5 9/5/63
4 9

LAMS-2845

MASTER

LOS ALAMOS SCIENTIFIC LABORATORY

OF THE UNIVERSITY OF CALIFORNIA ○ LOS ALAMOS NEW MEXICO

COMPUTER DATA PROCESSING SYSTEM

PROJECT ROVER

1962

Facsimile Price $ 12.50

Microfilm Price $ 5.45

Available from the
Office of Technical Services
Department of Commerce
Washington 25, D. C.

DISCLAIMER

LEGAL NOTICE

LAMS-2845
UC-32, MATHEMATICS
AND COMPUTERS
TID-4500 (19th Ed.)

LOS ALAMOS SCIENTIFIC LABORATORY

OF THE UNIVERSITY OF CALIFORNIA LOS ALAMOS NEW MEXICO

REPORT WRITTEN: January 15, 1963

REPORT DISTRIBUTED: March 13, 1963

COMPUTER DATA PROCESSING SYSTEM

PROJECT ROVER

1962

by

Francis Narin

Contract W-7405-ENG. 36 with the U. S. Atomic Energy Commission

Blank

Abstract

A system has been created for processing at the Los Alamos Scientific Laboratory large volumes of data from Project ROVER tests at the Nevada Test Site. The data are brought to Los Alamos as analogue, frequency modulated tape, which is translated in a Packard-Bell Tape-to-Tape converter into a binary coded decimal (BCD) IBM 7090 computer input tape. This input tape, tape A5, is processed on the 7090 by the RDH-D FORTRAN-II code and its 20 FAP and FORTRAN subroutines. Outputs from the 7090 run are tapes A3, which is a BCD tape used for listing on the IBM 1401 input-output computer, tape B5 which is a binary tape used as input to a Stromberg-Carlson 40/20 cathode ray tube (CRT) plotter, and tape B6 which is a binary tape used for permanent data storage and input to specialized subcodes. The information on tape B5 commands the 40/20 to write grids, data points and other information on the face of a CRT; the information on the CRT is photographed on 35 mm film which is subsequently developed; full-size (10" x 10") plots are made from the 35 mm film on a Xerox 1824 printer. The 7090 processes a data channel

(each individual measurement is a data channel) in approximately 4 seconds plus 4 seconds per plot to be made on the 40/20 for that channel. Up to 4500 data and calibration points on any one channel may be processed in one pass of the RDH-D code. This system has been used to produce more than 100,000 prints on the 1824 printer from more than 10,000 different 40/20 plots. At $6.00 per minute of 7090 time, it costs $1.60 to process a typical, 3 plot data channel on the 7090; each print on the 1824 costs between 5 and 10 cents including rental, supplies, and operator time. All automatic computer stops in the codes and subroutines are accompanied by on-line instructions to the operator. Extensive redundancy checking is incorporated in the FAP tape handling subroutines.

Acknowledgments

The contributions and efforts of a number of people were very helpful in the programming and debugging of this data processing system. Whenever possible, the individual coder responsible for each code is named; in addition, the advice and assistance of Messrs. Bob Hahn, Art Crosby and Connie Brummett of Edgerton, Germeshausen and Grier, Inc., (EGG) and Ivan Cherry of Group T-1, Los Alamos Scientific Laboratory, were particularly helpful in the writing and debugging of the system as a whole.

TABLE OF CONTENTS

Page

List of Figures

Blank

I

General Description - 1962 Data Processing System

A. INTRODUCTION

The Project Rover computer data processing system was designed to process at LASL large amounts of data from the Nevada Test Site with a minimum of manual data handling, consistent with the state-of-the-art and budgetary requirements. Figure 1 is a block diagram of the overall data processing system. The heart of the system is the IBM 7090 computer which converts a raw data tape and a set of calibration, control and labeling data into three output tapes, which are used for input to the Stromberg-Carlson 40/20 plotter (Tape B5), for generating listings through the IBM 1401 input-output computer (Tape A3), and for permanent data storage and input to specialized sub-codes (Tape B6). As in most data processing problems of this type, the data system and the computer program are "input-output" limited; that is, the amount of time spent in doing the computational steps is quite small when compared to the time spent getting the data into and out of the computer. In a production line type of operation approximately 30 seconds of manual operations are

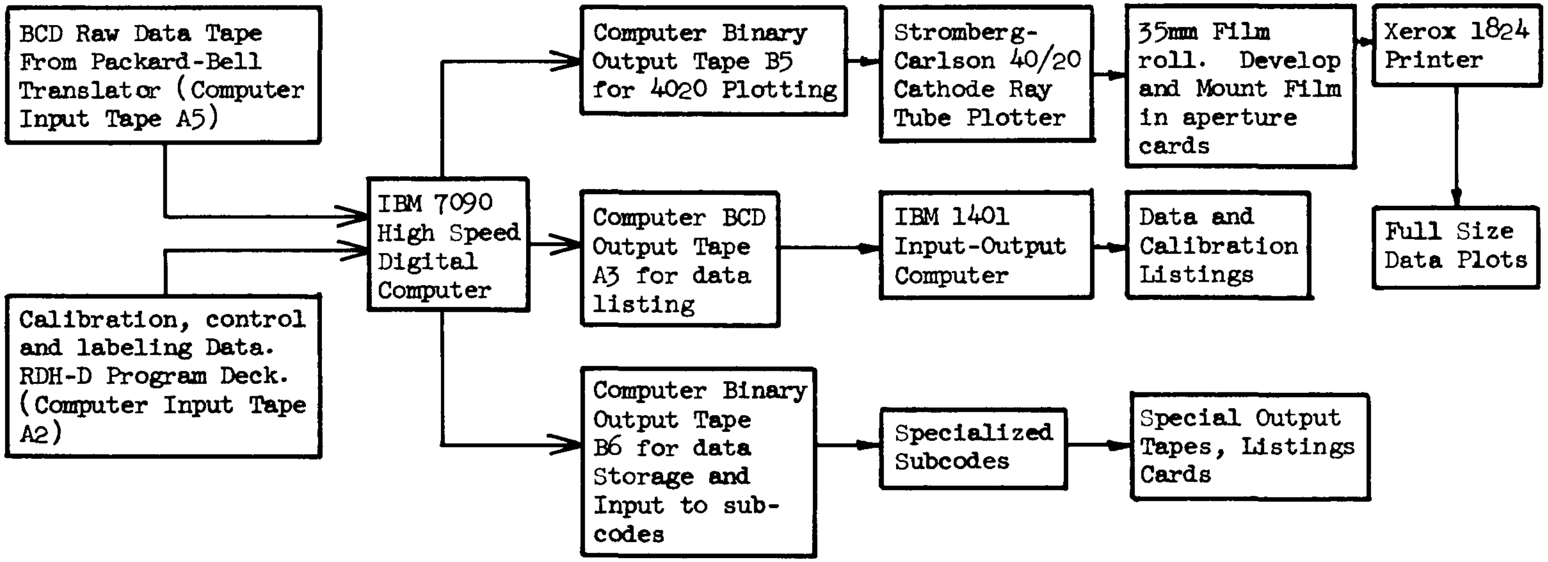


FIGURE 1

BLOCK DIAGRAM

OVERALL DATA PROCESSING SYSTEM

required to produce each final data plot; of this 30 seconds, roughly 1 second is in 7090 computation, 4 seconds in 7090 input-output, 5 seconds in miscellaneous handling of tapes and cards and in film developing and mounting, and 20 seconds in making the print in the Xerox 1824 Printer.

B. TAPE A5, THE RAW DATA TAPE

The 7090 input tape designated as tape A5 is a low density (200 bit per inch), binary coded decimal raw data tape produced in a Packard-Bell Tape-to-Tape converter. Each record on the tape corresponds to one frame from a 10, 20, or 40 frame per second multiplexer at the Nevada Test Site. On each record of this tape are time (6 digits), multiplexer frame full scale (3 digits), multiplexer frame zero scale (3 digits) and 1 to 47 data points (3 digits per point), each from a different data channel. The main data code, RDH-D can process up to 4500 records on a single file of tape A5 in any one pass through the 7090. There may be any number of files on tape A5 and any number of records in each file. There are time editing capabilities in the translator whereby only every nth multiplexer frame appears as a tape record, where n ranges from 1 to 256. The following is a set of typical 20 frame per second multiplexer records with 3 channels:

Time	Full Scale	Zero Scale	Channel 1	Channel 2	Channel 3
296018	985	015	115	210	301
296018	984	014	120	215	306
296018	985	016	125	218	315
296018	983	014	130	221	325
296020	983	013	135	224	330
296020	984	015	140	227	326
296020	986	017	145	231	320
296020	985	015	150	235	310
296022	982	013	155	237	306
296022	984	014	160	240	298

Note that a time of 296018 is 29601.8 seconds, and that the time changes only every 0.2 seconds. Therefore, a correction must be put in for the position of a frame in a set of frames with identical time. For example, the actual times of the first four frames in the example are: 29601.8, 29601.85, 29601.9, and 29601.95. This time correcting is done in the data codes internal to the CUST converting, unpacking and time-correcting subroutine. There is a parallel subroutine CUS which converts and unpacks tape A5 but does no time correcting. CUS and CUST (which uses a subroutine TIC for the time correcting) have identical calling sequences and may be used interchangeably with the restriction that CUST requires the TIC subroutine. The output of CUS or CUST is a binary, high density, unpacked and normalized tape B1 from which further data processing is done. Note that a normalized data point in this sense is defined by

$$\text{Normalized Data Point} = \text{NDP} = \frac{\text{Data Point - Frame Zero Scale}}{\text{Frame Full Scale - Frame Zero Scale}}$$

For each multiplexer of data being processed there is normally one file on tape A5.

C. CONVERSION OF RAW DATA TO REDUCED DATA

1. Methods of Applying Calibration Data

a. NCS = 0; no calibration case: In the case where there is no calibration on the input tape, specified by setting NCS = 0 for a given channel, a final data point FDP is generated from the normalized data point by means of the equation

$$FDP = \sum_{i=0}^{5} B_i (NDP)^i \quad .$$

Here, the coefficients B_i are computer input data.

b. NCS = 1; direct calibration case: In the case where there is a direct calibration on the input tape, specified by setting NCS = 1 for a given channel, a final data point is generated from a least square fit of the calibration steps. First, an average value AVG for the (normalized) amplitude of each calibration step is determined by averaging the central 80 percent of the span of each calibration step (or the first 50 points of the central 80 percent should there be more than 50 points). Then a table is formed of VTCS, the given value corresponding to each of the tape calibration steps, <u>in units of the physical variable</u>, versus AVG. Finally, by means of the LSF subroutine, a least square fit is done on the set of points (AVG, VTCS) to determine the coefficients B_i in the expression:

$$FDP = \sum_{i=0}^{5} B_i(NDP)^i.$$

Note that the fit can be from first to fifth power.

c. NCS = 2; voltage calibration plus transducer calibration case.

In the case where there is a voltage calibration on the tape, specified by setting NCS = 2 for a given channel, which is to be combined with a known transducer calibration curve, there is fed into the 7090 the voltage VTCS corresponding to each of the tape calibration steps, plus a set of coefficients B_i relating voltage V from the transducer with Y, the value of the physical variable. First the code finds the average values of the calibration steps and forms the table (AVG, VTCS). Next these values of VTCS are fed in as voltages, V, into the equation

$$Y = \sum_{i=0}^{5} B_i(V)^i$$

to generate a table (AVG, Y). Finally, by means of the LSF subroutine a least square fit is done to the set of points (AVG, Y) to determine a new set of coefficients B_i in the expression:

$$Y = FDP = \sum_{i=0}^{5} B_i (NDP)^i \quad .$$

2. Automatic Calibration Analysis

A simple calibration analysis is performed automatically for all channels for which there is a calibration (NCS = 1 or 2). First, an average for each calibration step is formed according to

$$AVG = \frac{1}{N} \sum_{i=1}^{N} NDP_i \qquad 1 \leqq N \leqq 50.$$

Next, a standard deviation for the step is formed according to

$$SIG = \sqrt{\frac{\sum_{i=1}^{N} (AVG-NDP_i)^2}{N}} \qquad 1 \leqq N \leqq 50.$$

In addition, a maximum deviation is formed from

$$DEVMAX = \text{Maximum of } |AVG - NDP_i| \text{ over } i = 1, N.$$

Finally, all the step averages are least square fitted to a straight line and the standard deviation of this fit, expressed as a percent of full scale, is determined. If this standard deviation is greater than an input number TSTSG, the channel is not processed. This automatically provides a linearity check on the channel calibration, providing TSTSG is reasonable. As an additional feature, all of the points used in the calibration analysis are written on tape B5 for plotting on the 40/20, at 5 channels of calibration data per 40/20 plot. Through the use of an input number NSIG the TSTSG test can be nullified for an individual channel.

D. IBM 7090 RUN

For a typical data run, the RDH-D code processes a data channel in N seconds, where N is given by

$$N = 4 + 4 \cdot \text{NSI seconds}$$

where NSI = number of scaling intervals = number of 40/20 plots to be prepared from tape B5. Note that for a KIWI B-1A type run, at 3 curves per channel, a 47 channel multiplexer would be processed in

$$\sim 47(4 + 4.3) = 752 \text{ seconds} = 12.5 \text{ minutes}.$$

The following are required for a 7090 run:

IBM 7090 computer, 32 K core, Fortran II Monitor System

Printer, Card Reader

Tapes A1, A2, A3, A4, A5, B1, B2, B5, B6

E. PRODUCTION OF FINAL DATA PLOTS

Tape B5, after removal from the computer, is fed into the Stromberg-Carlson 40/20 unit and, at a rate of about 1 film frame per second, the information on B5 is printed on 35 mm film reels. After developing, the individual film frames are mounted in "D" sized aperture cards in a Filmsort semi-automatic mounter at a rate of perhaps 10 frames per minute.

These aperture cards are then inserted into a Xerox 1824 printer to produce final (10" x 10") data plots at a rate of 3 plots per minute.

F. PRODUCTION OF FINAL DATA LISTINGS

Tape A3 contains all calibrations and least-square-fitting information, and selected data points. This tape is listed on the IBM 1401 computer and then re-used.

G. TAPE B6

Tape B6 contains identifying information from each channel and all reduced data points, stored as one record for each channel processed. This tape is written by the WRTB6 subroutine and may be read by the RDB6 subroutine. There is available a sub-code B6 to A3 which simply lists the information from B6 to A3 in BCD form. There are also a cross-plotting subroutine W3XPLT for making composite plots of various channels on B6, a code for calculating liquid hydrogen flow rates, a code for doing reactor power integrations and a few other codes.

II

Annotated Listings of Codes and Subroutines, and Subroutine Usage Table

A. ANNOTATED LISTING OF CODES AND SUBROUTINES

RDH-D, FORTRAN language, length 21567_{10}, written by F. Narin. RDH-D is the main code; input to RDH-D is a raw data tape A5 and calibration, control and labeling information on tape A2; output is reduced data on tape A3 in BCD form for listing, on tape B5 in binary form for input to the Stromberg-Carlson 40/20 cathode ray tube plotter and on tape B6 in binary form for permanent storage and input to subcodes.

CAN-D, FORTRAN language, length 21609_{10}, written by F. Narin. CAN-D is an auxiliary code which performs an analysis of the calibration on a raw data tape, checking each calibration step for noise and the calibration as a whole for linearity.

B6TØA3, FORTRAN language, length 9185_{10}, written by F. Narin. B6 to A3 is an auxiliary code which takes the information in permanent, binary storage on tape B6 and writes it on tape A3 in BCD form for listing.

CUS and CUST, FAP language,length 367_{10} and 728_{10} respectively, written by F. Narin. The CUS and CUST subroutines take the low density, packed, BCD, one record per multiplexer frame tape A5 and convert it into a high density, unpacked, binary, 200 multiplexer frame per record, normalized tape B1 for use in further data processing. In addition CUST, through the TIC subroutine, corrects the tape A5 by adding the appropriate time increments to compensate for the limited time resolution of 0.2 seconds on tape A5.

TIC, FAP language, length 388_{10}, written by T. Springer. The TIC subroutine is called by the CUST subroutine and replaces a list of consecutive floating point times in core storage and limited to 0.2 second time resolution with another set of corrected times. Any times which appear to be out of order are set to zero.

RDB1, FAP language, length 139_{10}, written by F. Narin. The RDB1 subroutine brings into core storage, normalized, raw data and time stored on tape B1 by the CUS or CUST subroutines.

LSF, FORTRAN language, length 523_{10}, written by M. Blaylock, EGG. The LSF subroutine least square fits a set of up to 50 data points to a polynomial of up to the fifth power.

WRTB6, FAP language, length 77_{10}, written by F. Narin. The WRTB6 subroutine writes all processed data points and times on tape B6 for permanent storage.

RDB6, FAP language, length 134_{10}, written by F. Narin. The RDB6 subroutine brings into core storage information written on B6 by the WRTB6 subroutine.

PØSB6, FAP language, length 35_{10}, written by F. Narin. The PØSB6 subroutine positions tape B6 for adding channels to a B6 with some channels already on it.

LSTB6, FAP language, length 60_{10}, written by F. Narin. The LSTB6 subroutine writes on tape A3 a list of the channel identification information for all of the channels on tape B6.

LIST, FORTRAN language, length 50_{10}, written by F. Narin. The LIST subroutine is called by the LSTB6 subroutine and merely formats the information found by LSTB6.

PRINT, FORTRAN language, length 251_{10}, written by F. Narin. PRINT is called by CUS, CUST, LSTB6, PØSB6, RDB1, RDB6 and WRTB6 and simply prints and writes on tape A3 information that redundancy checks or other difficulties were incurred by the subroutines.

LBLFGD,FAP language, length 398_{10}, written by K. Crandall. The LBLFGD subroutine writes information on tape B5 commanding the 40/20 to label a grid in fixed point (integer) notation.

LBLGRD, FAP language, length 503_{10}, written by K. Crandall. The LBLGRD subroutine writes information on tape B5 commanding the 40/20 to label a grid in floating point notation.

ADVFLM, FAP language, length 33_{10}, written by K. Crandall. The ADVFLM subroutine writes information on tape B5 commanding the 40/20 to advance its film frame.

CLAS, FAP language, length 154_{10}, written by K. Crandall. The CLAS subroutine writes information on tape B5 commanding the 40/20 to write "Confidential - Restricted Data" on a film frame.

LINCNT. This subroutine writes information on tape B5 commanding the 40/20 to set a line count for writing BCD information with a FORTRAN "WRITE ØUTPUT TAPE 12" statement.

PLFILM, FAP language, length 305_{10}, written by K. Crandall. The PLFILM subroutine writes information on tape B5 commanding the 40/20 to plot a set of points with a specified symbol.

SELGRD, FAP language, length 352_{10}, written by L. Egbert. The SELGRD subroutine writes information on tape B5 commanding the 40/20 to write various grid lines.

WFILM, FAP language, length 189_{10}, written by K. Crandall. The WFILM subroutine writes information on tape B5 commanding the 40/20 to write BCD information either horizontally or vertically.

PØLEV, FAP language, length 28_{10}, written by T. Jordan. PØLEV evaluates the polynomial which transforms a raw data point into a reduced data point.

MAX3, FAP language, length 53_{10}, written by B. Bacon. The MAX3 subroutine determines the algebraic maximum from an array of floating point numbers.

