PROGRAM DOCUMENTATION

CREW SYSTEM VIDAR DATA REDUCTION (CSVIDR) PROGRAM

Program Q939

Job Order 83-157

(NASA-CR-147709) PROGRAM DOCUMENTATION CREW SYSTEM VIDAR DATA REDUCTION (CSVIDR) PROGRAM (Lockheed Electronics Co.) 99 p HC $5.00 CSCL 09B

N76-23862

Unclas
G3/60 41450

Prepared By

Lockheed Electronics Company, Inc.
Aerospace Systems Division
Houston, Texas

Contract NAS 9-12200

For

INSTITUTIONAL DATA SYSTEMS DIVISION

DATA SYSTEMS AND ANALYSIS DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

March 1976

LEC-7253

JSC-10948

PROGRAM DOCUMENTATION

CREW SYSTEM VIDAR DATA REDUCTION (CSVIDR) PROGRAM

Job Order 83-157

PREPARED BY

H. B. Van Wie
Data Processing Systems Department

APPROVED BY

LEC

C. E. Hutchinson, Supervisor
Data Processing Systems Department

W. N. Fitzpatrick, Manager
Data Processing Systems Department

NASA

W. D. Poates, Technical Monitor
Data Processing Branch
4-19-76

Prepared By

Lockheed Electronics Company, Inc.

For

Institutional Data Systems Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

March 1976

LEC-7253

TECHNICAL REPORT INDEX/ABSTRACT

(See instructions on reverse side.)

1. TITLE AND SUBTITLE OF DOCUMENT PROGRAM DOCUMENTATION CREW SYSTEM VIDAR DATA REDUCTION (CSVIDR) PROGRAM	2. JSC NO. JSC-10948
3. CONTRACTOR/ORGANIZATION NAME Lockheed Electronics Company, Inc.	4. CONTRACT OR GRANT NO. NAS 9-12200
5. CONTRACTOR/ORIGINATOR DOCUMENT NO. LEC-7253	6. PUBLICATION DATE (THIS ISSUE) March 1976
7. SECURITY CLASSIFICATION Unclassifed	8. OPR (OFFICE OF PRIMARY RESPONSIBILITY) W. D. Poates/FD5
9. LIMITATIONS GOVERNMENT HAS UNLIMITED RIGHTS [X] YES [ ] NO	10. AUTHOR(S) H. B. Van Wie
11. NO, STATE LIMITATIONS AND AUTHORITY	
11. DOCUMENT CONTRACT REFERENCES	12. HARDWARE CONFIGURATION
WORK BREAKDOWN STRUCTURE NO. Not applicable	SYSTEM Not applicable
CONTRACT EXHIBIT NO. Not applicable	SUBSYSTEM Not applicable
DRL NO. AND REVISION Not applicable	MAJOR EQUIPMENT GROUP Not applicable
DRL LINE ITEM NO. Not applicable	

13. ABSTRACT

14. SUBJECT TERMS

JSC Form 833 (Rev Sep 74) NASA-JSC

CONTENTS

1. IDENTIFICATION

Title:	CREW SYSTEM VIDAR DATA REDUCTION PROGRAM
Acronym:	CSVIDR
Program number:	Q939
Author:	H. B. Van Wie
Date:	January 1976
Installation:	JSC, Houston, Texas
Authorization:	Contract number NAS 9-12200
Source language:	FORTRAN
Computer:	UNIVAC 1108/1110
Operating system:	EXEC 8
Location of source	decks and/or CUR/PUR tape records: Lockheed Electronics Company, Inc., Data Processing Systems Department maintains all program tapes associated with this program.

2. ABSTRACT

The Crew System Vidar Data Reduction (CSVIDR) program was written to process selected portions of data acquired from long term tests of Space Shuttle crew equipment. Data is recorded on a seven-track computer compatible tape (CCT) using the Vidar Autodata Eight Processing System located in Building 7. The input tape is in a six-bit binary coded decimal (BCD) format. The 1110 hardware conversion option is used to convert all data from a BCD format to Fieldata since the majority of the data is output without further conversion.

Data is retrieved from a character string, calibrated, tabulated, printed, and output on a fixed-sample-rate tape for use as input to a general-purpose plot program.

3. HISTORY

The CSVIDR program is authorized by job order 83-157, task agreement 20. The design for the CSVIDR program was approved in August 1975, and coding was initiated at that time. Existing subroutines for curve-fit and character-string manipulations were used as necessary by the driver program. The curve fit subroutine, CURV61, is documented by both the FIT73 program (Q417), LEC-1379, February 1974, and the BIOFIT program (J252), LEC-1380, January 1974. Character string manipulation subroutines were developed in November 1974 for use with the Phase A Integrated Structural Analysis System (Phase A ISAS).

4. ANALYSIS

The CSVIDR program was written to process selected portions of data acquired from long term testing of Space Shuttle crew equipment recorded on a Vidar Autodata Eight Processing System located in Building 7. A maximum of 40 channels of data may be recorded on each data tape, and selected processing of any or all of the channels can be provided by the CSVIDR program.

Data from each channel is retrieved from a character string, and the appropriate calibrations are applied. These calibrations may be supplied in the form of ordered pairs of voltages and engineering units (EU) or of coefficients for a polynomial equation. A maximum of 21 pairs of points or coefficients for as up to a fifth-order polynomial equation may be used. For each set of calibration points supplied, the CSVIDR program calls the CURV61 subroutine which performs a curve fit and calculates the coefficients for the equation of the line which contains those points.

The data acquisition system has the capability to record the data in two distinct modes. The limits for low, medium, and high ranges for each parameter are set by the operator prior to the test. During the test, as data is being recorded for each operating channel at a set sample rate, the data from one or more channels may exceed preset ranges. When this occurs, a scan of data from those parameters which exceeds the ranges is interspersed in the set sample rate data. The constant sample rate data is known as "log" data, and the interspersed data is called "limits" data. Each is output separately on a tabulated listing. The formats of the log and limits mode data are shown in appendix F. An output tape is generated containing the fixed-sample-rate data from which plots are made using a general-purpose plot program.

5. INPUT DESCRIPTION AND FORMATS

The input tapes are seven-track BCD tapes of 556 bits per inch (bpi) density recorded on the Vidar Autodata Eight Processing System. Each word on the tape consists of 14 characters, although the 14*th* character is a selected end-of-word (EOW) character and is not treated as a data character. Each character consists of six bits.

A scan of data consists of time in hours and minutes as the first word with 1 to 40 data words. There are 36 scans of data per record. The minimum record size is 72 words (36 scans of time plus 1 data word). The maximum record size is 1476 words (36 scans of time plus 40 data words).

The word format by character is shown below:

Character:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Time:	H	H	:	M	M	B	B	B	B	B	B	B	B	S
Log data:	T	B	I	I	I	P	D	D	D	D	D	E	U	S
Limits data:	T	.	I	I	I	P	D	D	D	D	D	E	U	S
Limits data:	T	*	I	I	I	P	D	D	D	D	D	E	U	S
Limits data:	T	-	I	I	I	P	D	D	D	D	D	E	U	S
Overload:	T	B	I	I	I	B	B	B	B	B	B	B	B	S

where:

B = blank
S = selected EOW character
D = data character
H = hours
M = minutes
T = type (1 to 7)
I = channel identity
P = polarity (- or blank)
E = negative exponent

U = units (V, °F, or °C)
* = high range
. = middle range
- = low range

The first word in each record contains time followed by data words from 1 to 40 until the data scan is ended, and another time word starts the next scan. The number of data words in a scan may vary, so the time word location is variable and is detected by examining the third character of each word for a colon, which is unique to the time word.

The following codes are used to determine the type and range of the data words depending on the value of character 1:

1 = ±0.1 V	5 = °C
2 = ±1.0 V	6 = °F
3 = ±10.0 V	7 = spare
4 = auto ranging	

The data within the same record may be recorded in the log and limits modes. In the log mode, only the second character of each data word is blank and all parameters that are operating are sampled at a given sample rate; e.g., 1 sample every 15 minutes. In the limits mode, the second character of each data word is either an asterisk, period, or minus, which indicates that the data value is in the high, middle, or low portion, respectively, of the range for which limits were set prior to recording the data. The limits data does not have a set sample rate but rather is output onto the data tape when any data value changes from one limits range to another.

If the input data exceeds the voltage range selected for a parameter, an overload condition exists in which case only the type and channel identity are provided, and the remaining characters are blank.

Characters 3, 4, and 5 of each data word represent the channel identity and may have a value from 000 to 039 to represent the 40 channels.

The polarity indicated by character 6 of each data word specifies whether a data value is negative or positive. For a positive number, the character is blank; for a negative number, a minus sign appears.

The magnitude of the data value is determined using the five data characters 7 through 11 and the negative exponent, character 12. The following example illustrates how this is calculated:

Character:	7	8	9	10	11	12
Value:	0	0	7	5	6	1

Means: 00756×10^{-1} or 75.6

The decimal is assumed to be at the right of character 11 and is moved to the left by the number of spaces specified in character 12. Character 13 of each word designates whether the engineering units are volts, degrees Farenheit, or degrees Centigrade.

6. OUTPUT DESCRIPTION AND FORMATS

Time history tabulations and a fixed-sample-rate tape are the required output of the program. Tabulations in different formats are provided for the limits data and the log data. The format for the limits data tabulation is illustrated below.

Time DDD:HH:MM	Channel	Value	EU	Code
001:08:11	004	74.8	°F	.
	012	77.4	°F	.
	013	7.107	V	*
	015	23.4	°C	.
001:08:12	004	76.6	°C	*
	013	6.865	V	.
	015	22.3	°C	-

The log data is fixed-sample-rate data and is illustrated in table I and tabulated in the format shown in appendix F.

The output tape is a fixed-sample-rate tape of the log data only, with 42 words per scan (time, 40 data words, and one dummy word); on this tape, only the requested channels have data, and the unrequested channels are filled with a default value (all 36 bits set to 1).

The output tape consists of the following characteristics:

- Standard seven-track digital magnetic tape
- 800 bpi packing density
- Odd parity
- Buffered (non-FORTRAN written)
- Forty-two 36-bit words per scan

- Time output in integer milliseconds
- Data in floating point EU
- Twelve scans per record

TABLE I. — LOG DATA TEST TITLE INFORMATION

Time	Channel	Units of measurement									
DDD:HH:MM	1-10	DEG C	DEG C	DEG F	DEG C	DEG C	DEG C	DEG F	DEG C	DEG C	DEG C
	11-20	DEG F	PSIA	DEG F	DEG F	DEG F	DEG F	DEG F	WATTS	DEG F	DEG F
	21-30	VOLTS	VOLTS	VOLTS	LB/HR	PSIG	VOLTS	VOLTS	VOLTS	PSID	VOLTS
	31-40	GAL/SEC	GAL/SEC	GAL/SEC	GAL/SEC	PSID	GAL/MIN	GAL/MIN	GAL/MIN	GAL/MIN	GAL/MIN
001:08:06	1-10	24.6	25.1	23.8	24.2	26.1	23.5	25.7	24.8	26.3	25.5
	11-20	77.9	76.3	75.4	75.8	77.2	76.5	78.1	76.6	75.7	74.0
	21-30	0.0634	0.0963	-0.245	0.418	0.623	2.852	-4.691	7.529	5.618	3.685
	31-40	0.523	0.689	0.419	0.732	0.613	29.53	31.76	33.24	32.68	30.07
001:08:21	1-10	25.1	25.3	23.9	24.0	26.3	23.3	25.9	24.7	26.5	25.4
	11-20	78.0	76.5	75.6	76.0	77.2	76.8	78.1	76.7	75.6	74.2
	21-30	0.0649	0.0932	-0.236	0.436	0.635	2.845	-4.723	7.613	5.630	3.695
	31-40	0.545	0.707	0.451	0.767	0.610	30.07	31.99	33.11	32.75	30.24
001:08:36	1-10	25.4	25.6	23.8	24.1	26.6	23.0	26.0	24.9	26.4	25.3
	11-20	78.2	76.5	75.7	76.2	77.1	76.7	78.1	76.8	75.7	74.3
	21-30	0.0657	0.0925	-0.222	0.441	0.648	2.863	-4.739	7.636	5.641	3.704
	31-40	0.563	0.722	0.476	0.789	0.604	30.15	32.12	32.96	32.81	30.35

7. RESTRICTIONS

The driver program, CSVIDR, is a special purpose program which uses parameter statements to permit minor changes, but the CSVIDR program was designed for the Vidar Autodata Eight Processing System. In its current configuration, the use of FASTRAND input/output (I/O) is discouraged since the record length of both the input and output tapes are not optimum for 36-bit words or 28-word sectors.

8. FACILITY USAGE REQUIREMENTS

The CSVIDR program requires less than 20,000 elements of core and uses external assignments for three tapes and one FASTRAND file. Provision is made for only one output tape. FORTRAN I/O and NTRAN languages are used.

9. RUNNING TIME

The CSVIDR program requires approximately 0.00315 seconds to process one data sample. If the typical input tape contains data for 30 channels for over 40 days with one scan of data every 15 minutes, then:

(1) Compute seconds per scan:

0.00315 seconds × 30 channels = 0.09450 seconds/scan

(2) Determine total scans:

From (40 days × 24 hr/day × 60 min/hr)/15 min/scan = 3840 scans in 40 days.

(3) Calculate EXEC 8 time in minutes:

0.09450 seconds/scan × 3840 scans = 6 minutes .

10. NONSTANDARD SYSTEM REQUIREMENTS

The CSVIDR program runs on the UNIVAC 1110, EXEC 8 system and utilizes the hardware BCD to Fieldata conversion feature.

APPENDIX A

RUNNING INSTRUCTIONS

APPENDIX A

RUNNING INSTRUCTIONS

The CSVIDR program reads and analyzes the lead cards to determine the channels to process, the types of calibrations necessary, and the assignment of the I/O devices. Time slice information and the EOW character are also input by lead cards. The necessary default values are calculated, and the initial conditions are printed.

If an error is detected while reading the lead cards, the remaining lead cards are read, and an error message is written. The default situation of no lead cards also causes a diagnostic message, but the program attempts to retrieve and output raw data for all channels for the entire data tape.

A.1 LEAD CARDS

- Type 1 (title card)

  The first lead card read is a title card and may contain up to 72 alphanumeric characters. The title card information is displayed as test title information for all tabular output.

- Type 2 (time card)

  Lead card type 2 has the word TIMEΔ left justified in the first five card columns and contains fields available for time in days, hours, and minutes (DDD:HH:MM) over a start/stop time interval. An additional field is available to provide the first time at the start of the tape. Since only hours and minutes are present on the input data tape, it is necessary to input a value for days to provide for a day rollover.

- Type 3 (tape I/O cards)

  Lead card type 3 has TAPEIO punched in card columns 1 through 6 and may contain the internal file names for two input tapes and

one output tape. Provision is also made to input the number of files to skip on the input tapes and the EOW character used to separate data words. The internal file name for the first input tape must be the same as on the @USE lead card. If a continuation input reel of tape is used, the internal file name must match its @USE lead card.

The fixed-sample-rate output tape also must have an internal file name supplied. Blank fields on the TAPEIO card or omission of the TAPEIO card result in the following default values:

TAPE1 = 15, input tape 1
TAPE2 = 0, no tape 2
TAPEF = 7, output tape
MOVE1 = 0, first file for first tape
MOVE2 = 0, first file for second input tape, if it is present
EOW = a blank EOW character defaults to M, which is input as @@@@@M.

- Types 4 and 5 (calibration cards)

Card types 4 and 5 are submitted together. Type 4 contains a two-digit channel number from 1 to 40 corresponding to the input tape identity fields 000 through 039. It also contains the degree (M) of the calibration curve which must be greater than 0 and less than 6 (0<M<6). The number of input calibration points is supplied on the type 5 lead card and ranges from 2 to 21. For every type 4 card, one or more type 5 cards follow.

Free field input is used to supply either input points for curve-fit or polynomial coefficients. In either case a search is made for the $XYCALS name and either XYPAIR= or POLYCO= will be found with input points or coefficients, respectively. (See also the JSC Procedures Manual Part 15 IDSD 15:60.030 for FORTRAN V guidelines, and appendix B.)

Omission of card types 4 and 5 permits the input of data from tape and tabular tape output of raw data.

A.2 CALIBRATION

If calibration is requested as flagged by input of a set of XYPAIR calibration points, the CURV61 subroutine is called to return the coefficients. These are stored in an array indexed by channel number. The remaining calibration sets are read in until coefficients for all requested channels are computed.

If the last lead card has been read and interpreted, the start/stop time and the first tape time values in DDD:HH:MM are converted to integer milliseconds. The DHM2MS subroutine is called to standardize time, since the time units for the fixed-sample-rate tape are integer milliseconds.

A.3 INITIALIZATION

Initial conditions for execution are printed, and if the lead card information is sensed as bad, the CSVIDR program error exits. If the bad input flag BADCRD is not set, the NTRAN subroutine is called to read the input tape. Validity checking of the input tape quality is limited to a maximum of 20 read errors, and word count is checked for a minimum of 168 words and a maximum of 3,444 to permit variable channel selection. The input of any number of channels from 1 to 40 is permitted.

A.4 RETRIEVAL OF TIME

If the input tape record is good, the SCHCHR subroutine is called to search for the first time code character, T1CHR, and the EOW character, EOWCHR. The third character of the first word of each tape input record contains character T1CHR, and EOWCHR is the 14*th* character from the start of the tape record. If the time word is confirmed, the SCHCHR subroutine is again called to find

the next time character, T2CHR, and the number of data words per scan is calculated. The number of days supplied by lead card plus the time word values in hours and minutes from the input scan are used by the DHM2MS subroutine to convert time to integer milliseconds.

If the time is less than the requested start time, as many additional scans of data as necessary are read at 36 scans per record until the start time is found or exceeded.

A.5 RETRIEVAL OF CHANNEL DATA

For each scan of data retrieved, the FD2INT subroutine is called to convert the channel characters from Fieldata to integer format in order to compare them to the card input or default list requested. If a data channel is not requested, the list is read until it is exhausted and a new time scan is examined. If the channel is requested, character one, CHR1, is checked for data types 1 through 7. If CHR1 is 1 through 4, the FD2FD subroutine is called to determine the Fieldata characters necessary to compute the floating point value of the raw data. The EVALFT subroutine is given the data value with the coefficients specified by the channel index to provide the computed dependent variable value, ZDATA. The ZDATA value is placed in the PLTBUF array, which is used to store data for the fixed-sample-rate output tape. The ZDATA array is also converted from floating point to Fieldata using the FP2FD subroutine to fill the location in either the log data line or the limits data line. The input tape format may be reviewed for log mode or limits mode.

A.6 TABULAR OUTPUT

The character string for tabular output of the limits mode data is constructed and stored in a page-sized buffer and output by the page onto a FASTRAND file according to the output format

specified in section 6. Each line of the log mode data is stored and spaced by channel number and output for each scan in the specified format.

A.7 FIXED-SAMPLE-RATE TAPE OUTPUT

For each 12 time scans, the PLTBUF buffer is output to the fixed-sample-rate tape. Additional input records are read and data retrieved, evaluated, and calibrated with log mode data tabbed; limits data stored; and fixed-sample-rate tape records written until the stop time is exceeded or the end-of-file (EOF) is sensed.

A.8 TERMINATION OF TAPE INPUT

If a continuation reel is required and the stop time has not been exceeded, the NTRAN subroutine rewinds and interlocks the first input tape, assigns the continuation reel according to the lead card and @USE card, and continues processing. If a continuation reel is not used and either stop time, EOF, or the maximum parity records are exceeded, the last buffer with PLTBUF data and/or fill is written. An EOF is written with interlock.

A.9 OUTPUT SUMMARY

The remaining data in the limits mode page buffer is written on the temporary status file (TSF). The TSF is read and the limits data printed by the page of 56 lines with 72 Fieldata characters each until all TSF blocks are read and written.

A summary is printed as follows:

IREC = number of input records
TREC = number of tape records written
DAY = same day as input unless incremented by detecting a backup in hours (i.e., day rollover)
HOUR = last hour on the input tape

MINUTE = last minute on the input tape
TIMOUT = last time converted to milliseconds

A.10 ERROR DETECTION

Error messages written by the CSVIDR program always start with **CSVIDR** and output, in addition, the last character position, LSTPTR; the starting character for a time scan, ISTCH; and the current scan, ISCAN. The UNIVAC EXEC 8 Hardware/Software Reference (UP7824) may be referenced for general error and status information.

APPENDIX B

LEAD CARD SETUPS

LEAD CARD SETUP

CARD NO. Type 1

JOB

PAGE 1 OF 6

NAME CSVIDR PROGRAMMER Van Wie DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1	1-72	12A6	TITLE	Tabular heading for log mode data and
				limits mode data

LEAD CARD SETUP

CARD NO. TYPE 2

JOB

NAME CSVIDR PROGRAMMER Van Wie DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1	1-6	A6	CARDIN	Punch TIMEΔΔ only in this field
2	17-19	I3	DDD(1)	Days
3	21-22	I2	HH(1)	Hours — Start time
4	24-25	I2	MM(1)	Minutes
5	37-39	I3	DDD(2)	Days
6	41-42	I2	HH(2)	Hours — Stop time
7	44-45	I2	MM(2)	Minutes
8	57-59	I3	DDD(3)	Days
9	61-62	I2	HH(3)	Hours — Start of tape (used in case
10	64-65	I2	MM(3)	Minutes — of rollover)

Comments: If the TIME card is missing, time is from first scan to last scan on the tape.

LEAD CARD SETUP

CARD NO. Type 3

JOB

NAME CSVIDR

PROGRAMMER Van Wie

DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1	1-6	A6	CARDIN	Lead card TAPEIO
2	14-15	I2	TAPE1	Internal file name for first input tape
3	24-25	I2	TAPE2	Internal file name for second input
				tape
4	34-35	I2	TAPEF	Internal file name for fixed-sample-
				rate tape
5	44-45	I2	MOVE1	Number of files to move over on TAPE1
6	54-55	I2	MOVE2	Number of files to move over on TAPE2
7	62	A1	EOW	EOW character; default or blank permits
				the character M.

Comments: Default for missing TAPEIO card causes TAPE1 = 15, TAPE2 = 0, TAPEF = 7, MOVE1 = 0, and MOVE2 = 0.

LEAD CARD SETUP

CARD NO. Type 4

JOB

NAME CSVIDR PROGRAMMER Van Wie DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
1	7-8	I2	IDCHAN	Channel number (2-digit integer from
				1 to 40)
2	10-18	A9	EUNIT	Output unit label
3	24	I1	M	Degree of curve (>0, <6)
4	29-30	I2	N	Number of calibration points
				(XYPAIR to follow) $2 \leq N \leq 21$

Comments: This lead card must be followed by a type 5 card.

