOR ALL VALID DATA
0007          A = IBUF(IVAR)
0008          IF(ABS(A) .GT. VAL) VAL = ABS( A )
0010    1     CONTINUE
0011          RETURN
0012          END
```

```
FORTRAN IV      STORAGE MAP

NAME    OFFSET  ATTRIBUTES

IVAR    000014  INTEGER*2 VARIABLE
A       000016  REAL*4    VARIABLE
ABS     000000  REAL*4    PROCEDURE


COMMON BLOCK /EXDCOM/   LENGTH 004040

IB      000000  INTEGER*2 ARRAY (7)
ISTRT   000016  INTEGER*2 VARIABLE
ISTP    000020  INTEGER*2 VARIABLE
IA      000022  INTEGER*2 ARRAY (7)
IBUF    000040  LOGICAL*1 ARRAY (2048)

COMMON BLOCK /FCCOM/    LENGTH 000014

IC      000000  INTEGER*2 ARRAY (4)
VAL     000010  REAL*4    VARIABLE
*
```

```
0001          SUBROUTINE SIM
0002          COMMON /EXDCOM/IA(16), IBUF(2048)
0003          LOGICAL*1 IBUF
0004          WRITE(7,100)
0005    100   FORMAT(3X,'ENTER LAMBDA, (10 TO 512)')
0006          READ(5,101)LAMBDA
0007    101   FORMAT(I3)
0008          WRITE(7,102)
0009    102   FORMAT(3X,'ENTER MAGNITUDE, (I1)')
0010          READ(5,103)MAG
0011    103   FORMAT(I2)
0012          I3 = 0
0013          I1 = 1
0014          DO 1 I = 1,2048
0015          IF(I3 .EQ. 0) GOTO 2
0017          IBUF(I) = MAG
0018          IF(I1 .LT. (LAMBDA/2)) GO TO 3
0020          I3 = 0
0021          I1 = 0
0022    3     I1 = I1 + 1
0023          GO TO 1
0024    2     IBUF(I) = 0
0025          IF(I1 .LT. (LAMBDA/2)) GO TO 3
0027          I3 = 1
0028          I1 = 0
0029    1     CONTINUE
0030          RETURN
0031          END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET   ATTRIBUTES

LAMBDA   000116   INTEGER*2 VARIABLE
MAG      000120   INTEGER*2 VARIABLE
I3       000122   INTEGER*2 VARIABLE
I1       000124   INTEGER*2 VARIABLE
I        000126   INTEGER*2 VARIABLE

COMMON BLOCK /EXDCOM/   LENGTH 004040

IA       000000   INTEGER*2 ARRAY (16)
IBUF     000040   LOGICAL*1 ARRAY (2048)
*
```

```
0001          SUBROUTINE SREAD  !READS DATA FILE
0002          COMMON /PEXCOM/ IA(3),IYPNT,IC, ICDB, IB(47), IDBUF(2048)
          C
          C   READ 4 FULL BLOCKS, SEQUENTIALLY ORDER DATA
          C
0003          IM = 1                !INITIALIZE INDEX VAR.
0004          DO 1 IL = ICDB,(ICDB + 4)    !FOUR FULL BLOCKS
0005          READ(2'IL,END=3) (IDBUF(I), I = IM,(IM + 255))       !READ A BLOCK
0006          IM = I
0007    1     CONTINUE
          C
          C   RETURN TO CALLING PROGRAM
          C
0008    2     RETURN
          C
          C   FLAG THE END OF THE FILE
          C
0009    3     IYPNT = 30000
0010          GO TO 2
0011          END
```

```
FORTRAN IV       STORAGE MAP

NAME     OFFSET  ATTRIBUTES

IM       000016  INTEGER*2 VARIABLE
IL       000020  INTEGER*2 VARIABLE
I        000022  INTEGER*2 VARIABLE


COMMON BLOCK /FEXCOM/   LENGTH 010152

IA       000000  INTEGER*2 ARRAY (3)
IYPNT    000006  INTEGER*2 VARIABLE
IC       000010  INTEGER*2 VARIABLE
ICDB     000012  INTEGER*2 VARIABLE
IB       000014  INTEGER*2 ARRAY (47)
IDBUF    000152  INTEGER*2 ARRAY (2048)
*C^C