MIN3, FAP language, length 54_{10}, written by B. Bacon. The MIN3 subroutine determines the algebraic minimum from an array of floating point numbers.

B. SUBROUTINE USAGE TABLE

The following are the subroutines used by the main data codes and the subroutines themselves, exclusive of the FORTRAN input-output routines. An "X" means the subroutine is used by the main code or its subroutines. An "O" means the subroutine is not used by the main code or its subroutines.

Subroutine	Coder	Code RDH-D	Code CAN-D	Code B6TØA3	Subroutines Used
ADVFLM	Crandall	X	X	O	-
CLAS	Crandall	X	O	O	-
CUST, (CUS)	Narin	X	X	O	PRINT, TIC
LBLFGD	Crandall	X	O	O	-
LBLGRD	Crandall	O	X	O	-
LINCNT		X	X	O	-
LSF	Blaylock,EGG	X	X	O	-
LIST	Narin	X	O	O	-
LSTB6	Narin	X	O	O	LIST, PRINT
MAX3	Bacon	X	X	O	-
MIN3	Bacon	X	X	O	-
PLFILM	Crandall	X	X	O	-
PØLEV	Jordan	X	O	O	-
PØSB6	Narin	X	O	O	PRINT
PRINT	Narin	X	X	X	-
RDB1	Narin	X	X	O	PRINT
RDB6	Narin	X	X	X	PRINT
SELGRD	Egbert	X	X	O	-
TIC	Springer	X	X	O	-
WFILM	Crandall	X	O	O	-
WRTB6	Narin	X	O	O	PRINT

III

Individual Code Write-Ups and Listings

A. RDH-D, Written by F. Narin

1. Introduction

The RDH-D (Rover Data Handling - D series) code is the main code in the data processing system. RDH-D itself handles the BCD input and output of data from the 7090 and the computations involved in transforming a raw data point into a reduced data point. In addition, it calls the various FAP subroutines which manipulate and transform blocks of data outside of the FORTRAN format restrictions. RDH-D also handles all of the routine manipulations involved in processing the data on the 7090. The following limitations are built into RDH-D:

47 channels per multiplexer

4500 data and calibration points used per channel per pass of the code

20 plotting time intervals, 500 points on any one plot

25 steps per calibration

50 data points used per calibration step

5th power least square fits to calibration data

The following tapes are used by RDH-D and its subroutines:

A1 is the utility tape (Logical tape 1)

A2 is the program and data tape (Logical tape 10)

A3 is the BCD output listing tape (Logical tape 9).

A4 is a temporary storage tape used by the CUST subroutine (Logical tape 4).

A5 is the tape unit for the Packard-Bell input tape (Logical tape 2).

B1 is a temporary storage tape written by the CUS and CUST subroutines and read by the RDB1 subroutine (Logical tape 5).

B2 is a temporary storage tape used by CUST and also by the main program (Logical tape 6).

B5 is the 40/20 output tape (Logical tape 12).

B6 is the permanent, binary, data storage tape (Logical tape 8).

A6, B3, B4 are not used. However, B4 is normally reserved for binary card output. Sense switch 6 is used to temporarily stop the data processing when tape B5 is full, in order to put on a new B5. Sense switches 1 through 5 are not used.

2. Tape A2 (Control, Calibration, Labeling) Input. The following are the formats for the RDH-D code input:

```
 1 Format (14I5)
 3 Format (12A6)
13 Format (7F10.5)
 7 Format (2F10.5,2I5)
 9 Format (48I1)
10 Format (7I5, 3F10.5, I5)
11 Format (1P7E10.5)
```

The following is the input for the RDH-D code:

1 per tape A2 (Read Input Tape 10, 1, NMPLX, IADDA5, IADVB6

```
              ┌ Read Input Tape 10, 3,(TITLE(I), I = 1, 6)
              │
              │ Read Input Tape 10, 1, NCH, NSI, NCHFM, NFMSEC, IMUC,
              │                        NFLAD, NFLBS, NRCSP, ICNFRD,
              │                        NA3SKP, NARIN
1 Set per     │
Multiplexer   │                        IMUC = IMUC + 1
              │
              │ Read Input Tape 10, 13, TMN, TMX, DELT, TSTSG
              │
              │ Read Input Tape 10, 7, (SMIN(I), SMAX(I), LXGD(I),
              │                        NXLB(I), I = 1, NSI)
              │
              └ Read Input Tape 10, 9, (ICS(I), I = 2, 48)

              ┌ DØ 185I = 1, NCH
              │
              │ Read Input Tape 10, 3 (REC(I, J),J = 1, 12)
              │
              │ Read Input Tape 10, 10, NCS(I), NTCC(I), NTCS(I), NMAX2(I),
              │                         ITAPØS(I), LYGD(I), NYLB(I), YMIN(I),
              │                         YMAX(I), PTLGMN(I), NSIG(I)
              │
              │       I3 = NCS(I) + 1    N1 = NTCC(I)     N2 = NTCS(I)
              │       GØ TØ (160, 170, 180), I3
              │
              │ 160 Read Input Tape 10, 11 (B1(I,J), J = 1, N1)
              │       GØ TØ 184
1 Set per     │
Channel       │ 170 Read Input Tape 10, 13, (VTCS(I,J),J = 1, N2)
              │
              │ 175 Read Input Tape 10, 13, (TSTCS(I,J), J=1, N2),SPAN
              │       GØ TØ 184
              │
              │ 180 Read Input Tape 10, 11, (B1(I,J), J = 1, N1)
              │     Read Input Tape 10, 13, (VTCS(I,J),J = 1, N2)
              │       GØ TØ (175, 181, 184), IMUC
              │
              │ 181 Read Input Tape 10, 13,(TSTCS(I,J), J = 1, N2), SPAN
              │       IMUC = 3
              │
              │ 184 Read Input Tape 10, 3,(VERLB(I,J), J = 1, 12)
              │
              └ 185 Read Input Tape 10, 3, (CHNM(I,J), J= 1, 12)
```

INPUT DEFINITIONS AND COMMENTS

NMPLX	Number of multiplexers to be processed on the input tape
IADDA5	= 0 if the current input tape is the last input tape, = 1 if there is an additional A5 tape.
IADVB6	= 0 if a new tape B6 is being used, = 1 if tape B6 is to be advanced and positioned before processing is started. This is only significant at the beginning of a computer run.
TITLE	= 36 columns of identifying information normally containing a. Reactor and Plan Number b. Multiplexer Number c. 7090 Run Identification For example: KIWI-BCF EP 2 LL10 7090 RUN 4/15/62 The title information appears on all (B5, B6, A3) the output tapes.
NCH	Number of channels on a multiplexer, any number between 1 and 47
NSI	Number of Scaling Intervals = number of 40/20 plots for each channel, any number between 0 and 20. Note that a. The code will plot only the last 500 points in a scaling interval b. The plots of calibration data are independent of NSI c. The code is severely output-limited by the writing of tape B5. The processing for a channel takes approximately 4 + 4 x NSI seconds
NCHFM	Number of channels in a multiplexer frame, typically 45 or 50 corresponding to 45 x 10 or 50 x 40 multiplexers
NFMSEC	Number of multiplexer frames per second, typically 10 or 40 corresponding to 45 x 10 or 50 x 40 multiplexers
IMUC	= 0 if a multiplexer is not uniformly calibrated, = 1 if a multiplexer is uniformly calibrated. If IMUC = 1 for a multiplexer, the times of the start of the calibration step (TSTCS) and the step span SPAN are read only for the first channel on the multiplexer for which NCS = 2, and are assumed the same for all other NCS = 2 channels. If IMUC = 0, times are read for all channels for which NCS = 1 or 2.
NFLAD	Number of Files (on tape A5) to advance before starting to search for data. Note that at the end of a search the tape is positioned at the beginning of the next file, so that for

a normal sequence of processing one file after the other, NFLAD = 0.

NFLBS — Number of files (on Tape A5) to backspace before starting to search for data.

NRCSP — Number of records to skip (on Tape A5) after finding a good record. This feature permits the time editing of the Packard-Bell tape on line. However, on-line time editing is an expensive waste of computer time; therefore, this should be used only in exceptional cases.

ICNFRD — = 1 for the 40/20 plots to be labeled "Confidential Restricted Data", = 0 for no labeling of this type on the plot

NA3SKP — Number of data points on Tape A3 to be skipped after writing a good point. Normally NA3SKP will be at least 10 to keep the length of the A3 listing reasonably short.

NARIN — Number of frames to advance on the 40/20 when ADVFLM is called between the data frames. If NARIN = 0, 2 frames are advanced.

TMN — Lower limit on good times for searching tape A5. All records for which the time is less than TMN are not processed.

TMX — Upper limit on good times for searching tape A5. All records for which the time is greater than TMX are not processed.

DELT — A constant time increment to be added to all times in processing. Normally zero.

TSTSG — The value against which sigma of the calibration linearity is tested. If the sigma of the calibration, in percent of full scale, is greater than TSTSG the channel is not processed. Normally, TSTSG will be ~1.0. See NSIG for a path around this test.

SMIN — Lower limit for times in a scaling interval.

SMAX — Upper limit for times in a scaling interval.

LXGD — Number specifying the writing of grid lines between SMIN and SMAX, LXGD may have the following values:

Linear Intervals	Cycle logarithmic intervals
1 = 1 interval	11 = 1 cycle
2 = 2	12 = 2
3 = 3	13 = 3
4 = 4	14 = 4
5 = 5	15 = 5
6 = 10	16 = 6
7 = 15	
8 = 20	
9 = 25	
10 = 50	

NXLB Number specifying the labeling of grid lines between SMIN and SMAX. NXLB may have the same values as given for LXGD, with the same significance.

ICS = 1 if a multiplexer pin (channel) is to be skipped, = 0 if the channel is to be processed. Note that ICS corresponds to the pin number as given in the channel data log; on a 47-channel multiplexer the data channel pins run from 2 to 48. To omit processing the 3rd channel on a multiplexer (pin No. 4), ICS (4) = 1.

REC 72 columns of information pertaining to an individual channel. This will normally be channel number, multiplexer assignment, and special comments. This information appears on all output tapes (B5, B6, A3). The REC card should not contain a channel name or similar information which would cause tapes B6 and A3 to become classified. The REC card should <u>not</u> contain information only applicable to a single experimental plan.

NCS = 0 for a channel with no calibration on it
= 1 for a directly calibrated channel
= 2 for a channel with a voltage calibration plus a known transducer calibration curve

NTCC Number of transducer calibration coefficients, used for NCS = 0 or 2, and ranging between 1 and 6.

NTCS Number of tape calibration steps, used for NCS = 1 or 2, and ranging between 1 and 25.

NMAX Power of the final least square fit relating raw data points to reduced data points, and ranging between 1 and 5.

ITAPOS Multiplexer pin number for the channel, ranging from 2 to 48.

LYGD — Number specifying the writing of grid lines between YMIN and YMAX. See LXGD for details.

NYLB — Number specifying the labeling of grid lines between YMIN and YMAX. See LXGD for details.

YMIN — Lower limit for data scaling on a 40/20 plot.

YMAX — Upper limit for data scaling on a 40/20 plot.

PTLGMX — The exponent corresponding to YMIN for labeling logarithmic plotting; for example, if PTLGMX = 3.0 and NYLB = 12, then the ordinate of the plot will be labeled:

```
 8
 6
 4
 2
10 + 04
 8
 6
 4
 2
10 + 03
```

NSIG — If NSIG is greater than 0, a channel will be processed even if the sigma of the calibration linearity is greater than TSTSG.

B1 — Coefficients of transducer calibration curve, for NCS = 2, or the coefficient of the final fit for NCS = 0.

VTCS — Value of the tape calibration steps,normally in millivolts for NCS = 2, in units of the physical variable for NCS = 1.

TSTCS — Time of the start of the calibration steps.

SPAN — Length of the calibration steps. Note that for the analysis of the calibration steps, 80 percent of the span is used, starting at the time TSTCS + 10 percent of the span.

VERLB — 72 columns of vertical labeling. Actually, only the first 64 appear on the 40/20 curve and the vertical labeling should be limited to columns 3 through approximately 55.

CHNM — Channel name in 72 columns. This quantity appears only on the 40/20 plot.

3. Running Description of the Code

Refer to the FORTRAN source deck listing for RDH-D. RDH-D first rewinds tapes 4, 5, 6 and 8 (A4, B1, B2 and B6), turns off the sense lights, and then sets the constants used in scaling the plots of calibration data. Next, it reads the tape control card, order 100, and tests SL4 (Sense Light 4). At the end of each processed tape, SL4 is turned on; thus advancing of B6 is avoided if there is already a B6 in use. If SL4 is not on, that is, this is a new computer run, IADVB6 is tested; if IADVB6 $\neq$ 0, PØSB6 is called. POSB6 advances B6 to the end-of-file and sets NCHB6 to the serial number of the last channel on B6. If IADVB6 = 0, NCHB6 is set = 0 and B6 is left alone (it has already been rewound). Note that NCHB6 is advanced 1 before each channel is written on tape B6 (order 605). Figure 2 is a block diagram of RDH-D.

Order 110 + 1, calling CUS, is the next significant step other than input. The CUS and CUST write-up gives details of what this subroutine does. However, in brief, either CUS or CUST converts, unpacks and normalizes the low density, BCD, 1 frame per record data on tape A5 and writes a high density, binary, approximately 200 frame per record tape B1 from which all further processing is done. In addition, CUST puts in a time correction, through the use of the TIC subroutine, for the Packard-Bell limitation of 0.2 seconds time resolution. After CUS or CUST is called, the number of records from A5 stored on B6 is in the core as NRCRD. Note that NRCRD $\leq$ 4500, that is, the code can handle up to 4500 data points on any one channel in any one pass through an A5 file.

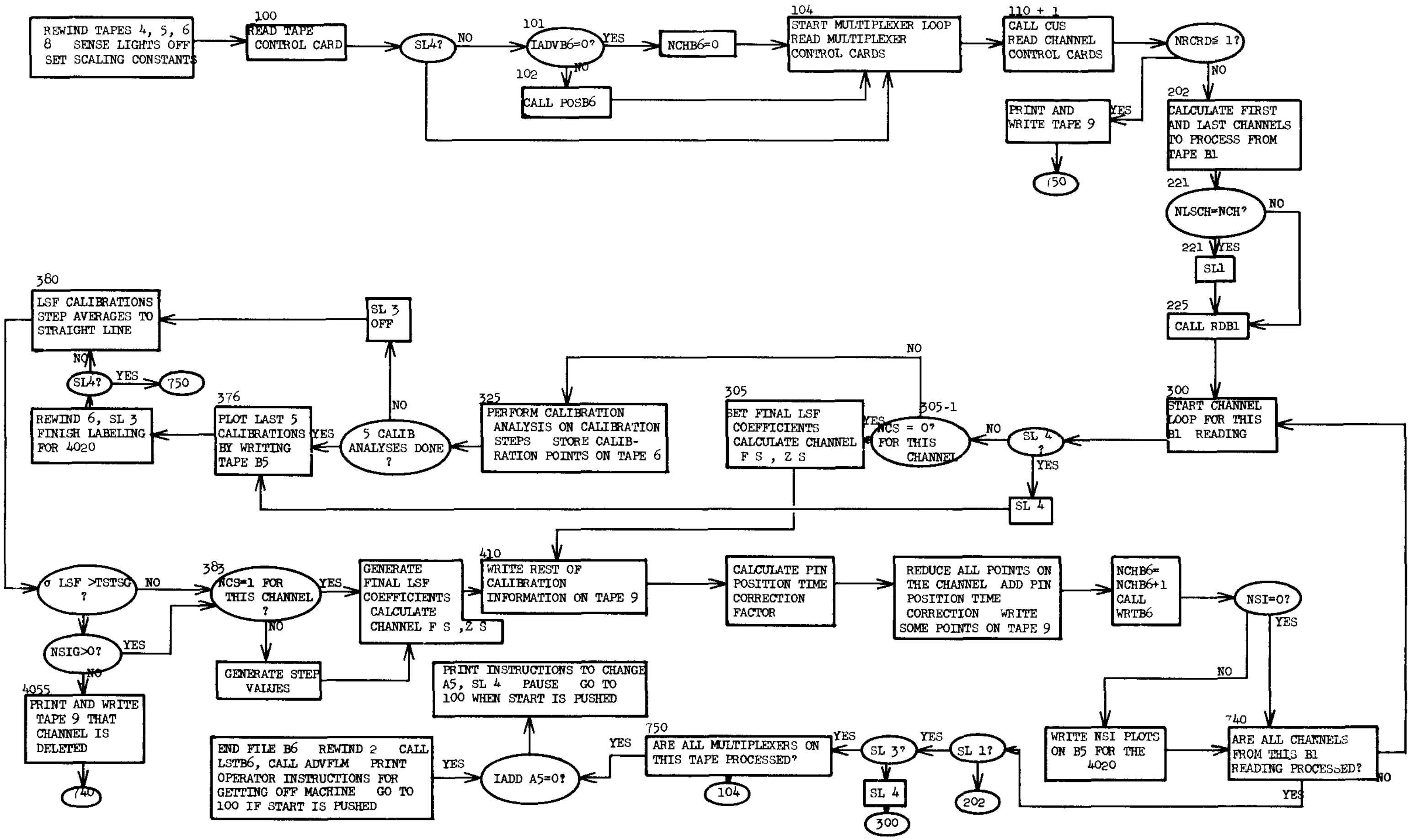


Figure 2. Simplified Block Diagram, RDH-D Code

The calculated quantity DTPPT is the time difference per multiplexer channel, for example, 0.5 msec on a 50 channel, 40 frame per second multiplexer.

At order 200, the maximum number of channels to process from tape B1 is calculated; note that if NRCRD = 4500, NCMX = 2, that is, at least 2 channels are always processed unless ICS, the channel skipping indicator, is not = 0. The next step checks to see that at least 2 records (frames) from A5 are on B1; if not, the multiplexer is skipped. Orders 202 to 222 calculate NFSCH and NLSCH, the numbers of the first and last channels to be processed from tape B1 in the next pass through B1. If the NLSCH is equal to NCH, the total number of channels, SL1 is turned on.

Order 225 calls RDB1 which brings the data channels from NFSCH to NCSCH and time T into the core; if we assume NFSCH = 4, NLSCH = 6 and designate the data point for the jth channel at the ith time as DP(i,j) then the storage block STG will have the following after RDB1 has been called:

```
STG(1) = T(1)
STG(2) = DP(1,4)
STG(3) = DP(1,5)
STG(4) = DP(1,6)
STG(5) = T(2)
STG(6) = DP(2,4)
STG(7) = DP(2,5)
STG(8) = DP(2,6)
     .
     .
     .
STG(NWDRD) = DP(NRCRD, 6)
```

Order 300 starts the individual channel processing loop. If SL4 is on, the processing is skipped, and the remaining calibration data stored on tape B2 is written on B5 for plotting; SL 4 is turned on at order 746 if the last channel on the multiplexer has been processed, but the calibration for that channel was not the 5th in a series stored temporarily on B2 and thus has not yet been written on B6.

At order 305-1, NCS for the channel is tested; if NCS = 0, that is, there is no calibration for the channel, the final coefficients B(I) are set, SIGMA is set to 0, the maximum and minimum channel values are calculated, and control is transferred to 410, the beginning of the reduction of the channel data points.