LEAD CARD SETUP

CARD NO. Type 5

JOB

NAME CSVIDR PROGRAMMER Van Wie DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
	1			This is a free field using the namelist.
				(See also IDSD 15:60.030.)
	2-9			$XYCALSΔ
	10-16			XYPAIR= or POLYCO=
	17-72			This is a free field.
				Input of calibration data for XYPAIR is
				as follows:
				The first character on each input
				record (card image) is not used by the
				program. The user may put his own
				sequence numbers here if desired.
				The first input record read using
				$XYCALSΔ must have a $ in the second
				character space followed by XYCALSΔ
				with a blank in card image column 9.
				Either the variable name XYPAIR= or
				POLYCO= is located and read with its
				corresponding data input constants
				separated by commas.

Comments: This lead card must be preceded by a type 4 card.

LEAD CARD SETUP

CARD NO. Type 5 (cont.)

JOB _____ PAGE 6 OF 6

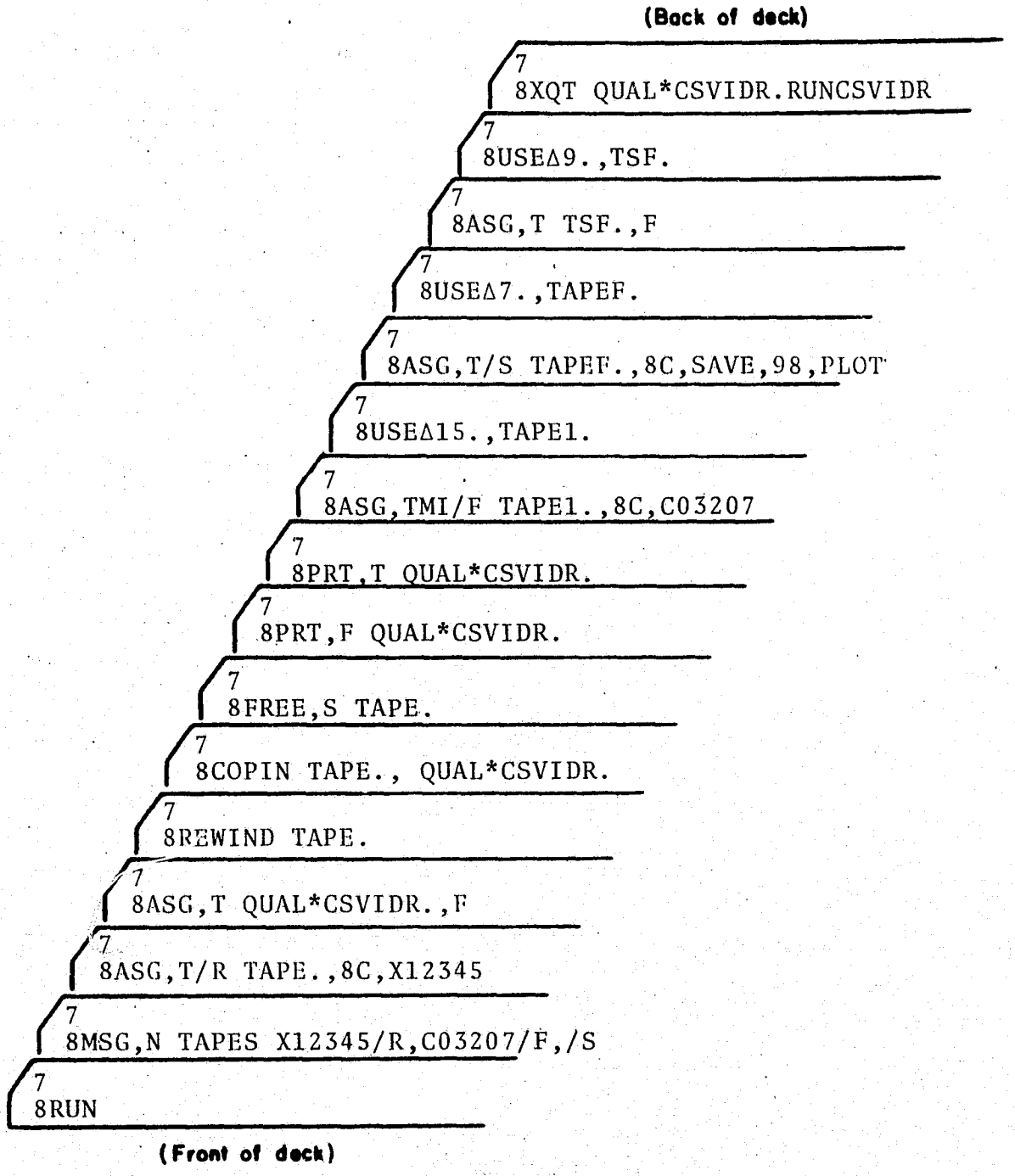
NAME CSVIDR PROGRAMMER Van Wie DATE 12/75

FIELD ID	CARD COLUMNS	FORMAT	SYMBOLIC NAME	IDENTIFICATION
				All subsequent blanks are ignored with
				only the comma or end of a card image
				denoting the end of a data field.
				The second variable name may continue
				on the same input record if at least
				one data item follows.
				The last item on an input record must
				be a constant followed by either a comma
				or the end of a card image. An identi-
				fication field may not be punched in
				columns 73 through 80.
				The end of a data set, not to exceed 21
				points, is signaled by a $END found any-
				where in a card image, except in the
				first character position.
				Decimal points may be used as necessary
				or omitted for integer input.

APPENDIX C

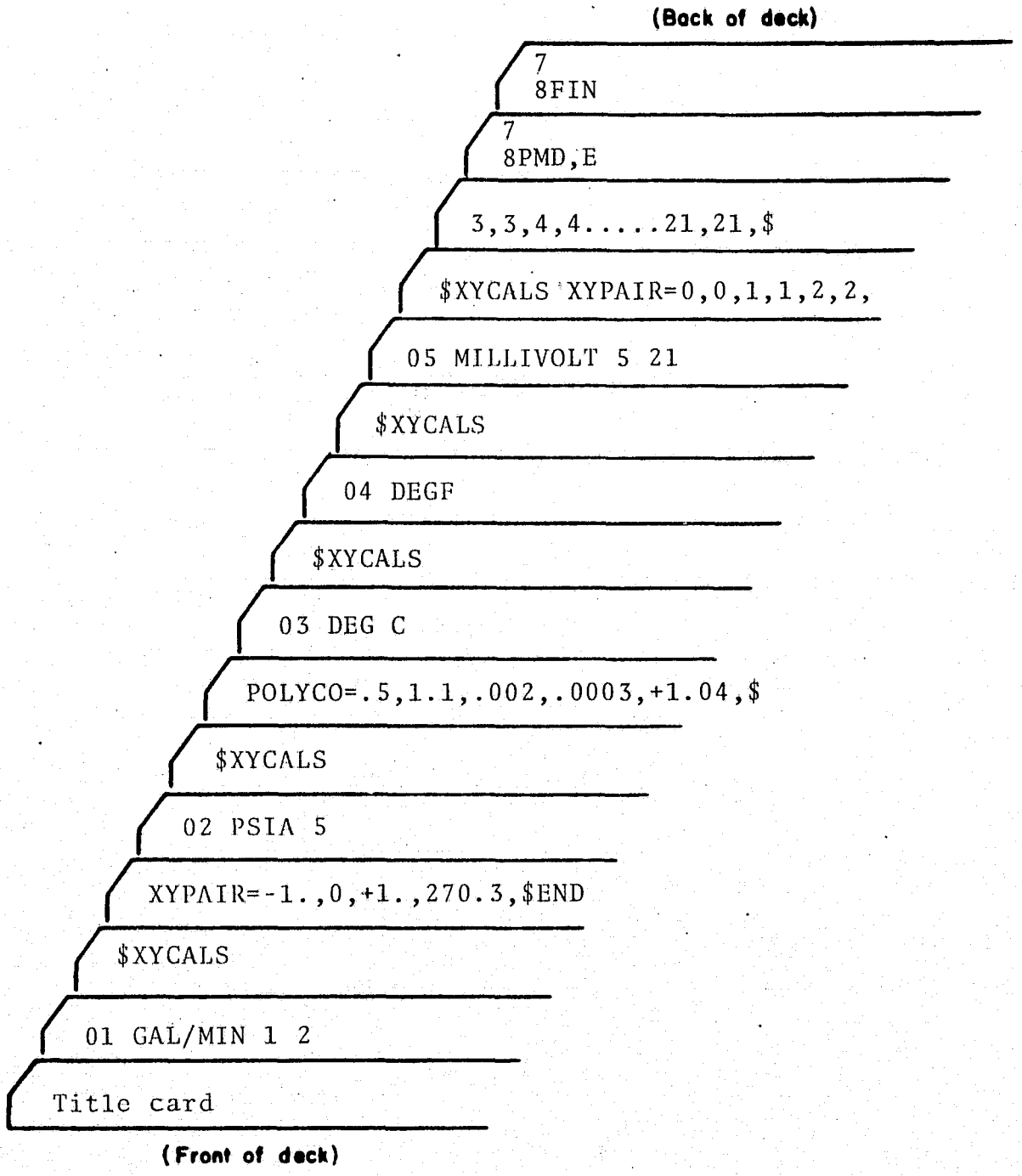
DECK SETUPS

DECK SETUP FOR PROGRAM CSVIDR PAGE NO. 1 OF 2



ASD 060

DECK SETUP FOR PROGRAM CSVIDR PAGE NO. 2 OF 2



APPENDIX D

DIAGNOSTIC MESSAGES

APPENDIX D

DIAGNOSTIC MESSAGES

CSVIDRHAS ENCOUNTERED A BAD INPUT RECORD.
THE ERROR OCCURRED TRYING TO READ RECORD

CSVIDRCOULD NOT FIND THE TIME CHARACTER

CSVIDRCOULD NOT FIND END OF WORD CHARACTER FOR SCAN

CSVIDRDOES NOT HAVE ANY LEAD CARDS. DEFAULTS USED

CSVIDRHAS AN ERROR IN CHANNEL NUMBER

CSVIDRHAS AN ERROR IN FREE FIELD INPUT

CSVIDRCOULD NOT FIND TYPE

CSVIDRUNABLE TO CONVERT DAYS TO MS

CSVIDRUNABLE TO CONVERT MS TO DAYS

CSVIDREXIT IREC=_____ LREC=_____
REC=_____ LSTPTR=_____ ISTC=_____ ISCAN=_____

APPENDIX E

SAMPLE INPUT

ORIGINAL PAGE IS OF POOR QUALITY

```
@PRT,S CALIEB.C03209
FURPUR 3076-10/31-17:06
FDS-L75C36*CALIEB.C03209
    1                    ACC.NO.  C03-2-000009
    TIME               000 09 32            FEB 13 76               000 09 32
    TAPEIC TAPLI=15  TAPET=00  TAPLF=07                              @@@@@M
          1 T/C            1      2
     $XYCALS XYPAIR=0,0,1,1,$
          2 T/C            1      2
     $XYCALS XYPAIR=0,0,1,1,$
          3 LABEL 002      1      2
     $XYCALS XYPAIR=0,0,1,1,$
         11 DEG F          1      2
     $XYCALS XYPAIR=0.,0.,5.,150.,$
         12 PSIA           1      2
     $XYCALS XYPAIR=0.,0.,5.,100.,$
         13 LB/HR          1      2
     $XYCALS XYPAIR=0,0,5,1200,$
         14 PSID           1      2
     $XYCALS XYPAIR=0,0,5,100,$
         15 DEG F          1      2
     $XYCALS XYPAIR=0,0,5,150,$
         16 LB/HR          1      2
     $XYCALS XYPAIR=0,0,5,1200,$
         17 PSIC           1      2
     $XYCALS XYPAIR=0,0,5,100,$
         18 VOLTS          1      2
     $XYCALS XYPAIR=0,0,.15,150,$
         19 WATTS          1      2
     $XYCALS XYPAIR=0,0,.1,1000,$
         20 PSIA           1      2
     $XYCALS XYPAIR=0,0,5,50,$
         21 LB/HR          1      2
     $XYCALS XYPAIR=0,0,5,1200,$
         22 LABEL 022      1      2
     $XYCALS XYPAIR=0,0,1,1,$
         23 LABEL 023      1      2
     $XYCALS XYPAIR=0,0,1,1,$
         24 PSIC           1      2
     $XYCALS XYPAIR=0.,0.,5.,100.,$
         25 PSIG           1      2
     $XYCALS XYPAIR=0,0,5,100,$
         26 LB/HR          1      2
     $XYCALS XYPAIR=0,0,5,1300,$
         27 WATTS          1      2
     $XYCALS XYPAIR=0,0,.1,1000,$
         28 VOLTS          1      2
     $XYCALS XYPAIR=0,0,.15,150,$
         29 LABEL 029      1      2
     $XYCALS XYPAIR=0,0,1,1,$
         30 VOLTS          1      2
     $XYCALS XYPAIR=0,0,.15,150,$
@XQT CSVIDR.RUNCSVIDR
```

```
          ACC.NO.  CG3-2-000009
TIME            DDD 09 32          DDD 13 26          DDD 09 32
TAPEIO TAPE1=15  TAPE2=00  TAPEF=07                    ¢¢¢¢¢¢M
          1 T/C              1       2
          2 T/C              1       2
          3 LABEL CG2        1       2
         11 DEG F            1       2
         12 PSIA             1       2
         13 LB/HR            1       2
         14 PSIB             1       2
         15 DEG F            1       2
         16 LB/HR            1       2
         17 PSIG             1       2
         18 VOLTS            1       2
         19 WATTS            1       2
         20 PSIA             1       2
         21 LB/HR            1       2
         22 LABEL C22        1       2
         23 LABEL C23        1       2
         24 PSIB             1       2
         25 PSIG             1       2
         26 LB/HR            1       2
         27 WATTS            1       2
         28 VOLTS            1       2
         29 LABEL C29        1       2
         30 VOLTS            1       2

 CSVIOR WILL EXECUTE USING INFO AS FOLLOWS:

 TITLE= 1              ACC.NO.  CG3-2-000009

 START TIME DDD=   0 HH=  9 MM= 32 MS=     34320000

 STOP  TIME DDD=   0 HH= 13 MM= 26 MS=     48360000

 TAPE  TIME DDD=   0 HH=  9 MM= 32 MS=     34320000

 TAPEIO TAPE1 WILL USE UNIT 15

 TAPEIO TAPE2 WILL USE UNIT  0

 TAPEIO TAPEF WILL USE UNIT  7

 END OF WORD CHARACTER EOW=¢¢¢¢¢M

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
  1 T/C             1

 COEFFICIENTS    .0000000   =A0    .1000000+01=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
  2 T/C             1

 COEFFICIENTS    .0000000   =A0    .1000000+01=A1
```

```
 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
  3 LABEL DG2       1

 COEFFICIENTS     .0000000   =A0     .1000000+01=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 11 DEG F           1

 COEFFICIENTS     .0000000   =A0     .3000000+02=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 12 PSIA            1

 COEFFICIENTS     .0000000   =A0     .2000000+02=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 13 LB/HR           1

 COEFFICIENTS     .0000000   =A0     .2400000+03=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 14 PSIG            1

 COEFFICIENTS     .0000000   =A0     .2000000+02=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 15 DEG F           1

 COEFFICIENTS     .0000000   =A0     .3000000+02=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 16 LB/HR           1

 COEFFICIENTS     .0000000   =A0     .2400000+03=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 17 PSIG            1

 COEFFICIENTS     .0000000   =A0     .2000000+02=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 18 VOLTS           1

 COEFFICIENTS     .1110223-15=A0     .1000000+04=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 19 WATTS           1
```

```
COEFFICIENTS     .8881784-15=A0     .1000000+05=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 20 PSIA           1

COEFFICIENTS     .0000000    =A0     .1000000+02=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 21 LE/HF          1

COEFFICIENTS     .0000000    =A0     .2400000+03=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 22 LABEL C22      1

COEFFICIENTS     .0000000    =A0     .1000000+01=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 23 LABEL C23      1

COEFFICIENTS     .0000000    =A0     .1000000+01=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 24 PSID           1

COEFFICIENTS     .0000000    =A0     .2000000+02=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 25 PSIG           1

COEFFICIENTS     .0000000    =A0     .2000000+02=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 26 LE/HF          1

COEFFICIENTS     .0000000    =A0     .2600000+03=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 27 WATTS          1

COEFFICIENTS     .8881784-15=A0     .1000000+05=A1

CHANNELS REQUESTED ARE:
PTR  LABEL        DEG
 28 VOLTS          1

COEFFICIENTS     .1110223-15=A0     .1000000+04=A1
```

```
 CHANNELS REQUESTED ARE:
PTR  LABEL         DEG
 29 LABEL 029        1

 COEFFICIENTS     .000000      =A0      .1000000+01=A1

 CHANNELS REQUESTED ARE:
PTR  LABEL         DEG
 30 VOLTS            1

 COEFFICIENTS     .111223-15=A0      .1000000+04=A1
```

APPENDIX F

SAMPLE OUTPUT

ACC.NO. CD3-2-00009

TIME CHANNEL CDD:HH:MM	1-10 11-20 21-30 31-40	1111111111 222222222 333333333 444444444 555555555 666666666 777777777 888888888 999999999 000000000										
		DEG F LB/HR	T/C PSIA LABEL 022	LABEL LB/HR LABEL 023	PSID PSID	DEG F PSIG	LB/HR LB/HR	PSIG WATTS	VOLTS VOLTS	WATTS LABEL 029	PSIA VOLTS
0: 9:33	1-10		74.3	73.4							
0: 9:33	11-20	0.000	3.540	0.240	2.660	-0.030	-0.240	3.380	0.000	0.100	1.800
0: 9:33	21-30	0.240	0.000	0.000	0.440	-14.600	0.760	0.100	0.000	0.000	0.000
0: 9:34	1-10	73.0	70.4	73.4							
0: 9:34	11-20	-0.030	3.460	0.240	2.740	0.000	0.240	3.360	0.000	0.100	1.780
0: 9:34	21-30	0.240	0.000	0.000	0.420	-14.600	0.520	0.100	0.000	0.000	0.000
0: 9:35	1-10	73.0	70.9	73.4							
0: 9:35	11-20	0.000	3.460	0.240	2.760	-0.030	0.240	3.320	0.100	0.000	1.780
0: 9:35	21-30	0.240	0.000	0.000	0.440	-14.600	0.780	0.100	0.000	0.001	0.000
0: 9:36	1-10	73.0	70.4	73.0							
0: 9:36	11-20	0.000	3.440	0.000	2.800	0.000	0.000	3.280	0.000	0.100	1.770
0: 9:36	21-30	0.000	0.000	0.000	0.420	-14.600	0.520	0.100	0.100	0.001	0.000
0: 9:37	1-10	69.0	70.4	73.4							
0: 9:37	11-20	-0.030	3.380	0.240	2.840	0.000	0.000	3.240	0.100	0.100	1.750
0: 9:37	21-30	0.240	0.000	0.000	0.420	-14.600	0.780	0.000	0.000	0.000	0.000
0: 9:39	1-10	73.0	73.0	73.9							
0: 9:39	11-20	0.000	3.360	-0.240	2.900	-0.030	0.000	3.240	0.000	0.200	1.750
0: 9:39	21-30	0.240	0.000	0.000	0.400	-14.600	0.780	0.200	0.000	0.001	0.000
0: 9:41	1-10	73.4	71.2	73.9							
0: 9:41	11-20	0.000	3.340	0.000	2.940	0.030	0.240	3.260	0.000	0.100	1.720
0: 9:41	21-30	0.000	0.000	0.000	0.420	-14.600	0.520	0.000	0.100	0.001	0.100
0: 9:42	1-10	73.4	71.7	73.9							
0: 9:42	11-20	0.000	3.360	0.000	2.960	0.000	0.000	3.260	0.000	0.100	1.720
0: 9:42	21-30	0.000	0.000	0.000	0.400	-14.600	0.520	0.100	0.000	0.001	0.000
0: 9:43	1-10	73.9	72.1	74.3							
0: 9:43	11-20	-0.030	3.320	0.240	3.000	-0.030	0.000	3.260	0.000	0.000	1.710
0: 9:43	21-30	0.000	0.000	0.000	0.420	-14.600	0.780	0.200	0.000	0.001	0.100
0: 9:44	1-10	73.4	71.2	74.3							
0: 9:44	11-20	0.000	3.320	0.000	3.020	0.000	0.000	3.220	-0.100	0.100	1.700
0: 9:44	21-30	0.000	0.000	0.000	0.420	-14.600	1.040	0.100	0.000	0.001	0.100
0: 9:45	1-10	73.4	72.6	73.9							
0: 9:45	11-20	0.000	3.300	0.000	3.000	0.000	0.000	3.240	0.100	0.100	1.700
0: 9:45	21-30	0.000	0.000	0.000	0.420	-14.580	0.520	0.100	0.000	0.001	0.000
0: 9:46	1-10	73.9	72.6	74.8							
0: 9:46	11-20	0.030	3.280	0.240	3.040	0.000	0.000	3.220	0.000	0.100	1.690
0: 9:46	21-30	0.000	0.000	0.000	0.440	-14.580	0.520	0.200	0.000	0.001	0.000

ACC.NO. L03-2-C00015

111111111.222222222 333333333 444444444 555555555 666666666 777777777 888888888 999999999 000000000

TIME DDD:HH:MM	CHANNEL 1-10 11-20 21-30 31-40	T/C DEG F LB/HR	T/C PSIA LABEL 022	LABEL 002 LB/HR LABEL 023	PSID PSID	DEG F PSIG	LB/HR LB/HR	PSIG WATTS	VOLTS VOLTS	WATTS LABEL 029	PSIA VOLTS
0: 9:48	1-10	73.9	74.3	73.9							
0: 9:48	11-20	0.030	3.280	-0.240	3.640	0.000	0.240	3.220	-0.100	0.100	1.690
0: 9:48	21-30	0.000	0.000	0.000	0.440	-14.680	1.040	0.200	0.000	0.000	0.100
0: 9:49	1-10	73.4	72.6	74.3							
0: 9:49	11-20	0.000	3.280	0.000	3.080	0.000	0.000	3.240	0.000	0.100	1.700
0: 9:49	21-30	0.440	0.000	0.000	0.420	-14.680	0.520	0.100	0.100	0.002	-0.100
0: 9:50	1-10	73.9	73.9	73.9							
0: 9:50	11-20	0.000	3.300	0.000	3.080	0.000	0.000	3.220	-0.100	0.000	1.680
0: 9:50	21-30	0.100	0.000	0.000	0.420	-14.680	0.780	0.200	0.000	0.000	0.000
0: 9:50	1-10	73.9	71.7	74.3							
0: 9:50	11-20	0.000	3.300	0.000	3.080	0.000	0.240	3.200	0.000	0.200	1.660
0: 9:50	21-30	0.000	0.000	0.000	0.420	-14.680	0.520	0.100	0.000	0.001	0.000
0: 9:52	1-10	73.9	71.2	74.3							
0: 9:52	11-20	0.000	3.280	0.000	3.100	0.000	0.000	3.200	0.000	0.200	1.670
0: 9:52	21-30	0.000	0.000	0.000	0.440	-14.680	0.780	0.100	0.000	0.001	0.000
0: 9:52	1-10	73.9	73.4	74.3							
0: 9:52	11-20	0.000	3.260	0.240	3.100	0.000	0.240	3.180	0.000	0.000	1.670
0: 9:52	21-30	0.000	0.000	0.000	0.420	-14.640	0.760	0.200	-0.100	0.000	0.000
0: 9:54	1-10	73.9	72.6	74.8							
0: 9:54	11-20	0.030	3.240	0.240	3.140	-0.030	0.240	3.140	0.100	0.100	1.650
0: 9:54	21-30	0.240	0.000	0.000	0.420	-14.600	0.520	0.100	0.000	0.000	-0.100
0: 9:55	1-10	73.9	71.7	74.3							
0: 9:55	11-20	0.000	3.260	0.000	3.140	-0.030	0.000	3.160	0.000	0.100	1.660
0: 9:55	21-30	0.000	0.000	0.000	0.440	-14.600	0.520	0.200	0.100	0.000	-0.100
0: 9:56	1-10	73.9	72.1	74.3							
0: 9:56	11-20	0.000	3.260	0.000	3.120	0.000	0.000	3.160	0.000	0.200	1.660
0: 9:56	21-30	0.000	0.000	0.000	0.440	-14.620	0.520	0.200	0.000	0.000	0.100
0: 9:57	1-10	73.9	71.2	74.8							
0: 9:57	11-20	0.000	-0.140	0.000	-0.680	0.000	0.240	-0.600	0.000	0.100	0.720
0: 9:57	21-30	0.000	0.000	0.000	0.440	-14.620	0.520	0.100	0.000	0.000	0.000
0: 9:58	1-10	73.9	72.6	74.3							
0: 9:58	11-20	0.000	1.760	0.000	0.840	0.000	0.000	1.160	0.100	0.000	1.250
0: 9:58	21-30	0.000	0.000	0.000	0.420	-14.600	0.760	0.100	0.100	0.000	0.000
0: 9:59	1-10	73.9	73.0	74.3							
0: 9:59	11-20	0.000	2.540	0.000	1.120	0.000	0.000	2.000	0.000	0.100	1.410
0: 9:59	21-30	0.000	0.000	0.000	0.440	-14.600	0.760	0.200	0.000	0.001	0.000