.
```

```
                                                        .TITLE TEKMOD
                                                        .MCALL .REGDEF,..V2...
                001000                          STBASE=1000
                                                        .GLOBL ALPHA,CRLF,LFCR,CR,LF,HOME,BELL,IPLOT
                                                        .GLOBL CHOUT,PRTV,PRTH,GRIN,ERASE,TEKIN,ALPHA2
                                                        .GLOBL  WRTLSI,CNTCAL,CHOUTS
                                                        ..V2...
                                                        .REGDEF
 000000                                         BUF:    .BYTE   14,0,0,0,0,0,0,0,0,0,0,0,0,0,37,0,31,0
 000000     014     000     000
 000003     000     000     000
 000006     000     000     000
 000011     000     000     000
 000014     000     000     037
 000017     000     031     000
                175610                          WCSR=   175610
                175612                          DATI=   175612
                175614                          CSR=    175614
                175616                          DATO=   175616
                000370                          TTY=    370
                                                ;  CLEAR SCREEN
                                                ;   CALL ERASE
 000022                                         ERASE:
 000022  010046                                        MOV   R0,-(SP)
 000024  010146                                        MOV     R1,-(SP)
 000026  005067  000002'                               CLR     LBLUP
 000032  012700  000000'                               MOV     #BUF,R0
 000036  012701  000020                                MOV     #16.,R1
 000042  000167  000214                                JMP     DOIO
                                                ;  PLACE TERMINAL IN CHARACTER MODE
                                                ;   CALL ALPHA
 000046                                         ALPHA:
 000046  010046                                        MOV     R0,-(SP)
 000050  010146                                        MOV     R1,-(SP)
 000052  012700  000015'                               MOV     #BUF+13.,R0
 000056  012701  000002                                MOV     #2,R1
 000062  000167  000174                                JMP     DOIO
                                                ;SELECT DOUBLE SIZE VECTORS
                                                ;CALL ALPHA2
 000066  010046                                 ALPHA2: MOV     R0,-(SP)
 000070  010146                                        MOV     R1,-(SP)
 000072  012700  000015'                               MOV     #BUF+13.,R0
 000076  012701  000004                                MOV     #4,R1
 000102  000167  000154                                JMP     DOIO
 000106     015     000     000  BUF1:   .BYTE   15,0,0,0,0,0,12,0
 000111     000     000     000
 000114     012     000
                                                ;  LINE FEED, CARRIAGE RETURN
                                                ;   CALL LFCR
 000116                                         CRLF:
 000116  010046                                 LFCR:   MOV     R0,-(SP)
 000120  010146                                        MOV     R1,-(SP)
 000122  012700  000106'                               MOV     #BUF1-R0
 000126  012701  000010                                MOV     #10,R1
 000132  000167  000124                                JMP     DOIO
 000136                                         CR:
 000136  010046                                        MOV     R0,-(SP)
 000140  010146                                        MOV     R1,-(SP)
```