For NCS $\neq$ 0, the process of finding the data points in the calibration steps is started at order 325. The quantities A and C are the times corresponding to 0.01 seconds less than 10 percent of the calibration step and 0.01 seconds more than 90 percent of the calibration step. The DØ375 loop is over all calibration steps and the DØ360 loop is over all the points on each step. Good times (between A and C) and corresponding calibration data points are stored in TMP and TEMP respectively. The average for each step is stored in AVG, and after writing times and points on B2 the deviations of each point from the step average are stored in TEMP. Then the maximum of these deviations is calculated and a standard deviation SIG calculated from

$$\mathrm{SIG} = \sqrt{\frac{\Sigma(\mathrm{CDP} - \mathrm{AVG})^2}{\mathrm{NCDP}}}$$

where CDP = calibration data point

AVG = average of CDP's for that step

NCDP = Number of CDP's for that step, ≤ 50.

This information is written out and the number of points stored on B2 for this calibration and step are stored in NCLPTS. Control then goes to 375.

If the DØ360 loop is exhausted J4 is tested; if J4 = 0, that is, no points were found (calibration times not in STG block) it is assumed that 50 points of value 0 were found. If J4 $\neq$ 0 , the computation of the next step proceeds.

After all calibration steps have been analyzed, a check is made to see if 5 channels of calibration analyses have been completed; if not, SL3 is turned off and control goes to 380. If so, the calibration points are written on B6 for plotting together with labeling and grid writing information and SL3 is turned on. Then SL4 is tested; if on, that is if this was an end of multiplexer set of calibration plots, control goes to 750 to start the next multiplexer. If SL4 is not on, control goes to 380 where the calibration step averages are fit to a straight line by LSF, and the standard deviation A is calculated as a percent of full scale, and compared to TSTSG. If A is greater than TSTSG and NSIG for the channel is zero, the channel is assumed to be bad and control goes to 740, thus omitting the channel. If A is less than TSTSG, or if A is greater than TSTSG and NSIG > 0, the channel processing continues at 383.

At 384, if NCS = 1, the calibration step averages AVG and VTCS, the values of the tape calibration steps (in units of the physical variable) are least square fit to determine the coefficients B of the final fit. The maximum and minimum channel values are calculated at 317 and control is transferred to 410, the beginning of the reduction of the channel data points.

At 390, if NCS = 2, the transducer calibration coefficients are used to generate a set of values of the physical variable corresponding to the calibration steps. Control then goes to 384 to generate the final set of fit coefficients as before.

At 410, after setting to zero any coefficients B not generated in the final fit and writing some calibration data on Tape A3, the data reduction begins. Time is corrected for the position of a channel in a data frame by adding a quantity

(ITAPØS + 1)*DTPPT

to each time in storage for each point. For example, to the time for each point on the first channel, at tape position ITAPOS = 2, is added a time correction 3*DTPPT, since it occurs after the double width full scale and single width zero scale pulses. Note that from each time the previous channels' time correction is subtracted as the current channels' time correction is added in. Each raw data point is inserted in the expansion

$$\text{Reduced Data Point} = \sum_{i=0}^{\text{NMAX}} B_i(\text{Raw Data Point})$$

to generate the final data point.

Data to be written on tape A3 are stored temporarily and/or written out at this time. At order 605, the serial number for the next tape B6 record is advanced 1 and all the times and data points, plus the TITLE and REC information are written on B6 by the WRTB6 subroutine. Orders 625 to 700 write the information on tape B5 for plotting on the 4020 plotter.

At 700, if sense switch 6 is down, tape B5 is prepared for removal and the code pauses for a new B5 to be put on. Then, if SL 1 is not on, control returns to 202 to process the rest of the channels on the multiplexer. If SL 1 is on, that is, if the last channels processed were the last ones to be processed on the multiplexer, SL3 is tested. If SL3 is on, that is, the calibrations from the last channels processed were written on B5 for plotting, control goes to 750, the end of the multiplexer loop. If SL3 is not on, SL4 is turned on and if there are calibrations to be written on B5 for plotting, control goes to 300. If there are no calibrations to be plotted, SL4 is turned off and control goes to 750.

After all multiplexers on the input tape have been processed, IADDA5 is tested; if IADDA5 $\neq$ 0, SL4 is turned on, instructions to change A5 are printed and the 7090 pauses. If IADDA5 = 0, the channel identification for the channels on B6 is listed on A3 and instructions for getting off the machine are printed on line.

4. Source Deck (FORTRAN) Listing

```
CRDHD
CLIMITED TO 47 CHANNELS,4500 POINTS PER CHANNEL,50 POINTS USED PER
CCALIBRATION STEP,TIMES IN ASCENDING ORDER ON INPUT TAPE,20 SCALING
CINTERVALS,500 POINTS ON ANY ONE PLOT, 5TH POWER LEAST SQUARE FITS
C25 CALIBRATION STEPS ON ANY ONE CHANNEL
C SENSELIGHT 1 INDICATES NLSCH=NCH
C SENSELIGHT 2 INDICATES BAD CHANNEL
C SENSELIGHTS 3 AND 4 ARE USED FOR PLOTTING CALIBRATION DATA
C A1 IS THE UTILITY TAPE 1
C A2 IS THE PROGRAM TAPE 10
C A3 IS THE LISTING TAPE 9
C A4 IS INTERMEDIATE TIME STORAGE TAPE 4
C A5 IS THE INPUT RAW DATA TAPE 2
C B1 IS THE INTERMEDIATE TAPE 5
C B2 IS THE INTERMEDIATE CALIBRATION DATA STORAGE TAPE 6
C B5 IS THE 40/20 PLOTTING TAPE 12
C B6 IS THE PERMANENT DATA STORAGE TAPE 8
C      SENSE SWITCH SIX DOWN GIVES PAUSE TO CHANGE TAPE B-5
      DIMENSIONSTG(13500),TITLE(6),REC(47,12),NCS(47),NTCC(47),AVG(25),
     1NTCS(48),NMAX2(47),ITAPOS(47),LYGD(47),NYLB(47),YMIN(47),YMAX(47),
     2B1(47,6),VTCS(47,25),VERLB(47,12),CHNM(47,12),TMP(500),LXGD(20),
     3NXLB(20),NPTS(25),B(6),TSTCS(47,25),SPAN(47),SMIN(20),SMAX(20),
     4TMIN(5),TMAX(5),XYBND(4),NCLPTS(5,25),NSTP(5),NOCLCH(5),TEMP(50),
     5ICS(47),XYPLT(4),PTLGMN(47) ,NSIG(47)
    1 FORMAT(14I5)
    3 FORMAT(12A6)
    4 FORMAT(13X,6A6,32H CHANNEL CALIBRATIONS,CODE RDH-D)
    5 FORMAT(8I5,3F10.5,I2)
    7 FORMAT(2F10.5,2I5)
    9 FORMAT(48I1)
   10 FORMAT(7I5,3F10.5,I5)
   11 FORMAT(1P7E10.5)
   13 FORMAT(7F10.5)
   15 FORMAT(50H0...PUT NEXT REEL OF TAPE A5 ON...THEN PRESS START)
   17 FORMAT(1H0)
   18 FORMAT(1H1)
   19 FORMAT(1H 6A6,I5,3H X I2,12H MULTIPLEXER)
   21 FORMAT(1H 12A6,6H  NCS=I2,17H FOR THIS CHANNEL)
   23 FORMAT(1H112A6,6A6,4HNCS=I1,I3,1HXI2)
   24 FORMAT(25H0INPUT COEFFICIENTS B(I)=1P6E12.3)
   25 FORMAT(25H FINAL COEFFICIENTS B(I)=1P6E12.3,7H SIGMA=E12.3)
   27 FORMAT(25H SCALING COEFFICIENTS Y =1P2E12.3,
     133H  CALCULATED FULL AND ZERO SCALE=2E12.3)
   29 FORMAT(30H CALIBRATION STEP TIMES,SPAN =11F8.1)
   30 FORMAT(15H SCALING LIMITS5(F8.1,4H TO F8.1,1H,))
   31 FORMAT(6H STEP I2,6H NPTS=I4,5H AVG=0PF6.3,
     15H SIG=0PF7.4, 9H MAX DEV=0PF6.3)
   33 FORMAT(24H NOMINAL VALUE OF STEPS=16F6.1)
   34 FORMAT(7H SIGMA=F9.4,7H TSTSG=F9.4,
     150H FOR THE TAPE CALIBRATION,IN PERCENT OF FULL SCALE)
   37 FORMAT(46H0THIS CHANNEL APPEARS BAD AND IS BEING OMITTED)
   38 FORMAT(46H THIS CHANNEL APPEARS BAD AND IS BEING OMITTED)
   43 FORMAT(4(0PF12.2,F7.3,1PE11.3))
   45 FORMAT(14H0TOTAL NO PTS=I5,8H NA3SKP=I5,11H CODE RDH-D)
   47 FORMAT(35X,33HCALCULATED FULL,ZERO SCALE,SIGMA=2(1PE10.3,1H,),
     1E10.3,11H CODE RDH-D)
   49 FORMAT(55H0OPERATOR...THIS RUN OF RDH-D IS OVER...SAVE THIS LIST )
   50 FORMAT(13A6,16HFIRST STEP TMIN=F10.3,6H TMAX=F10.3)
   51 FORMAT(32H0UNLOAD AND SAVE TAPES B6 AND A5)
   53 FORMAT(35H0THERE IS OUTPUT ON TAPES A3 AND B5)
   55 FORMAT(40H TAPES A3 AND B5 CAN BE HANDLED NORMALLY)
   57 FORMAT(32H TAPE A3 CAN BE HANDLED NORMALLY)
```

```
   58 FORMAT(55H TAPE B5 IS CLASSIFIED AND MUST BE HANDLED WITH CARE   )
   60 FORMAT(6X,7HPIN NO I2,12A6,15H STEP 1 LIMITS=2F9.2)
   61 FORMAT(1H110X,38H THE FOLLOWING CHANNELS ARE ON TAPE B6)
   62 FORMAT(39H0MORE THAN 500 PTS IN SCALING INTERVAL  F10.2,4H TO
     1F10.2)
   63 FORMAT(55H0LESS THAN 2 RECORDS FOUND BY CUS. THIS MULTIPLEXER OUT)
   65 FORMAT(18H0CALIBRATION STEP 2F10.2,   8H MISSING)
      REWIND 6
      REWIND4
      REWIND5
      REWIND8
      SENSELIGHT0
      XYBND(1)=0.0
      XYBND(2)=100.0
      XYBND(3)=0.0
      XYBND(4)=100.
      XYPLT(1)=0.0
      XYPLT(2)=100.0
       XYPLT(3)=0.0
      XYPLT(4)=1.0
  100 READINPUTTAPE10,1,NMPLX,IADDA5,IADVB6
      IF(SENSELIGHT4)104,101
  101 IF(IADVB6)102,103,102
  102 CALLPOSB6(NCHB6)
      GOTO104
  103 NCHB6=0
  104 DO7501I=1,NMPLX
      NLSCH=0
      NCLSEI=1
      READINPUTTAPE10,3,(TITLE(I),I=1,6)
      READINPUTTAPE10,1,NCH,NSI,NCHFM,NFMSEC,IMUC,NFLAD,
     1NFLBS,NRCSP,ICNFRD,NA3SKP,NARIN
      IF(NARIN)1041,1040,1041
 1040 NARIN=2
 1041 PRINT 18
      PRINT18
      PRINT19,(TITLE(J),J=1,6),NCHFM,NFMSEC
      READINPUTTAPE10,13,TMN,TMX,DELT,TSTSG
      IMUC=IMUC+1
      READINPUTTAPE10,7,(SMIN(I),SMAX(I),LXGD(I),NXLB(I),I=1,NSI)
      READINPUTTAPE10, 9,(NTCS(I),I=1,48)
      DO110I=2,48
  110 ICS(I-1)=NTCS(I)
      CALLCUS(NCH, 4500,NFLAD,NFLBS,TMN,TMX,NRCRD,STG,NRCSP)
      REWIND6
      A=NCHFM
      C=NFMSEC
      DTPPT=1.0/(A*C)
  150 DO185I=1,NCH
      READINPUTTAPE10,3,(REC(I,J),J=1,12)
      READINPUTTAPE10,10,NCS(I),NTCC(I),NTCS(I),NMAX2(I),ITAPOS(I),
     1LYGD(I),NYLB(I),YMIN(I),YMAX(I),PTLGMN(I),NSIG(I)
      N1=NTCC(I)
      N2=NTCS(I)
      I3=NCS(I)+1
      GOTO(160,170,180),I3
  160 READINPUTTAPE10,11,(B1(I,J),J=1,N1)
      GOTO184
  170 READINPUTTAPE10,13,(VTCS(I,J),J=1,N2)
  175 READINPUTTAPE10,13,(TSTCS(I,J),J=1,N2),SPAN(I)
      GOTO184
  180 READINPUTTAPE10,11,(B1(I,J),J=1,N1)
      READINPUTTAPE10,13,(VTCS(I,J),J=1,N2)
```