ACC.NO. CD3-2-00004

DDD:HR:MM	CHANNEL NO.	VALUE	E.U.	CODE
0: 9:32	001	70.8	T/C	•
0: 9:32	023	0.	PSID	-
0: 9:32	028	0.	LABEL 029	-
0: 9:34	028	0.	LABEL 029	•
0: 9:34	028	0.	LABEL 029	-
0: 9:35	028	0.	LABEL 029	•
0: 9:35	028	0.	LABEL 029	-
0: 9:35	028	0.	LABEL 029	•
0: 9:35	028	0.	LABEL 029	-
0: 9:35	028	0.	LABEL 029	•
0: 9:35	028	0.	LABEL 029	-
0: 9:35	028	0.	LABEL 029	•
0: 9:35	028	0.	LABEL 029	-
0: 9:36	028	0.	LABEL 029	•
0: 9:36	028	0.	LABEL 029	-
0: 9:36	028	0.	LABEL 029	•
0: 9:36	028	0.	LABEL 029	-
0: 9:36	028	0.	LABEL 029	•
0: 9:36	028	0.	LABEL 029	-
0: 9:38	028	0.	LABEL 029	•
0: 9:38	028	0.	LABEL 029	-
0: 9:44	028	0.	LABEL 029	-
0: 9:44	028	0.	LABEL 029	•
0: 9:44	028	0.	LABEL 029	-
0: 9:44	028	0.	LABEL 029	•
0: 9:44	028	0.	LABEL 029	-
0: 9:44	028	0.	LABEL 029	•
0: 9:44	028	0.	LABEL 029	-
0: 9:45	028	0.	LABEL 029	-
0: 9:45	028	0.	LABEL 029	•
0: 9:45	028	0.	LABEL 029	-
0: 9:45	028	0.	LABEL 029	•
0: 9:45	028	0.	LABEL 029	-
0: 9:48	028	0.	LABEL 029	•
0: 9:48	028	0.	LABEL 029	-
0: 9:48	028	0.	LABEL 029	•
0: 9:49	028	0.	LABEL 029	-
0: 9:49	028	0.	LABEL 029	•
0: 9:49	028	0.	LABEL 029	-
0: 9:49	028	0.	LABEL 029	•
0: 9:49	028	0.	LABEL 029	-
0: 9:49	028	0.	LABEL 029	•
0: 9:50	028	0.	LABEL 029	-
0: 9:50	028	0.	LABEL 029	•
0: 9:53	028	0.	LABEL 029	-
0: 9:54	028	0.	LABEL 029	•

FIXED SAMPLE RATE RECORD 34360000

.0000000	.7430000E+02	.7340000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3540000E+01	.2460000E+00	.2660000E+01	-.3800000E-01	.2460000E+00	.3380000E+01	.1110223-15	.1000000E+00	.1800000E+01
.2400000E+00	.1000000E-04	.1000000E-04	.4400000E+00	-.1460000E+02	.7800000E+00	.1000000E+00	.1110223-15	.0000000	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34440000

.7330000E+02	.7040000E+02	.7340000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
-.3600000E-01	.3460000E+01	.2460000E+00	.2740000E+01	.0000000	.2460000E+00	.3360000E+01	.1110223-15	.1600000E+00	.1780000E+01
.2400000E+00	.1000000E-04	.1000000E-04	.4200000E+00	-.1460000E+02	.5200000E+00	.1000000E+00	.1110223-15	.0000000	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34500000

.7330000E+02	.7040000E+02	.7340000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3460000E+01	.2460000E+00	.2760000E+01	-.3600000E-01	.2460000E+00	.3320000E+01	.1000000E+00	.8881784-15	.1760000E+01
.2400000E+00	.1000000E-04	.1000000E-04	.4400000E+00	-.1460000E+02	.7600000E+00	.1000000E+00	.1110223-15	.1000000-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34560000

.0000000	.7040000E+02	.7330000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3440000E+01	.2460000E+00	.2800000E+01	.0000000	.0000000	.3280000E+01	.1110223-15	.1000000E+00	.1770000E+01
.0000000	.1000000E-04	.1000000E-04	.4200000E+00	-.1460000E+02	.5200000E+00	.1000000E+00	.1000000E+00	.1000000-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34620000

.6970000E+02	.0000000	.7340000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
-.3580000E-01	.3380000E+01	.2460000E+00	.2840000E+01	.0000000	.0000000	.3240000E+01	.1000000E+00	.1000000E+00	.1750000E+01
.2400000E+00	.1000000E-04	.1000000E-04	.4200000E+00	-.1460000E+02	.7800000E+00	.8881784-15	.1110223-15	.0000000	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34740000

.7330000E+02	.7330000E+02	.7340000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3380000E+01	.2460000E+00	.2900000E+01	-.3600000E-01	.0000000	.3240000E+01	.1110223-15	.2000000E+00	.1750000E+01
.2400000E+00	.0000000	.0000000	.4600000E+00	-.1460000E+02	.7800000E+00	.2000000E+00	.1110223-15	.1000000-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	
.0000000									

34860000

.7340000E+02	.7120000E+02	.7390000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3340000E+01	.2460000E+00	.2940000E+01	.3000000E-01	.2400000E+00	.3260000E+01	.1110223-15	.1000000E+00	.1720000E+01
.0000000	.0000000	.2000000E-04	.4200000E+00	-.1460000E+02	.5200000E+00	.8881784-15	.1000000E+00	.1000000-02	.1000000E+00
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									

34920000

.0001000	.7117000E+02	.7390000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000	.3360000E+01	-.0000000	.2960000E+01	.0000000	.0000000	.3260000E+01	.1110223-15	.1000000E+00	.1720000E+01
.0000000	.1000000E-04	.1000000E-04	.4000000E+00	-.1460000E+02	.5200000E+00	.1000000E+00	.1110223-15	.1000000E-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									
34960000									
.7390000E+02	.7210000E+02	.7430000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
-.3000000E-01	.3320000E+01	.2470000E+00	.3000000E+01	-.7000000E-01	.0000000	.3260000E+01	.1110223-15	.8881784-15	.1710000E+01
.0000000	.1000000E-04	.2000000E-04	.4200000E+00	-.1460000E+02	.7500000E+00	.2000000E+00	.1110223-15	.1000000E-02	.1000000E+00
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									
35040000									
.7340000E+02	.7120000E+02	.7430000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0001000	.3320000E+01	.0000000	.3020000E+01	.0000000	.0000000	.3220000E+01	-.1000000E+00	.1000000E+00	.1700000E+01
.0000000	.1000000E-04	.1000000E-04	.4200000E+00	-.1460000E+02	.1040000E+01	.1000000E+00	.1110223-15	.1000000E-02	.1000000E+00
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									
35100000									
.7340000E+02	.7260000E+02	.7390000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0001000	.3300000E+01	.0000000	.3000000E+01	.0000000	.0000000	.3240000E+01	.1000000E+00	.1000000E+00	.1700000E+01
.0000000	.1000000E-04	.1000000E-04	.4200000E+00	-.1458000E+02	.5260000E+00	.2000000E+00	.1110223-15	.1000000E-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									
35160000									
.7390000E+02	.7260000E+02	.7480000E+02	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.3000000E-01	.3280000E+01	.2400000E+00	.3040000E+01	.0000000	.0000000	.3220000E+01	.1110223-15	.1000000E+00	.1690000E+01
.0000000	.1000000E-04	.1000000E-04	.4400000E+00	-.1480000E+02	.5200000E+00	.2000000E+00	.1110223-15	.1000000E-02	.1110223-15
.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000	.0000000
.0000000									

APPENDIX G

PROGRAM ELEMENTS AND LISTINGS

APPENDIX G

PROGRAM ELEMENTS AND LISTINGS

G.1 INTERNAL SUBROUTINES

The subroutines with the $ suffix are internal EXEC 8 system subroutines and are as follows:

NINTR$
NRDUS
NIO3$
NIO2$
NWDU$
NIO1$
NRNL$
XPII
NSTOP$

G.2 EXTERNAL SUBROUTINES

External subroutines other than EXEC 8 system subroutines are grouped according to function as follows:

CURV61
FD2FD
NTRAN
DHM2MS
SCHCHR
FD2INT
EVALFT
FP2FD
MS2DAY
INT2FD

These external subroutines are described in detail, according to their function, in the following sections.

G.2.1 Curve Fit and Evaluation Subroutines

- Subroutine CURV61 (SPX, SPY, M, N, SPA, A)

  The CURV61 subroutine uses the least-squares method to find the M degree polynomial

$$Y = A_0 + A_1X + A_2X^2 + \cdots + A_MX^M$$

which approximates a curve through N given points (X_i, Y_i) where i = 1, N. The subroutine uses six calling arguments to interface with the CSVIDR program

where:

SPX = floating point independent variable

SPY = floating point dependent variable

M = degree of equation to be fitted

N = number of points in the data array

SPA = single precision coefficients for terms A_0 through A_M

A = double precision coefficients for terms A_0 through A_M

- Subroutine EVALFT (X, M, A, Y_c)

  EVALFT uses the coefficients input from lead cards or from the CURV61 subroutine to generate a set of dependent-variable values (Y_c) corresponding to a set of predetermined values (X) provided by the CSVIDR program

  where:

  X = independent variables from VIDAR

  M = degree of the calibration

  A = coefficients for terms A_0 through A_M

  Y_c = calculated dependent variable

G.2.2 Time Conversion Subroutines

- Subroutine DHM2MS (IDAY, IHR, MIN, MS, $)

  The DHM2MS subroutine converts time to milliseconds

  where:

  IDAY = days in integer format

  IHR = hours in integer format

MIN = minutes in integer format

MS = milliseconds in integer format

$ = statement return if integer milliseconds are less than zero

- Subroutine MS2DAY (ITIM, IDAY, JHRS, JMIN, SEC, $)
  The MS2DAY subroutine converts milliseconds to DDD:HH:MM:SEC

  where:

  ITIM = time in milliseconds

  IDAY = days in integer format

  JHRS = hours in integer format

  JMIN = minutes in integer format

  SEC = seconds in floating point format

  $ = statement return if milliseconds supplied are less than zero

G.2.3 Character String Manipulation Subroutines

- Subroutine SCHCHR (IFIELD, ISTCH, NOCH, ICHAR, NTHCHR, $)

  where:

  IFIELD = field to be searched for a comparison

  ISTCH = starting character position in IFIELD where the search is to begin

  NOCH = number of characters beginning in IFIELD at character position ISTCH to be searched

  ICHAR = character for which IFIELD is to be searched

  NTHCHR = if a comparison is made, the character position in IFIELD where the comparison is stored

  $ = statement return if a comparison was not made

- Subroutine FD2FD (NOCH, LINE1, ISTCH, LINE2, JSTCH)

  where:

  NOCH = number of characters to move

  LINE1 = first word address of the source line

  ISTCH = starting character position in the source line from which to begin moving

  LINE2 = first word address of destination line

  JSTCH = starting character position in the destination line

- Subroutine FD2INT (LINE, ISTCH, NOCH, VALUE, $)

  where:

  LINE = first word address of the source line

  ISTCH = starting character position in the source line

  NOCH = number of characters to be converted to an integer

  VALUE = location where the integer value is to be stored

  $ = return taken when a character other than a +, -, blank, or numeric is detected

- Subroutine FP2FD (FP, NOCH, NORDP, FIELD, ISTCH)

  where:

  FP = location where the floating point value is stored

  NOCH = number of characters to be converted to Fieldata

  NORDP = number of digits to be placed at the right of the decimal point

  FIELD = first word address of the destination line

  ISTCH = starting character position in the destination line

- Subroutine INT2FD (INT, NOCH, FIELD, ISTCH)

  where:

  INT = integer value to be converted to Fieldata

  NOCH = number of characters

  FIELD = field in which the Fieldata integer is to be placed

  ISTCH = starting character position in the field where the Fieldata integer is to be placed

```
*FOR,S FD5-L75036*CALCUR.CSVIDR
FOR SE2C-10/31/75-17:31:36 (3,)

   MAIN PROGRAM

   STORAGE USED: CODE(1) 004174; DATA(0) 001322; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   BUFFER 007555
    0004   TIMES  000014
    0005   IO     000520
    0006   OPCOEF 000770
    0007   COMOUT 001406
    0010   TABLOG 000374


   EXTERNAL REFERENCES (BLOCK, NAME)

    0011   CURV61
    0012   FD2FD
    0013   NTRAN
    0014   DHM2MS
    0015   SCHCHR
    0016   FD2INT
    0017   EVALFT
    0020   FP2FD
    0021   MS2DAY
    0022   INT2FD
    0023   NINTR$
    0024   NRDU$
    0025   NIO3$
    0026   NIO2$
    0027   NWDU$
    0030   NIO1$
    0031   NRNL$
    0032   XPII
    0033   NSTOP$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   002061 1000L     0001   001603 1034G     0001   002101 1050L     0001   002271 1076L     0001   002305 1085L
    0001   000637 110L      0000   000232 11111F    0001   000653 115L      0001   002246 1156G     0001   000664 117L
    0001   000664 119L      0001   002306 1200L     0001   002323 1205L     0001   002351 1215G     0001   002363 1230L
    0001   002423 1233G     0001   002412 1235L     0001   000705 125L      0001   002435 1250L     0001   002475 1251G
    0001   002464 1255L     0001   002547 1267G     0001   002507 1270L     0001   002536 1275L     0001   002561 1290L
    0001   002610 1295L     0001   000707 130L      0001   002521 1305G     0001   002633 1310L     0001   002662 1315L
    0001   002673 1323G     0001   002705 1330L     0001   002734 1335L     0001   002745 1341G     0001   000722 135L
    0001   002757 1350L     0001   003006 1355L     0001   003017 1357G     0001   003031 1370L     0001   003071 1375G
    0001   003060 1375L     0001   003103 1390L     0001   003132 1395L     0001   000724 140L      0001   003155 1410L
    0001   003143 1413G     0001   003203 1425L     0001   003224 1435G     0001   000753 145L      0001   003227 1450L
    0001   003241 1451L     0001   003255 1452L     0001   003265 1454L     0001   003312 1466G     0001   003316 1469L
    0001   003312 1470G     0001   003370 1480L     0001   003412 1483L     0001   003423 1485L     0001   003430 1500L
    0001   003337 1502G     0001   003337 1504G     0001   003500 1510L     0001   003511 1515L     0001   003520 1519L
    0001   003550 1521L     0001   003577 1528L     0001   003615 1530L     0001   003420 1535G     0001   003627 1540L
    0001   003702 1545L     0001   003711 1546L     0001   003660 1630G     0001   003726 1646G     0001   003726 1650G
```

```
0001   004004 1740G    0001   001071 185L     0001   004141 1850L    0000   001124 1851F    0001   004147 1860L
0000   001134 1861F    0001   001111 188L     0001   004155 1900L    0001   001265 200L     0000   000252 2000F
0000   000253 2002F    0000   000255 2004F    0000   000264 2006F    0000   000270 2008F    0001   000053 204G
0001   000060 211G     0001   000061 214G     0001   001310 220L     0001   000074 224G     0001   001374 230L
0001   000102 231G     0001   001402 231L     0001   000103 234G     0001   001427 240L     0001   001437 250L
0001   001551 263L     0000   000275 3000F    0000   000276 3001F    0000   000300 3002F    0000   000311 3003F
0000   000315 3004F    0000   000331 3005F    0000   000345 3006F    0000   000361 3007F    0000   000370 3008F
0000   000377 3009F    0000   000406 3010F    0000   000617 3011F    0000   000643 3012F    0000   000666 3013F
0000   000675 3014F    0000   000676 3015F    0000   000716 3016F    0000   000725 3017F    0000   000727 3018F
0001   000220 304G     0000   000423 3065F    0000   000521 3070F    0001   000225 311G     0001   000234 317G
0001   000323 346G     0001   001557 350L     0001   001564 355L     0001   000264 40L      0001   001615 400L
0001   003755 4000L    0000   000756 4001F    0000   001060 4003F    0000   000774 4020F    0000   001005 4025F
0001   003765 4030L    0000   001026 4035F    0001   004030 4040L    0000   001045 4045F    0001   004050 4049L
0001   004070 4050L    0000   001073 4051F    0001   004102 4052L    0000   001105 4053F    0001   004113 4056L
0001   004125 4057L    0000   001116 4058F    0000   001144 4059F    0000   001170 4060F    0001   000522 443G
0001   000525 446G     0001   000540 457G     0001   000574 470G     0001   000332 50L      0001   000621 500G
0000   000230 5000F    0001   000633 507G     0001   000646 516G     0001   000702 554G     0000   000250 5555F
0001   000736 571G     0001   000402 60L      0001   000767 604G     0001   001641 605L     0001   000416 61L
0001   001674 610L     0001   000774 611G     0001   001044 631G     0001   001731 650L     0001   001220 710G
0001   001252 723G     0001   001273 736G     0001   000546 90L      0005 R 000425 A        0006 D 000000 ADP
0006 D 000754 ADPSET   0000 I 000063 BADCHN   0000 I 000215 BADCHR   0000 I 000062 BADCRD   0000 I 000037 BLANK
0000 I 000100 BLNKED   0003 I 000000 BUF      0000 I 000076 CALIB    0005 I 000002 CARDIN   0005 I 000156 CHAN
0007 I 001267 CHANLC   0000 I 000147 CHNL     0000 I 000135 CHRALL   0000 I 000152 CHR1     0000 I 000155 CHR13
0000 I 000153 CHR2     0000 I 000160 CHR6     0000 I 000154 CHR7     0000 I 000032 COLON    0000 I 000033 COLONZ
0000 I 000146 DATWRD   0000 I 000163 DAY      0004 I 000000 DDD      0005 I 000433 DEG      0000 I 000122 DO105J
0000 D 000001 DPDATA   0000 I 000115 EOW      0000 I 000142 EOWCHR   0000 I 000035 EOW1     0000 I 000036 EOW2
0005 I 000000 EUNIT    0007 I 001340 FDOUT    0000 I 000172 GRP      0004 I 000003 HH       0000 I 000041 HILIM
0000 I 000141 HOUR     0000 I 000061 HZERO    0000 I 000050 H1       0000 I 000051 H2       0000 I 000052 H3
0000 I 000053 H4       0000 I 000054 H5       0000 I 000055 H6       0000 I 000056 H7       0000 I 000057 H8
0000 I 000060 H9       0000 I 000121 I        0000 I 000134 IBAD     0000 I 000046 IBIG     0000 I 000103 ICOL
0000 I 000116 IDCHAN   0000 I 000174 IDMP     0000 I 000213 III      0000 I 000211 III      0000 I 000212 III
0000 I 000064 INCHAN   0000 I 000101 INSTAT    0000 I 000131 IREC     0000 I 000104 IROW     0000 I 000132 ISCAN
0000 I 000047 ISMALL   0000 I 000136 ISTCH     0000 I 000105 J        0000 I 000150 JC       0000 I 000151 JCC
0000 I 000171 JCHNL    0000 I 000205 JCK      0000 I 000173 JCOL     0000 I 000201 JGRP     0000 I 000127 JJ
0000 I 000102 JJL      0000 I 000125 JP       0000 I 000204 JPL      0000 I 000124 JPRT     0000 I 000107 JT
0000 I 000210 JX       0000 I 000202 J12      0000 I 000177 J21      0000 I 000200 J3       0000 I 000106 K
0007 I 000000 LABEL    0005 I 000034 LABL     0007 I 000027 LIMBUF   0000 I 000013 LIMLIN   0000 I 000065 LIMROW
0000 I 000070 LIN      0000 I 000071 LINHDG   0000 I 000176 LINSAV   0007 I 000002 LOGLIN   0000 I 000042 LOLIM
0000 I 000073 LREC     0000 I 000207 LROW     0000 I 000067 LSCAN    0000 I 000166 LSTAT    0000 I 000137 LSTPTR
0000 I 000027 LUNIT    0000 I 000206 LWSTAT   0000 I 000117 M        0000 I 000040 MIDLIM   0000 I 000000 MIN
0004 I 000006 MM       0000 I 000113 MOVE1    0000 I 000114 MOVE2    0000 I 000123 MP1      0000 I 000120 N
0000 I 000034 NEG      0000 I 000162 NEGEXP   0000 I 000043 NEGZRO   0000 L 000004 NOCARD   0000 L 000003 NOTAB
0000 I 000130 NP       0000 I 000175 NXTPTR   0000 I 000170 PBUF     0003 I 006565 PLTBUF   0005 R 000504 POLYCO
0000 I 000133 PSCAN    0000 I 000157 RDATA    0000 I 000145 SAVCHR   0000 I 000143 SAVHR    0007 I 000057 STITLE
0004   000012 STPTME   0004   000011 STRTME   0010 I 000000 TABBUF   0000 I 000112 TAPEF    0000 I 000126 TAPEIN
0000 I 000077 TAPEIO   0000 L 000005 TAPEO    0000 I 000110 TAPE1    0000 I 000111 TAPE2    0005 I 000512 TFILL
0000 I 000075 TIME     0005 I 000515 TIMEMS   0007 I 001342 TIMOUT   0005 I 000020 TITLE    0004   000013 TPITME
0000 I 000072 TREC     0000 I 000064 TSCAN    0000 I 000074 TSF      0000 I 000167 TSLOT    0000 L 000006 TYPE1
0000 L 000007 TYPE2    0000 L 000010 TYPE3    0000 L 000011 TYPE4    0000 L 000012 TYPE5    0000 I 000140 T1CHR
0000 I 000144 T2CHR    0000 I 000214 WBAD     0000 I 000165 WRD      0000 I 000203 WSTAT    0005 D 000301 XPT
0000   000220 XYCALS   0005 R 000353 XYPAIR   0005 D 000227 YPT      0000 R 000044 ZBIG     0000 R 000156 ZDATA
0000 R 000161 ZMULT    0003 R 006565 ZPLTBU   0000 R 000164 ZSEC     0000 R 000045 ZSMALL
```



```
00101      1*          PARAMETER CHRWRD=14,MINREC=168,MXBUF=3445,MXCHAN=41,MXDEG=5                    000000
```

```
00101     2*          *,MXSCAN=36,NDEGP1=6,NPTXY=21,PLIMIT=10,PRTROW=56,PRTCOL=20                    000000
00101     3*          *,LUCARD=5,LUOUT=6,MXLIM=12,TABCOL=10                                          000000
00101     4*          *,MXBAD=20,NDEG=5,MXLOG=21,BUFSIZ=504                                          000000
00103     5*           DOUBLE PRECISION ADP,DPDATA,ADPSET,XPT,YPT                                    000001
00104     6*           REAL XYPAIR,POLYCO,A                                                          000001
00105     7*           LOGICAL NOTAB,NOCARD,TAPEO                                                    000001
00106     8*           LOGICAL TYPE1,TYPE2,TYPE3,TYPE4,TYPE5                                         000001
00107     9*           IMPLICIT INTEGER(A-Y)                                                         000001
00110    10*           COMMON                                                                        000001
00110    11*          */BUFFER/BUF(MXBUF),PLTBUF(BUFSIZ)                                            000001
00110    12*          */TIMES/DDD(3),HH(3),MM(3),STRTME,STPTME,TP1TME                               000001
00110    13*          */IO/EUNIT(2),CARDIN(14),TITLE(12),LABL(2,MXCHAN)                             000001
00110    14*          *,CHAN(MXCHAN)                                                                000001
00110    15*          *,YPT(NPTXY),XPT(NPTXY),XYPAIR(2,NPTXY),A(NDEGP1),DEG(MXCHAN)                 000001
00110    16*          *,POLYCO(NDEGP1),TFILL(3),TIMEMS(3)                                           000001
00110    17*          */DPCOEF/ADP(NDEGP1,MXCHAN),ADPSET(NDEGP1)                                    000001
00110    18*          */COMOUT/LABEL(2),LOGLIN(MXLOG),LIMBUF(MXLIM,PRTROW),CHANLC(MXCHAN)           000001
00110    19*          *,FDOUT(2),TIMOUT(MXSCAN)                                                     000001
00110    20*          */TABLOG/TABBUF(MXLOG,12)                                                     000001
00110    21*     C                                                                                  000001
00111    22*           DIMENSION STITLE(10),ZPLTBU(1),LIMLIN(MXLIM),LUNIT(3)                         000001
00111    23*     C                                                                                  000001
00112    24*           EQUIVALENCE (ZPLTBU,PLTBUF)                                                   000001
00113    25*           EQUIVALENCE (STITLE(1),LIMBUF(1,3))                                           000001
00114    26*           DATA                                                                          000001
00114    27*          *COLON/'     :'/                                                              000001
00114    28*          *COLONZ/'@@@@@:'/                                                             000001
00114    29*          *NEG/'@@@@@-'/                                                                000001
00114    30*          *EOW1/'@@@@@9'/                                                               000001
00114    31*          *EOW2/'@@@@@Z'/                                                               000001
00114    32*          *BLANK/'      '/                                                              000001
00114    33*          *MIDLIM/'@@@@@.'/                                                             000001
00114    34*          *HILIM/'@@@@@*'/                                                              000001
00114    35*          *LOLIM/'@@@@@-'/                                                              000001
00114    36*          *NEGZRO/-0/                                                                   000001
00114    37*          *ZBIG/1.0E+37/                                                                000001
00114    38*          *ZSMALL/1.0E-37/                                                              000001
00114    39*          *IBIG/0377777777777/                                                          000001
00114    40*          *ISMALL/0400000000000/                                                        000001
00133    41*           DATA                                                                         000001
00133    42*          *H1/'1'/                                                                      000001
00133    43*          *H2/'2'/                                                                      000001
00133    44*          *H3/'3'/                                                                      000001
00133    45*          *H4/'4'/                                                                      000001
00133    46*          *H5/'5'/                                                                      000001
00133    47*          *H6/'6'/                                                                      000001
00133    48*          *H7/'7'/                                                                      000001
00133    49*          *H8/'8'/                                                                      000001
00133    50*          *H9/'9'/                                                                      000001
00133    51*          *HZERO/'0'/                                                                   000001
00133    52*          *STITLE/                                                                      000001
00133    53*          *' DDD:HH:MM   CHANNEL NO.       VALUE         E.U.        CODE'/             000001
00133    54*     C         NAMELIST INPUT                                                          000001
00133    55*     C                                                                                  000001
00147    56*           NAMELIST/XYCALS/XYPAIR,POLYCO                                                000001
00150    57*           BADCRD=0                                                                     000001
00151    58*           BADCHN=0                                                                     000002
```

```
00152     59*          NOTAB=.FALSE.                                    000003
00153     60*          NOCARD=.FALSE.                                   000004
00154     61*          TAPEO=.TRUE.                                     000005
00155     62*          TYPE1=.FALSE.                                    000007
00156     63*          TYPE2=.FALSE.                                    000010
00157     64*          TYPE3=.FALSE.                                    000011
00160     65*          TYPE4=.FALSE.                                    000012
00161     66*          TYPE5=.FALSE.                                    000013
00162     67*          TFILL(1)=ISMALL                                  000014
00163     68*          TFILL(2)=IBIG                                    000016
00164     69*          TFILL(3)=ISMALL                                  000020
00165     70*          INCHAN=0                                         000022
00166     71*          LIMROW=3                                         000023
00167     72*          TSCAN=0                                          000025
00170     73*          LSCAN=0                                          000026
00171     74*          LIN=0                                            000027
00172     75*          LINHDG=56                                        000030
00173     76*          TREC=0                                           000032
00174     77*          LREC=0                                           000033
00175     78*          TSF=9                                            000034
00176     79*          TIME='TIME  '                                    000036
00177     80*          CALIB='      '                                   000040
00200     81*          TAPEID='TAPEID'                                  000042
00201     82*          BLNKED='@@@@@ '                                  000044
00202     83*          INSTAT=MXBUF                                     000046
00203     84*          DO 24 JJL=1,MXLIM                                000053
00206     85*       24 LIMLIN(JJL)=BLANK                                000053
00210     86*          DO 26 ICOL=1,MXLOG                               000061
00213     87*          DO 25 IROW=1,12                                  000061
00216     88*       25 TABBUF(ICOL,IROW)=BLANK                          000061
00220     89*       26 CONTINUE                                         000067
00222     90*          LIMLIN=3                                         000067
00223     91*          DO 27 JJL=1,MXLOG                                000074
00226     92*       27 LOGLIN(JJL)=BLANK                                000074
00230     93*          DO 29 J=1,MXLIM                                  000103
00233     94*          DO 28 K=4,PRTROW                                 000103
00236     95*       28 LIMBUF(J,K)=BLANK                                000103
00240     96*       29 LIMBUF(J,2)=BLANK                                000105
00242     97*          LIMBUF(11,3)=BLANK                               000113
00243     98*          LIMBUF(12,3)=BLANK                               000115
00244     99*          TABBUF(2,3)='TIME  '                             000117
00245    100*          TABBUF(3,3)='CHANNE'                             000121
00246    101*          TABBUF(4,3)='L     '                             000123
00247    102*          TABBUF(1,4)=' DDD:H'                             000125
00250    103*          TABBUF(2,4)='H:MM  '                             000127
00251    104*          TABBUF(3,4)=' 1-10 '                             000131
00252    105*          TABBUF(3,5)='11-20 '                             000133
00253    106*          TABBUF(3,6)='21-30 '                             000135
00254    107*          TABBUF(3,7)='31-40 '                             000137
00255    108*          TABBUF(4,4)='      '                             000141
00256    109*          TABBUF(4,5)='      '                             000143
00257    110*          TABBUF(4,6)='      '                             000145
00260    111*          TABBUF(4,7)='      '                             000147
00261    112*          TABBUF(4,2)='    11'                             000151
00262    113*          TABBUF(5,2)='111111'                             000153
00263    114*          TABBUF(6,2)='1 2222'                             000155
00264    115*          TABBUF(7,2)='22222 '                             000157
```

```
00265   116*          TABBUF(8,2)='333333'                                                    000161
00266   117*          TABBUF(9,2)='333 44'                                                    000163
00267   118*          TABBUF(10,2)='444444'                                                   000165
00270   119*          TABBUF(11,2)='4 5555'                                                   000167
00271   120*          TABBUF(12,2)='55555 '                                                   000171
00272   121*          TABBUF(13,2)='666666'                                                   000173
00273   122*          TABBUF(14,2)='666 77'                                                   000175
00274   123*          TABBUF(15,2)='777777'                                                   000177
00275   124*          TABBUF(16,2)='7 8888'                                                   000201
00276   125*          TABBUF(17,2)='88888 '                                                   000203
00277   126*          TABBUF(18,2)='999999'                                                   000205
00300   127*          TABBUF(19,2)='999 00'                                                   000207
00301   128*          TABBUF(20,2)='000000'                                                   000211
00302   129*          TABBUF(21,2)='0     '                                                   000213
00303   130*          DO 30 J=1,MXCHAN                                                        000220
00306   131*       30 CHAN(J)=BLANK                                                           000220
00310   132*          DO 33 J=1,MXCHAN                                                        000225
00313   133*          LABL(1,J)=6HOUTPUT                                                      000225
00314   134*       33 LABL(2,J)=6H LABEL                                                      000226
00316   135*          DO   34 JT=1,BUFSIZ                                                     000234
00321   136*       34 PLTBUF(JT)=NEGZRO                                                       000234
00321   137*     C                                                                           000234
00321   138*     C    READ IN LEAD CARDS                                                     000234
00321   139*     C                                                                           000234
00323   140*          READ(LUCARD,2000,ERR=1900,END=115 )TITLE                                000236
00326   141*          TAPE0=.TRUE.                                                            000250
00327   142*          TYPE1=.TRUE.                                                            000252
00330   143*          WRITE(LUOUT,3000)TITLE                                                  000253
00333   144*       40 READ(LUCARD,2002,ERR=1900,END=117 )CARDIN                               000264
00336   145*          WRITE(LUOUT,3001)CARDIN                                                 000275
00341   146*          IF(CARDIN(1).NE.TIME)GO TO 50                                           000305
00343   147*          TYPE2=.TRUE.                                                            000310
00344   148*          READ(30,2004,ERR=1900,END=117)(DDD(J),HH(J),MM(J),J=1,3)                000312
00354   149*          GO TO 40                                                                000330
00355   150*       50 IF(CARDIN(1).NE.TAPEID)GO TO 60                                         000332
00357   151*          TYPE3=.TRUE.                                                            000334
00360   152*          READ(30,2006,ERR=1900,END=117)TAPE1,TAPE2,TAPEF,MOVE1,MOVE2,EOW         000336
00370   153*          IF(TAPE1.EQ.0)TAPE1=15                                                  000353
00372   154*          IF(TAPEF.EQ.0)TAPEF=7                                                   000357
00374   155*          IF(TAPEF.LT.0)TAPE0=.FALSE.                                             000363
00376   156*          IF(.NOT.TAPE0)TAPEF=7                                                   000367
00400   157*          IF(EOW.EQ.BLANK)EOW='@@@@@M'                                            000373
00402   158*          GO TO 40                                                                000400
00403   159*       60 IF(CARDIN(1).EQ.CALIB)GO TO 61                                          000402
00405   160*          BADCRD=BADCRD+1                                                         000404
00406   161*          WRITE(LUOUT,4020)                                                       000407
00410   162*          GO TO 40                                                                000414
00411   163*       61 READ(30,2008,ERR=1900,END=117)IDCHAN,EUNIT,M,N                          000416
00417   164*          TYPE4=.TRUE.                                                            000433
00420   165*          IF(M.EQ.0)M=1                                                           000435
00422   166*          IF(N.EQ.0)N=2                                                           000441
00424   167*          INCHAN=INCHAN+1                                                         000445
00425   168*          IF(INCHAN.GT.MXCHAN)GO TO 4056                                          000450
00427   169*          CHANLC(INCHAN)=IDCHAN                                                   000453
00430   170*          DEG(INCHAN)=M                                                           000456
00431   171*          IF(IDCHAN.LT.1.OR.IDCHAN.GT.MXCHAN)GO TO 4050                           000460
00433   172*          CHAN(INCHAN)=INCHAN                                                     000500
```

```
00434   173*          LABL(1,INCHAN)=EUNIT(1)                                  000502
00435   174*          LABL(2,INCHAN)=EUNIT(2)                                  000504
00436   175*          READ( 5,XYCALS,END=119,ERR=4052)                         000506
00441   176*          TYPE5=.TRUE.                                             000514
00442   177*          DO 80 J=1,NPTXY                                          000525
00445   178*          DO 79 I=1,2                                              000525
00450   179*          IF(ABS(XYPAIR(I,J)).GT.0)GO TO 90                        000525
00452   180*       79 CONTINUE                                                 000540
00454   181*       80 CONTINUE                                                 000540
00456   182*          DO 82 J=1,NDEGP1                                         000540
00461   183*          IF(ABS(POLYCO(J)).GT.0)GO TO 110                         000540
00463   184*       82 CONTINUE                                                 000546
00463   185*    C        CHECK FOR DEGREE AND NUMBER OF POINTS                 000546
00465   186*       90 IF(M.GT.NDEG.OR.N.GT.NPTXY)GO TO 4000                    000546
00467   187*          DO 95 J=1,N                                              000563
00472   188*          XPT(J)=XYPAIR(1,J)                                       000574
00473   189*       95 YPT(J)=XYPAIR(2,J)                                       000575
00475   190*          CALL CURV61(XPT,YPT,M,N,A,ADPSET)                        000600
00476   191*          DO 105 J=NPTXY                                           000610
00477   192*          DO 100 I=1,2                                             000621
00502   193*      100 XYPAIR(I,J)=0.                                           000621
00504   194*      105 CONTINUE                                                 000622
00505   195*          MP1=M+1                                                  000622
00506   196*          DO 107 J=1,MP1                                           000625
00511   197*      107 ADP(J,INCHAN)=ADPSET(J)                                  000633
00513   198*          GO TO 40                                                 000635
00513   199*    C                                                              000635
00514   200*      110 CONTINUE                                                 000637
00514   201*    C                                                              000637
00514   202*    C     PUT CALS IN PROPER ARRAY                                 000637
00515   203*          DO 111 J=1,NDEGP1                                        000637
00520   204*          ADP(J,INCHAN)=POLYCO(J)                                  000646
00521   205*      111 POLYCO(J)=0.                                             000647
00523   206*          GO TO 40                                                 000651
00523   207*    C                                                              000651
00524   208*      115 CONTINUE                                                 000653
00525   209*          IF(NOTAB)GO TO 1900                                      000653
00527   210*          NOCARD=.TRUE.                                            000654
00530   211*          WRITE(LUOUT,4003)                                        000656
00532   212*      117 CONTINUE                                                 000664
00533   213*      119 CONTINUE                                                 000664
00534   214*          IF(TYPE1)GO TO 125                                       000664
00536   215*          TITLE(1)=' VIDAR'                                        000665
00537   216*          TITLE(2)=' DATA '                                        000667
00540   217*          TITLE(3)='ACQUIS'                                        000671
00541   218*          TITLE(4)='ITION '                                        000673
00542   219*          TITLE(5)='SYSTEM'                                        000675
00543   220*          DO 120 J=6,12                                            000702
00546   221*      120 TITLE(J)=BLANK                                           000702
00550   222*      125 IF(TYPE2)GO TO 130                                       000705
00552   223*      130 IF(TYPE3)GO TO 135                                       000707
00554   224*          TAPE1=15                                                 000710
00555   225*          TAPE2=0                                                  000712
00556   226*          TAPEF=7                                                  000713
00557   227*          MOVE1=0                                                  000715
00560   228*          MOVE2=0                                                  000716
00561   229*          EOW='@@@@@@'                                             000717
```

```
00562   230*       135 IF(TYPE4)GO TO 140                                                  000722
00564   231*       140 IF(TYPE5)GO TO 145                                                  000724
00566   232*           INCHAN=MXCHAN                                                       000725
00567   233*           M=1                                                                 000727
00570   234*           DO 144 J=1,INCHAN                                                   000736
00573   235*           DEG(J)=M                                                            000736
00574   236*           CHANLC(J)=J                                                         000740
00575   237*           CHAN(J)=J                                                           000742
00576   238*           ADP(1,J)=0.D0                                                       000744
00577   239*       144 ADP(2,J)=1.0D0                                                      000746
00601   240*       145 CONTINUE                                                            000753
00602   241*           CALL FD2FD(1,H1,1,TITLE(1),1)                                       000753
00603   242*           DO 147 JPRT=1,12                                                    000767
00606   243*       147 TABBUF(JPRT,1)=TITLE(JPRT)                                          000767
00610   244*           DO 170 JP=1,MXLIM                                                   000774
00613   245*           LIMBUF(JP,1)=TITLE(JP)                                              000774
00614   246*       170 TABBUF(JP,1)=TITLE(JP)                                              000775
00614   247*     C     POSITION INPUT TAPES                                              000775
00614   248*     C                                                                       000775
00616   249*           IF(TAPE1.GT.0)CALL NTRAN(TAPE1,22)                                  001000
00620   250*           IF(TAPE1.GT.0)CALL NTRAN(TAPE1,10)                                  001007
00622   251*           CALL NTRAN(TAPE1,22)                                                001016
00623   252*           IF(MOVE1.GT.0)CALL NTRAN(TAPE1,8,MOVE1)                             001022
00625   253*           CALL NTRAN(TAPE1,22)                                                001032
00626   254*           TAPEIN=TAPE1                                                        001036
00626   255*     C        CONVERT START/STOP AND FIRST TIME CHECKS TO MILLISECONDS         001036
00627   256*           J=0                                                                 001040
00630   257*           DO 188 JJ=1,3                                                       001044
00633   258*           J=J+1                                                               001045
00634   259*           IF(DDD(J).NE.0.OR. HH(J).NE.0.OR. MM(J).NE.0)GO TO 185              001047
00636   260*           TIMEMS(J)=TFILL(J)                                                  001065
00637   261*           GO TO 188                                                           001067
00640   262*       185 CALL DHM2MS(DDD(J),HH(J),MM(J),TIMEMS(J),.1850)                     001071
00641   263*       188 CONTINUE                                                            001112
00643   264*           EOW1=EOW                                                            001112
00644   265*           WRITE(LUOUT,3002)                                                   001114
00644   266*     C                                                                       001114
00644   267*     C                                                                       001114
00644   268*     C    PRINT INITIAL CONDITIONS FOR EXECUTION                             001114
00646   269*           WRITE(LUOUT,3003)TITLE                                              001121
00651   270*           WRITE(LUOUT,3004)DDD(1),HH(1),MM(1),TIMEMS(1)                       001131
00657   271*           WRITE(LUOUT,3005)DDD(2),HH(2),MM(2),TIMEMS(2)                       001142
00665   272*           WRITE(LUOUT,3006)DDD(3),HH(3),MM(3),TIMEMS(3)                       001153
00673   273*           WRITE(LUOUT,3007)TAPE1                                              001164
00676   274*           WRITE(LUOUT,3008)TAPE2                                              001172
00701   275*           WRITE(LUOUT,3009)TAPEF                                              001200
00704   276*           WRITE(LUOUT,3016)EOW1                                               001206
00704   277*     C     DO 193 JJJ=1,INCHAN                                               001206
00707   278*           DO 193 J=1,INCHAN                                                   001220
00707   279*     C     J=CHANLC(JJJ)                                                     001220
00712   280*           NP=DEG(J)+1                                                         001224
00713   281*           WRITE(LUOUT,3010)CHANLC(J),LABL(1,J),LABL(2,J),DEG(J)               001227
00721   282*       193 WRITE(LUOUT,3011)(ADP(K,J),K=1,NP)                                  001242
00721   283*     C                                                                       001242
00721   284*     C     READ INPUT TAPE                                                   001242
00730   285*           IF(BADCRD.GT.0)GO TO 1900                                           001260
00732   286*           IREC=0                                                              001263
```

```
00732    287*     C                                                                  001263
00733    288*       200 ISCAN=0                                                      001265
00734    289*           PSCAN=0                                                      001265
00735    290*           DO 205 JJ=1,INSTAT                                           001273
00740    291*       205 BUF(JJ)=NEGZRO                                               001273
00742    292*           CALL NTRAN(TAPEIN,2,MXBUF,BUF,INSTAT)                        001275
00743    293*           IREC=IREC+1                                                  001304
00744    294*       220 CALL NTRAN(TAPEIN,22)                                        001310
00745    295*           IF(INSTAT.EQ.-1)GO TO 220                                    001313
00747    296*           IF(INSTAT.LT.   MXBUF.AND.INSTAT.GE.MINREC) GO TO 230        001316
00751    297*           IBAD=IBAD+1                                                  001334
00752    298*           WRITE(LUOUT,4020)                                            001337
00754    299*           WRITE(LUOUT,4025)IREC,INSTAT                                 001344
00760    300*           IF(IBAD.GT.MXBAD)GO TO 1500                                  001353
00762    301*           IF(INSTAT.EQ.-2)IREC=IREC-1                                  001357
00764    302*           IF(INSTAT.EQ.-2)GO TO 1500                                   001365
00766    303*           IF(INSTAT.EQ.-3)GO TO 1500                                   001370
00770    304*       230 CHRALL=INSTAT*6                                              001374
00771    305*           ISTCH=1                                                      001376
00772    306*           LSTPTR=0                                                     001400
00773    307*       231 T1CHR=0                                                      001402
00774    308*           CALL SCHCHR(BUF,ISTCH,3,COLONZ,T1CHR,$4030)                  001402
00775    309*           HOUR=0                                                       001412
00776    310*           MIN=0                                                        001413
00777    311*           EOWCHR=0                                                     001414
01000    312*           CALL SCHCHR(BUF(1),ISTCH,18,EOW1, EOWCHR,$240)               001415
01001    313*           GO TO 250                                                    001425
01002    314*       240 CALL SCHCHR(BUF(1),ISTCH,18,EOW2,EOWCHR,$4040)               001427
01003    315*       250 CALL FD2INT(BUF(1),T1CHR-2,2,HOUR,$1850)                     001437
01004    316*           CALL FD2INT(BUF(1),T1CHR+1,2,MIN ,$1850)                     001450
01005    317*           ISCAN=ISCAN+1                                                001462
01006    318*           IF(HOUR.LT.SAVHR)DDD(3)=DDD(3)+1                             001465
01010    319*           SAVHR=HOUR                                                   001474
01011    320*           CALL DHM2MS(DDD(3),HOUR,MIN,TIMOUT(ISCAN),$1850)             001476
01011    321*     C                                                                  001476
01012    322*           LSTPTR=EOWCHR                                                001510
01013    323*           ISTCH=EOWCHR+1                                               001512
01014    324*           IF(TIMOUT(ISCAN).GT.TIMEMS(2))GO TO 1500                     001514
01014    325*     C                                                                  001514
01014    326*     C     GET NEXT TIME CHARACTER TO GET DATA WORDS PER SCAN           001514
01016    327*           IF(ISCAN.EQ.MXSCAN)GO TO 263                                 001520
01020    328*           CALL SCHCHR(BUF(1),ISTCH,CHRALL-ISTCH,COLONZ,T2CHR,$4030)    001523
01021    329*           SAVCHR=T2CHR                                                 001535
01021    330*     C                                                                  001535
01021    331*     C    GET DATA WORDS                                                001535
01021    332*     C                                                                  001535
01022    333*           DATWRD=(T2CHR-T1CHR)/CHRWRD-1                                001537
01023    334*           ISTCH=T2CHR-2                                                001544
01024    335*           GO TO 350                                                    001547
01025    336*       263 DATWRD=(CHRALL-LSTPTR)/CHRWRD-1                              001551
01025    337*     C      GET CHANNEL FROM CHR POS 3-5                                001551
01026    338*       350 IF(TIMOUT(ISCAN).LT.TIMEMS(1))GO TO 1485                     001557
01030    339*       355 CHNL=0                                                       001564
01031    340*           CALL FD2INT(BUF(1),LSTPTR+3,3,CHNL,$4049)                    001564
01031    341*     CX    WRITE(LUOUT,5000)CHNL,DATDUM,JCC,FDOUT(1),FDOUT(2)           001564
01032    342*      5000 FORMAT(1X,10013)                                             001603
01033    343*           DO 360 JC=1,INCHAN                                           001603
```

```
01036    394*            JCC=JC                                                  001603
01036    395*      CX     CHDUMP                                                 001603
01036    396*      CX     IF(CHDUMP.GT.100)GO TO 5001                            001603
01036    397*      CX     WRITE(LUOUT,5000)JC,CHANLC(JC),CHNL,ISCAN,IREC         001603
01036    398*      CX     CHDUMP=CHDUMP+1                                        001603
01037    399*       5001 CONTINUE                                                001605
01040    350*            IF(CHNL.EQ.CHANLC(JC)-1)GO TO 400                      001605
01040    351*      C                                                            001605
01040    352*      C     CHNL 000 ON INPUT TAPE USES 001 ON LEAD CARD INPUT & OUT LABEL  001605
01040    353*      C                                                            001605
01042    354*        360 CONTINUE                                               001613
01049    355*            GO TO 1485                                             001613
01049    356*      C       GO TO NEW SCAN                                       001613
01045    357*        400 CONTINUE                                               001615
01046    358*            CHR1=0                                                 001615
01047    359*            CALL FD2INT(BUF(1),LSTPTR+1,1,CHR1,$4057)              001615
01050    360*            IF(CHR1.EQ.0)GO TO 1450                                001627
01052    361*            IF(CHR1.GT.6)GO TO 1450                                001631
01054    362*            IF(CHR1.LT.0)GO TO 1450                                001635
01056    363*        605 CHR2=0                                                 001641
01057    364*            CALL FD2FD(1,BUF(1),LSTPTR+2,CHR2,6)                   001641
01057    365*      CX    IF(CHR2.NE.BLNKED)GO TO 650                            001641
01060    366*            IF(CHR2.EQ.MIDLIM)GO TO 650                            001653
01062    367*            IF(CHR2.EQ.LOLIM)GO TO 650                             001656
01064    368*            IF(CHR2.EQ.HILIM)GO TO 650                             001661
01066    369*            IF(CHR2.EQ.BLNKED)GO TO 610                            001664
01070    370*            LSTPTR=LSTPTR+1                                        001667
01071    371*            GO TO 605                                              001672
01071    372*      C                                                            001672
01071    373*      C   LIMIT DATA PATCH TO SHIFT EXTRA CHARACTERS AND GET LOG DATA  001672
01071    374*      C                                                            001672
01072    375*        610 CONTINUE                                               001674
01072    376*      C                                                            001674
01072    377*      C     SHOULD HAVE LOG MODE OR OVERLOAD DATA                  001674
01072    378*      C                                                            001674
01073    379*            CHR7=0                                                 001674
01074    380*            CALL FD2FD(6,BUF(1),LSTPTR+7,CHR7,6)                   001674
01075    381*            CHR13=BLANK                                            001706
01076    382*            CALL FD2FD(1,BUF(1),LSTPTR+13,CHR13,6)                 001710
01077    383*            IF(CHR7.NE.BLANK)GO TO 650                             001722
01101    384*            ZDATA=ZBIG                                             001725
01102    385*            GO TO 1000                                             001727
01103    386*        650 CONTINUE                                               001731
01103    387*      C     FLOAT DATA                                             001731
01103    388*      C                                                            001731
01104    389*            CALL FD2INT(BUF(1),LSTPTR+7,5,RDATA,$4057)             001731
01105    390*            CHR6=0                                                 001742
01106    391*            CALL FD2FD(1,BUF(1),LSTPTR+6,CHR6,6)                   001743
01107    392*            IF(CHR6.EQ.NEG)ZMULT=-1.                               001755
01111    393*            IF(CHR6.EQ.BLNKED)ZMULT=1.                             001762
01113    394*            CALL FD2INT(BUF(1),LSTPTR+12,1,NEGEXP,$4057)           001774
01114    395*            ZDATA=(ZMULT*RDATA)/10**NEGEXP                         002006
01115    396*            FDOUT(1)=6H                                            002022
01116    397*            FDOUT(2)=6H                                            002024
01117    398*            IF(CHR1.GT.4)GO TO 1000                                002026
01121    399*            DPDATA=ZDATA                                           002032
01121    400*      CX    IF(CHNL.NE.24.AND. CHNL.NE.25)GO TO 11113              002032
```

```
01121    401*     Cx    IF(NEGCK.GT.10)GO TO 11113                                                    002032
01121    402*     Cx    NEGCK=NEGCK+1                                                                 002032
01121    403*     Cx    IF(CHNL.EQ.24)WRITE(LUOUT,11111)CHNL,ZMULT,DPDATA,FDOUT,ZDATA                 002032
01121    404*     Cx    IF(CHNL.EQ.25)WRITE(LUOUT,11111)CHNL,ZMULT,DPDATA,FDOUT,ZDATA                 002032
01122    405*     11111 FORMAT('0CHNL=',I6,' ZMULT=',E15.8,'DPDATA=',E15.8,                           002034
01122    406*          *' FDOUT=',2A6,'ZDATA=',E15.8)                                                002034
01122    407*     Cx    WRITE(LUOUT,5555)CHR1,CHR2,CHR6,CHR7,CHR13,CHNL                               002034
01123    408*     11113 CONTINUE                                                                      002034
01124    409*           CALL EVALFT(DPDATA,DEG(JCC),ADP(1,JCC),ZDATA)                                 002034
01125    410*           CALL FP2FD(ZDATA,12,3,FDOUT(1),1)                                             002050
01125    411*     Cx    IF(CHNL.NE.24.AND.CHNL.NE.25)GO TO 11116                                      002050
01125    412*     Cx    IF(NEGCK.GT.10)GO TO 11116                                                    002050
01125    413*     Cx    IF(CHNL.EQ.24)WRITE(LUOUT,11111)CHNL,ZMULT,DPDATA,FDOUT,ZDATA                 002050
01125    414*     Cx    IF(CHNL.EQ.25)WRITE(LUOUT,11111)CHNL,ZMULT,DPDATA,FDOUT,ZDATA                 002050
01125    415*     Cx    WRITE(LUOUT,5555)CHR1,CHR2,CHR6,CHR7,CHR13,CHNL                               002050
01126    416*     11116 CONTINUE                                                                      002057
01127    417*           GO TO 1050                                                                    002057
01130    418*      1000 IF(LABL(1,JCC).EQ.BLANK)LABL(1,JCC)=CHR13                                     002061
01132    419*           CALL FP2FD(ZDATA,6,1,FDOUT,4)                                                 002071
01133    420*      1050 LABEL(1)=LABL(1,JCC)                                                          002101
01134    421*           LABEL(2)=LABL(2,JCC)                                                          002104
01134    422*     C                                                                                   002104
01134    423*     C        CHECK TIMOUT                                                               002104
01134    424*     C                                                                                   002104
01135    425*           CALL MS2DAY(TIMOUT(ISCAN),DAY,HOUR,MIN,ZSEC,$1860)                            002106
01135    426*     C       BUILD TIME OUTPUT WORD FOR TABS                                             002106
01136    427*           LIMLIN(2)=BLANK                                                               002121
01137    428*           CALL INT2FD(DAY,3,LIMLIN(1),2)                                                002123
01140    429*           CALL FD2FD(1,COLON,6,LIMLIN(1),5)                                             002131
01141    430*           CALL INT2FD(HOUR,2,LIMLIN(1),6)                                               002140
01142    431*           CALL FD2FD(1,COLON,6,LIMLIN(2),2)                                             002146
01143    432*           CALL INT2FD(MIN,2,LIMLIN(2),3)                                                002155
01144    433*           LOGLIN(1)=LIMLIN(1)                                                           002163
01145    434*           LOGLIN(2)=LIMLIN(2)                                                           002165
01145    435*     C                                                                                   002165
01145    436*     C     CHECK FOR LOG MODE OR OVERLOAD                                              002165
01145    437*     C                                                                                   002165
01146    438*           IF(CHR2.EQ.BLNKED)GO TO 1200                                                  002167
01146    439*     C                                                                                   002167
01146    440*     C       SPACE LIMIT TAB LINE                                                        002167
01146    441*     C                                                                                   002167
01150    442*           CALL FD2FD(3,BUF(1),LSTPTR+3,LIMLIN(3),4)                                     002172
01151    443*           CALL FD2FD(9,FDOUT(1),1,LIMLIN(5),4)                                          002204
01152    444*           CALL FD2FD(9,LABEL(1),1,LIMLIN(8),1)                                          002213
01153    445*           CALL FD2FD(1,BUF(1),LSTPTR+2,LIMLIN(10),6)                                    002222
01153    446*     C                                                                                   002222
01153    447*     C     PUT LIMBUF ON MASS STORAGE                                                    002222
01154    448*           LIMROW=LIMROW+1                                                               002234
01155    449*           DO 1075 J=1,MXLIM                                                             002237
01160    450*           LIMBUF(J,LIMROW)=LIMLIN(J)                                                    002246
01161    451*      1075 LIMLIN(J)=BLANK                                                               002247
01163    452*           WRD=MXLIM*LIMROW                                                              002252
01164    453*           IF(LIMROW.LT.PRTROW)GO TO 1085                                                002255
01166    454*           CALL NTRAN(TSF,1,WRD,LIMBUF(1,1),LSTAT)                                       002261
01167    455*      1076 CALL NTRAN(TSF,22)                                                            002271
01170    456*           IF(LSTAT.EQ.-1)GO TO 1076                                                     002274
01172    457*           LREC=LREC+1                                                                   002277
```

```
01173   458*          LIMROW=3                                                  002302
01174   459*       1085 CONTINUE                                                002305
01175   460*          GO TO 1450                                                002305
01175   461*     C                                                              002305
01175   462*     C      FIRST WORD PER SCAN IS TIME                             002305
01175   463*     C                                                              002305
01175   464*     C    STORE AND OUTPUT FOR FIXED SAMPLE RATE TAPE               002305
01175   465*     C                                                              002305
01175   466*     C                                                              002305
01175   467*     C    DATA USES SLOTS 2 THROU 42 PER SCAN AND 12 SCANS          002305
01176   468*       1200 IF(ISCAN.EQ.LSCAN)GO TO 1205                            002306
01200   469*          LSCAN=ISCAN                                               002310
01201   470*          TSLOT=TSCAN*42+1                                          002312
01202   471*          PLTBUF(TSLOT)=TIMOUT(ISCAN)                               002320
01203   472*       1205 PBUF=CHANLC(JCC)+TSLOT                                  002323
01204   473*          IF(PBUF.GT.BUFSIZ)GO TO 1900                              002326
01206   474*          ZPLTBU(PBUF)=ZDATA                                        002331
01206   475*     C                                                              002331
01206   476*     C    SET UP AND SPACE LOG TAB LINE BY THE SCAN                 002331
01207   477*          LUNIT(1)=BLANK                                            002334
01210   478*          LUNIT(2)=BLANK                                            002336
01211   479*          LUNIT(3)=BLANK                                            002337
01212   480*          JCHNL=1                                                   002340
01213   481*          GRP=0                                                     002342
01214   482*           DO 1225  JCOL=JCHNL,40,TABCOL                            002351
01217   483*          GRP=GRP+1                                                 002351
01220   484*          IF(CHANLC(JCC).EQ.JCOL)GO TO 1230                         002354
01222   485*       1225 CONTINUE                                                002361
01224   486*          GO TO 1235                                                002361
01225   487*       1230 CALL FD2FD(9,FDOUT(1),4,TABBUF(4,GRP+8),5)              002363
01226   488*          CALL FD2FD(9,LABEL(1),1,TABBUF(4,GRP+3),5)                002376
01227   489*          GO TO 1425                                                002410
01230   490*       1235 JCHNL=2                                                 002412
01231   491*          GRP=0                                                     002413
01232   492*           DO 1245  JCOL=JCHNL,MXCHAN,TABCOL                        002423
01235   493*          GRP=GRP+1                                                 002423
01236   494*          IF(CHANLC(JCC).EQ.JCOL)GO TO 1250                         002426
01240   495*       1245 CONTINUE                                                002433
01242   496*          GO TO 1255                                                002433
01243   497*       1250 CALL FD2FD(9,FDOUT(1),4,TABBUF(6,GRP+8),3)              002435
01244   498*          CALL FD2FD(9,LABEL(1),1,TABBUF(6,GRP+3),3)                002450
01245   499*          GO TO 1425                                                002462
01246   500*       1255 JCHNL=3                                                 002464
01247   501*          GRP=0                                                     002465
01250   502*           DO 1265  JCOL=JCHNL,MXCHAN,TABCOL                        002475
01253   503*          GRP=GRP+1                                                 002475
01254   504*          IF(CHANLC(JCC).EQ.JCOL)GO TO 1270                         002500
01256   505*       1265 CONTINUE                                                002505
01260   506*          GO TO 1275                                                002505
01261   507*       1270 CALL FD2FD(9,FDOUT(1),4,TABBUF(8,GRP+8),1)              002[illegible]
01262   508*          CALL FD2FD(9,LABEL(1),1,TABBUF(8,GRP+3),1)                002[illegible]
01263   509*          GO TO 1425                                                002534
01264   510*       1275 JCHNL=4                                                 002536
01265   511*          GRP=0                                                     002537
01266   512*           DO 1285  JCOL=JCHNL,MXCHAN,TABCOL                        002547
01271   513*          GRP=GRP+1                                                 002547
01272   514*          IF(CHANLC(JCC).EQ.JCOL)GO TO 1290                         002552
```

```
01274   515*      1285 CONTINUE                                                    002557
01276   516*           GO TO 1295                                                  002557
01277   517*      1290 CALL FD2FD(9,FDOUT(1),4,TABBUF(9,GRP+8),5)                  002561
01300   518*           CALL FD2FD(9,LABEL(1),1,TABBUF(9,GRP+3),5)                  002574
01301   519*           GO TO 1425                                                  002606
01302   520*      1295 JCHNL=5                                                     002610
01303   521*           GRP=0                                                       002611
01304   522*            DO 1305  JCOL=JCHNL,MXCHAN,TABCOL                          002621
01307   523*           GRP=GRP+1                                                   002621
01310   524*           IF(CHANLC(JCC).EQ.JCOL)GO TO 1310                           002624
01312   525*      1305 CONTINUE                                                    002631
01314   526*           GO TO 1315                                                  002631
01315   527*      1310 CALL FD2FD(9,FDOUT(1),4,TABBUF(11,GRP+8),3)                 002633
01316   528*           CALL FD2FD(9,LABEL(1),1,TABBUF(11,GRP+3),3)                 002646
01317   529*           GO TO 1425                                                  002660
01320   530*      1315 JCHNL=6                                                     002662
01321   531*           GRP=0                                                       002663
01322   532*            DO 1325  JCOL=JCHNL,MXCHAN,TABCOL                          002673
01325   533*           GRP=GRP+1                                                   002673
01326   534*           IF(CHANLC(JCC).EQ.JCOL)GO TO 1330                           002676
01330   535*      1325 CONTINUE                                                    002703
01332   536*           GO TO 1335                                                  002703
01333   537*      1330 CALL FD2FD(9,FDOUT(1),4,TABBUF(13,GRP+8),1)                 002705
01334   538*           CALL FD2FD(9,LABEL(1),1,TABBUF(13,GRP+3),1)                 002720
01335   539*           GO TO 1425                                                  002732
01336   540*      1335 JCHNL=7                                                     002734
01337   541*           GRP=0                                                       002735
01340   542*            DO 1340  JCOL=JCHNL,MXCHAN,TABCOL                          002745
01343   543*           GRP=GRP+1                                                   002745
01344   544*           IF(CHANLC(JCC).EQ.JCOL)GO TO 1350                           002750
01346   545*      1340 CONTINUE                                                    002755
01350   546*           GO TO 1355                                                  002755
01351   547*      1350 CALL FD2FD(9,FDOUT(1),4,TABBUF(14,GRP+8),5)                 002757
01352   548*           CALL FD2FD(9,LABEL(1),1,TABBUF(14,GRP+3),5)                 002772
01353   549*           GO TO 1425                                                  003004
01354   550*      1355 JCHNL=8                                                     003006
01355   551*           GRP=0                                                       003007
01356   552*            DO 1365  JCOL=JCHNL,MXCHAN,TABCOL                          003017
01361   553*           GRP=GRP+1                                                   003017
01362   554*           IF(CHANLC(JCC).EQ.JCOL)GO TO 1370                           003022
01364   555*      1365 CONTINUE                                                    003027
01366   556*           GO TO 1375                                                  003027
01367   557*      1370 CALL FD2FD(9,FDOUT(1),4,TABBUF(16,GRP+8),3)                 003031
01370   558*           CALL FD2FD(9,LABEL(1),1,TABBUF(16,GRP+3),3)                 003044
01371   559*           GO TO 1425                                                  003056
01372   560*      1375 JCHNL=9                                                     003060
01373   561*           GRP=0                                                       003061
01374   562*            DO 1385  JCOL=JCHNL,MXCHAN,TABCOL                          003071
01377   563*           GRP=GRP+1                                                   003071
01400   564*           IF(CHANLC(JCC).EQ.JCOL)GO TO 1390                           003074
01402   565*      1385 CONTINUE                                                    003101
01404   566*           GO TO 1395                                                  003101
01405   567*      1390 CALL FD2FD(9,FDOUT(1),4,TABBUF(18,GRP+8),1)                 003103
01406   568*           CALL FD2FD(9,LABEL(1),1,TABBUF(18,GRP+3),1)                 003116
01407   569*           GO TO 1425                                                  003130
01410   570*      1395 JCHNL=10                                                    003132
01411   571*           GRP=0                                                       003133
```

```
01412    572*           DO 1405  JCOL=JCHNL,MXCHAN,TABCOL                    003143
01415    573*          GRP=GRP+1                                             003143
01416    574*          IF(CHANLC(JCC).EQ.JCOL)GO TO 1410                     003146
01420    575*     1405 CONTINUE                                              003153
01422    576*          GO TO 1425                                            003153
01423    577*     1410 CALL FD2FD(9,FDOUT(1),4,TABBUF(19,GRP+8),5)           003155
01429    578*          CALL FD2FD(9,LABEL(1),1,TABBUF(19,GRP+3),5)           003170
01425    579*     1425 CONTINUE                                              003203
01425    580*     C                                                          003203
01425    581*     C       PUT TIME IN TABBUF                                 003203
01425    582*     C                                                          003203
01426    583*          TABBUF(1,GRP+8)=LOGLIN(1)                             003203
01427    584*          TABBUF(2,GRP+8)=LOGLIN(2)                             003206
01427    585*     C                                                          003206
01427    586*     C      CHANNEL ROW TABBED OUT 1-10...31-40                 003206
01427    587*     C                                                          003206
01430    588*          TABBUF(3,GRP+8)=TABBUF(3,GRP+3)                       003210
01431    589*          LIN=LIN+1                                             003212
01432    590*          TABBUF(1,GRP+8)=LOGLIN(1)                             003215
01433    591*          TABBUF(2,GRP+8)=LOGLIN(2)                             003217
01434    592*          DO 1426 JJL=1,MXLOG                                   003224
01437    593*     1426 LOGLIN(JJL)=BLANK                                     003224
01437    594*     C                                                          003224