```
     51 000142  012700  000106'                    MOV     #BUF1,R0
     52 000146  012701  000006                     MOV     #6,R1
     53 000152  000167  000104                     JMP     DOIO
     54 000156                             LF:
     55 000156  010046                             MOV     R0,-(SP)
     56 000160  010146                             MOV     R1,-(SP)
     57 000162  012700  000114'                    MOV     #BUF1+6,R0
     58 000166  012701  000002                     MOV     #2,R1
     59 000172  000167  000064                     JMP     DOIO
     60 000176  000000  000206' 000210' BUF2:     .WORD   0,BUF2+10,BUF2+12,BUF2+14,0,767.,0
        000204  000212' 000000  001377
        000212  000000
     61                                            ;  POSITION CURSER UPPER LEFT SCREEN
     62                                            ;    CALL HOME
     63 000214                             HOME:
     64 000214  010546                             MOV     R5,-(SP)
     65 000216  012705  000176'                    MOV     #BUF2,R5
     66 000222  004767  000066                     JSR     PC,IPLOT
     67 000226  004767  177614                     JSR     PC,ALPHA
     68 000232  012605                             MOV     (SP)+,R5
     69 000234  000207                             RTS     PC
     70 000236     007     000      000  BUF3:     .BYTE   7,0,0,0
        000241     000
     71                                            ;  RING BELL
     72                                            ;    CALL BELL
     73 000242                             BELL:
     74 000242  010046                             MOV     R0,-(SP)
     75 000244  010146                             MOV     R1,-(SP)
     76 000246  012700  000236'                    MOV     #BUF3,R0
     77 000252  012701  000004                     MOV     #4,R1
     78 000256  000167  000000                     JMP     DOIO
     79 000262                             DOIO:
     80 000262  105737  175614                     TSTB    @#CSR
     81 000266  100375                             BPL     DOIO
     82 000270  112037  175616                     MOVB    (R0)+,@#DATO
     83 000274  077106                             SOB     R1,DOIO
     84 000276  012601                             MOV     (SP)+,R1
     85 000300  012600                             MOV     (SP)+,R0
     86 000302  000207                             RTS     PC
     87 000304  000000                     LEN:    .WORD   0
     88 000306                             BUFF:   .BLKB   6
     89                                            ;  POSITION GRAPHICS CURSOR
     90                                            ;    CALL PLOT(IX,IY,IPEN)
     91                                            ;    IX,IY=SCREEN COORDINATES 0-1024
     92                                            ;    IPEN=1 PEN DOWN
     93                                            ;        =0 PEN UP
     94                                            ;    FIRST POINT MUST BE PEN UP
     95 000314                             IPLOT:
     96 000314  010046                             MOV     R0,-(SP)
     97 000316  010146                             MOV     R1,-(SP)
     98 000320  010346                             MOV     R3,-(SP)
     99 000322  010446                             MOV     R4,-(SP)
    100 000324  012704  000313'                    MOV     #BUFF+5,R4
    101 000330  012767  000004  177746             MOV     #4,LEN
    102 000336  012746  000001                     MOV     #1,-(SP)
    103 000342  017500  000002                     MOV     @2(R5),R0
    104 000346  010001                     1$:     MOV     R0,R1
```

```
105 000350  042700  176037                  BIC     #176037,R0
106 000354  012703  000005                  MOV     #5,R3
107 000360  006200                  2$:     ASR     R0
108 000362  077302                          SOB     R3,2$
109 000364  052700  000040                  BIS     #40,R0
110 000370  042701  177740                  BIC     #177740,R1
111 000374  052701  000100                  BIS     #100,R1
112 000400  005716                          TST     (SP)
113 000402  100002                          BPL     5$
114 000404  052701  000040                  BIS     #40,R1
115 000410  110144                  5$:     MOVB    R1,-(R4)
116 000412  110044                          MOVB    R0,-(R4)
117 000414  005716                          TST     (SP)
118 000416  100405                          BMI     10$
119 000420  005416                          NEG     (SP)
120 000422  017500  000004                  MOV     @4(R5),R0
121 000426  000167  177714                  JMP     1$
122 000432  017500  000006          10$:    MOV     @6(R5),R0
123 000436  001004                          BNE     20$
124 000440  005267  177640                  INC     LEN
125 000444  112744  000035                  MOVB    #35,-(R4)
126 000450  016700  177630          20$:    MOV     LEN,R0
127 000454  105737  175614          25$:    TSTB    @#CSR
128 000460  100375                          BPL     25$
129 000462  112437  175616                  MOVB    (R4)+,@#DATO
130 000466  077006                          SOB     R0,25$
131 000470  005726                          TST     (SP)+
132 000472  012604                          MOV     (SP)+,R4
133 000474  012603                          MOV     (SP)+,R3
134 000476  012601                          MOV     (SP)+,R1
135 000500  012600                          MOV     (SP)+,R0
136 000502  000207                          RTS     PC
137                                         ;  OUTPUT ASCII CHARACTERS
138                                         ;    CALL CHOUT(IBUF)
139                                         ;    IBUF=N,ARRAY OF CHARACTERS
140                                         ;    N=NUMBER OF CHARACTERS TO BE OUTPUT
141 000504                          CHOUT:
142 000504  004767  177336                  JSR     PC,ALPHA
143 000510  010046                          MOV     R0,-(SP)
144 000512  010146                          MOV     R1,-(SP)
145 000514  016500  000002                  MOV     2(R5),R0
146 000520  062700  000002                  ADD     #2,R0
147 000524  017501  000002                  MOV     @2(R5),R1
148 000530  000167  177526                  JMP     DOIO
149                                 ;
150                                 ;       PRINT NORMAL SIZED CHARACTER STRINGS
151                                 ;       CALL CHOUTS(IBUF,N)
152                                 ;       IBUF = CHARACTER STRING
153                                 ;       N = NUMBER OF ASCII CHARACTERS
154                                 ;
155 000534  004767  177306          CHOUTS: JSR     PC,ALPHA
156 000540  010046                          MOV     R0,-(SP)
157 000542  010146                          MOV     R1,-(SP)
158 000544  016500  000002                  MOV     2(R5),R0
159 000550  017501  000004                  MOV     @4(R5),R1
160 000554  000167  177502                  JMP     DOIO
161 000560  000000                  CHS:    .WORD
```