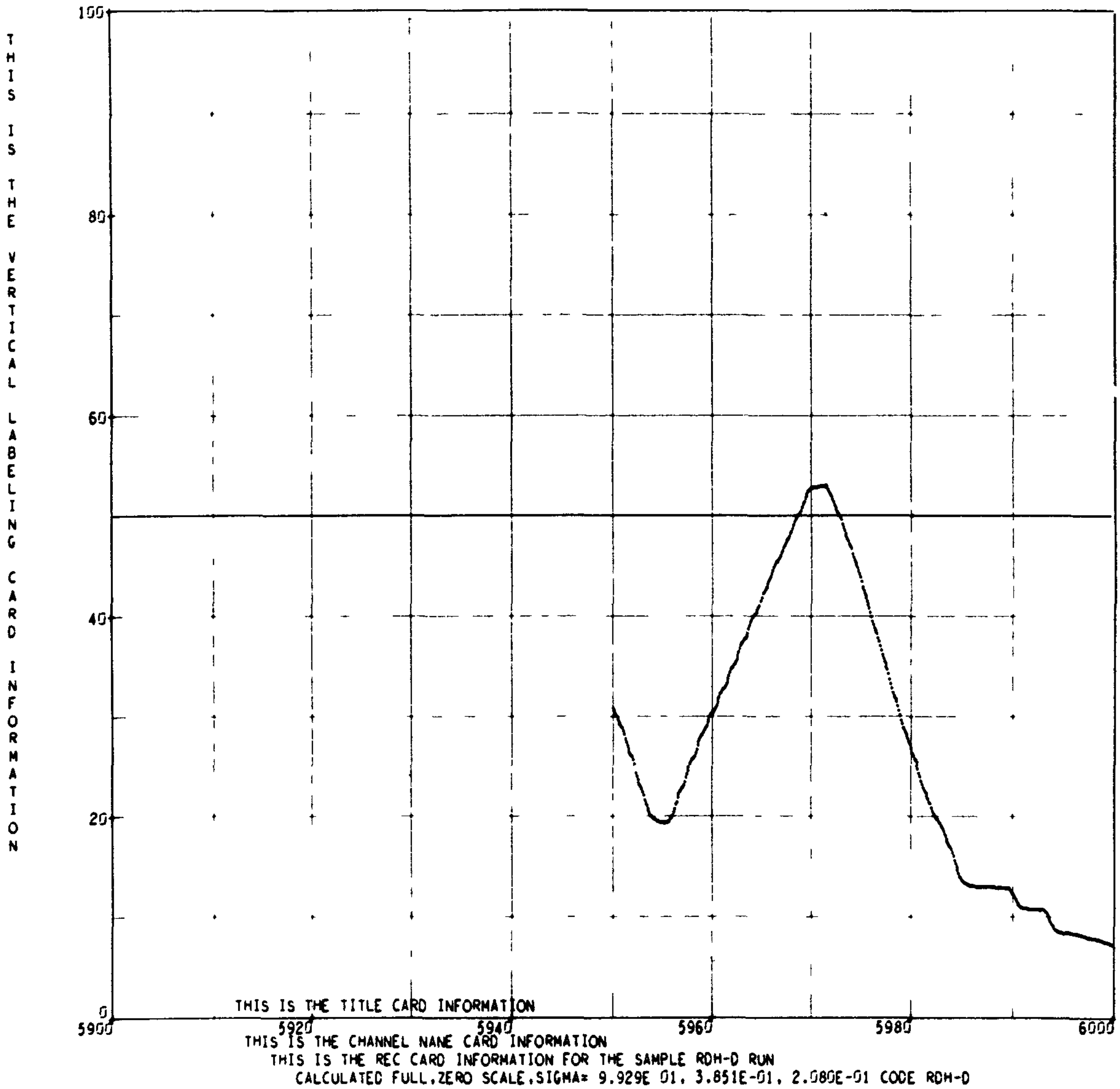
```
      GOTO(175,181,182),IMUC
181   READINPUTTAPE10,13,(TSTCS(I,J),J=1,N2),SPAN(I)
      IMUC=3
      K=I
      N3=N2
      GOTO184
182   DO183J=1,N3
183   TSTCS(I,J )=TSTCS(K,J )
      SPAN(I)=SPAN(K)
184   READINPUTTAPE10,3,(VERLB(I,J),J=1,12)
185   READINPUTTAPE10,3,(CHNM(I,J),J=1,12)
200   NCMX=13500/NRCRD-1
      IF(NRCRD-1)201,201,202
201   PRINT63
      WRITEOUTPUTTAPE9,63
      GOTO750
202   NFSCH=NLSCH+1
203   IF(NFSCH-NCH)204,204,745
204   IF(ICS(NFSCH))206,208,206
206   NFSCH=NFSCH+1
      GOTO203
208   NLSCH=NFSCH
      I=1
210   NLSCH=NLSCH+1
      IF(NLSCH-NCH)211,211,220
211   I=I+1
212   IF(ICS(NLSCH))220,214,220
214   IF(I-NCMX)210,221,220
220   NLSCH=NLSCH-1
221   IF(NLSCH-NCH)225,222,220
222   SENSELIGHT1
225   CALLRDB1(NCH,NRCRD,NFSCH,NLSCH,STG)
      NWRC=NLSCH-NFSCH+2
      NWORD=NRCRD*NWRC
      J=0
300   DO740I2=NFSCH,NLSCH
      IF(SENSELIGHT4)301,302
301   SENSELIGHT4
      GOTO376
302   J=J+1
      WRITEOUTPUTTAPE9,23,(REC(I2,I),I=1,12),(TITLE(I3),I3=1,6),NCS(I2)
     1,NCHFM,NFMSEC
      J1=NMAX2(I2)+1
      PRINT21,(REC(I2,I),I=1,12),NCS(I2)
      IF(NCS(I2))325,305,325
305   DO315I3=1,J1
315   B(I3)=B1(I2,I3)
      SIGMA=0.0
317   CHMIN=POLEV(0.,J1,B,TMP)
      CHMAX=POLEV(1.,J1,B,TMP)
      GOTO410
325   X=SPAN(I2)*.1-.01
      Y=SPAN(I2)*.8+.02
      J2=NTCS(I2)
      WRITEOUTPUTTAPE9,29,(TSTCS(I2,I),I=1,J2),SPAN(I2)
      J3=1
      DO375I3=1,J2
      A=TSTCS(I2,I3)+X
      C=A+Y
      J4=0
      DO328I=1,50
      TMP(I)=0.0
328   TEMP(I)=0.0
```

```
      D=0.0
      J4IND=0
  329 D0360I4=J3,NWDRD,NWRC
      IF(J4IND)3292,3292,345
 3292 IF(STG(I4)-A)360,340,330
  330 IF(STG(I4)-C)340,340,335
  335 IF(J4)360,360,345
  340 I5=I4+J
      D=D+STG(I5)
      J4=J4+1
      TEMP(J4)=STG(I5)
      TMP(J4)=STG(I4)
      IF(J4-50)360,345,345
  345 A=J4
      AVG(I3)=D/A
      C=0.0
      WRITETAPE6,(TEMP(I),I=1,50),(TMP(K),K=1,50)
      D0350I5=1,J4
      TEMP(I5)=ABSF(TEMP(I5)-AVG(I3))
  350 C=C+TEMP(I5)*TEMP(I5)
      CALLMAX3(TEMP,-1,J4,I5,XAM)
      SIG=SQRTF(C/A)
      WRITEOUTPUTTAPE9,31,I3,J4,AVG(I3),SIG,XAM
      NCLPTS(NCLSET,I3)=J4
      GOT0375
  360 CONTINUE
      IF(J4)362,362,375
  362 J4=50
      WRITEOUTPUTTAPE9,65,A,C
      J4IND=1
      J3=1
      GO TO 329
  375 J3=I4-NWRC
      NOCLCH(NCLSET)=I2
      NSTP(NCLSET)=J2
      NCLSET=NCLSET+1
      IF(NCLSET-5)3755,3755,376
 3755 IF(SENSELIGHT3)380,380
  376 I=NCLSET-1
      NCLSET=1
      REWIND6
      CALL ADVFLM(0,NARIN)
      CALLSELGRD(5,6)
      CALLLBLFGD(1,6,XYRND)
      D0379I3=1,I
      I5=NSTP(I3)
      D0378I4=1,I5
      I7=NCLPTS(I3,I4)
      READTAPE6,(TEMP(I6),I6=1,50),(TMP(I8),I8=1,50)
      A=TMP(1)
      C=(TMP(I7)-A)/18.0
      D=1+20*(I3-1)
      IF(I4-1)3768,3765,3768
 3765 TMIN(I3)=TMP(1)
      TMAX(I3)=TMP(I7)
 3768 D0377I6=1,I7
  377 TMP(I6)=(TMP(I6)-A)/C+D
  378 CALLPLFILM(I7,TMP(100),TMP,-1,TEMP,-1,XYPLT,42,0)
  379 SENSELIGHT3
      REWIND6
      CALLLINCNT(58)
      WRITEOUTPUTTAPE12,4,(TITLE(I),I=1,6)
      D03795I3=1,I
```

```
      I4=NOCLCH(I3)
      I7=ITAPOS(I4)
 3795 WRITEOUTPUTTAPE12,60,I7,(CHNM(I4,I5),I5=1,12),TMIN(I3),TMAX(I3)
      IF(SENSELIGHT4)750,380
  380 DO382I3=1,J2
  382 TEMP(I3)=VTCS(I2,I3)
      WRITEOUTPUTTAPE9,33,(TEMP(I),I=1,J2)
      CALLLSF(J2,1,AVG,TEMP,B,SIGMA)
      CALLMAX3(TEMP,-1,J2,I,A)
      CALLMIN3(TEMP,-1,J2,I,C)
      A=100.*SIGMA/(A-C)
      WRITEOUTPUTTAPE9,34,A,ISTSG
      IF(A-ISTSG)383,383,405
  383 IF(NCS(I2)-1)390,384,390
  384 CALLLSF(J2,NMAX2(I2),AVG,TEMP,B,SIGMA)
      GOTO317
  390 J3=NICC(I2)
      DO392I3=1,J3
  392 B(I3)=B1(I2,I3)
      WRITEOUTPUTTAPE9,24,(B(I),I=1,J3)
  404 DO4045I3=1,J2
      A=TEMP(I3)
 4045 TEMP(I3)=POLEV(A,J3,B,TMP)
      GOTO384
  405 IF(NSIG(I2))4055,4055,383
 4055 WRITEOUTPUTTAPE9,37
      PRINT38
      GOTO740
  410 IF(J1-6)412,420,420
  412 J4=J1+1
      DO415I=J4,6
  415 B(I)=0.0
  420 WRITEOUTPUTTAPE9,25,(B(I),I=1,6),SIGMA
      WRITEOUTPUTTAPE9,27,YMIN(I2),YMAX(I2),CHMAX,CHMIN
      WRITEOUTPUTTAPE9,30,(SMIN(I),SMAX(I),I=1,NSI)
  500 IF(I2-NFSCH)502,502,525
  502 TINCP=0.0
  525 WRITEOUTPUTTAPE9,45,NRCRD,NA3SKP
      I=NA3SKP
      J3=1
      J4=NMAX2(I2)+1
      X=ITAPOS(I2)+1
      TINC=X*DTPPT
      X=TINC-TINCP
      TINCP=TINC
      DO600I3=1,NWDRD,NWRC
      I=I+1
      J2=I3+J
      STG(I3)=STG(I3)+X
      A=STG(J2)
      STG(J2)=POLEV(A,J4,B,TMP)
      IF(I-NA3SKP)600,600,530
  530 I=0
      TEMP(J3)=STG(I3)
      TEMP(J3+1)=A
      TEMP(J3+2)=STG(J2)
      J3=J3+3
      IF(J3-12)600,600,532
  532 WRITEOUTPUTTAPE9,43,(TEMP(J3),J3=1,12)
      J3=1
  600 CONTINUE
      DO601I=1,12
      TEMP(I)=REC(I2,I)
```

```
      TEMP(I+13)=CHNM(I2,I)
  601 TEMP(I+25)=VERLB(I2,I)
  605 NCHB6=NCHB6+1
      CALLWRTB6(NCHB6,NRCRD,TEMP,TITLE,STG,J,NWRC)
      TEMP(43)=YMAX(I2)
      TEMP(42)=YMIN(I2)
      J2=0
      IF(NSI)740,740,625
  625 DO700I3=1,NSI
      A=SMIN(I3)
      C=SMAX(I3)
      J2IND=1
      DO 645 I4=1,NWORD,NWRC
      IF(STG(I4)-A)645,640,630
  630 IF(STG(I4)-C)640,640,631
  631 J2IND=J2IND+1
      IF(J2IND-10)645,645,650
  640 J2=J2+1
      J2IND=1
  645 I5=I4
  650 CALL ADVFLM(0,NARIN)
      I5=I5-(J2IND-1)*NWRC
      CALLSELGRD(LXGD(I3),LYGD(I2))
      IF(LYGD(I2)-11)652,654,654
  652 I=LYGD(I2)-4
      CALLSELGRD(1,I)
  654 TEMP(40)=A
      TEMP(41)=C
      CALLLINCNT(60)
      WRITEOUTPUTTAPE12,47,CHMAX,CHMIN,SIGMA
      CALLWFILM(1,63,TEMP(26),0,1.0,5.0)
      CALLWFILM(0,36,TITLE,0,56.0,30.0)
      CALLWFILM(0,72,TEMP(14),0,58.0,30.0)
      CALLWFILM(0,72,TEMP,0,59.,33.0)
      J3=I5
      J4=J3+J
      IF(J2-500)657,657,656
  656 J2=500
      WRITEOUTPUTTAPE 9,62,TEMP(40),TEMP(41)
  657 CALLPLFILM(J2,TMP   ,STG(J3), NWRC,STG(J4), NWRC,TEMP(40),42,0)
      IF(NYLB(I2)-10)659,659,658
  658 TEMP(42)=PTLGMN(I2)
  659 CALLLBLFGD(NXLB(I3),NYLB(I2),TEMP(40))
      IF(ICNFRD)660,700,660
  660 CALLCLAS
  700 J2=0
      IF ( SENSE SWITCH 6 ) 710, 740
  710 CALL ADVFLM (0,2)
      END FILE 12
      END FILE 12
      REWIND 12
      PAUSE 77777
  740 CONTINUE
      IF(SENSELIGHT1)745,202
  745 IF(SENSELIGHT3)750,746
  746 SENSELIGHT4
      IF(NCLSET-1)748,748,300
  748 IF(SENSELIGHT4)750,750
  750 CONTINUE
      IF(IADDA5)760,800,760
  760 PRINT15
      PRINT17
      PRINT17
```

```
      SENSELIGHT4
      PAUSE77777
      GOTO100
  800 ENDFILE8
      REWIND2
      WRITEOUTPUTTAPE9,61
      CALLLSTB6
      REWIND8
      PRINT49
      PRINT51
      PRINT53
      CALLADVFLM(0,2)
      IF(ICNFRD)810,805,810
  805 PRINT55
      GOTO815
  810 PRINT57
      PRINT58
  815 PRINT17
      PRINT17
      PAUSE77777
      GOTO100
      END
```



5. Sample Output
(Xerox Plot)

```
THIS IS THE REC CARD INFORMATION FOR THE SAMPLE RDH-D RUN          THIS IS THE TITLE CARD INFORMATION  NCS=2 50X20
CALIBRATION STEP TIMES,SPAN =   5774.0  5784.0  5794.0  5804.0  5814.0  5824.0  5834.0  5844.0  5854.0  5864.0  5874.0
CALIBRATION STEP TIMES,SPAN =      10.0
STEP  1 NPTS=  50 AVG= 1.005 SIG= 0.0007 MAX DEV= 0.001
STEP  2 NPTS=  50 AVG= 0.907 SIG= 0.0040 MAX DEV= 0.017
STEP  3 NPTS=  50 AVG= 0.806 SIG= 0.0020 MAX DEV= 0.008
STEP  4 NPTS=  50 AVG= 0.705 SIG= 0.0017 MAX DEV= 0.007
STEP  5 NPTS=  50 AVG= 0.603 SIG= 0.0031 MAX DEV= 0.013
STEP  6 NPTS=  50 AVG= 0.504 SIG= 0.0073 MAX DEV= 0.031
STEP  7 NPTS=  50 AVG= 0.400 SIG= 0.0042 MAX DEV= 0.017
STEP  8 NPTS=  50 AVG= 0.299 SIG= 0.0049 MAX DEV= 0.021
STEP  9 NPTS=  50 AVG= 0.196 SIG= 0.0028 MAX DEV= 0.011
STEP 10 NPTS=  50 AVG= 0.094 SIG= 0.0028 MAX DEV= 0.011
STEP 11 NPTS=  50 AVG= 0.     SIG= 0.     MAX DEV= 0.
NOMINAL VALUE OF STEPS=   1.0   0.9   0.8   0.7   0.6   0.5   0.4   0.3   0.2   0.1   0.
SIGMA=   0.2095 TSTSG=   1.0000 FOR THE TAPE CALIBRATION,IN PERCENT OF FULL SCALE

INPUT COEFFICIENTS B(I)=  -0.            1.000E 02
FINAL COEFFICIENTS B(I)=   3.851E-01    9.891E 01   0.          0.          0.          0.          SIGMA=   2.080E-01
SCALING COEFFICIENTS Y =  -0.            1.000E 02  CALCULATED FULL AND ZERO SCALE=   9.929E 01   3.851E-01
SCALING LIMITS  5900.0 TO   6000.0,

TOTAL NO PTS= 2260 NA3SKP=    9 CODE RDH-D
```

5774.20	1.004	9.971E 01	5775.20	1.003	9.960E 01	5776.20	1.004	9.971E 01	5777.20	1.003	9.960E 01
5778.20	1.005	9.981E 01	5779.20	1.005	9.981E 01	5780.20	1.005	9.981E 01	5781.20	1.005	9.981E 01
5782.20	1.005	9.981E 01	5783.20	1.005	9.981E 01	5784.20	1.005	9.981E 01	5785.20	0.917	9.112E 01
5786.20	0.907	9.008E 01	5787.20	0.906	8.998E 01	5788.20	0.905	8.988E 01	5789.20	0.905	8.988E 01
5790.20	0.905	8.988E 01	5791.20	0.905	8.988E 01	5792.20	0.905	8.988E 01	5793.20	0.905	8.988E 01
5794.20	0.861	8.553E 01	5795.20	0.811	8.057E 01	5796.20	0.806	8.015E 01	5797.20	0.805	8.005E 01
5798.20	0.804	7.994E 01	5799.20	0.804	7.994E 01	5800.20	0.804	7.994E 01	5801.20	0.804	7.994E 01
5802.20	0.804	7.994E 01	5803.20	0.804	7.994E 01	5804.20	0.752	7.477E 01	5805.20	0.709	7.053E 01
5806.20	0.704	7.001E 01	5807.20	0.704	7.001E 01	5808.20	0.704	7.001E 01	5809.20	0.704	7.001E 01
5810.20	0.704	7.001E 01	5811.20	0.703	6.991E 01	5812.20	0.703	6.991E 01	5813.20	0.703	6.991E 01
5814.20	0.696	6.919E 01	5815.20	0.611	6.080E 01	5816.20	0.604	6.008E 01	5817.20	0.603	5.998E 01
5818.20	0.601	5.987E 01	5819.20	0.601	5.987E 01	5820.20	0.601	5.987E 01	5821.20	0.601	5.987E 01
5822.20	0.602	5.994E 01	5823.20	0.602	5.994E 01	5824.20	0.602	5.994E 01	5825.20	0.523	5.207E 01
5826.20	0.504	5.020E 01	5827.20	0.501	4.989E 01	5828.20	0.501	4.989E 01	5829.20	0.501	4.989E 01
5830.20	0.501	4.989E 01	5831.20	0.500	4.984E 01	5832.20	0.500	4.984E 01	5833.20	0.499	4.973E 01
5834.20	0.499	4.973E 01	5835.20	0.410	4.094E 01	5836.20	0.400	3.991E 01	5837.20	0.399	3.980E 01
5838.20	0.399	3.980E 01	5839.20	0.397	3.970E 01	5840.20	0.397	3.970E 01	5841.20	0.397	3.970E 01
5842.20	0.397	3.970E 01	5843.20	0.397	3.970E 01	5844.20	0.397	3.970E 01	5845.20	0.312	3.122E 01
5846.20	0.298	2.990E 01	5847.20	0.298	2.986E 01	5848.20	0.296	2.969E 01	5849.20	0.296	2.969E 01
5850.20	0.296	2.969E 01	5851.20	0.296	2.969E 01	5852.20	0.296	2.966E 01	5853.20	0.296	2.966E 01
5854.20	0.281	2.822E 01	5855.20	0.203	2.046E 01	5856.20	0.196	1.973E 01	5857.20	0.195	1.963E 01
5858.20	0.195	1.963E 01	5859.20	0.194	1.952E 01	5860.20	0.195	1.963E 01	5861.20	0.194	1.952E 01
5862.20	0.194	1.952E 01	5863.20	0.194	1.952E 01	5864.20	0.177	1.787E 01	5865.20	0.100	1.032E 01
5866.20	0.093	9.593E 00	5867.20	0.093	9.593E 00	5868.20	0.092	9.489E 00	5869.20	0.092	9.489E 00
5870.20	0.092	9.489E 00	5871.20	0.092	9.489E 00	5872.20	0.092	9.489E 00	5873.20	0.092	9.489E 00
5874.20	0.091	9.386E 00	5875.20	0.	3.851E-01	5876.20	0.	3.851E-01	5877.20	0.	3.851E-01
5878.20	0.	3.851E-01	5879.20	0.	3.851E-01	5880.20	0.	3.851E-01	5881.20	0.	3.851E-01
5882.20	0.	3.851E-01	5883.20	0.	3.851E-01	5884.20	0.	3.851E-01	5885.20	0.	3.851E-01
5886.20	0.	3.851E-01	5887.20	0.	3.851E-01	5888.20	0.	3.851E-01	5889.20	0.	3.851E-01
5890.20	0.	3.851E-01	5891.20	0.	3.851E-01	5892.20	0.	3.851E-01	5893.20	0.	3.851E-01
5894.20	0.	3.851E-01	5895.20	0.	3.851E-01	5896.20	0.	3.851E-01	5897.20	0.	3.851E-01
5898.20	0.	3.851E-01	5899.20	0.	3.851E-01	5900.20	0.049	5.248E 00	5901.20	0.172	1.735E 01
5902.20	0.189	1.911E 01	5903.20	0.272	2.728E 01	5904.20	0.369	3.691E 01	5905.20	0.440	4.394E 01
5906.20	0.491	4.891E 01	5907.20	0.496	4.942E 01	5908.20	0.497	4.953E 01	5909.20	0.498	4.963E 01
5910.20	0.512	5.098E 01	5911.20	0.540	5.377E 01	5912.20	0.569	5.667E 01	5913.20	0.603	5.998E 01
5914.20	0.624	6.215E 01	5915.20	0.649	6.453E 01	5916.20	0.673	6.691E 01	5917.20	0.701	6.970E 01
5918.20	0.728	7.239E 01	5919.20	0.754	7.498E 01	5920.20	0.775	7.705E 01	5921.20	0.798	7.932E 01
5922.20	0.820	8.150E 01	5923.20	0.843	8.377E 01	5924.20	0.865	8.595E 01	5925.20	0.887	8.812E 01
5926.20	0.913	9.070E 01	5927.20	0.934	9.277E 01	5928.20	0.904	8.977E 01	5929.20	0.859	8.532E 01
5930.20	0.821	8.160E 01	5931.20	0.788	7.829E 01	5932.20	0.759	7.550E 01	5933.20	0.730	7.260E 01

5934.20	0.703	6.991E 01	5935.20	0.678	6.743E 01	5936.20	0.657	6.536E 01	5937.20	0.632	6.287E 01
5938.20	0.607	6.039E 01	5939.20	0.579	5.770F 01	5940.20	0.556	5.543E 01	5941.20	0.535	5.325E C1
5942.20	C.510	5.087E 01	5943.20	0.492	4.901E 01	5944.20	0.471	4.694E 01	5945.20	0.441	4.404E 01
5946.20	0.414	4.135E 01	5947.20	0.384	3.835F 01	5948.20	0.355	3.546E 01	5949.20	0.324	3.246E 01
5950.20	0.303	3.C39E 01	5951.20	0.277	2.780E 01	5952.20	0.247	2.480E 01	5953.20	0.215	2.170E 01
5954.20	0.196	1.973E 01	5955.20	0.191	1.932E 01	5956.20	0.206	2.077E 01	5957.20	0.233	2.346E 01
5958.20	0.259	2.604E 01	5959.20	0.286	2.866E 01	5960.20	0.306	3.070E 01	5961.20	0.328	3.287E 01
5962.20	0.353	3.525E 01	5963.2C	0.377	3.763E 01	5964.20	0.402	4.011E 01	5965.20	0.426	4.249E 01
5966.20	0.450	4.492E 01	5967.20	0.47C	4.684E 01	5968.20	0.493	4.911E 01	5969.20	0.514	5.118E 01
5970.20	0.530	5.284E 01	5971.20	0.532	5.305E 01	5972.20	0.519	5.170E 01	5973.20	0.494	4.922E C1
5974.20	C.463	4.622E C1	5975.20	0.431	4.301E 01	5976.20	0.396	3.960E C1	5977.20	0.362	3.622E 01
5978.20	0.324	3.246E 01	5979.20	0.289	2.894E 01	5980.20	0.260	2.607E 01	5981.20	0.228	2.294E 01
5982.20	0.202	2.C35E 01	5983.20	0.185	1.870E 01	5984.20	0.159	1.611E 01	5985.20	0.132	1.342E 01
5986.20	0.129	1.311E 01	5987.20	C.128	1.302E 01	5988.20	0.128	1.301E 01	5989.20	C.127	1.290E C1
5990.20	0.117	1.198E 01	5991.20	0.106	1.085E 01	5992.20	0.105	1.074E 01	5993.20	0.103	1.053E 01
5994.20	0.085	8.774E 00	5995.20	0.082	8.455E 00	5996.20	0.081	8.360E 00	5997.20	0.079	8.153E 00

MCRE THAN 500 PTS IN SCALING INTERVAL 59CC.00 TO 6000.00

B. CAN-D, Written by F. Narin

1. Introduction

CAN-D (Calibration Analysis - D series) is a FORTRAN IBM 7090 code which simply does a calibration analysis for each channel on a multiplexer in the same manner as is done in RDH-D. The RDH-D write-up specifies the method in detail. Figure 3 is a block diagram of the code. As an example of the speed of the code an 11-step calibration was analyzed on 47 channels of data in the following times:

1.5 minutes with only a summary of each step and channel written on the output tape.