01437    595*     C    MORE CHANNELS ?                                       003224
01437    596*     C                                                          003224
01441    597*     1450 DATWRD=DATWRD-1                                       003227
01441    598*     CX     IDMP CHR                                            003227
01442    599*          IF(IDMP.GT.72)GO TO 1451                              003231
01442    600*     CX     WRITE(LUOUT,5555)CHR1,CHR2,CHR6,CHR7,CHR13,CHNL     003231
01444    601*     5555 FORMAT(1X,6012)                                       003235
01445    602*          IDMP=IDMP+1                                           003235
01446    603*     1451 CONTINUE                                              003241
01447    604*          LSTPTR=LSTPTR+12                                      003241
01450    605*          CALL SCHCHR(BUF(1),LSTPTR,6,EOW1,NXTPTR,$1452)        003243
01451    606*          GO TO 1454                                            003253
01452    607*     1452 CALL SCHCHR(BUF(1),LSTPTR,6,EOW2,NXTPTR,$4040)        003255
01453    608*     1454 LSTPTR=NXTPTR                                         003265
01454    609*          IF(DATWRD.GT.0)GO TO 355                              003266
01454    610*     C                                                          003266
01454    611*     C       CHECK TO SEE IF ANY LOG DATA PRESENT FOR THIS SCAN  003266
01456    612*          IF(LINSAV.EQ.LIN)GO TO 1485                           003271
01460    613*          LINSAV=LIN                                            003274
01460    614*     C                                                          003274
01460    615*     C    TAB OUT LOG DATA BY SCAN                              003274
01460    616*     C                                                          003274
01461    617*          IF(LINHDG.LT.48)GO TO 1469                            003276
01463    618*          LINHDG=0                                              003302
01464    619*          WRITE(LUOUT,3014)((TABBUF(J21,J8),J21=1,21),J8=1,8)   003303
01475    620*     1469 CONTINUE                                              003316
01476    621*          JGRP=GRP+8                                            003316
01477    622*          LINHDG=LINHDG+JGRP-7                                  003320
01500    623*          WRITE(LUOUT,3014)((TABBUF(J21,J12),J21=1,21),J12=9,JGRP)  003327
01511    624*          WRITE(LUOUT,3014)LOGLIN                               003342
01511    625*     C   FIXED SAMPLE RATE 42 DATA * 12 SCANS=504 WORDS         003342
01511    626*     C                                                          003342
01514    627*          TSCAN=TSCAN+1                                         003352
01515    628*          IF(TSCAN.LT.12)GO TO 1485                             003355
```

```
01517   629*          CALL NTRAN(TAPEF,1,BUFSIZ,PLTBUF(1),WSTAT)                    003360
01520   630*    1480  CALL NTRAN(TAPEF,22)                                         003370
01521   631*          IF(WSTAT.EQ.-1)GO TO 1480                                     003373
01523   632*          IF(WSTAT.EQ.BUFSIZ)GO TO 1490                                 003376
01523   633*    C                                                                   003376
01523   634*    C     FATAL ERROR                                                   003376
01525   635*          WRITE(LUOUT,4060)TREC,TSCAN                                   003401
01531   636*          GO TO 1500                                                    003410
01532   637*    1483  TSCAN=0                                                       003412
01533   638*          TREC=TREC+1                                                   003412
01534   639*          DO 1484 JT=1,BUFSIZ                                           003420
01537   640*    1484  PLTBUF(JT)=NEGZRO                                           003420
01537   641*    C                                                                   003420
01537   642*    C      MORE SCANS?  IF NOT READ ANOTHER INPUT RECORD                003420
01541   643*    1485  IF(ISCAN.LT.MXSCAN)GO TO 231                                003423
01541   644*    C      THROUGH WITH DATA SCAN                                       003423
01543   645*          GO TO 200                                                     003426
01544   646*    1500  CONTINUE                                                     003430
01544   647*    C                                                                   003430
01544   648*    C      TAB OUT LAST DATA                                            003430
01544   649*    C                                                                   003430
01545   650*          WRITE(LUOUT,3018)                                             003430
01547   651*          WRITE(LUOUT,3014)TABBUF                                       003434
01547   652*    CX    CALL NTRAN(TAPEIN,11,22)                                      003434
01552   653*          CALL NTRAN(TAPEIN,10,22)                                      003444
01553   654*          IF(TAPE2.EQ.0)GO TO 1510                                      003451
01553   655*    C                                                                   003451
01553   656*    C    CONTINUATION TAPE                                            003451
01553   657*    C                                                                   003451
01555   658*          TAPEIN=TAPE2                                                  003453
01556   659*          TAPE2=0                                                       003455
01557   660*          CALL NTRAN(TAPEIN,10,22)                                      003456
01560   661*          IF(MOVE2.EQ.0)GO TO 200                                       003463
01562   662*          CALL NTRAN(TAPEIN,8,MOVE2)                                    003465
01563   663*          CALL NTRAN(TAPEIN,22)                                         003472
01564   664*          GO TO 200                                                     003476
01565   665*    1510  IF(TSCAN.EQ.0)GO TO 1519                                     003500
01567   666*          CALL NTRAN(TAPEF,1,BUFSIZ,PLTBUF(1),WSTAT)                    003501
01570   667*    1515  CALL NTRAN(TAPEF,22)                                        003511
01571   668*          IF(WSTAT.EQ.-1)GO TO 1515                                     003514
01571   669*    C                                                                   003514
01571   670*    C     DO EOF,REWIND, AND RELEASE OF FIXED SAMPLE RATE TAPE          003514
01571   671*    C                                                                   003514
01573   672*    1519  IF(TREC.EQ.0)GO TO 1528                                     003520
01575   673*          IF(.NOT.TAPEO)CALL NTRAN(TAPEF,10,22)                         003521
01577   674*          IF(TAPEO)CALL NTRAN(TAPEF,9,10,22)                            003530
01577   675*    C                                                                   003530
01601   676*          CALL NTRAN(TAPEF,2,BUFSIZ,PLTBUF,INSTAT)                      003540
01602   677*    1521  CALL NTRAN(TAPEF,22)                                        003550
01603   678*          IF(INSTAT.EQ.-1)GO TO 1521                                    003553
01605   679*          IF(INSTAT.NE.BUFSIZ)GO TO 1528                                003556
01607   680*          WRITE(LUOUT,3015)PLTBUF                                       003561
01607   681*    CX    CALL NTRAN(TAPEF,11,22)                                       003561
01612   682*          CALL NTRAN(TAPEF,10,22)                                       003571
01612   683*    C      WRITE LIMIT DATA ON TEMP STATUS FILE (TSF)                   003571
01613   684*    1528  IF(LIMROW.LT.4)GO TO 1540                                   003577
01615   685*          WRD=MXLIM*LIMROW                                              003602
```

```
01616   686*            CALL NTRAN(TSF,1,WRD,LIMBUF(1,1),LSTAT)                                  003605
01617   687*     1530   CALL NTRAN(TSF,22)                                                      003615
01620   688*            IF(LSTAT.EQ.-1)GO TO 1530                                                003620
01622   689*            LREC=LREC+1                                                              003623
01623   690*     1540   WRD=MXLIM*PRTROW                                                        003627
01624   691*            CALL NTRAN(TSF,10,22)                                                    003630
01624   692*     C                                                                               003630
01624   693*     C       PRINT LIMIT DATA FROM TSF FILE                                          003630
01625   694*            WRITE(LUOUT,3017)                                                        003645
01625   695*     C                                                                               003645
01627   696*            DO 1550 JPL=1,LREC                                                       003660
01632   697*            JCK=JPL                                                                  003660
01633   698*            IF(LREC.EQ.0)GO TO 1546                                                  003662
01635   699*            IF(JCK.GE.LREC)WRD=MXLIM*LIMROW                                          003664
01637   700*            CALL NTRAN(TSF,2,WRD,LIMBUF(1,1),LWSTAT)                                 003672
01640   701*     1545   CALL NTRAN(TSF,22)                                                      003702
01641   702*            IF(LWSTAT.EQ.-1)GO TO 1545                                               003705
01643   703*     1546   LROW=WRD/MXLIM                                                         003711
01644   704*     1550   WRITE(LUOUT,3000)((LIMBUF(J12,JX),J12=1,12),JX=1,LROW)                 003714
01656   705*            WRITE(LUOUT,3013)                                                        003733
01660   706*            WRITE(LUOUT,3012)IREC,TREC,DAY,HOUR,MIN,TIMOUT(ISCAN)                    003740
01670   707*            GO TO 1900                                                               003753
01670   708*     C      FORMATS INPUT                                                            003753
01670   709*     C                                                                               003753
01671   710*      2000  FORMAT(12A6)                                                             003755
01672   711*      2002  FORMAT(13A6,A2)                                                          003755
01673   712*      2004  FORMAT(16X,I3,1X,I2,1X,I2,2(11X,I3,1X,I2,1X,I2))                         003755
01674   713*      2006  FORMAT(13X,I2,4(8X,I2),A6)                                            003755
01675   714*      2008  FORMAT(6X,I2,1X,2A6,2X,I1,4X,I2)                                       003755
01675   715*     C                                                                               003755
01675   716*     C      FORMATS OUTPUT                                                           003755
01675   717*     C                                                                               003755
01676   718*      3000  FORMAT(12A6)                                                           003755
01677   719*      3001  FORMAT(1X,13A6,A2)                                                     003755
01700   720*      3002  FORMAT('0  CSVIDR WILL EXECUTE USING INFO AS FOLLOWS:')               003755
01701   721*      3003  FORMAT ('0 TITLE= ',13A6,A2)                                         003755
01702   722*      3004  FORMAT('0 START TIME DDD= ',I3,' HH= ',I2,' MM= ',I2,' MS= ',I12)     003755
01703   723*      3005  FORMAT('0 STOP  TIME DDD= ',I3,' HH= ',I2,' MM= ',I2,' MS= ',I12)     003755
01704   724*      3006  FORMAT('0 TAPE  TIME DDD= ',I3,' HH= ',I2,' MM= ',I2,' MS= ',I12)     003755
01705   725*      3007  FORMAT('0 TAPEIO TAPE1 WILL USE UNIT',I3)                             003755
01706   726*      3008  FORMAT('0 TAPEIO TAPE2 WILL USE UNIT',I3)                             003755
01707   727*      3009  FORMAT('0 TAPEIO TAPEF WILL USE UNIT',I3)                             003755
01710   728*      3010  FORMAT('0 CHANNELS REQUESTED ARE:'/                                   003755
01710   729*           *' PTR  LABEL       DEG'/                                              003755
01710   730*           *(1X,I3,1X,2A6,1X,I3))                                                 003755
01711   731*      3065  FORMAT('  ABSCISSA',7XE12.7,'=X1',3XE12.7,'=X2',3XE12.7,'=X3',3X      003755
01711   732*           1E12.7,'=X4',3XE12.7,'=X5',3XE12.7,'=X6',/17XE12.7,'=X7',3XE12.7, '    003755
01711   733*           2=X8',3XE12.7,'=X9',3XE12.7,'=X10',2XE12.7,'=X11',2XE12.7,'=X12',/1    003755
01711   734*           37XE12.7,'=X13',2XE12.7,'=X14',2XE12.7,'=X15',2XE12.7,'=X16',2XE12.   003755
01711   735*           47,'=X17',2XE12.7,'=X18',/17XE12.7,'=X19',2XE12.7,'=X20',2XE12.7,'=   003755
01711   736*           5X21')                                                                 003755
01712   737*      3070  FORMAT('0 ORDINATE',7XE12.7,'=Y1',3XE12.7,'=Y2',3XE12.7,'=Y3',3X     003755
01712   738*           1E12.7,'=Y4',3XE12.7,'=Y5',3XE12.7,'=Y6',/17XE12.7,'=Y7',3XE12.7, '    003755
01712   739*           2=Y8',3XE12.7,'=Y9',3XE12.7,'=Y10',2XE12.7,'=Y11',2XE12.7,'=Y12',/1    003755
01712   740*           37XE12.7,'=Y13',2XE12.7,'=Y14',2XE12.7,'=Y15',2XE12.7,'=Y15',2XE12.   003755
01712   741*           47,'=Y17',2XE12.7,'=Y18',/17XE12.7,'=Y19',2XE12.7,'=Y20',2XE12.7,'=   003755
01712   742*           5Y21')                                                                 003755
```

```
01713    793*      3011 FORMAT(/'  COEFFICIENTS',3XE12.7,'=A0',3XE12.7,'=A1',3XE12.7,'=A2'      003755
01713    794*          1,3XE12.7,'=A3',3XE12.7,'=A4',3XE12.7,'=A5')                            003755
01714    795*      3012 FORMAT(' INPUT RECORDS=',I6,' OUTPUT RECORDS=',I6,' LAST TIME OUT      003755
01714    796*          * IS',1X,I3,2(1X,I2),' MILLISECONDS= ',I12)                             003755
01715    797*      3013 FORMAT( '1 NORMAL EXIT FOR PROGRAM CSVIDR')                            003755
01716    798*      3014 FORMAT(21A6)                                                           003755
01717    799*      3015 FORMAT('1FIXED SAMPLE RATE RECORD ',(1X,I12,/,10                       003755
01717    800*          *E13.7,/,10E13.7,/,10E13.7,/,10E13.7,/,E13.7,//))                       003755
01720    801*      3016 FORMAT('0 END OF WORD CHARACTER EOW=',A6)                              003755
01721    802*      3017 FORMAT('1')                                                            003755
01722    803*      3018 FORMAT('1CSVIDR WILL OUTPUT LAST LOG DATA SAVED',//                    003755
01722    804*          *'0DUMP THE FIRST RECORD OF FIXED S/R TAPE',//                          003755
01722    805*          *'0PRINT OUT LIMIT MODE DATA IF PRESENT')                               003755
01722    806*     C                                                                            003755
01722    807*     C      DIAGNOSTIC MSG AND ERROR MSG                                         003755
01722    808*     C                                                                            003755
01723    809*      4000 WRITE(LUOUT,4001)                                                      003755
01725    810*      4001 FORMAT('0 **CSVIDR**POINTS OR DEGREE IN ERROR. WILL READ MORE DATA     003761
01725    811*          * SETS IF POSSIBLE')                                                    003761
01726    812*           NOTAB=.TRUE.                                                           003761
01727    813*           GO TO 40                                                               003763
01730    814*      4020 FORMAT('0**CSVIDR**HAS ENCOUNTERED A BAD INPUT RECORD')                003765
01731    815*      4025 FORMAT('  THE ERROR OCCURRED TRYING TO READ RECORD',I7,'.',I6,'THI     003765
01731    816*          *RTY SIX BIT WORDS WERE RETURNED')                                      003765
01732    817*      4030 III=ISTCH                                                              003765
01733    818*           II2=ISTCH+2                                                            003766
01734    819*           WRITE(LUOUT,4035)COLON,ISCAN,(BUF(III),III=III,II2)                    003770
01744    820*      4035 FORMAT('0**CSVIDR**COULD NOT FIND THE TIME CHARACTER (',A6,')          004007
01744    821*          * SCAN ',I3,1X,/,1X,15A6)                                               004007
01745    822*           PSCAN=PSCAN+1                                                          004007
01746    823*           ISCAN=ISCAN+1                                                          004012
01747    824*           ISTCH=ISTCH+1                                                          004015
01750    825*           IF(ISCAN.GT.36)GO TO 200                                               004020
01752    826*           IF(PSCAN.GT.36)GO TO 200                                               004023
01754    827*           GO TO 231                                                              004026
01755    828*      4040 WRITE(LUOUT,4045)ISCAN                                                 004030
01760    829*      4045 FORMAT('0**CSVIDR**COULD NOT FIND END OF WORD CHARACTER FOR SCAN '     004035
01760    830*          *,I3)                                                                   004035
01761    831*           WBAD=WBAD+1                                                            004035
01762    832*           ISTCH=ISTCH+1                                                          004040
01763    833*           IF(WBAD.GT.37)GO TO 200                                                004043
01765    834*      4003 FORMAT('0**CSVIDR**DOES NOT HAVE ANY LEAD CARDS.  DEFAULTS USED')      004046
01766    835*           GO TO 231                                                              004046
01767    836*      4049 WRITE(LUOUT,4051)CHNL                                                  004050
01772    837*           BADCHN=BADCHN+1                                                        004055
01773    838*           IF (BADCHN.GT.100)GO TO 1500                                           004060
01775    839*           LSTPTR=LSTPTR+1                                                        004063
01776    840*           GO TO 355                                                              004066
01777    841*      4050 WRITE(LUOUT,4051)IDCHAN                                                004070
02002    842*      4051 FORMAT('0**CSVIDR**HAS AN ERROR IN FINDING CHANNEL NUMBER',I6)         004075
02003    843*           BADCRD=BADCRD+1                                                        004075
02004    844*           GO TO 40                                                               004100
02005    845*      4052 WRITE(LUOUT,4053)                                                      004102
02007    846*      4053 FORMAT('0**CSVIDR**HAS AN ERROR IN FREE FIELD INPUT')                  004106
02010    847*           BADCRD=BADCRD+1                                                        004106
02011    848*           GO TO 40                                                               004111
02012    849*      4056 WRITE(LUOUT,4051)INCHAN                                                004113
```

```
02015    800*            BADCRD=BADCRD+1                                        004120
02016    801*            GO TO 40                                               004123
02017    802*       4057 WRITE(LUOUT,4058)                                      004125
02021    803*            BADCHR=BADCHR+1                                        004131
02022    804*            IF( BADCHR.GT.100)GO TO 1500                           004134
02024    805*            GO TO 400                                              004137
02025    806*       4058 FORMAT('0**CSVIDR**COULD NOT FIND TYPE')               004141
02026    807*       1850 WRITE(LUOUT,1851)                                      004141
02030    808*       1851 FORMAT('0**CSVIDR**UNABLE TO CONVERT DAYS TO MS')      004145
02031    809*            GO TO 1900                                             004145
02032    810*       1860 WRITE(LUOUT,1861)                                      004147
02034    811*       1861 FORMAT('0**CSVIDR**UNABLE TO CONVERT MS TO DAYS')      004153
02035    812*            GO TO 1900                                             004153
02036    813*       1900 CONTINUE                                               004155
02037    814*            WRITE(LUOUT,4059)IREC,LREC,TREC,LSTPTR,ISTCH,ISCAN     004155
02047    815*       4059 FORMAT(' **CSVIDR**      EXIT IREC= ',I4,' LREC= ',I4,  004167
02047    816*           *' TREC= ',I4,' LSTPTR= ',I4,' ISTCH= ',I4,' ISCAN= ',I4)004167
02050    817*       4060 FORMAT('0**CSVIDR**BAD FIXED SAMPLE RATE OUTPUT RECORD.004167
02050    818*           *  LAST REC=',I4,' SCAN=',I3)                           004167
02050    819*      C                                                            004167
02050    820*      C                                                            004167
02051    821*            STOP                                                   004167
02052    822*            END                                                    004173
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.CURV61
FOR SE2C-10/31/75-17:32:26 (1,)

   SUBROUTINE CURV61    ENTRY POINT 000416

   STORAGE USED: CODE(1) 000446; DATA(0) 000437; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000055 114G      0001   000064 122G      0001   000077 130G      0001   000105 135G      0001   000114 142G
    0001   000131 151G      0001   000145 160G      0001   000150 163G      0001   000172 172G      0001   000200 177G
    0001   000220 203G      0001   000255 214G      0001   000260 220G      0001   000335 232G      0001   000370 244G
    0001   000301 33L       0001   000314 35L       0000 D 000212 B         0000 D 000350 DIVB      0000 D 000353 FMULTB
    0000 I 000343 I         0000 I 000341 II        0000   000373 INJP$     0000 I 000352 I1        0000 I 000342 J
    0000 I 000346 K         0000 I 000347 L         0000 I 000337 LB        0000 I 000336 LS        0000 I 000340 LV
    0000 D 000344 P         0000 D 000355 SIGMA     0000 D 000124 SUM       0000 D 000176 V         0000 D 000000 X
    0000 D 000052 Y

00100      1*     C                                                                                          000010
00101      2*            SUBROUTINE CURV61(SPX,SPY,M,N,SPA,A)                                                000010
00101      3*     C                                                                                          000010
00103      4*            PARAMETER NPTXY=21,NDEGP1=6,NDEGP2=7                                                000010
00104      5*            IMPLICIT DOUBLE PRECISION (A-H,O-Z)                                                 000010
00105      6*            REAL SPA                                                                            000010
00105      7*     C      X AND Y ARE DATA ARRAYS.                                                            000010
00105      8*     C      M IS THE DEGREE OF EQUATION TO BE FITTED.                                           000010
00105      9*     C     N IS THE NUMBER OF POINTS IN THE DATA ARAY.                                          000010
00105     10*     C      A IS THE COEFFICIENTS OF THE EQUATION STARTING WITH THE CONSTANT TERM.              000010
00106     11*            DIMENSION SPA(1),SPX(1),SPY(1),X(NPTXY),Y(NPTXY),SUM(NPTXY)                         000010
00107     12*            DIMENSION V(NDEGP1),A(NDEGP1),B(NDEGP1,NDEGP2)                                      000010
00110     13*            LS=2*M+1                                                                            000010
00111     14*            LB=M+2                                                                              000014
00112     15*            LV=M+1                                                                              000017
00113     16*            DO 3 II=1,N                                                                         000031
00116     17*            X(II)=SPX(II)                                                                       000055
00117     18*          3 Y(II)=SPY(II)                                                                       000056
00121     19*            DO 5 J=2,LS                                                                         000064
00124     20*          5 SUM(J)=0.0D0                                                                        000064
00126     21*            SUM(1)=N                                                                            000066
00127     22*            DO 6 J=1,LV                                                                         000077
00132     23*          6 V(J)=0.0D0                                                                          000077
00134     24*            DO16 I=1,N                                                                          000105
00137     25*            P=1.0D0                                                                             000105
00140     26*            V(1)=V(1)+Y(I)                                                                      000106
00141     27*            DO 13 J=2,LV                                                                        000114
00144     28*            P=X(I)*P                                                                            000114
```

```
00145     29*          SUM(J)=SUM(J)+P                                    000116
00146     30*       13 V(J)=V(J)+Y(I)*P                                   000120
00150     31*          DO 16 J=LB,LS                                      000131
00153     32*          P=X(I)*P                                           000131
00154     33*       16 SUM(J)=SUM(J)+P                                    000133
00157     34*       17 DO 20 I=1,LV                                       000150
00162     35*          DO 20 K=1,LV                                       000150
00165     36*          J=K+I                                              000150
00166     37*       20 B(K,I)=SUM(J-1)                                    000153
00171     38*          DO 22 K=1,LV                                       000172
00174     39*       22 B(K,LB)=V(K)                                       000172
00176     40*       23 DO 31 L=1,LV                                       000200
00201     41*          DIVB=B(L,L)                                        000212
00202     42*          DO 26 J=L,LB                                       000220
00205     43*       26 B(L,J)=B(L,J)/DIVB                                 000220
00207     44*          I1=L+1                                             000223
00210     45*          IF (I1-LB)28,33,33                                 000226
00213     46*       28 DO 31 I=I1,LV                                      000231
00216     47*          FMULTB=B(I,L)                                      000255
00217     48*          DO 31 J=L,LB                                       000260
00222     49*       31 B(I,J)=B(I,J)-B(L,J)*FMULTB                        000260
00226     50*       33 A(LV)=B(LV,LB)                                     000301
00227     51*          I=LV                                               000311
00230     52*       35 SIGMA=0.0D0                                        000314
00231     53*          DO 37 J=I,LV                                       000315
00234     54*       37 SIGMA=SIGMA+B(I-1,J)*A(J)                          000335
00236     55*          I=I-1                                              000341
00237     56*          A(I)=B(I,LB)-SIGMA                                 000344
00240     57*       40 IF (I-1)41,41,35                                   000356
00243     58*       41 DO 42 II=1,NDEGP                                   000370
00246     59*       42 SPA(II)=A(II)                                      000370
00250     60*          RETURN                                             000372
00251     61*          END                                                000445
END FOR
```

```
*FOR,S F05-L75036*CALCUR.DHM2MS
FOR SE2C-10/31/75-17:33:17 (1,)

   SUBROUTINE DHM2MS    ENTRY POINT 000032

   STORAGE USED: CODE(1) 000041; DATA(0) 000006; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR4$
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000002 INJP$



00101      1*            SUBROUTINE DHM2MS(IDAY,IHR,MIN,MS,S)                    000000
00103      2*            MS=IDAY*86400000+IHR*3600000+MIN*60000                  000000
00104      3*            IF(MS.LT.0)RETURN 5                                     000010
00106      4*            RETURN                                                  000017
00107      5*            END                                                     000040
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.EVALFT
FOR SE2C-10/31/75-17:33:54 (1,)

   SUBROUTINE EVALFT    ENTRY POINT 000044

   STORAGE USED: CODE(1) 000060; DATA(0) 000022; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   XPDI
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000015 107G      0000   000004 INJP$     0000 I 000002 J          0000 D 000000 YD


00101      1*            SUBROUTINE EVALFT(Z,NDEG,A,YC)                                  000006
00101      2*     C                                                                      000006
00101      3*     C     USED BY CURVE FIT DEMONSTRATION PROGRAM BIOFIT (J252-LEC 1380)   000006
00101      4*     C                                                                      000006
00103      5*            DOUBLE PRECISION A,Z,YD                                         000006
00104      6*            DIMENSION A(1)                                                  000006
00105      7*            YD=A(1)                                                         000006
00106      8*            DO 35 J=1,NDEG                                                  000015
00111      9*         35 YD=YD+A(J+1)*Z**J                                               000015
00113     10*            YC=YD                                                           000026
00114     11*            RETURN                                                          000030
00115     12*            END                                                             000057
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.FD2FD
FOR SE2C-10/31/75-17:39:08 (0,)

   SUBROUTINE FD2FD     ENTRY POINT 000265

   STORAGE USED: CODE(1) 000300; DATA(0) 000044; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR2$
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000244 100L      0001   000132 144G      0001   000051 90L       0001   000176 82L       0001   000202 84L
    0001   000206 86L       0001   000214 90L       0001   000226 92L       0001   000232 94L       0001   000236 96L
    0000 I 000006 I         0000   000014 INJP$     0000 I 000007 J         0000 I 000010 K         0000 I 000001 KSTCH
    0000 I 000011 L         0000 I 000002 LSTCH     0000 I 000012 M         0000 I 000003 ONE       0000 I 000000 SIGN
    0000 I 000004 SIX       0000 I 000005 THREE6

00101      1*           COMPILER (FLD=R)                                                              000012
00103      2*           SUBROUTINE FD2FD (NOCH, LINE1, ISTCH, LINE2, JSTCH)                           000012
00103      3*     C     CALLING SEQUENCE:                                                             000012
00103      4*     C        CALL FD2FD (NOCH, LINE1, ISTCH, LINE2, JSTCH)                              000012
00103      5*     C     WHERE:                                                                        000012
00103      6*     C        NOCH   = NO. OF CHARACTERS TO MOVE.                                        000012
00103      7*     C        LINE1  = FIRST WORD ADDRESS OF SOURCE LINE.                                000012
00103      8*     C        ISTCH  = STARTING CHARACTER POSITION IN SOURCE LINE TO BEGIN               000012
00103      9*     C                 MOVING FROM.                                                      000012
00103     10*     C        LINE2  = FIRST WORD ADDRESS OF DESTINATION LINE.                           000012
00103     11*     C        ISTCH  = STARTING CHARACTER POSITION IN DESTINATION LINE TO                000012
00103     12*     C                 BEGIN MOVING TO.                                                  000012
00103     13*     C                                                                                   000012
00105     14*           IMPLICIT INTEGER (A-Z)                                                        000012
00106     15*           DIMENSION LINE1(1), LINE2(1)                                                  000012
00107     16*           KSTCH = ISTCH                                                                 000012
00110     17*           LSTCH = JSTCH                                                                 000014
00111     18*           SIGN = +1                                                                     000016
00112     19*           ONE = 1                                                                       000020
00113     20*           SIX = 6                                                                       000021
00114     21*           THREE6 = 36                                                                   000023
00115     22*           IF (LOC(LINE1(1)).NE.LOC(LINE2(1))) GO TO 80                                  000025
00117     23*           IF (ISTCH.GT.JSTCH) GO TO 80                                                  000031
00121     24*           KSTCH = KSTCH + NOCH - 1                                                      000034
00122     25*           LSTCH = LSTCH + NOCH - 1                                                      000040
00123     26*           SIGN = 2                                                                      000043
00124     27*           ONE = -ONE                                                                    000045
00125     28*           SIX = -SIX                                                                    000046
00126     29*           THREE6 = -THREE6                                                              000047
00127     30*        80 I = KSTCH/6 + 1                                                               000051
```

```
00130     31*           IF (MOD (KSTCH, 6).EQ.0) I = I - 1                                000054
00132     32*           J = 41 - MOD (KSTCH, 6)*6                                         000064
00133     33*           IF (MOD (KSTCH, 6).EQ.0) J = J - 36                               000073
00135     34*           K = LSTCH/6 + 1                                                   000100
00136     35*           IF (MOD (LSTCH, 6).EQ.0) K = K - 1                                000104
00140     36*           L = 41 - MOD (LSTCH, 6)*6                                         000114
00141     37*           IF (MOD (LSTCH, 6).EQ.0) L = L - 36                               000123
00143     38*           DO 100 M = 1, NOCH                                                000132
00146     39*           FLD (L, 6, LINE2(K)) = FLD (J, 6, LINE1(I))                       000142
00147     40*           J = J - SIX                                                       000163
00150     41*           GO TO (82, 84), SIGN                                              000166
00151     42*        82 IF (J.GT.0) GO TO 90                                              000176
00153     43*           GO TO 86                                                          000200
00154     44*        84 IF (J.LT.36) GO TO 90                                             000202
00156     45*        86 J = J + THREE6                                                    000206
00157     46*           I = I + ONE                                                       000210
00160     47*        90 L = L - SIX                                                       000214
00161     48*           GO TO (92, 94), SIGN                                              000216
00162     49*        92 IF (L.GT.0) GO TO 100                                             000226
00164     50*           GO TO 96                                                          000230
00165     51*        94 IF (L.LT.36) GO TO 100                                            000232
00167     52*        96 L = L + THREE6                                                    000236
00170     53*           K = K + ONE                                                       000240
00171     54*       100 CONTINUE                                                         000244
00173     55*           RETURN                                                            000244
00174     56*           END                                                               000277
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.FP2FD
FOR SE2C-10/31/75-17:34:17 (0,)

   SUBROUTINE FP2FD     ENTRY POINT 000527

   STORAGE USED: CODE(1) 000545; DATA(0) 000037; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   XPII
    0004   NERR2$
    0005   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000222 100L      0001   000242 110L      0001   000102 127G      0001   000301 177G      0001   000347 200L
    0001   000351 210L      0001   000366 220L      0001   000402 230L      0001   000457 235G      0001   000423 300L
    0001   000513 310L      0001   000121 50L       0001   000130 60L       0001   000165 70L       0000 I 000010 BYTE
    0000 I 000003 I         0000 I 000002 ICNT      0000   000016 INJP$     0000 I 000005 INTSV     0000 I 000001 ISW
    0000 I 000004 J         0000 I 000007 K         0000 I 000006 NOCHM1    0000 I 000011 NOCK      0000 I 000000 060

00101      1*            COMPILER (FLD=R)                                                          000002
00103      2*            SUBROUTINE FP2FD (FP, NOCH, NORDP, FIELD, ISTCH)                          000002
00103      3*      C     CALLING SEQUENCE:                                                         000002
00103      4*      C        CALL FP2FD (FP, NOCH, NORDP, FIELD, ISTCH)                             000002
00103      5*      C     WHERE:                                                                    000002
00103      6*      C        FP     = LOCATION WHERE THE FLOATING POINT VALUE IS STORED.           000002
00103      7*      C        NOCH   = NO. OF CHARCTERS TO BE CONVERTED TO FIELD DATA.              000002
00103      8*      C        NORDP  = NO. OF DIGITS TO BE PLACED RIGHT OF THE DECIMAL POINT.       000002
00103      9*      C        FIELD  = FIRST WORD ADDRESS OF THE DESTINATION LINE.                  000002
00103     10*      C        ISTCH  = STARTING CHARACTER POSITION IN THE DESTINATION LINE.         000002
00103     11*      C                                                                              000002
00105     12*            IMPLICIT INTEGER (A-Z)                                                    000002
00106     13*            REAL FP                                                                   000002
00107     14*            DIMENSION FIELD(1)                                                        000002
00110     15*            DATA 060/060/                                                             000002
00112     16*            IF (NOCH.LT.(NORDP+2)) GO TO 300                                          000002
00114     17*            ISW = 1                                                                   000007
00115     18*            ICNT = 0                                                                  000011
00116     19*            I = (ISTCH + NOCH - 1)/6 + 1                                              000012
00117     20*            IF (MOD ((ISTCH + NOCH - 1), 6).EQ.0) I = I - 1                           000021
00121     21*            J = 41 - MOD ((ISTCH + NOCH - 1), 6)*6                                    000031
00122     22*            IF (MOD ((ISTCH + NOCH - 1), 6).EQ.0) J = J - 36                          000042
00124     23*            INTSV = ABS (FP)*(10**NORDP) + 0.5                                        000047
00125     24*            NOCHM1 = MIN ((NOCH - 1), 11)                                             000070
00126     25*            DO 100 K = 1, NOCHM1                                                      000077
00131     26*            BYTE = MOD (INTSV, 10)                                                    000102
00132     27*            INTSV = INTSV/10                                                          000105
00133     28*            GO TO (50, 70), ISW                                                       000111
00134     29*         50 IF (ICNT.EQ.NORDP) GO TO 60                                               000121
```

```
00136     30*          ICNT = ICNT + 1                                                    000123
00137     31*          GO TO 70                                                           000126
00140     32*       60 ISW = 2                                                            000130
00141     33*          FLD (J, 6, FIELD(I)) = 6H@@@@@@.                                   000131
00142     34*          J = J + 6                                                          000150
00143     35*          IF (J.LE.35) GO TO 70                                              000153
00145     36*          J = J - 36                                                         000156
00146     37*          I = I - 1                                                          000161
00147     38*       70 FLD (J, 6, FIELD(I)) = OR (060, BYTE)                              000165
00150     39*          J = J + 6                                                          000205
00151     40*          IF (J.LE.35) GO TO 100                                             000210
00153     41*          J = J - 36                                                         000213
00154     42*          I = I - 1                                                          000216
00155     43*      100 CONTINUE                                                           000222
00157     44*          IF (INTSV.GT.0) GO TO 300                                          000222
00161     45*          NOCK = NOCH - (NORDP + 2)                                          000225
00162     46*          IF (NOCK.GT.0) GO TO 110                                           000231
00164     47*          IF (FP.LT.0) GO TO 300                                             000233
00166     48*          RETURN                                                             000236
00167     49*      110 I = ISTCH/6 + 1                                                    000242
00170     50*          IF (MOD(ISTCH,6).EQ.0) I = I - 1                                   000245
00172     51*          J = 41 - MOD (ISTCH, 6)*6                                          000255
00173     52*          IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                000264
00175     53*          ISW = 1                                                            000274
00176     54*          DO 200 K = 1, NOCK                                                 000301
00201     55*          IF (FLD (J, 6, FIELD(I)).NE.6H@@@@@@) GO TO 210                    000304
00203     56*          ISW = 2                                                            000314
00204     57*          FLD (J, 6, FIELD(I)) = 6H@@@@@@                                    000320
00205     58*          J = J - 6                                                          000333
00206     59*          IF (J.GT.0) GO TO 200                                              000336
00210     60*          J = J + 36                                                         000340
00211     61*          I = I + 1                                                          000343
00212     62*      200 CONTINUE                                                           000351
00214     63*      210 IF (FP.GE.0) RETURN                                                000351
00216     64*          GO TO (300, 220), ISW                                              000356
00217     65*      220 J = J + 6                                                          000366
00220     66*          IF (J.LE.35) GO TO 230                                             000370
00222     67*          J = J - 36                                                         000373
00223     68*          I = I - 1                                                          000376
00224     69*      230 FLD (J, 6, FIELD(I)) = 6H@@@@@-                                    000402
00225     70*          RETURN                                                             000417
00226     71*      300 I = ISTCH/6 + 1                                                    000423
00227     72*          IF (MOD(ISTCH,6).EQ.0) I = I - 1                                   000426
00231     73*          J = 41 - MOD (ISTCH, 6)*6                                          000436
00232     74*          IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                000445
00234     75*          DO 310 K = 1, NOCH                                                 000457
00237     76*          FLD (J, 6, FIELD(I)) = 6H@@@@@*                                    000464
00240     77*          J = J - 6                                                          000477
00241     78*          IF (J.GT.0) GO TO 310                                              000502
00243     79*          J = J + 36                                                         000504
00244     80*          I = I + 1                                                          000507
00245     81*      310 CONTINUE                                                           000513
00247     82*          RETURN                                                             000513
00250     83*          END                                                                000544
END FOR
```

```
*FOR,S FDS-L75036*CALCUR.FD2INT
FOR SE2C-10/31/75-17:34:38 (0,)

   SUBROUTINE FD2INT    ENTRY POINT 000215

   STORAGE USED: CODE(1) 000225; DATA(0) 000023; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR2$
    0004   NERR4$
    0005   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000164 100L      0001   000174 110L      0001   000176 120L      0001   000044 124G      0001   000062 50L
    0001   000107 60L       0001   000123 62L       0001   000135 65L       0001   000141 70L       0001   000144 75L
    0001   000151 80L       0000 I 000010 BYTE      0000 I 000003 I         0000   000012 INJP$     0000 I 000001 ISW
    0000 I 000004 J         0000 I 000002 JSW       0000 I 000007 K         0000 I 000005 NUM       0000 I 000000 017
    0000 I 000006 XTEN
```



```
00101      1*          COMPILER (FLD=R)                                                    000005
00103      2*          SUBROUTINE FD2INT (LINE, ISTCH, NOCH, VALUE, $)                     000005
00103      3*     C    CALLING SEQUENCE:                                                   000005
00103      4*     C      CALL FD2INT (LINE, ISTCH, NOCH, VALUE, $)                         000005
00103      5*     C    WHERE:                                                              000005
00103      6*     C      LINE   = FIRST WORD ADDRESS OF THE SOURCE LINE.                   000005
00103      7*     C      ISTCH  = STARTING CHARACTER POSITION IN THE SOURCE LINE.          000005
00103      8*     C      NOCH   = NO. OF CHARCTERS TO BE CONVERTED TO AN INTEGER.          000005
00103      9*     C      VALUE  = LOCATION WHERE THE INTEGER VALUE IS TO BE STORED.        000005
00103     10*     C      $      = RETURN TAKEN WHEN A CHARACTER OTHER THAN A +, -, BLANK   000005
00103     11*     C                 OR NUMERIC IS DETECTED.                               000005
00103     12*     C                                                                       000005
00105     13*          IMPLICIT INTEGER (A-Z)                                             000005
00106     14*          DIMENSION LINE(14)                                                 000005
00107     15*          DATA 017/017/                                                      000005
00111     16*          ISW = 1                                                            000005
00112     17*          JSW = 1                                                            000007
00113     18*          I = ISTCH/6 + 1                                                    000010
00114     19*          IF (MOD (ISTCH, 6).EQ.0) I = I - 1                                 000014
00116     20*          J = 41 - MOD (ISTCH, 6)*6                                          000024
00117     21*          IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                000033
00121     22*          NUM = 0                                                            000040
00122     23*          XTEN = 0                                                           000041
00123     24*          DO 100 K = 1, NOCH                                                 000044
00126     25*          BYTE = FLD (J, 6, LINE(I))                                         000046
00127     26*          IF (BYTE.NE.6H@@@@@@ ) GO TO 50                                   000055
00131     27*          BYTE = 0                                                           000057
00132     28*          GO TO 75                                                           000060
00133     29*       50 IF (BYTE.GE.6H@@@@@0.AND.BYTE.LE.6H@@@@@9) GO TO 70                000062
```

```
00135     30*          GO TO (60, 65), ISW                                  000077
00136     31*       60 IF (BYTE.EQ.6H@@@@@+) GO TO 62                       000107
00140     32*          IF (BYTE.NE.6H@@@@@-) RETURN 5                       000111
00142     33*          JSW = 2                                              000120
00143     34*       62 IF (NUM.GT.0) RETURN 5                               000123
00145     35*          ISW = 2                                              000131
00146     36*          GO TO 80                                             000133
00147     37*       65 RETURN 5                                             000135
00150     38*       70 BYTE = AND (017, BYTE)                               000141
00151     39*       75 NUM = XTEN + BYTE                                    000144
00152     40*          XTEN = NUM*10                                        000146
00153     41*       80 J = J - 6                                            000151
00154     42*          IF (J.GT.0) GO TO 100                                000153
00156     43*          J = J + 36                                           000155
00157     44*          I = I + 1                                            000160
00160     45*      100 CONTINUE                                             000164
00162     46*          GO TO (120, 110), JSW                                000164
00163     47*      110 NUM = -NUM                                           000174
00164     48*      120 VALUE = NUM                                          000176
00165     49*          RETURN                                               000177
00166     50*          END                                                  000224
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.INT2FD
FOR SE2C-10/31/75-17:34:43 (0,)

   SUBROUTINE INT2FD    ENTRY POINT 000513
              JNT2FD    ENTRY POINT 000526

   STORAGE USED: CODE(1) 000541; DATA(0) 000040; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   FILLCH 000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004   NERR2$
    0005   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000131 100L      0001   000147 125L      0001   000060 133G      0001   000205 150L      0001   000220 170G
    0001   000266 200L      0001   000432 240G      0001   000270 300L      0001   000302 325L      0001   000316 350L
    0001   000347 365L      0001   000354 375L      0001   000374 400L      0001   000455 500L      0001   000005 90L
    0000 I 000001 BNKSW     0000 I 000006 BYTE      0000 I 000002 I         0003 I 000000 IFILL     0000   000013 INJP$
    0000 I 000004 INTSV     0000 I 000010 ISW       0000 I 000003 J         0000 I 000005 K         0000 I 000007 NOCH$1
    0000 I 000000 060
```

```
00101      1*          COMPILER (FLD=R)                                                    000000
00103      2*          SUBROUTINE INT2FD (INT, NOCH, FIELD, ISTCH)                         000000
00103      3*     C    CALLING SEQUENCE:                                                   000000
00103      4*     C       CALL INT2FD (INT, NOCH, FIELD, ISTCH)                            000000
00103      5*     C    WHERE:                                                              000000
00103      6*     C       INT    - INTEGER VALUE TO BE CONVERTED TO FIELD DATA.           000000
00103      7*     C       NOCH   - NUMBER OF CHARACTERS.                                  000000
00103      8*     C       FIELD  - FIELD THAT THE FIELD DATA INTEGER IS TO BE PLACED IN.  000000
00103      9*     C       ISTCH  - STARTING CHARACTER POSITION IN THE FIELD THAT THE      000000
00103     10*     C                   FIELD DATA INTEGER IS TO BE PLACED IN.           000000
00103     11*     C                                                                       000000
00105     12*          IMPLICIT INTEGER (A-Z)                                             000000
00106     13*          DIMENSION FIELD(1)                                                 000000
00107     14*          DATA 060/060/                                                      000000
00111     15*          COMMON /FILLCH/ IFILL                                              000000
00112     16*          DATA IFILL/6H@@@@@ /                                               000000
00114     17*          BNKSW = 1                                                          000000
00115     18*          GO TO 90                                                           000001
00116     19*          ENTRY JNT2FD (INT, NOCH, FIELD, ISTCH)                             000002
00120     20*          BNKSW = 2                                                          000002
00120     21*     C    CALCULATE POINTERS TO THE END OF THE FIELD DATA FIELD.             000002
00121     22*       90 I = (ISTCH + NOCH - 1)/6 + 1                                       000005
00122     23*          IF (MOD ((ISTCH + NOCH - 1), 6).EQ.0) I = I - 1                    000013
00124     24*          J = 41 - MOD ((ISTCH + NOCH - 1), 6)*6                             000026
```

```
00125     25*          IF (MOD ((ISTCH + NOCH - 1), 6).EQ.0) J = J - 36                    000037
00127     26*          INTSV = INT                                                         000049
00130     27*          IF (INTSV.LT.0) INTSV = -INTSV                                      000096
00132     28*          DO 100 K = 1, NOCH                                                  000060
00135     29*          BYTE = MOD (INTSV, 10)                                              000065
00136     30*          INTSV = INTSV/10                                                    000071
00137     31*          FLD (J, 6, FIELD(I)) = OR (060, BYTE)                               000075
00140     32*          J = J + 6                                                           000114
00141     33*          IF (J.LE.35) GO TO 100                                              000117
00143     34*          J = J - 36                                                          000122
00144     35*          I = I - 1                                                           000125
00145     36*        100 CONTINUE                                                          000131
00147     37*          IF (NOCH.GT.1) GO TO 125                                            000131
00147     38*    C     NO. OF CHARACTERS = 1. JUMP IF THE ORIGIONAL NO. IS NEGATIVE.       000131
00151     39*          IF (INT.LT.0) GO TO 400                                             000135
00151     40*    C     JUMP, IF THE WORKING NO. HAS NOT BEEN REDUCED TO ZERO.              000135
00153     41*          IF (INTSV.GT.0) GO TO 400                                           000140
00155     42*          RETURN                                                              000143
00155     43*    C     CALCULATE THE POINTERS TO THE BEGINNING OF THE FIELD DATA FIELD.    000143
00156     44*        125 I = ISTCH/6                                                       000147
00157     45*          IF (MOD (ISTCH, 6).NE.0) I = I + 1                                  000151
00161     46*          J = 41 - MOD (ISTCH, 6)*6                                           000161
00162     47*          IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                 000170
00164     48*          GO TO (150, 375), BNKSW                                             000175
00164     49*    C     REPLACE LEADING ZEROS WITH BLANKS.                                  000175
00165     50*        150 NOCHM1 = NOCH - 1                                                 000205
00166     51*          ISW = 1                                                             000207
00167     52*          DO 200 K = 1, NOCHM1                                                000220
00172     53*          IF (FLD (J, 6, FIELD(I)).NE.6H@@@@@0) GO TO 300                     000223
00174     54*          ISW = 2                                                             000233
00175     55*          FLD (J, 6, FIELD(I)) = IFILL                                        000237
00176     56*          J = J - 6                                                           000252
00177     57*          IF (J.GT.0) GO TO 200                                               000255
00201     58*          J = J + 36                                                          000257
00202     59*          I = I + 1                                                           000262
00203     60*        200 CONTINUE                                                          000270
00205     61*        300 IF (INTSV.GT.0) GO TO 400                                         000270
00207     62*          GO TO (350, 325), ISW                                               000272
00210     63*        325 J = J + 6                                                         000302
00211     64*          IF (J.LE.35) GO TO 350                                              000304
00213     65*          J = J - 36                                                          000307
00214     66*          I = I - 1                                                           000312
00215     67*        350 IF (INT.GE.0) GO TO 365                                           000316
00217     68*          IF (FLD (J, 6, FIELD(I)).NE.IFILL) GO TO 400                        000320
00221     69*          FLD (J, 6, FIELD(I)) = 6H@@@@@-                                     000332
00222     70*        365 IFILL = 6H@@@@@                                                   000347
00223     71*          RETURN                                                              000350
00224     72*        375 IF (INT.GE.0) GO TO 365                                           000354
00224     73*    C     JUMP TO @**@ FILL, IF LEADING DIGIT IS NOT ZERO.                    000354
00226     74*          IF (FLD (J, 6, FIELD(I)).NE.6H@@@@@0) GO TO 400                     000356
00226     75*    C     REPLACE THE LEADING ZERO WITH A MINUS SIGN.                         000356
00230     76*          RETURN                                                              000370
00230     77*    C     NO. IS TOO LARGE TO FIT IN THE NO. OF CHARACTER SPACES.             000370
00231     78*        400 I = ISTCH/6                                                       000374
00232     79*          IF (MOD (ISTCH, 6).NE.0) I = I + 1                                  000376
00234     80*          J = 41 - MOD (ISTCH, 6)*6                                           000406
00235     81*          IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                 000415
```

```
00237     82*          DO 500 K = 1, NOCH                                  000432
00242     83*          FLD (J, 6, FIELD(I)) = 6H*******                    000437
00243     84*          J = J - 6                                           000452
00244     85*          IF (J.GT.0) GO TO 500                               000455
00246     86*          J = J + 36                                          000457
00247     87*          I = I + 1                                           000462
00250     88*     500 CONTINUE                                             000466
00252     89*          RETURN                                              000466
00253     90*          END                                                 000540
END FOR
```

```
*FOR,S FD5-L75036*CALCUR.MS2DAY
FOR SE2C-10/31/75-17:34:49 (0,)

   SUBROUTINE MS2DAY    ENTRY POINT 000076

   STORAGE USED: CODE(1) 000124; DATA(0) 000020; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR4$
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000011 10L       0000 I 000000 IA        0000 I 000001 IB        0000   000010 INJP$


00101      1*            SUBROUTINE MS2DAY(ITIM,IDAY,JHRS,JMIN,SEC,$)                    000000
00101      2*     C         CONVERT MILLISECONDS TO DDD:HH:MM:SEC                        000000
00103      3*             IA  =  ITIM                                                     000000
00104      4*             IB  =  0                                                        000001
00105      5*             IF(IA.GE.0) GO TO 10                                            000002
00107      6*             IB  =  1                                                        000004
00110      7*             IA  =  IABS(IA)                                                 000006
00111      8*      10     CONTINUE                                                        000011
00112      9*            IDAY = IA/86400000                                              000011
00113     10*            JHRS = MOD(IA,86400000)/3600000                                 000014
00114     11*            JMIN = MOD(IA,3600000)/60000                                    000023
00115     12*            SEC = MOD(IA,60000)/1000.                                       000032
00116     13*             IF(IB.EQ.0) RETURN                                             000042
00120     14*            IDAY=-IDAY                                                      000047
00121     15*             JHRS  =  -  JHRS                                               000051
00122     16*             JMIN  =  -JMIN                                                 000053
00123     17*             SEC   =  -SEC                                                  000055
00124     18*            RETURN 6                                                        000057
00125     19*            END                                                             000123
END FOR
```