```
000562                          PRBF:   .BLKW   38.
                                ;PRINT VERTICAL LABEL OR HORIZONTAL DOUBLE SPACED LABEL
                                ;       CALL PRTV(IBUF,N)
                                ;       CALL PRTH(IBUF,N)
                                ;       IBUF=CHARACTER BUFFER
                                ;       N=NUMBER OF CHARACTERS
000676  010046                  PRTV:   MOV     R0,-(SP)
000700  010146                          MOV     R1,-(SP)
000702  010246                          MOV     R2,-(SP)
000704  012767  005010  177646          MOV     #5010,CHS
000712  000167  000014                  JMP     OUT
000716  010046                  PRTH:   MOV     R0,-(SP)
000720  010146                          MOV     R1,-(SP)
000722  010246                          MOV     R2,-(SP)
000724  012767  000040  177626          MOV     #40,CHS
000732  004767  177130          OUT:    JSR     PC,ALPHA2
000736  017500  000004                  MOV     @4(R5),R0
000742  012702  000562'                 MOV     #PRBF,R2
000746  016501  000002                  MOV     2(R5),R1
000752  112122                  L10:    MOVB    (R1)+,(R2)+
000754  116722  177600                  MOVB    CHS,(R2)+
000760  116722  177575                  MOVB    CHS+1,(R2)+
000764  077006                          SOB     R0,L10
000766  017501  000004                  MOV     @4(R5),R1
000772  070127  000003                  MUL     #3,R1
000776  012602                          MOV     (SP)+,R2
001000  012700  000562'                 MOV     #PRBF,R0
001004  005267  000002'                 INC     LBLUP
001010  000167  177246                  JMP     DOIO
001014  005267  000242          GINT:   INC     WTSE
001020  000002                          RTI
001022  000000                  PSTAT:  .WORD   0
001024  000000                  INTLOC: .WORD   0
                                ;  ALLOW GRAPHIC INPUT FROM JOYSTICK
                                ;    CALL GRIN(IALPHA,IX,IY)
                                ;    PLACES GRAPHIC CROSSHAIR ON SCREEN
                                ;    COORDINATE IS TRANSMITED WHEN ANY KEYBOARD CHARACTER IS ENTERED
                                ;    RETURNS:  IALPHA=KEYBOARD CHARACTER
                                ;              IX,IY=SCREEN COORDINATES 0-1024
001026  010046                  GRIN:   MOV     R0,-(SP)
001030  010146                          MOV     R1,-(SP)
001032  005067  000214                  CLR     GBF
001036  005067  000220                  CLR     WTSE
001042  052737  000100  175610          BIS     #100,@#WCSR
001050  105737  175614          G1:     TSTB    @#CSR
001054  100375                          BPL     G1
001056  112737  000021  175616          MOVB    #21,@#DATO
001064  012737  001014' 000370          MOV     #GINT,@#TTV
001072  012701  000007                  MOV     #7,R1
001076  012700  001253'                 MOV     #GBF+1,R0
001102  005767  000154          G2:     TST     WTSE
001106  001775                          BEQ     G2
001110  113720  175612                  MOVB    @#DATI,(R0)+
001114  005067  000142                  CLR     WTSE
001120  052737  000100  175610          BIS     #100,@#WCSR
001126  077103                          SOB     R1,G2
001130  012737  001320' 000370          MOV     #TTINT,@#TTV
```