2.0 minutes with a summary plus a listing of all points

2.5 minutes with a summary plus 40/20 output for all points

3.0 minutes with a summary plus 40/20 output and a listing of all points.

The CAN-D code has the following limitations:

1. 25 steps per calibration
2. 50 data points per calibration step
3. 4500 good records on any file of input tape.

Note that at 50 points x 25 steps the maximum number of records that CAN-D can utilize is 1250; there should be no reason to ever have 4500 records on a file.

The following tapes are used by CAN-D and its subroutines:

A1 is the utility tape (Logical tape 1)

A2 is the program and data tape (Logical tape 10)

BLOCK DIAGRAM, CAN-D CODE FOR THE 7090

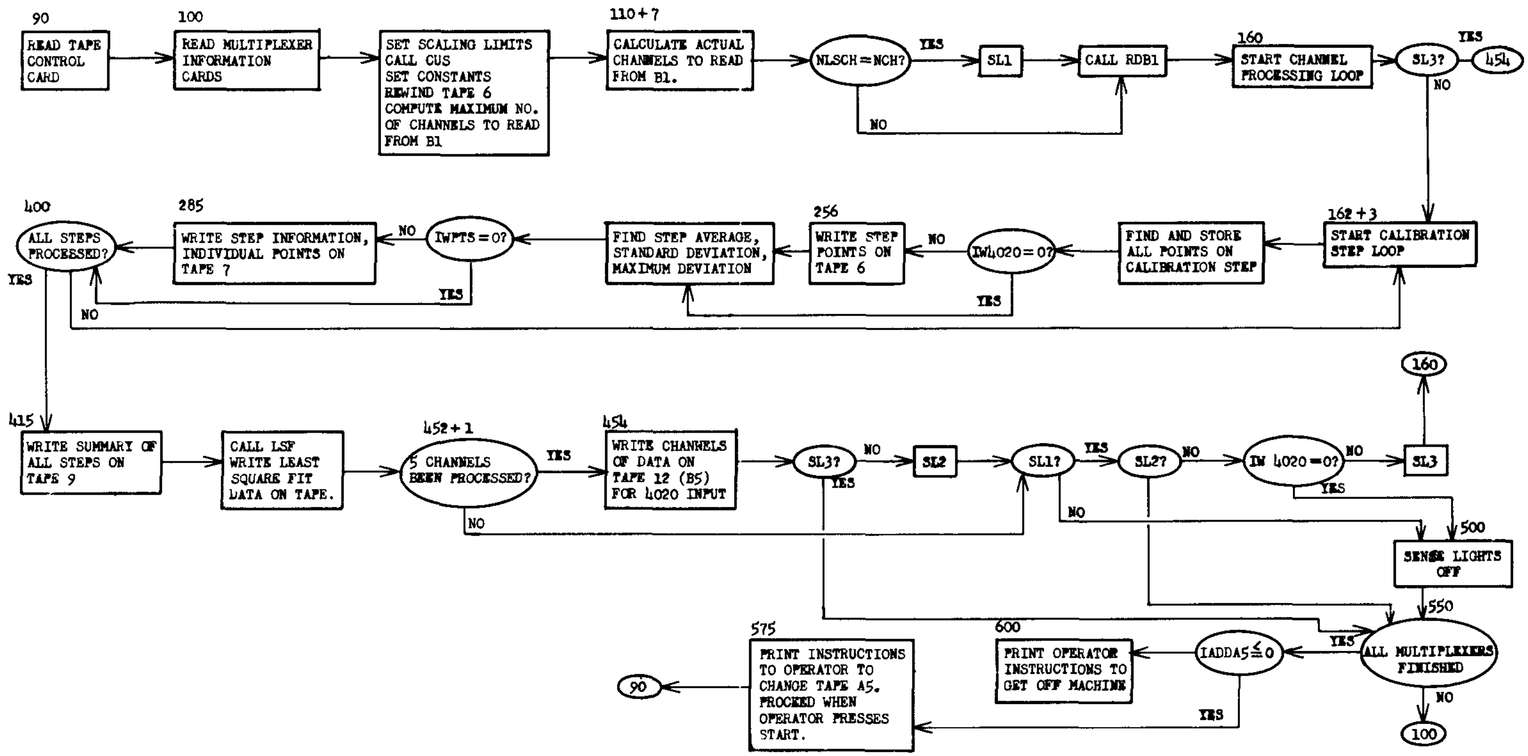


Figure 3

A3 is the BCD output listing tape (Logical tape 9).

A4 is a temporary storage tape used by the CUST subroutine (Logical tape 4).

A5 is the tape unit for the Packard-Bell input tape (Logical tape 2).

B1 is a temporary storage tape written by the CUS and CUST subroutines and read by the RDB1 subroutine (Logical tape 5).

B2 is a temporary storage tape used by CUST and also by the main program (Logical tape 6).

B5 is the 40/20 output tape (Logical tape 12).

A6, B3, B4, B6 are not used. However, B4 is normally reserved for binary card output. Sense Switches are not used.

2. Tape A2 (Control, Labeling) Input

The following are the formats for the CAN-D code:

```
1.  Format (2F10.5, 10I5)
5.  Format (12A6)
7.  Format (7F10.5)
9.  Format (14I5).
```

The following is the input for the CAN-D code:

One card per tape

```
READ INPUT TAPE 10, 9, NMPLX, IADDA5
```

One set per Calibration

```
READ INPUT TAPE 10, 5,(TITLE(I), I = 1, 12)

READ INPUT TAPE 10, 1, TMN, TMX, NCH, NSTP, IWPTS, IW4020,
                       NFLAD, NFLBS, NRCSP

READ INPUT TAPE 10, 7, (TBSTP(I), I = 1, NSTP), SPAN

READ INPUT TAPE 10, 7, (VSTP(I), I = 1, NSTP)
```

Definitions and Comments:

NMPLX = the number of multiplexer calibrations on the tape.

IADDA5 = 0 if this is the last tape, = 1 if there is an additional tape A5.

TITLE = 72-columns of identifying information.

TMN = Minimum time limit for good data times.

TMX = Maximum time limit for good data times.

NCH = Number of channels on the tape (≤ 47).

NSTP = Number of steps in the calibration (≤ 25).

IWPTS = 0 to omit listing individual points, $\neq 0$ to list individual points.

IW4020 = 0 to omit 40/20 output, $\neq 0$ to include 40/20 output.

NFLAD = Number of files to advance on tape A5 before processing.

NFLBS = Number of files to backspace on tape A5 before processing.

NRCSP = Number of records to skip on tape A5 after finding a good record.

TBSTP = Time of the beginning of the I^{th} calibration step.

SPAN = Length of the calibration steps

VSTP = Value of the I^{th} calibration steps, typically 100, 90, 80...

3. Running Description of the Code

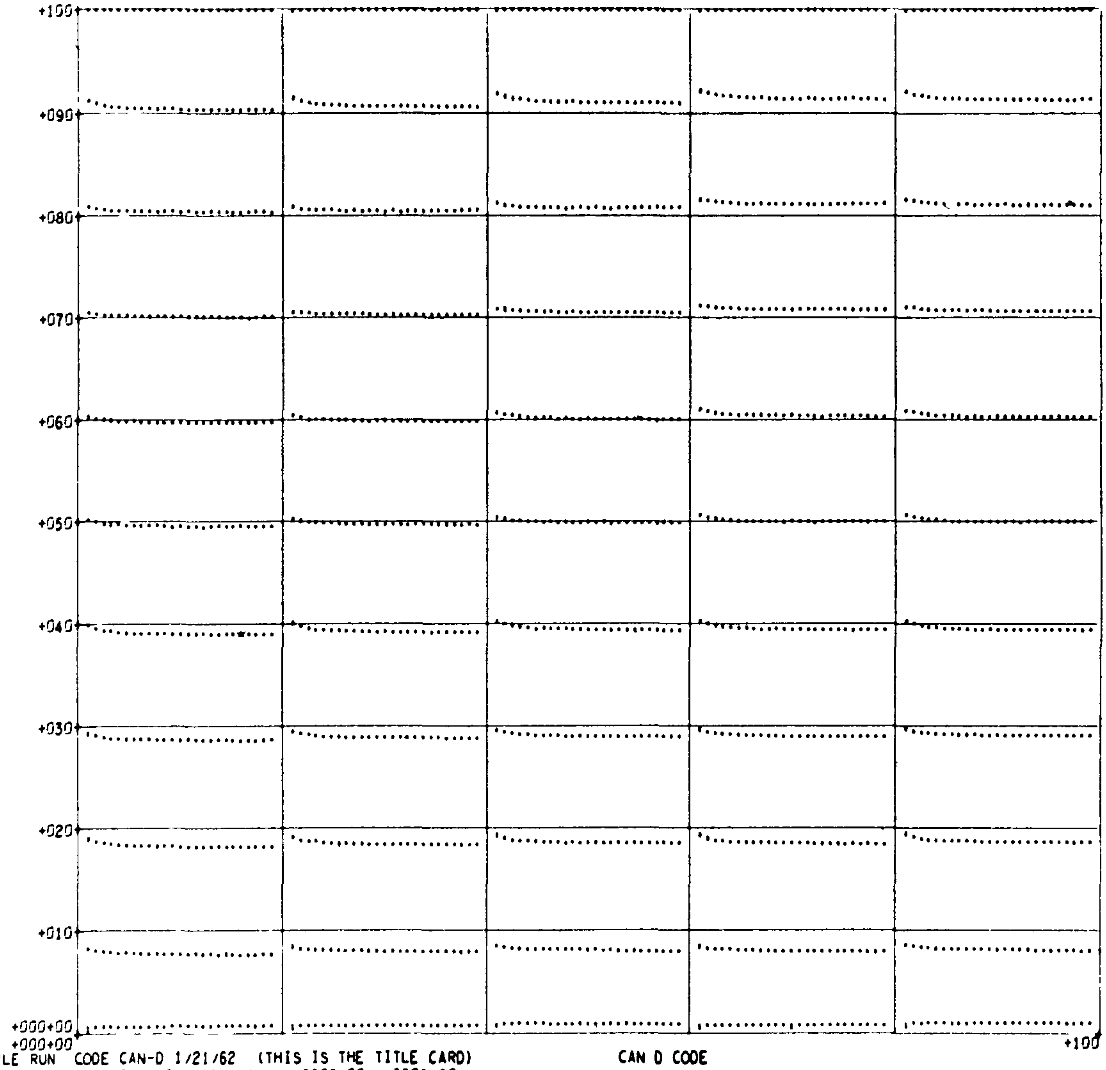
Refer to the CAN-D FORTRAN listings. After reading the input information and setting the 40/20 plotting scaling constants CUS is called by order 110 to convert, normalize, unpack and, if CUST is used, time correct the data from tape A5 and write tape B1. If less than two good records are found on tape A5, this multiplexer is skipped by transferring to order 550. If two or more good records are found, the maximum number of channels to process per tape B1 reading is calculated (NCHMX) and the numbers of the last (NLSCH) channel and first (NFSCH) channel to be processed in their next reading of B1 are calculated. If NLSCH = NCH, that is, if the next reading of B1 is to be last for this multiplexer, SL 1 is turned on. The DØ 472 loop, orders 160 through 472, is very similar to the corresponding loop in RDH-D for calibration analysis; refer to the RDH-D write up for details. After the loop is completed for all channels in this RDB1 reading, SL 1 is tested; if on, SL 2 is tested and if SL 2 is not on, SL 3 is put on and the remaining channels not written on B5 for plotting are put on B5 if IW4020 ≠ 0. If SL 1 is not on, processing continues for the rest of the channels in the multiplexer. If SL 1 and SL 2 are on, or if SL 1 is on and IW4020 = 0, control goes to 550, the end of the multiplexer loop. After all multiplexers on the tape are processed, IADDA5 is tested; if 0, instructions to the operator to get off the machine are printed; if ≠ 0 instructions to change A5 and continue processing are printed.

4. Source Deck (FORTRAN) Listing

```
CCAN D
CTAPE A1 IS THE UTILITY TAPE
CTAPE A2 IS THE PROGRAM TAPE
CTAPE A5 IS THE INPUT TAPE
CTAPES A3 AND B5 ARE THE OUTPUT TAPES
CTAPES A4, B1 AND B2 ARE THE INTERMEDIATE TAPES
CSHARE TWO BOARD
CLIMITED TO 25 STEPS, 47 CHANNELS , 4500 GOOD RECORDS PER FILE
      DIMENSIONTITLE(12),STG(20000),TEMP(50),AVG(25)
     1,TEMP2(100),TBSTP(25),VSTP(25),NPTS(25),SIG(25),XAM(25),B(2)
     2,NCLPTS(5,25),TEMP3(50),TEMP4(50),TMIN(5),TMAX(5),XYBND(4),
     3NCHPLT(5),T1(25),T2(25),XYPLT(4)
    1 FORMAT(2F10.5,10I5)
    2 FORMAT (55H0LESS THAN 2 RECORDS FOUND BY CUS. THIS MULTIPLEXER OUT
     1)
    3 FORMAT(47I1)
    5 FORMAT(12A6)
    6 FORMAT(1H012A6,11H CODE CAN D)
    7 FORMAT(7F10.5)
    8 FORMAT(6H STEP I2,7H TBSTP=F8.1,6H NPTS=I3,6H VSTP=F7.2,
     15H AVG=F7.3,9H MAX DEV=F7.3,5H SIG=F8.4,9H TBG+END=2F9.2)
    9 FORMAT(14I5)
   10 FORMAT(17F7.3)
   11 FORMAT(1H112A6,11H CODE CAN D)
   14 FORMAT(20H0SUMMARY FOR PIN NO.I3,11H CAN D CODE)
   16 FORMAT(1H0I2,20H STEPS    FIT COEFFS=2F9.4,8H  SIGMA=F8.4)
   18 FORMAT(55H0OPERATOR...PUT NEXT REEL OF A5 ON, THEN PRESS START    )
   20 FORMAT(1H1)
   22 FORMAT(55H0OPERATOR...THIS CAN-D RUN IS COMPLETE...SAVE TAPE A5   )
   25 FORMAT(1H 12A6,11H CAN D CODE)
   27 FORMAT(7H PIN NOI3,26H FIRST STEP TMIN AND TMAX=2F10.2)
   30 FORMAT(9H PIN NO. I3)
   90 READINPUTTAPE10,9,NMPLX,IADDA5
      DO5501O=1,NMPLX
  100 READINPUTTAPE10,5,(TITLE(I),I=1,12)
      READINPUTTAPE10,1,TMN,TMX,NCH,NSTP,IWPTS,IW4020,NFLAD,NFLBS,NRCSP
       WRITEOUTPUTTAPE9,11,(TITLE(I),I=1,12)
      XYBND(1)=0.0
      XYBND(2)=100.0
      XYBND(3)=0.0
      XYBND(4)=100.0
      XYPLT(1)=0.0
      XYPLT(2)=100.0
      XYPLT(3)=0.0
      XYPLT(4)=1.0
      PRINT6,(TITLE(I),I=1,12)
      READINPUTTAPE10,7,(TBSTP(I),I=1,NSTP),SPAN
      READINPUTTAPE10,7,(VSTP(I),I=1,NSTP)
  110 CALLCUS(NCH,4500,NFLAD,NFLBS,TM N,TM X,NRCRD,STG,NRCSP)
      IF (NRCRD-1) 111, 111, 112
  111 PRINT 2
      WRITE OUTPUT TAPE 9,2
      GO TO 550
  112 NLSCH=0
      NCLSET=1
      REWIND6
      NCHMX=20000/NRCRD-1
      X=.1*SPAN-.01
      Y=.8*SPAN+.02
      DO500I1=1,47
      NFSCH=NLSCH+1
      NLSCH=NLSCH+NCHMX
```

```
      J=NCHMX
      GOTO145
  140 J=J-1
      NLSCH=NLSCH-1
  145 IF(NLSCH-NCH)150,147,140
  147 SENSELIGHT1
  150 CALLRDB1(NCH,NRCRD,NFSCH,NLSCH,STG)
      NCHP=NLSCH-NFSCH+1
      NWDS=NCHP+1
  160 DO472I2=1,NCHP
      IF(SENSELIGHT3)161,162
  161 SENSELIGHT3
      GOTO454
  162 NPIN=NFSCH+I2
      IB=1
      NO=NWDS*NRCRD
      DO400I3=1,NSTP
      TM N=TBSTP(I3)+X
      TM X=TM N+Y
      I=0
      DO252I4=IB,NO,NWDS
      IF(STG(I4)-TM N)252,200,175
  175 IF(STG(I4)-TM X)200,200,250
  200 I=I+1
      IF(I-50)205,205,203
  203 I=I-1
      GOTO255
  205 J=I4+I2
      TEMP(I)=STG(J)
      TEMP3(I)=STG(I4)
      GOTO252
  250 IF(I)255,252,255
  252 CONTINUE
  255 SUM=0.0
      T1(I3)=TEMP3(1)
      T2(I3)=TEMP3(I)
      IF(IW4020)256,258,256
  256 WRITETAPE6,(TEMP(J),J=1,50),(TEMP3(K),K=1,50)
      NCLPTS(NCLSET,I3)=I
      NCHPLT(NCLSET)=NPIN
  258 Z=I
      NPTS(I3)=I
      DO260I5=1,I
  260 SUM=SUM+TEMP(I5)
      AVG(I3)=SUM/Z
      SUM=0.0
      DO265I5=1,I
      TEMP2(I5)=ABSF(AVG(I3)-TEMP(I5))
  265 SUM=SUM+TEMP2(I5)*TEMP2(I5)
      SIG(I3)=SQRTF(SUM/Z)
      CALLMAX3(TEMP2,-1,I,I5,SUM)
      XAM(I3)=SUM
      IF(IWPTS)275,400,275
  275 IF(I3-1)285,283,285
  283 WRITEOUTPUTTAPE9,30,NPIN
  285 WRITEOUTPUTTAPE9,8,I3,TBSTP(I3),NPTS(I3),VSTP(I3),AVG(I3),XAM(I3),
     1SIG(I3),T1(I3),T2(I3)
      WRITEOUTPUTTAPE9,10,(TEMP(J),J=1,I)
  400 IB=I4-NWDS
  410 WRITEOUTPUTTAPE9,14,NPIN
  415 DO420I=1,NSTP
  420 WRITEOUTPUTTAPE9,8,I,TBSTP(I),NPTS(I),VSTP(I),AVG(I),XAM(I),SIG(I)
     1,T1(I),T2(I)
```

```
      CALLLSF(NSTP,1,AVG,VSTP,B,SIGMA)
  450 WRITEOUTPUTTAPE9,16,NSTP,B(1),B(2),SIGMA
      IF(IW4020)452,472,452
  452 NCLSET=NCLSET+1
      IF(NCLSET-6)472,454,454
  454 I=NCLSET-1
      NCLSET=1
      REWIND6
      CALLADVFLM(0,2)
      CALLSELGRD(5,6)
      CALLLBLGRD(1,6,XYBND)
      DO465I2=1,I
      DO460I3=1,NSTP
      I4=NCLPTS(I2,I3)
      READTAPE6,(TEMP(I5),I5=1,50),(TEMP3(I6),I6=1,50)
      A=TEMP3(1)
      C=(TEMP3(I4)-A)/18.0
      D=1+20*(I2-1)
      IF(I3-1)457,456,457
  456 TMIN(I2)=TEMP3(1)
      TMAX(I2)=TEMP3(I4)
  457 DO458I5=1,I4
  458 TEMP3(I5)=D+(TEMP3(I5)-A)/C
  460 CALLPLFILM(I4,TEMP4,TEMP3,-1,TEMP,-1,XYPLT,42,0)
  465 CONTINUE
      REWIND6
      CALLLINCNT(58)
      WRITEOUTPUTTAPE12,25,(TITLE(J),J=1,12)
      DO468I4=1,I
  468 WRITEOUTPUTTAPE12,27,NCHPLT(I4),TMIN(I4),TMAX(I4)
      IF(SENSELIGHT3)550,470
  470 SENSELIGHT2
  472 CONTINUE
  475 IF(SENSELIGHT1)480,500
  480 IF(SENSELIGHT2)550,482
  482 IF(IW4020)484,550,484
  484 SENSELIGHT3
      GOTO160
  500 SENSELIGHT0
  550 CONTINUE
      IF(IADDA5)600,600,575
  575 PRINT18
      WRITEOUTPUTTAPE9,18
      PRINT20
      PAUSE77777
      GOTO90
  600 PRINT22
      PRINT20
      WRITEOUTPUTTAPE9,22
      PAUSE77777
      GOTO90
      END
```