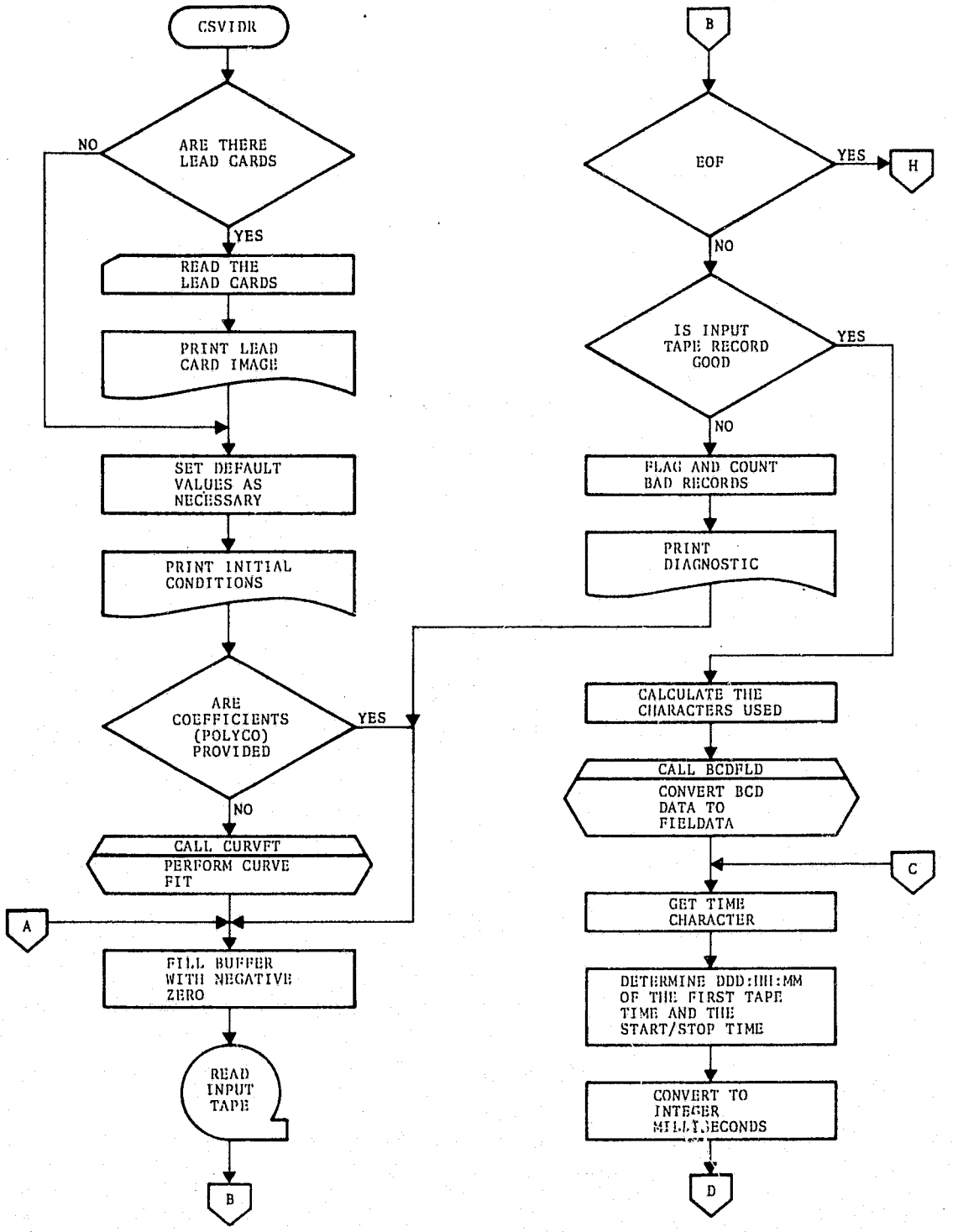
```
*FOR,S FD5-L75036*CALCUR.SCHCHR
FOR SE2C-10/31/75-17:34:56 (0,)

   SUBROUTINE SCHCHR    ENTRY POINT 000126

   STORAGE USED: CODE(1) 000146; DATA(0) 000022; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR4$
    0004   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000067 100L       0001   000040 115G       0001   000100 150L       0000 I 000000 I          0000   000004 INJP$
    0000 I 000001 J          0000 I 000002 K
```