```
                                                SWAB    GBF
                                                MOV     #2,R1
                                                MOV     GBF+2,R0
                                        G3:     SWAB    R0
                                                ASLB    R0
                                                ASLB    R0
                                                ASLB    R0
                                                ASH     #-3,R0
                                                BIC     #176000,R0
                                                MOV     @6(R5),@4(R5)
                                                MOV     R0,@6(R5)
                                                MOV     GBF+4,R0
                                                SOB     R1,G3
                                                MOV     GBF,@2(R5)
                                                CMP     GBF,#33 ;CHECK FOR ESCAPE
                                                BEQ     ESCAPE
                                                SUB     #7,@4(R5)
                                                BGT     1$
                                                CLR     @4(R5)
                                        1$:     MOV     (SP)+,R1
                                                MOV     (SP)+,R0
                                                RTS     PC
                                        GBF:    .BLKW   4
                                        WTSE:   .WORD   0
                                        ;  ENABLE UNSOLICITED SINGLE CHARACTER INPUT FROM TERMINAL TO TEKCOM
                                        ;    COMMON/TEKCOM/ICHBUF
                                        ;    CALL TEKIN
                                        TEKIN:  MOV     #TTINT,@#TTV
                                                CLR     @#TTV+2
                                                BIS     #100,@#WCSR
                                                BIS     #200,@#CSR
                                                CLR     ICHBUF
                                                RTS     PC
                                        TTINT:  MOVB    @#DATI,ICHBUF
                                                CMP     ICHBUF,#33      ;CHECK FOR ESCAPE
                                                BEQ     ESCAPE
                                                CMP     ICHBUF,#21      ;CHECK FOR DCR1(CNTRL Q)
                                                BEQ     STOP
                                                CMP     ICHBUF,#22      ;CHECK FOR DCR2(CNTRL R)
                                                BEQ     STOP
                                        TTNT1:  BIS     #100,@#WCSR
                                                RTI
                                        ESCAPE: MOV     #110,ICHBUF     ;SET H
                                                MOV     (SP)+,INTLOC
                                                MOV     (SP)+,PSTAT
                                                MOV     #STBASE,SP      ;RESET STACK TO TOP
                                                MOV     PSTAT,-(SP)     ;PROSSESOR STATUS
                                                MOV     #CNTCAL,-(SP)   ;ADDRESS OF MMS CALL TO CNTRL
                                                BR      TTNT1
                                        STOP:   TST     @#167740        ;CHECK INTERUPT JUST ARRIVED
                                                BLT     TTNT2
                                                MOV     R0,-(SP)
                                                MOV     #1,R0           ;ONE WORD COMMAND
                                                CMP     ICHBUF,#21
                                                BEQ     1$
                                        1$:     JSR     PC,WRTLSI       ;STOP SCANNER
                                                MOV     (SP)+,R0
```

```
    276 001454  005067  000000'         TTNT2:  CLR     ICHBUF
    277 001460  052737  000100  175610         BIS     #100,@#WCSR
    278 001466  000002                         RTI
    279         000000'                        .CSECT  TEKCOM
    280 000000  000000                 ICHBUF: .WORD   0
    281 000002  000000                 LBLUP:  .WORD   0
    282         000001'                        .END
```

```
TEKMOD  RT-11 MACRO VM02-12   15-SEP-80 PAGE 1+
SYMBOL TABLE

ALPHA   000046RG        ALPHA2  000066RG        BELL    000242RG        BUF     000000R         BUFF    000306R
BUF1    000106R         BUF2    000176R         BUF3    000236R         CHOUT   000504RG        CHOUTS  000534RG
CHS     000560R         CNTCAL= ****** G        CR      000136RG        CRLF    000116RG        CSR   = 175614
DATI  = 175612          DATO  = 175616          DOIO    000262R         ERASE   000022RG        ESCAPE  001366R
GBF     001252R         GINT    001014R         GRIN    001026RG        G1      001050R         G2      001102R
G3      001152R         HOME    000214RG        ICHBUF  000000R   002   INTLOC  001024R         IPLOT   000314RG
LBLUP   000002R   002   LEN     000304R         LF      000156RG        LFCR    000116RG        L10     000752R
OUT     000732R         PC    =%000007          PRBF    000562R         PRTH    000716RG        PRTV    000676RG
PS1AT   001022R         R0    =%000000          R1    =%000001          R2    =%000002          R3    =%000003
R4    =%000004          R5    =%000005          SP    =%000006          STBASE= 001000          STOP    001422R
TEKIN   001264RG        TTINT   001320R         TTNT1   001356R         TTNT2   001454R         TTV   = 000370
WCSR  = 175610          WRTLSI= ****** G        WTSE    001262R         ...V2 = 000001
. ABS.  000000    000
        001470    001
TEKCOM  000004    002
ERRORS DETECTED: 0
FREE CORE: 17777. WORDS

,TT:/N:TTM=TEKMDX

ERRORS DETECTED: 0
FREE CORE: 17777. WORDS

*
```