SAMPLE RUN CODE CAN-D 1/21/62 (THIS IS THE TITLE CARD)
PIN NO 2 FIRST STEP TMIN AND TMAX= 5505.00 5509.80
PIN NO 3 FIRST STEP TMIN AND TMAX= 5505.00 5509.80
PIN NO 4 FIRST STEP TMIN AND TMAX= 5505.00 5509.80
PIN NO 5 FIRST STEP TMIN AND TMAX= 5505.00 5509.80
PIN NO 6 FIRST STEP TMIN AND TMAX= 5505.00 5509.80

5. Sample Output
(Xerox Plot)

```
 SAMPLE RUN  CODE CAN-D 1/21/63  (THIS IS THE TITLE CARD)                CODE CAN D
PIN NO.    2
STEP  1 TBSTP=  5504.0 NPTS= 50 VSTP= 100.00 AVG=  1.019 MAX DEV=  0.006 SIG=  0.0020 TBG+END=  5505.00  5509.80
 1.020  1.020  1.019  1.019  1.020  1.019  1.020  1.020  1.020  1.020  1.021  1.021  1.020  1.020  1.020  1.020  1.020
 1.019  1.019  1.020  1.021  1.020  1.020  1.021  1.021  1.021  1.019  1.019  1.019  1.020  1.020  1.020  1.019  1.019
 1.020  1.019  1.019  1.019  1.018  1.018  1.018  1.015  1.017  1.017  1.016  1.015  1.015  1.014  1.014  1.013
STEP  2 TBSTP=  5514.0 NPTS= 50 VSTP=  90.00 AVG=  0.906 MAX DEV=  0.007 SIG=  0.0020 TBG+END=  5515.00  5519.80
 0.913  0.912  0.911  0.910  0.909  0.908  0.907  0.907  0.907  0.907  0.906  0.906  0.906  0.906  0.906  0.906  0.906
 0.906  0.905  0.906  0.906  0.906  0.906  0.906  0.905  0.905  0.905  0.905  0.905  0.905  0.905  0.905  0.905  0.905
 0.905  0.905  0.905  0.905  0.905  0.905  0.905  0.904  0.905  0.905  0.905  0.905  0.905  0.905  0.905  0.904
STEP  3 TBSTP=  5524.0 NPTS= 50 VSTP=  80.00 AVG=  0.805 MAX DEV=  0.005 SIG=  0.0012 TBG+END=  5525.00  5529.80
 0.810  0.809  0.808  0.808  0.807  0.807  0.806  0.806  0.806  0.806  0.805  0.805  0.805  0.806  0.805  0.805  0.805
 0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805  0.805
 0.804  0.805  0.804  0.804  0.804  0.803  0.804  0.803  0.804  0.804  0.804  0.805  0.804  0.805  0.803  0.804
STEP  4 TBSTP=  5534.0 NPTS= 50 VSTP=  70.00 AVG=  0.702 MAX DEV=  0.003 SIG=  0.0012 TBG+END=  5535.00  5539.80
 0.705  0.705  0.705  0.704  0.703  0.703  0.703  0.703  0.703  0.703  0.703  0.703  0.702  0.702  0.702  0.702  0.702
 0.702  0.702  0.702  0.702  0.702  0.702  0.701  0.702  0.702  0.702  0.702  0.702  0.702  0.701  0.702  0.702  0.702
 0.702  0.701  0.701  0.701  0.701  0.701  0.701  0.701  0.700  0.700  0.705  0.701  0.702  0.701  0.702  0.701
STEP  5 TBSTP=  5544.0 NPTS= 50 VSTP=  60.00 AVG=  0.599 MAX DEV=  0.006 SIG=  0.0013 TBG+END=  5545.00  5549.80
 0.605  0.603  0.601  0.602  0.600  0.602  0.601  0.597  0.599  0.599  0.599  0.599  0.600  0.599  0.598  0.599  0.599
 0.598  0.598  0.598  0.598  0.598  0.598  0.598  0.598  0.599  0.598  0.598  0.598  0.598  0.598  0.598  0.598  0.598
 0.598  0.598  0.598  0.578  0.598  0.598  0.598  0.598  0.598  0.598  0.598  0.598  0.599  0.598  0.598  0.599
STEP  6 TBSTP=  5554.0 NPTS= 50 VSTP=  50.00 AVG=  0.496 MAX DEV=  0.006 SIG=  0.0017 TBG+END=  5555.00  5559.80
 0.502  0.502  0.501  0.500  0.498  0.498  0.499  0.497  0.498  0.498  0.497  0.496  0.496  0.496  0.496  0.495  0.496
 0.496  0.496  0.496  0.495  0.496  0.494  0.495  0.496  0.495  0.495  0.495  0.495  0.495  0.495  0.495  0.495  0.495
 0.495  0.496  0.495  0.495  0.495  0.496  0.496  0.495  0.495  0.495  0.495  0.495  0.496  0.495  0.496  0.495
STEP  7 TBSTP=  5564.0 NPTS= 50 VSTP=  40.00 AVG=  0.391 MAX DEV=  0.008 SIG=  0.0022 TBG+END=  5565.00  5569.80
 0.400  0.399  0.396  0.395  0.394  0.393  0.394  0.393  0.392  0.392  0.392  0.391  0.391  0.391  0.391  0.391  0.391
 0.391  0.391  0.391  0.371  0.391  0.390  0.391  0.391  0.390  0.391  0.390  0.390  0.391  0.391  0.391  0.390  0.390
 0.390  0.390  0.390  0.391  0.391  0.390  0.390  0.390  0.390  0.391  0.390  0.390  0.390  0.390  0.390  0.390
STEP  8 TBSTP=  5574.0 NPTS= 50 VSTP=  30.00 AVG=  0.287 MAX DEV=  0.007 SIG=  0.0017 TBG+END=  5575.00  5579.80
 0.294  0.292  0.292  0.291  0.290  0.289  0.288  0.288  0.288  0.288  0.288  0.287  0.288  0.287  0.287  0.288  0.288
 0.287  0.287  0.287  0.287  0.287  0.287  0.287  0.287  0.286  0.287  0.288  0.287  0.287  0.286  0.287  0.287  0.286
 0.287  0.287  0.287  0.287  0.287  0.286  0.286  0.286  0.286  0.287  0.286  0.286  0.287  0.286  0.287  0.287
STEP  9 TBSTP=  5584.0 NPTS= 50 VSTP=  20.00 AVG=  0.183 MAX DEV=  0.008 SIG=  0.0021 TBG+END=  5585.00  5589.80
 0.191  0.189  0.188  0.187  0.186  0.186  0.185  0.185  0.184  0.184  0.183  0.184  0.183  0.183  0.183  0.183  0.183
 0.183  0.183  0.182  0.183  0.183  0.183  0.183  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182
 0.183  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182  0.182
STEP 10 TBSTP=  5594.0 NPTS= 50 VSTP=  10.00 AVG=  0.078 MAX DEV=  0.006 SIG=  0.0016 TBG+END=  5595.00  5599.80
 0.084  0.083  0.082  0.081  0.081  0.081  0.080  0.080  0.079  0.079  0.079  0.080  0.079  0.079  0.079  0.077  0.077
 0.077  0.079  0.077  0.077  0.077  0.077  0.077  0.077  0.077  0.077  0.077  0.078  0.077  0.077  0.076  0.077  0.077
 0.076  0.076  0.078  0.076  0.077  0.076  0.076  0.076  0.076  0.076  0.076  0.076  0.077  0.077  0.077  0.076
STEP 11 TBSTP=  5604.0 NPTS= 50 VSTP=   0.    AVG=  0.008 MAX DEV=  0.005 SIG=  0.0008 TBG+END-  5605.00  5609.80
 0.003  0.007  0.007  0.007  0.008  0.007  0.008  0.007  0.007  0.007  0.007  0.007  0.007  0.007  0.007  0.007  0.007
 0.007  0.008  0.007  0.007  0.007  0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.007  0.008  0.007  0.008
 0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.008  0.007  0.008  0.008  0.008  0.008  0.008

SUMMARY FOR PIN NO.  2 CAN D CODE
STEP  1 TBSTP=  5504.0 NPTS= 50 VSTP= 100.00 AVG=  1.019 MAX DEV=  0.006 SIG=  0.0020 TBG+END=  5505.00  5509.80
STEP  2 TBSTP=  5514.0 NPTS= 50 VSTP=  90.00 AVG=  0.906 MAX DEV=  0.007 SIG=  0.0020 TBG+END=  5515.00  5519.80
STEP  3 TBSTP=  5524.0 NPTS= 50 VSTP=  80.00 AVG=  0.805 MAX DEV=  0.005 SIG=  0.0012 TBG+END=  5525.00  5529.80
STEP  4 TBSTP=  5534.0 NPTS= 50 VSTP=  70.00 AVG=  0.702 MAX DEV=  0.003 SIG=  0.0012 TBG+END=  5535.00  5539.80
STEP  5 TBSTP=  5544.0 NPTS= 50 VSTP=  60.00 AVG=  0.599 MAX DEV-  0.006 SIG=  0.0013 TBG+END=  5545.00  5549.80
STEP  6 TBSTP=  5554.0 NPTS= 50 VSTP=  50.00 AVG=  0.496 MAX DEV=  0.006 SIG=  0.0017 TBG+END=  5555.00  5559.80
STEP  7 TBSTP=  5564.0 NPTS= 50 VSTP=  40.00 AVG=  0.391 MAX DEV=  0.008 SIG=  0.0022 TBG+END=  5565.00  5569.80
STEP  8 TBSTP=  5574.0 NPTS= 50 VSTP=  30.00 AVG=  0.287 MAX DEV=  0.007 SIG=  0.0017 TBG+END=  5575.00  5579.80
STEP  9 TBSTP=  5584.0 NPTS= 50 VSTP=  20.00 AVG=  0.183 MAX DEV=  0.008 SIG=  0.0021 TBG+END=  5585.00  5589.80
STEP 10 TBSTP=  5594.0 NPTS= 50 VSTP=  10.00 AVG=  0.078 MAX DEV=  0.006 SIG=  0.0016 TBG+END=  5595.00  5599.80
STEP 11 TBSTP=  5604.0 NPTS= 50 VSTP=   0.    AVG=  0.008 MAX DEV=  0.005 SIG=  0.0008 TBG+END=  5605.00  5609.80

11 STEPS    FIT COEFFS=    1.4507   97.5418   SIGMA=   0.8202
PIN NO.    3
STEP  1 TBSTP=  5504.0 NPTS= 50 VSTP= 100.00 AVG=  1.017 MAX DEV=  0.001 SIG=  0.0008 TBG+END=  5505.00  5509.80
```

C. B6TØA3, Written by F. Narin

1. Description

B6TØA3 is a simple code for writing on tape A3 for listing the (binary) contents of any channel on tape B6. Input is by means of the FORTRAN statement

```
READ INPUT TAPE 10, 1, NCHB6, TMIN, TMAX
1 = 1 FØRMAT (I5, 2F10.5)
```

NCHB6 = Serial number of the desired channel on tape B6

TMIN = Lower limit of times (and corresponding data points) to be written on A3.

TMAX = Upper limit of times (and corresponding data points) to be written on A3.

Channel after channel is written on A3 until NCHB6 = 0 on an input card. Note that there is no restriction on the order of reading channels from B6; however, much computer time may be wasted by reading out of serial order.

2. Source Deck (FORTRAN) Listing

```
CB6T0A3
      DIMENSIONTSTG(4500),YSTG(4500),REC(12),TITLE(6)
    1 FORMAT(I5,2F10.5)
    2 FORMAT(I5,1H 12A6,1H 6A6)
    3 FORMAT(5(0PF13.3,1PE11.3))
    4 FORMAT(55H0 OPERATOR...THIS B6T0A3 RUN IS OVER...SAVE TAPE B6     )
    5 FORMAT(1H0)
    6 FORMAT(21H0CODE B6 TO A3 NCHB6=I5,6H TMIN=F10.2,6H TMAX=F10.2)
  100 READINPUTTAPE10,1,NCHB6,TMIN,TMAX
      WRITEOUTPUTTAPE9,6,NCHB6,TMIN,TMAX
      IF(NCHB6)105,200,105
  105 DO110I=1,4500
      YSTG(I)=0.0
  110 TSTG(I)=0.0
  112 CALLRDB6(NCHB6,NPTS,TMIN,TMAX,TSTG,YSTG,REC,TITLE,1)
      WRITEOUTPUTTAPE9,2,NPTS,REC,TITLE
      DO125I=1,NPTS,5
      SUM=0.0
      I1=I
      I2=I+4
      DO116I3=I1,I2
  116 SUM=SUM+TSTG(I3)
      IF(SUM)118,125,118
  118 WRITEOUTPUTTAPE9,3,(TSTG(I3),YSTG(I3),I3=I1,I2)
  125 CONTINUE
      GOTO100
  200 WRITEOUTPUTTAPE9,4
      PRINT4
      PRINT5
      PRINT5
      PAUSE77777
      GOTO100
      END
```

D. CUS AND CUST SUBROUTINES, Written by F. Narin

1. Description

a. Purpose: To convert, unpack, normalize, and, for CUST only, time correct the Packard-Bell data input tape A5 and write an intermediate tape B1 for use in data processing.

b. Calling Sequence: CALL CUS(NCH, NRCMX, NFLAD, NFLBS, TMIN, TMAX, NRCRD, STG, NRCSP)

NCH = Number of channels on tape A5 to be processed, ranging between 1 and 47.

NRCMX = Maximum number of tape A5 records to be written on the intermediate tape, ranging from 1 to 9600.

NFLAD = Number of files to advance on tape A5 before starting to search for data. Note that at the end of a search the tape is positioned at the beginning of the next file, so that for a normal sequence of processing one file after the other NFLAD = 0.

NFLBS = Number of files to backspace on tape A5 before starting to search for data.

TMIN = Lower limit on good times for searching tape A5. All records for which the time is less than TMIN are not processed.

TMAX = Upper limit on good times for searching tape A5. All records for which the time is greater than TMAX are not processed.

NRCRD = Number of records actually converted by CUS. This is an output number.

STG = A location in the main program used by CUS for temporary storage. STG must be dimensioned 9601 or larger in the calling program. Contents of locations STG(1) to STG(9601) will be destroyed by CUS.

NRCSP = Number of records to skip (on tape A5) after finding a good record. This feature permits the time editing of the Packard-Bell tape on line. However, on-line time editing is an expensive waste of computer time; therefore, this should be used only on exceptional cases.

c. Length and Language: CUS $557_8 = 367_{10}$; FAP language

CUST $1330_8 = 728_{10}$; FAP language

d. Details: CUS takes the low density, packed, BCD, one-record per multiplexer frame Packard-Bell tape A5 and converts it to a high density, unpacked, binary, relatively end-of-record gap-free, normalized tape B1 for use in further data processing. Tape B1 is then read by the subroutine RDB1. Consider the following example:

Tape A5 (Low Density, BCD)

Record 1:	396402	990	085	101	155	164	318
Record 2:	396402	989	086	102	153	165	320
3:	396402	989	086	105	155	166	322
4:	396402	991	083	104	156	167	324
5:	396404	990	079	102	157	168	326
6:	296486	800	210	290	261	283	201
7:	396404	991	081	104	158	169	328
8:	396404	992	084	105	159	170	330
9:	396404	995	083	106	160	171	332
10:	396406	990	083	107			
11:	396406	991	090	108	162	173	330
12:	396406	989	085	109	163	174	328
13:	396406	988	086	110	164	175	326
14:	396408	986	087	112	165	176	324
15:	396408	990	085	112	166	177	322
16:	396408	990	081	113	167	178	321
17:	396408	991	091	114	168	179	320

Here the first 6 digits represent time (396402 = 39640.2 seconds), digits 7 to 9 are full scale, digits 10 to 12 are zero scale, digits 13 to 15 are data channel 1, digits 16 to 18 are data channel 2, etc. After CUST has been called, with the following calling sequence,

CALL CUS(3, 6, 0, 0, 39000.,40000.,NRCRD, STG, 1)

the following will be written on tape B1 as a single record:

39640.2	.0178	.0773	.0873	
39640.3[1]	.0275	.0825	.0946	
39640.4	.0252	.0856	.0977	(all numbers to full 7090 accuracy)
39640.5[2]	.0231	.0826	.0947	
39640.6	.0265	-.0915	-.0915	
39640.7[3]	.0265	.0863	.0985	

Note that a normalized data point is the data point minus zero scale divided by full scale minus zero scale. Note also that short records are filled in with zero as the data point, and long records are truncated. Normally, there will be one end-of-record gap on tape B1 per 200 records on tape A5. However, if there are many bad times on tape A5 and CUST is used, there can be some additional end-of-record gaps written on B1. The subroutine RDB1 ignores all end-of-record gaps. Figures 4, 5 and 6 are block diagrams of CUS subroutine, CUST subroutine part 1, and CUST subroutine, part 2 respectively.

e. Data Code Usage: CUS or CUST are used by the main data codes as the first steps in data processing. Both CAN-D and RDH-D call CUS or CUST once for each multiplexer of data; all further processing is from the CUS or CUST output tape B1 by means of the RDB1 subroutine.