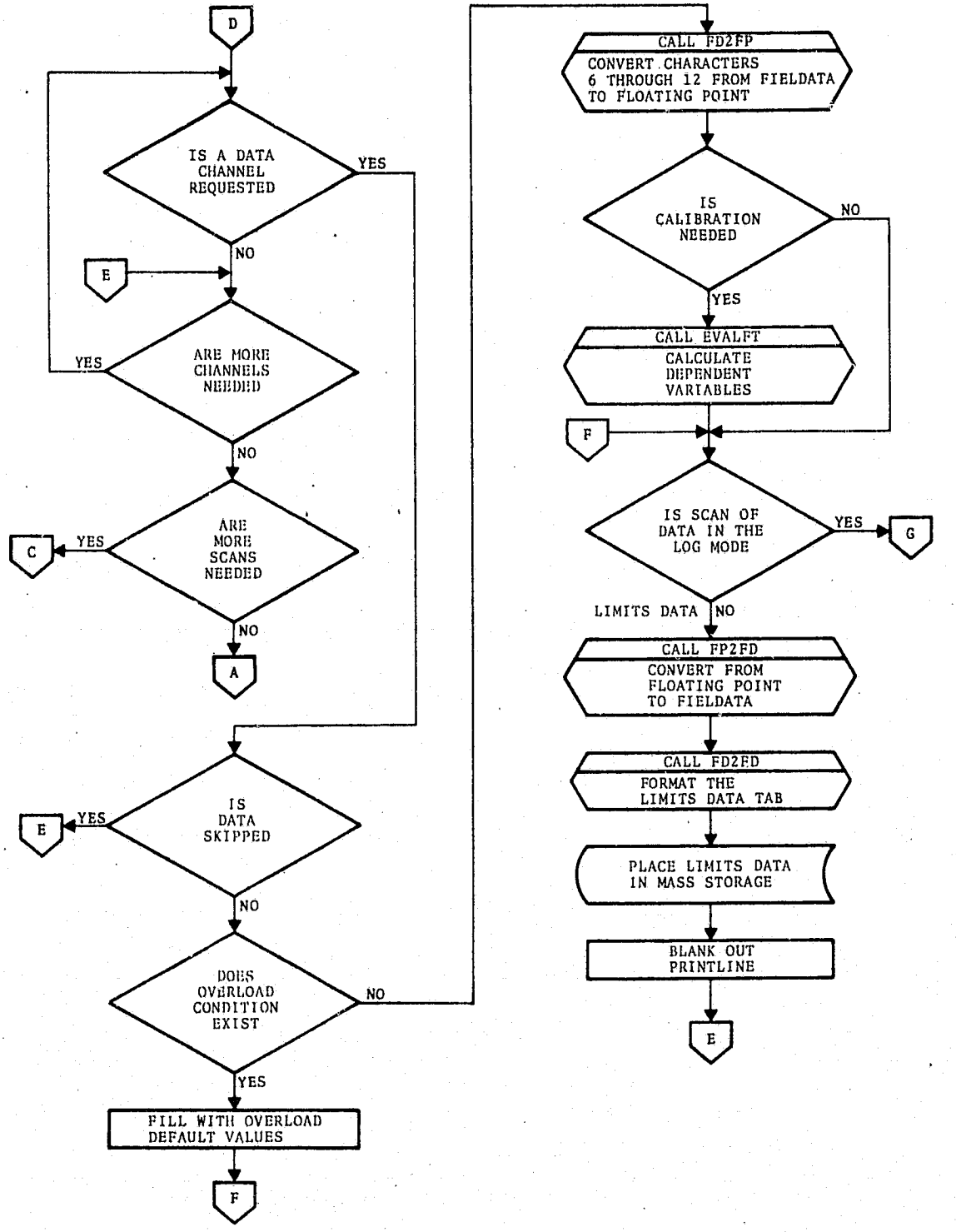
```
00101      1*            COMPILER (FLD=R)                                                    000005
00103      2*            SUBROUTINE SCHCHR (IFIELD, ISTCH, NOCH, ICHAR, NTHCHR, $)           000005
00103      3*      C                                                                         000005
00103      4*      C     CALLING SEQUENCE:                                                   000005
00103      5*      C        CALL SCHCHR (IFIELD, ISTCH, NOCH, ICHAR, NTHCHR, $)              000005
00103      6*      C     WHERE:                                                              000005
00103      7*      C     IFIELD - THE FIELD TO BE SEARCHED FOR A COMPARE.                    000005
00103      8*      C     ISTCH  - THE STARTING CHARACTER POSITION IN IFIELD WHERE THE        000005
00103      9*      C                 SEARCH IS TO BEGIN.                                     000005
00103     10*      C     NOCH   - THE NO. OF CHARACTERS BEGINNING IN IFIELD AT CHARACTER     000005
00103     11*      C                 POSITION ISTCH TO SEARCH.                               000005
00103     12*      C     ICHAR  - THE CHARACTER FOR WHICH IFIELD IS TO BE SEARCHED FOR.      000005
00103     13*      C     NTHCHR - IF A COMPARISION IS MADE, THE CHARACTER POSITION IN        000005
00103     14*      C                 IFIELD WHERE THE COMPARISION WAS MADE IS STORED.        000005
00103     15*      C     $      - THE STATEMENT RETURN IF A COMPARISION WAS NOT MADE.        000005
00103     16*      C     NORMAL RETURN.                                                      000005
00103     17*      C                                                                         000005
00105     18*            DIMENSION IFIELD(1)                                                 000005
00106     19*            I = ISTCH/6 + 1                                                     000005
00107     20*            IF (MOD (ISTCH, 6).EQ.0) I = I - 1                                  000011
00111     21*            J = 41 - MOD (ISTCH, 6)*6                                           000021
00112     22*            IF (MOD (ISTCH, 6).EQ.0) J = J - 36                                 000030
00114     23*            DO 100 K = 1, NOCH                                                  000040
00117     24*            IF (FLD (J, 6, IFIELD(I)).EQ.ICHAR) GO TO 150                       000043
00121     25*            J = J - 6                                                           000053
00122     26*            IF (J.GT.0) GO TO 100                                               000056
00124     27*            J = J + 36                                                          000060
00125     28*            I = I + 1                                                           000063
00126     29*        100 CONTINUE                                                            000070
00130     30*            NTHCHR = ISTCH + NOCH                                               000070
00131     31*            RETURN 6                                                            000073
00132     32*        150 NTHCHR = (I - 1)*6 + (41 - J)/6                                     000100
00133     33*            RETURN                                                              000107
```