---

1. CUS will leave this number as 39640.2
2. CUS will leave this number as 39640.4
3. CUS will leave this number as 39640.6

BLOCK DIAGRAM OF CUS SUBROUTINE

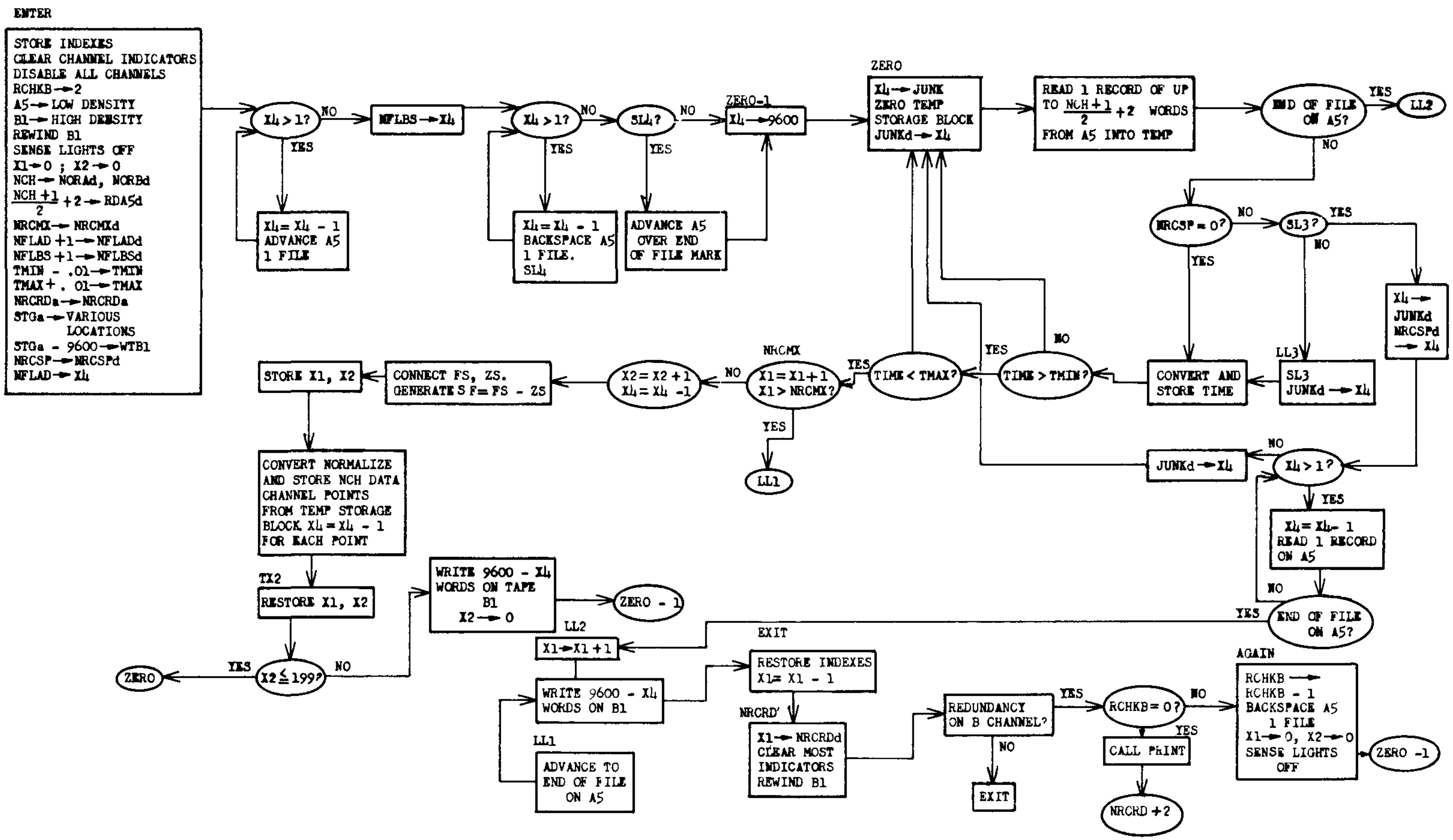


Figure 4
(Some Redundancy Checking Omitted)

BLOCK DIAGRAM OF CUST SUBROUTINE

PART 1 — UNTIL TIME CORRECTING

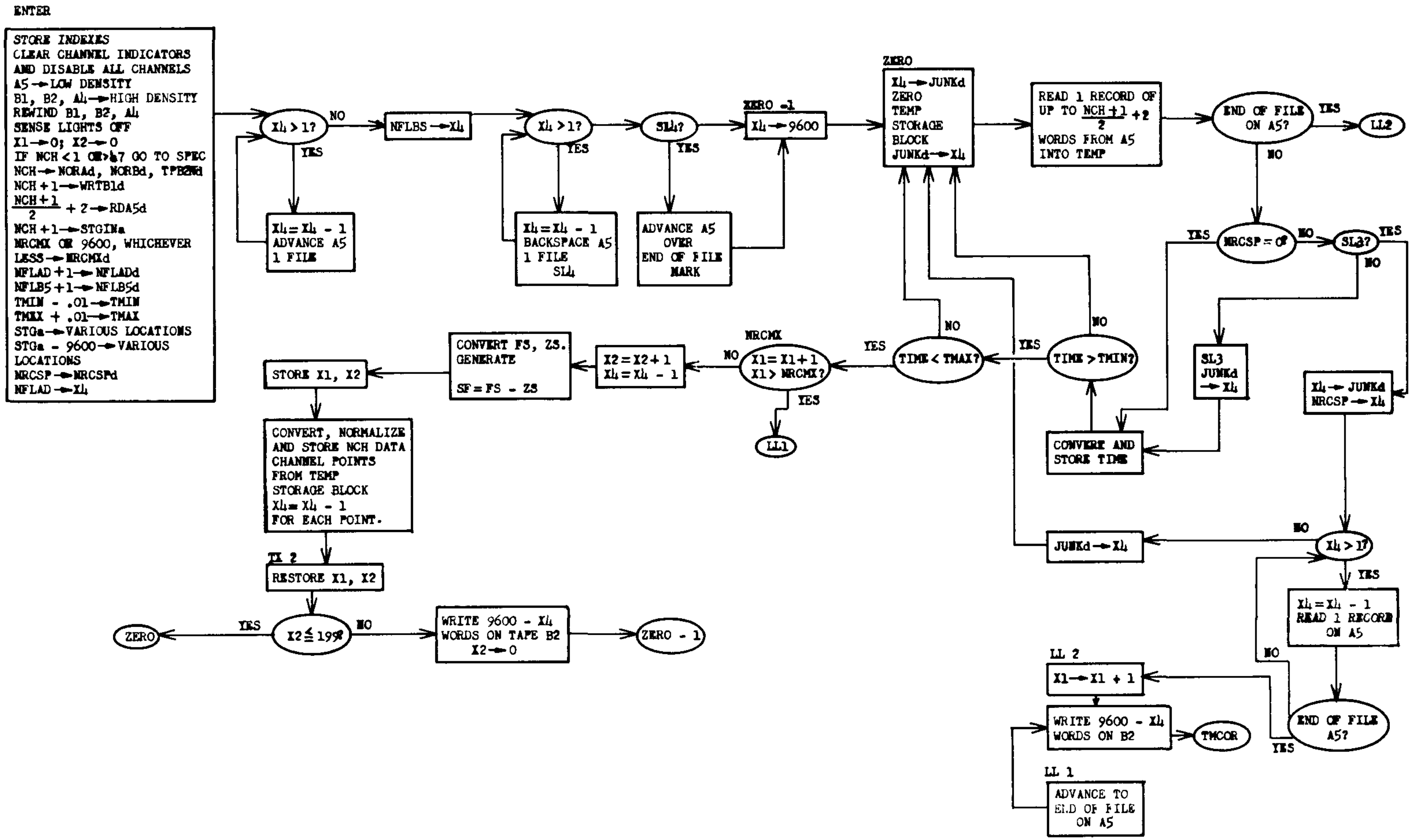


Figure 5
(Some Redundancy Checking Omitted)

BLOCK DIAGRAM, CUST SUBROUTINE

PART 2: TIME CORRECTING TO END OF PROGRAM

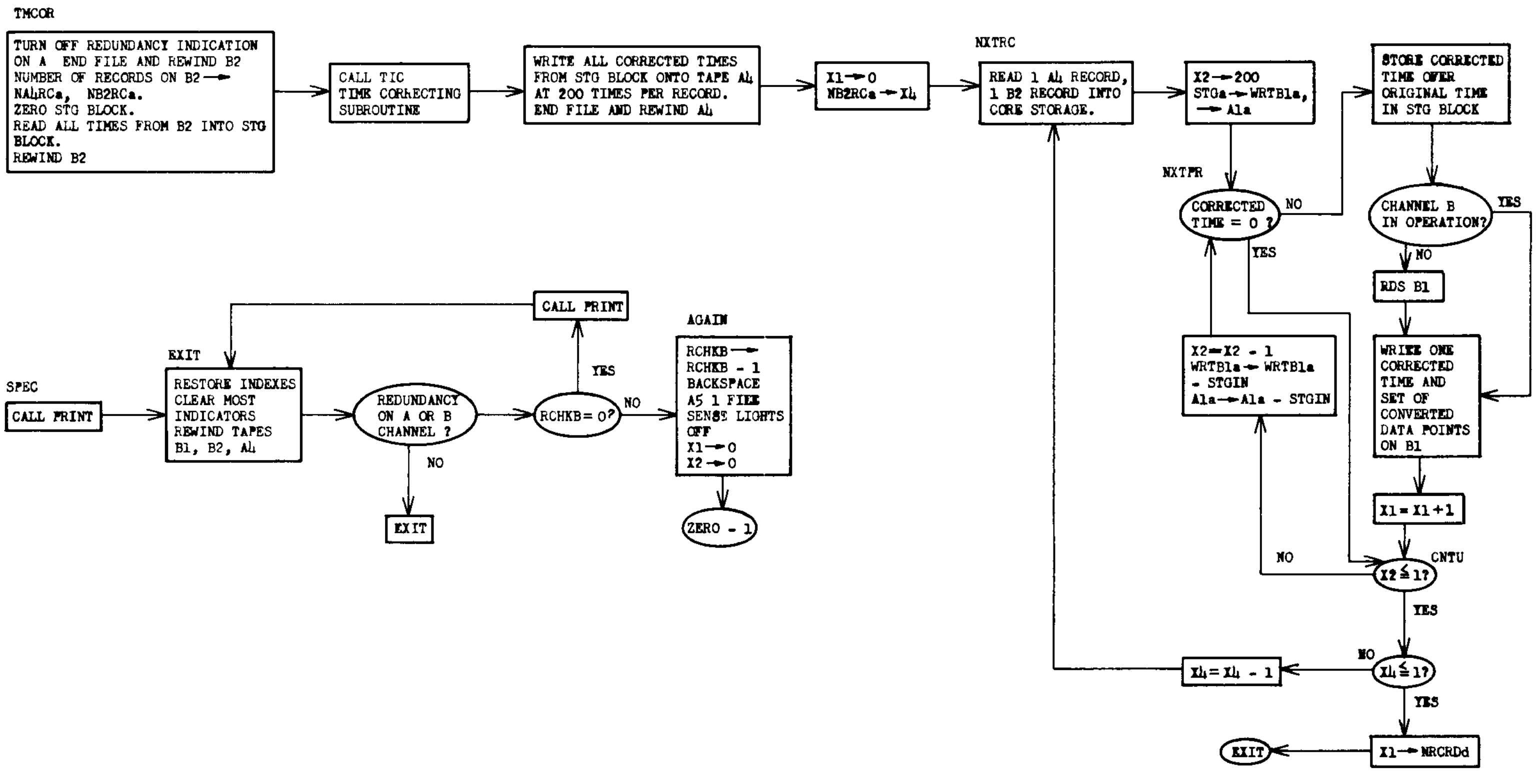


Figure 6
(Some Redundancy Checking Omitted)

In a typical case tape B1 is approximately 1/10th the length of tape A5; this, plus the fact that B1 is converted to binary and unpacked, saves a large amount of 7090 time. Either CUS or CUST may be used at any time. CUST takes slightly longer in that it makes two extra passes through the output data in order to do the time correcting, and it calls the time correcting subroutine TIC once. It must be pointed out that CUST eliminates all tape A5 records for which the times appear to be bad, and there have been tapes from the Packard-Bell with odd time patterns which TIC thought were all bad, with the result that TIC eliminated almost all data. Thus, if a tape can not be read by CUST, and if the time correcting is not important, the use of CUS is indicated. If time correcting is important, a new tape A5 should be made.