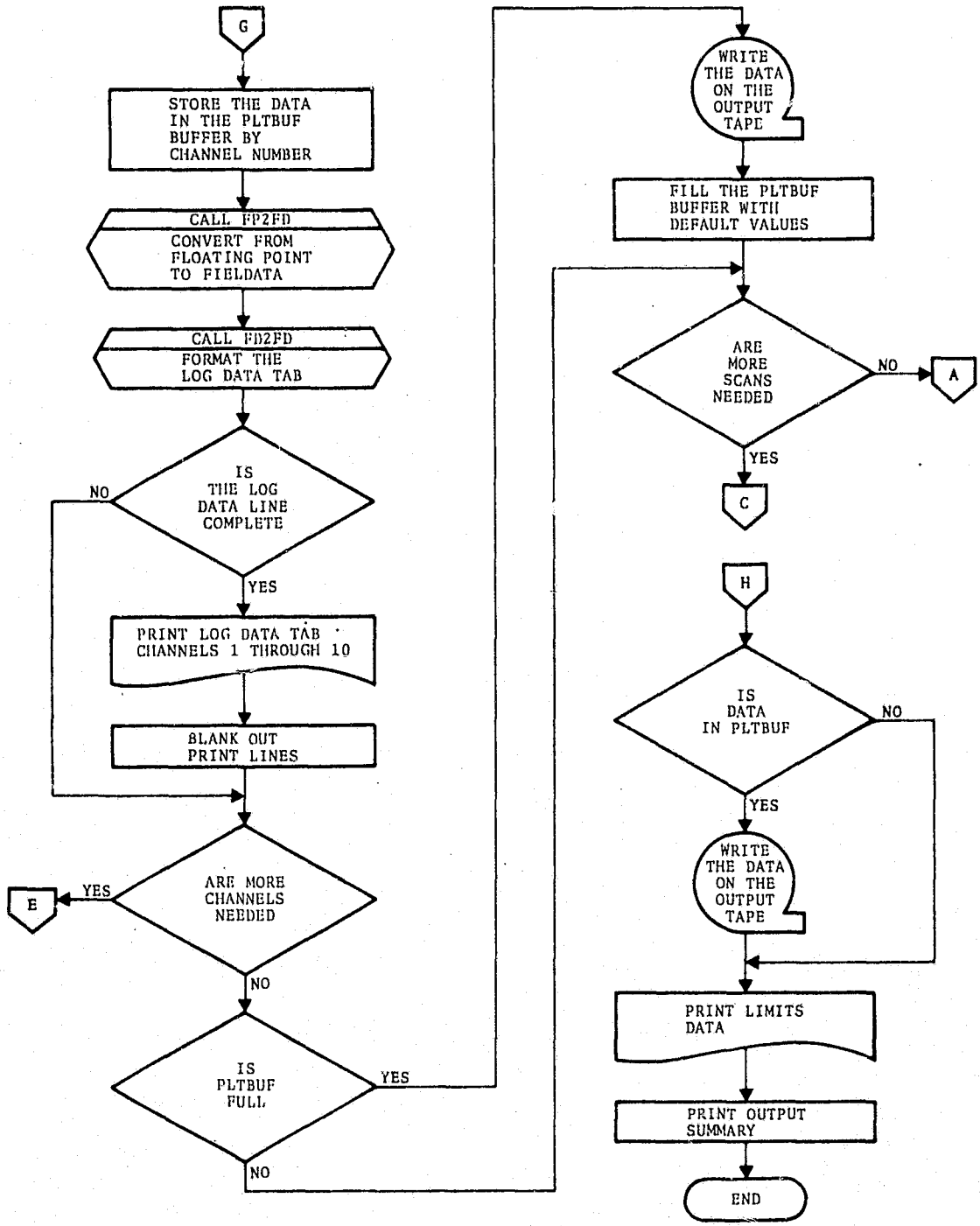
```
00139      39*          END                                          000145
END FOR

*BRKPT PRINTS
```

APPENDIX II

FLOW CHARTS




D
IS A DATA CHANNEL REQUESTED
YES
NO
E
ARE MORE CHANNELS NEEDED
YES
NO
ARE MORE SCANS NEEDED
C
YES
NO
A
IS DATA SKIPPED
E
YES
NO
DOES OVERLOAD CONDITION EXIST
NO
YES
FILL WITH OVERLOAD DEFAULT VALUES
F
CALL FD2FP
CONVERT CHARACTERS 6 THROUGH 12 FROM FIELDATA TO FLOATING POINT
IS CALIBRATION NEEDED
NO
YES
CALL EVALFT
CALCULATE DEPENDENT VARIABLES
F
IS SCAN OF DATA IN THE LOG MODE
YES
G
LIMITS DATA
NO
CALL FP2FD
CONVERT FROM FLOATING POINT TO FIELDATA
CALL FD2FD
FORMAT THE LIMITS DATA TAB
PLACE LIMITS DATA IN MASS STORAGE
BLANK OUT PRINTLINE
E



APPENDIX I

CORRESPONDENCE

THIS FORM MUST BE COMPLETED BY TYPEWRITER

01 4	01 7 PROGRAM NO	JSC COMPUTER PROGRAM ABSTRACT	01 14 DATE (MMDDYY)
JSC	Q939		12-3-75

01 20 TITLE OF PROGRAM (62 CHARACTERS MAXIMUM)	01 72 SYMBOLIC NAME (9 CHARACTERS MAXIMUM)	PARENT PROGRAM		
		02 14 CATEGORY	02 16 SITE	02 19 PROGRAM NO
CREW SYSTEMS VIDAR DATA REDUCTION	CSVIDR			

02 26 CATEGORY	02 27 LANGUAGE NO. 1	02 32 LANGUAGE NO. 2	02 37 KEY WORDS (8 MAXIMUM SEPARATED BY COMMAS)
Q	FOR V		Data Reduction, Crew Systems, UNIVAC EXEC 8, Calibration

WHOM TO CONTACT ABOUT THE PROGRAM					05 48 STATUS	05 49
05 14 CONTACT (LAST NAME)	05 28 SITE	05 31 ORGN CODE	05 39 PROJECT NO	05 45 NASA CENTER	☐ A. UNDER DEVELOPMENT ☐ B. OPERATIONAL ☒ C. COMPLETED	☐ A. THIS PROGRAM IS NOT FOR SHARING ☐ B. LIMITED SHARING (SEE ABSTRACT)
W. R. Lacy	JSC	FD5	4201			

DATES		05 58 REVISION CODE	TIME AND COST FOR DEVELPOMENT			
05 50 INITIATED MMYY	05 54 COMPLETED MMYY		05 59 MAN-MONTHS	05 64 MACHINE HOURS	05 69 COMPUTER TYPE	05 74 TOTAL COST (DOLLARS)
0975	1275	☐ A. REVISION ☐ B. CANCELLATION	3.0 (59 60 61 62 63)	2.0 (64 65 66 67 68)		. (74 75 76 77 78 79 80)

CARD NUMBER	COLS.	ABSTRACT	ELITE MARGIN	PICA MARGIN
06		CSVIDR is a special purpose data reduction program written		
07		to retrieve, calibrate and tabulate data acquired from the		
08		Vidar Autodata Eight Processing System located in Building		
09		7.		
10				
11		CSVIDR is designed to require less than 20K core and uses		
12		mass storage to reduce use of tape drives and permit optimum		
13		access to the UNIVAC EXEC VIII system.		
14				
15		The input tape format is 6 bit BCD. All data is converted		
16		from BCD to field data since the majority of the data are		
17		output without further conversion.		
18				
19		Optional output is a fixed sample rate tape containing scans		
20		of time with calibrated data.		
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31				
32				
33				
34				
35		ORIGINAL PAGE IS		
36		OF POOR QUALITY		
37				
38				
39				
40				
41				

RELATED DOCUMENTATION (66 CHARACTERS MAXIMUM SEPARATE EACH REF BY COMMAS)	
42	Q417, FIT73, J252, BIOFIT

JSC Form 142 (Aug 73) NASA-JSC

NASA/LEC TASK AGREEMENT

Job Order 83-157

Title: Software Development for Vidar Data Acquisition System

T. A. No.: 20

TA #20

Approval:

C. E. Hutchinson — 8/18/75
C. E. Hutchinson
Supervisor, Engineering Test Data Processing Section

Date

Walter D. Poates — 9-3-75
W. D. Poates
NASA Technical Monitor
Data Processing Branch

Date

W. N. Fitzpatrick — 8/18/75
W. N. Fitzpatrick
Manager, Data Processing Systems Department

Date

F. Fulton — 9/5/75
F. Fulton, Chief
Data Processing Branch

Date

1.0 IDENTIFICATION

Title: Software Development for Vidar Data Acquisition System

T.A. No.: 20

T.O./J.O.: 83-157

2.0 RESPONSIBILITIES

Lockheed Electronics Company will gather the requirements and provide the analysis and programming associated with this task.

3.0 DESCRIPTION

3.1 Task Purpose and Scope

The purpose of this task is to develop the capability of processing data recorded at the Crew Systems Test Facility in Building 7. Data from long-term tests of Shuttle Crew Equipment will be recorded on 7-track computer compatible tapes by the Vidar Autodata Processing System. The capability will be developed to tabulate selected parameters and to generate a fixed sample rate tape from which plots can be made.

3.2 Planned Approach

A new program will be written to retrieve, calibrate and tabulate the data. A fixed sample rate tape will also be generated and used as input to a general purpose plot program. The new program will be written in FORTRAN to be run on the UNIVAC 1108/1110 computers and will use the EXEC 8 operating system.

Specifications for the input tape format, calibrations to be made, and program outputs are attached.

4.0 SCHEDULES AND MILESTONES

Program Development

Oral design review	8/28/75
Coding and checkout complete	9/24/75
Validation complete	10/1/75

Program Documentation

To Tech Pubs	10/15/75
To NASA (Review)	10/29/75
Publication	11/17/75

5.0 RESOURCES

Estimated resources required are:

Man-Hours

Analysis	80
Programming	240
Validation	20
Tech Pubs	50
	390

Computer Hours

Univac 1110/1108 EXEC 8 - 3.0 hrs.