2. Source Deck (FAP) Listing

```
*CUST
        COUNT   400
*CALLING SEQ. CALLCUS(NCH,NRCMX,NFLAD,NFLBS,TMIN,TMAX,NRCRD,STG,NRCSP)
*LIMITED TO 47 CHANNELS,9600 RECORDS ACCEPTED IN FIRST PASS
*ALL SENSE LIGHTS ARE TURNED OFF BY CUS
*SL3=RECORD SKIPPING INDICATOR
*SL4=BSFA 5 RECORD SKIP INDICATOR
*STG MUST BE DIMENSIONED 9601 OR GREATER IN CALLING PROGRAM
*INDEXA=TOTAL NO OF RECORDS COUNT
*INDEXB=TEMPORARY RECORD COUNT
*INDEXC=WORD COUNT FOR STORAGE PLUS INCIDENTALS
*A5 IS THE INPUT TAPE
*B1 IS THE OUTPUT TAPE
*B2 AND A4 ARE THE INTERMEDIATE TAPES
        ENTRY   CUS
 CUS    SXA     EXIT,1
        SXA     EXIT+1,2
        SXA     EXIT+2,4
        SXA     REPT,4
        STZ     JUNK
        ENB     JUNK
        CLA     OCT2
        STO     RCHKB
 BEG2   TRCA    *+1
        TRCB    *+1
        TEFA    *+1
        TEFB    *+1
        SDLA    5
        SDHB    1
        SDHB    2
        SDHA    4
        REWB    1
        REWB    2
        REWA    4
        SLF
        AXT     0,1
        AXT     0,2
        CLA*    1,4
        TZE     SPEC
        TMI     SPEC
        SUB     OCT47
        TPL     SPEC
        CLA*    1,4
        STD     NORA
        STD     NORB
        STD     TPB2N
        ADD     OCT1
        STD     WRTB1
        ARS     1
        ADD     OCT2
        STD     RDA5
        CLA*    1,4
        ADD     OCT1
        ARS     18
        STA     STGIN
        CLA*    2,4
        SUB     OCTN
        TPL     *+4
        CLA*    2,4
        STD     NRCMX
        TRA     *+3
        CLA     OCTN
```

```
        STD     NBCMX
        CLA*    3,4
        ADD     OCT1
        STD     NFLAD
        CLA*    4,4
        ADD     OCT2
        STD     NFLBS
        CLA*    5,4
        FSB     TENM2
        STO     TMIN
        CLA*    6,4
        FAD     TENM2
        STO     TMAX
        CLA     7,4
        STA     NRCRD
        CLA     8,4
        STA     STG1
        STA     STG2
        STA     STG3
        STA     STG4
        STA     STG5
        STA     STG6
        STA     STG7
        STA     TICL+1
        SUB     OCTM
        STA     STGAD
        STA     WRTA4
        STA     WRTB2
        STA     RDB2
        STA     WRTB1
        CLA*    9,4
        STD     NRCSP
        LXD     NFLAD,4
        TNX     *+5,4,1
        RTDA    5
        TCOA    *
        TEFA    *-3
        TRA     *-3
        LXD     NFLBS,4
        TNX     *+5,4,1
        SLN     4
        BSFA    5
        TCOA    *
        TRA     *-4
        SLT     4
        TRA     *+4
        RTDA    5
        TCOA    *
        TEFA    *+1
        AXT     9600,4
ZERO    SXD     JUNK,4
        AXT     26,4
        STZ     TEMP+26,4
        TIX     *-1,4,1
        LXD     JUNK,4
        TRCA    *+1
        AXT     1,4
XX      RTDA    5
        RCHA    RDA5
        TCOA    *
        TRCA    *+2
        TRA     X4S
        TXH     *+3,4,1
```

```
        BSRA    5
        TXI     XX,4,1
        TSX     $(SPH),4
        NOP     FMA
        SXD     IVAN,1
        AXT     27,1
        LDQ     TEMP+27,1
        STR
        TIX     *-2,1,1
        LXD     IVAN,1
        TSX     $(FIL),4
X4S     LXD     JUNK,4
        TEFA    LL2
        NZT     NRCSP
        TRA     CONV
        SLT     3
        TRA     LL3
        SXD     JUNK,4
        LXD     NRCSP,4
        TIX     *+3,4,1
        LXD     JUNK,4
        TRA     ZERO
        RTDA    5
        TCOA    *
        TEFA    *+2
        TRA     *-6
        LXD     JUNK,4
        TRA     LL2
LL3     SLN     3
        LXD     JUNK,4
CONV    STZ     JUNK
        PXD     ,,
        LDQ     TEMP
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        XCA
        FMP     TEN2
        STO     SF
        STZ     JUNK
        PXD     ,,
        LDQ     TEMP
        RQL     18
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        XCA
        FMP     TENMO
        FAD     SF
STG1    STO     ,4
        FSB     TMIN
        TMI     ZERO
STG2    CLS     ,4
        FAD     TMAX
        TMI     ZERO
        TXI     *+1,1,1
NRCMX   TXH     LL1,1,0
        TXI     *+1,2,1
        TIX     *+1,4,1
```

```
        STZ     JUNK
        PXD     ,,
        LDQ     TEMP+1
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        STO     FS
        PXD     ,,
        STZ     JUNK
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        STO     ZS
        FSB     FS
        CHS
        STO     SF
        SXD     TEMP,1
        SXD     TEMP+1,2
        AXT     0,1     XA=CHANNEL COUNT
        AXT     24,2    XB=TEMP STG COUNT
        TXI     *+1,1,1
NORA    TXH     TX2,1,0
        TXI     *+1,2,-1
        STZ     JUNK
        PXD     ,,
        LDQ     TEMP+25,2
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        FSB     ZS
        FDP     SF
STG3    STQ     ,4
        TIX     *+2,4,1
        AXT     0,4
        TXI     *+1,1,1
        STZ     JUNK
NORB    TXH     TX2,1,0
        PXD     ,,
        LDQ     TEMP+25,2
        RQL     18
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        FSB     ZS
        FDP     SF
STG4    STQ     ,4
        TIX     *+1,4,1
        TXI     NORA,1,1
TX2     LXD     TEMP,1
        LXD     TEMP+1,2
        TXL     ZERO,2,199
        STZ     JUNK
        SXD     JUNK,4
        CLA     OCTN
        SUB     JUNK
```

```
        STO     WRTB2
        TSX     B2WT,4
        AXT     0,2
        TRA     ZERO-1
LL2     TXI     *+1,1,1
        TIX     *+1,1,1
        STZ     JUNK
        SXD     JUNK,4
        CLA     OCTN
        SUB     JUNK
        STO     WRTB2
        TSX     B2WT,4
        TRA     TMCOR
LL1     RTDA    5
        TCOA    *
        TEFA    LL2+1
        TRA     *-3
TMCOR   TCOB    *
        WEFB    2
        REWB    2
        TRCA    *+1
        AXT     0,2
        PXA     ,1
        SUB     OC310
        TPL     *+2
        TRA     *+2
        TXI     *-3,2,1
        TXI     *+1,2,1
        SXA     NA4RC,2
        SXA     NB2RC,2
        AXT     9600,4
STG5    STZ     ,4
        TIX     *-1,4,1
STG6    STZ     ,
        AXT     9600,4
        RTBB    2
        RCHB    TIME
TMRD    LCHB    TPB2N
        CLA     TEMP
STG7    STO     ,4
        TIX     *+2,4,1
        AXT     0,4
        TNX     *+3,1,1
        LCHB    TIME
        TRA     TMRD
        TCOB    *
        REWB    2
TICL    CALL    TIC,0
NA4RC   AXT     ,4
        AXT     5,1
WA4     WTBA    4
        RCHA    WRTA4
        TCOA    *
        TRCA    A4CK
        CLA     WRTA4
        ADD     OC310
        STO     WRTA4
        TNX     B4WE,4,1
        TRA     WA4
A4CK    BSRA    4
        WTBA    4
        TCOA    *
        TRCA    *+2
```

```
         TIX    WA4,1,1
         SXA    EXLT,4
         TSX    $(SPH),4
         NOP    FMS
         TSX    $(FIL),4
         CLA    STGAD
         STA    WRTA4
EXLT     AXT    XX,4
         RUNA   4
         HTR    NA4RC
FMS      BCI    7,(36H  TAPE A4 BAD. CHANGE AND PUSH START )
B4WE     WEFA   4
         REWA   4
         AXT    0,1
NB2RC    AXT    ,4
NXTRC    TCOA   *
         TCOB   *
         RTBA   4
         RCHA   RDA4
         RTBB   2
         RCHB   RDB2
         TCOA   *
         TCOB   *
         AXT    200,2
         CLA    STGAD
         STA    WRTB1
         STA    A1
NXTPR    CLA    TEMP+200,2
         TZE    CNTU
A1       STO    ,
         TCOB   *+4
         WTBB   1
         RCHB   WRTB1
         TRA    *+4
         LCHB   WRTB1
         IOT
         TRA    *-5
         TXI    *+1,1,1
CNTU     TNX    EOF,2,1
         CLA    WRTB1
         SUB    STGIN
         STO    WRTB1
         CLA    A1
         ADD    STGIN
         STO    A1
         TRA    NXTPR
EOF      TCOB   *
         TRCB   *+3
         TNX    NRCRD,4,1
         TRA    NXTRC
         LXD    WRTB1,2          NCH+1
         SXD    *+5,2
         LXA    STGAD,2          STG-9600
         SXA    *+4,2
         SXA    WRTB1,2      WRITE ADDRESS
         LXA    A1,2
XNC      TXI    *+1,2,**        SET TO NCH + 1
         TIX    *+1,2,**          SET TO STGAD
         SXD    WRTB1,2
RSTT     AXT    5,2
         BSRB   1
         WTBB   1
         TCOB   *
```

```
         TRCB    TRBLE
         WTBB    1
         RCHB    WRTB1
         TCØB    *
         TRCB    *+4
         LXD     XNC,2
         SXD     WRTB1,2
         TRA     EØF+2
         TIX     RSTT+1,2,1
TRBLE    SXA     XR4,4
         TSX     (SPH),4
         NØP     BFMT
         TSX     (FIL),4
XR4      AXT     **,4
         REWB    2
         CLA     STGAD
         STA     WRTA4
         RUNB    1
         HTR     NB2RC-2
BFMT     BCI     7,(36H  TAPE B1 BAD. CHANGE AND PUSH START )
NRCRD    SXD     ,1
EXIT     AXT     0,1
         AXT     0,2
         AXT     0,4
         SLF
         TCØB    *
         WEFB    1
         REWB    1
         REWB    2
         REWA    4
         TEFA    *+1
         TEFB    *+1
         TRCA    *+3
         TRCB    *+2
         TRA     10,4
         ZET     RCHKB
         TRA     AGAIN
         CALL    PRINT,ØCT3
         TRA     EXIT
SPEC     CALL    PRINT,ØCT1
         TRA     EXIT
AGAIN    CLA     RCHKB
         SUB     ØCT1
         STØ     RCHKB
         BSFA    5
         BSFA    5
         RTDA    5
         TCØA    *
REPT     AXT     0,4
         STZ*    3,4
         STZ*    4,4
         TRA     BEG2
RDA5     IØST    TEMP,,1
WRTA4    IØCD    ,,200
WRTB1    IØCT    ,,
WRTB2    IØCD    ,,
RDA4     IØCD    TEMP,,200
RDB2     IØST    ,,9600
TIME     IØCT    TEMP,,1
TPB2N    IØCTN   TEMP+1,,0
TEMP     BSS     200
 IVAN    BSS     1
SIGIN    HTR     ,
```

```
STGAO  HTR    ,
NFLAD  HTR    ,
NFLBS  HTR    ,
TMIN   HTR    ,
TMAX   HTR    ,
JUNK   HTR    ,
FS     HTR    ,
ZS     HTR    ,
SF     HTR    ,
NRCSP  HTR    ,
RCHKB  HTR    ,
H      HTR    TEN
       HTR    TEN,,100
       HTR    TEN,,200
       HTR    TEN,,300
       HTR    TEN,,400
       HTR    TEN,,500
       HTR    TEN,,600
       HTR    TEN,,700
       HTR    TEN,,800
       HTR    TEN,,900
TEN    HTR    ONE
       HTR    ONE,,10
       HTR    ONE,,20
       HTR    ONE,,30
       HTR    ONE,,40
       HTR    ONE,,50
       HTR    ONE,,60
       HTR    ONE,,70
       HTR    ONE,,80
       HTR    ONE,,90
ONE    HTR
       HTR    ,,1
       HTR    ,,2
       HTR    ,,3
       HTR    ,,4
       HTR    ,,5
       HTR    ,,6
       HTR    ,,7
       HTR    ,,8
       HTR    ,,9
TEN2   DEC    100.
TENMO  DEC    .1
TENM2  DEC    .01
EXP    OCT    221000000000
OCT1   OCT    000001000000
OCT2   OCT    000002000000
OCT3   OCT    000003000000
OCT47  OCT    000057000001
OC310  OCT    000000000310
OCTN   OCT    022600000000
OCTM   OCT    000000022600
B2WT   TCOB   *
       TRCB   *+1
       AXT    5,2
       WTBB   2
       RCHB   WRTB2
       TCOB   *
       TRCB   *+2
       TRA    1,4
       BSRB   2
       WTBB   2
       TCOB   *
```

```
        TRCB    *+2
        TIX     B2W1+3,2,1
        TSX     $(SPH),4
        NOP     FMT
        TSX     $(FIL),4
        RUNB    2
        HTR     AGAIN
FMT     BCI     7,(36H  TAPE B2 BAD. CHANGE AND PUSH START )
FMA     BCI     6,(31H 2ND RED. CK A5 COND IGNORED.
        BCI     3,//1H 14A6/1H 13A6)
        END
```

```
*CUS
         COUNT   350
*CALLING SEQ. CALLCUS(NCH,NRCMX,NFLAD,NFLBS,TMIN,TMAX,NRCRD,STG,NRCSP)
*SL3=RECORD SKIPPING INDICATOR
*SL4=BSFA 5 RECORD SKIP INDICATOR
*STG MUST BE DIMENSIONED 9601 OR GREATER IN CALLING PROGRAM
*INDEXA=TOTAL NO OF RECORDS COUNT
*INDEXB=TEMPORARY RECORD COUNT
*INDEXC=WORD COUNT FOR STORAGE PLUS INCIDENTALS
         ENTRY   CUS
 CUS     SXA     EXIT+3,1
         SXA     EXIT,2
         SXA     EXIT+4,4
         SXA     REPT,4
         STZ     JUNK
         ENB     JUNK
         CLA     OCT2
         STO     RCHKB
 BEG2    TRCA    *+1
         TRCB    *+1
         TEFA    *+1
         TEFB    *+1
         SDLA    5
         SDHB    1
         REWB    1
         SLF
         AXT     0,1
         AXT     0,2
         CLA*    1,4
         STD     NORA
         STD     NORB
         ADD     OCT1
         ARS     1
         ADD     OCT2
         STD     RDA5
         CLA*    2,4
         STD     NRCMX
         CLA*    3,4
         ADD     OCT1
         STD     NFLAD
         CLA*    4,4
         ADD     OCT2
         STD     NFLBS
         CLA*    5,4
         FSB     TENM2
         STO     TMIN
         CLA*    6,4
         FAD     TENM2
         STO     TMAX
         CLA     7,4
         STA     NRCRD
         CLA     8,4
         STA     STG1
         STA     STG2
         STA     STG3
         STA     STG4
         SUB     OCTM
         STA     WTB1
         STZ     NRCSP
         CLA*    9,4
         STD     NRCSP
         LXD     NFLAD,4
```

```
        TNX    *+5,4,1
        RTDA   5
        TCOA   *
        TEFA   *-3
        TRA    *-3
        LXD    NFLBS,4
        TNX    *+5,4,1
        SLN    4
        BSFA   5
        TCOA   *
        TRA    *-4
        SLT    4
        TRA    *+4
        RTDA   5
        TCOA   *
        TEFA   *+1
        AXT    9600,4
ZERO    SXD    JUNK,4
        AXT    26,4
        STZ    TEMP+26,4
        TIX    *-1,4,1
        LXD    JUNK,4
        TRCA   *+1
        AXT    1,4
XX      RTDA   5
        RCHA   RDA5
        TCOA   *
        TRCA   *+2
        TRA    X4S
        TXH    *+3,4,1
        BSRA   5
        TXI    XX,4,1
        TSX    $(SPH),4
        NOP    FMA
        SXD    IVAN,1
        AXT    27,1
        LDQ    TEMP+27,1
        STR
        TIX    *-2,1,1
        LXD    IVAN,1
        TSX    $(FIL),4
X4S     LXD    JUNK,4
        TEFA   LL2
        NZT    NRCSP
        TRA    CONV
        SLT    3
        TRA    LL3
        SXD    JUNK,4
        LXD    NRCSP,4
        TIX    *+3,4,1
        LXD    JUNK,4
        TRA    ZERO
        RTDA   5
        TCOA   *
        TEFA   *+2
        TRA    *-6
        LXD    JUNK,4
        TRA    LL2
LL3     SLN    3
        LXD    JUNK,4
CONV    STZ    JUNK
        PXD
        LDQ    TEMP
```

```
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        XCA
        FMP     TEN2
        STO     SF
        STZ     JUNK
        PXD
        LDQ     TEMP
        RQL     18
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        XCA
        FMP     TENMO
        FAD     SF
STG1    STO     ,4
        FSB     TMIN
        TMI     ZERO
STG2    CLS     ,4
        FAD     TMAX
        TMI     ZERO
        TXI     *+1,1,1
NRCMX   TXH     LL1,1,0
        TXI     *+1,2,1
        TIX     *+1,4,1
        STZ     JUNK
        PXD
        LDQ     TEMP+1
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        STO     FS
        PXD
        STZ     JUNK
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        STO     ZS
        FSB     FS
        CHS
        STO     SF
        SXD     TEMP,1
        SXD     TEMP+1,2
        AXT     0,1      XA=CHANNEL COUNT
        AXT     24,2     XB=TEMP STG COUNT
        TXI     *+1,1,1
NORA    TXH     TX2,1,0
        TXI     *+1,2,-1
        STZ     JUNK
        PXD
        LDQ     TEMP+25,2
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
```

```
        ARS     8
        ORA     EXP
        FSB     ZS
        FDP     SF
STG3    STQ     ,4
        TIX     *+2,4,1
        AXT     0,4
        TXI     *+1,1,1
        STZ     JUNK
NORB    TXH     TX2,1,0
        PXD
        LDQ     TEMP+25,2
        RQL     18
        CAQ     H,,3
        STD     JUNK
        CLA     JUNK
        ARS     8
        ORA     EXP
        FSB     ZS
        FDP     SF
STG4    STQ     ,4
        TIX     *+1,4,1
        TXI     NORA,1,1
TX2     LXD     TEMP,1
        LXD     TEMP+1,2
        TXL     ZERO,2,199
        STZ     JUNK
        SXD     JUNK,4
        CLA     OCTN
        SUB     JUNK
        STD     WTB1
        TSX     B1WT,4   REPLACES B1 WRITE
        AXT     0,2
        TRA     ZERO-1
LL2     TXI     *+1,1,1
        STZ     JUNK
        SXD     JUNK,4
        CLA     OCTN
        SUB     JUNK
        STD     WTB1
        TSX     B1WT,4
        TRA     EXIT
LL1     RTDA    5
        TCOA    *
        TEFA    LL2+1
        TRA     *-3
EXIT    AXT     0,2
        TIX     *+1,1,1
NRCRD   SXD     ,1
        AXT     0,1
        AXT     0,4
        TCOB    *
        WEFB    1
        REWB    1
        TEFA    *+1
        TEFB    *+1
        TRCA    *+1
        TRCB    *+2
        TRA     10,4
        ZET     RCHKB
        TRA     AGAIN
        CALL    PRINT,OC13
        TRA     NRCRD+2
```

```
AGAIN  CLA    RCHKB
       SUB    OCT1
       STO    RCHKB
       BSFA   5
       BSFA   5
       RTDA   5
       TCOA   *
REPT   AXT    0,4
       STZ*   3,4
       STZ*   4,4
       TRA    BEG2
RDA5   IOST   TEMP,,0
WTB1   IOCD   ,,0
TEMP   BSS    26
IVAN   BSS    1
NFLAD  HTR
NFLBS  HTR
TMIN   HTR
TMAX   HTR
JUNK   HTR
FS     HTR
ZS     HTR
SF     HTR
NRCSP  HTR
RCHKB  HTR
H      HTR    TEN
       HTR    TEN,,100
       HTR    TEN,,200
       HTR    TEN,,300
       HTR    TEN,,400
       HTR    TEN,,500
       HTR    TEN,,600
       HTR    TEN,,700
       HTR    TEN,,800
       HTR    TEN,,900
TEN    HTR    ONE
       HTR    ONE,,10
       HTR    ONE,,20
       HTR    ONE,,30
       HTR    ONE,,40
       HTR    ONE,,50
       HTR    ONE,,60
       HTR    ONE,,70
       HTR    ONE,,80
       HTR    ONE,,90
ONE    HTR
       HTR    ,,1
       HTR    ,,2
       HTR    ,,3
       HTR    ,,4
       HTR    ,,5
       HTR    ,,6
       HTR    ,,7
       HTR    ,,8
       HTR    ,,9
TEN2   DEC    100.
TENM0  DEC    .1
TENM2  DEC    .01
EXP    OCT    221000000000
OCT1   OCT    000001000000
OCT2   OCT    000002000000
OCT3   OCT    000003000000
OCTN   OCT    022600000000
```

```
OCTM   OCT     000000022600
FMA    BCI     6,(31H 2ND RED. CK AS COND IGNORED.
       BCI     3,//1H 14A6/1H 13A6)
BIWT   TCOB    *
       AXT     5,2
       TRCB    *+1
       WTBB    1
       RCHB    WTB1
       TCOB    *
       TRCB    *+2
       TRA     1,4
       BSRB    1
       WTBB    1
       TCOB    *
       TRCB    *+2
       TIX     BIWT+3,2,1
       TSX     $(SPH),4
       NOP     FMT
       TSX     $(FIL),4
       RUNB    1
       HTR     AGAIN
FMT    BCI     7,(36H  TAPE B1 BAD. CHANGE AND PUSH START )
       END
```

E. TIC SUBROUTINE, Written by T. E. Springer

1. Description

a. Purpose: To replace a list of consecutive floating point times in storage obtained from the translated tape A5 with corrected times.

b. Calling Sequence: CALL TIC(T), where T is one less than the lowest of 9600 words of storage (times) stored backwards.

c. Length and Language: $604_8 = 388_{10}$; FAP language.

d. Details: The time code generator in the Packard-Bell translator updates time every 0.2 seconds. Sampling at 40 frames per second, the identical time will be written at the beginning of 8 consecutive multiplexer frames. When the analog tape is translated, any number of frame skips (IFT) from 1 to 256 may be made. Since only one time out of 8 consecutive translated times may be the correct time, the question arises which of the 8 times is the correct one and what corrections should be made to the other times. With the only requirement that the translator be synchronized at the beginning to a true .4 second time (it could be any true time), the TIC subroutine can look at a set of 9 consecutive times, can determine the IFT, the time between frames, and a correct time in the group. With that information, it can correct the incorrect times. Table I illustrates the operation of TIC.

Table I

	Before Calling TIC(T)	After Calling TIC(T)
Contents of Location T-9	39000.40	39000.40
T-8	39000.20	39000.35
T-7	39000.20	39000.30
T-6	39000.20	39000.25
T-5	39000.20	39000.2
T-4	39000.00	39000.15
T-3	39000.00	39000.10
T-2	39000.00	39000.05
T-1	39000.00	39000.0

There is no difference in the operation of the TIC subroutine for multiplexer frame rates of 40, 20 or 10 frames per second. For example, a multiplexer run at 10 frames per second with every frame translated (IFT = 1), looks just like a multiplexer run at 40 frames per second translated with an IFT of 4.

The operation of TIC can best be seen from empirical observation of a set of times, corrected and uncorrected. One looks at the pattern of first differences and observes at most two numbers, one of which can be designated as H for high, the other as L for low. If all differences are equal, one can consider the differences all high. When thought of in this manner, it can be seen that all IFT's can be reduced to 8 different patterns corresponding to the patterns for IFT's from 1 to 8. Any higher IFT's are reducible to the pattern of $IFT_{Mod\ 8}$. Thus the pattern of lows and highs of the first differences for an IFT of 19 looks just like the pattern for an IFT of $19_{Mod\ 8} = 3$.

Table II shows a typical set of numbers, corrected and uncorrected, and the first differences for the 8 classes IFT = 1 to 8. Note that after 8 first differences the pattern will repeat itself so that only 9 consecutive numbers at most are necessary to apply time correcting. An examination of Table II will reveal in all cases except for IFT = 3 or 5 that the good number corresponds to a high followed by a low. When all are high, IFT = 8, any number is a good one. When IFT = 3, the good number corresponds to a high preceded by a single low; when IFT = 5, the good number is a high followed by a low-high-low. The logic of these three cases is contained in TIC to determine the first good number in the list.

TIC is called by the CUST subroutine. It replaces a block of 9600 times in floating point with the correct times in floating point. The following is a discussion of the code in operation.

At first all times are converted to fixed point with the binary point between bits 20 and 21. This leaves 15 bits for the fraction or 4-1/2 decimal places. Since the times may be as high as 99999.8, the lower part of the fraction may not have been included in the original floating point number. Enough of the fraction exists, however, to place the fraction as 0, .2, .4, .6, or .8. Thus the fractional part of the uncorrected times is replaced with the correct binary configuration. This greatly reduces round-off errors in the following procedures.

Since the difference between any time and a time 8 times later is always the correct time difference for 8 time steps, the code begins to

TABLE II

IFT = 1		ΔT=.025	
3.175	3.0	0	L
3.200	3.2	.2	H
3.225	3.2	0	L
3.250	3.2	0	L
3.275	3.2	0	L
3.300	3.2	0	L
3.325	3.2	0	L
3.350	3.2	0	L
3.375	3.2	0	L
3.400	3.4	.2	H
3.425	3.4	0	L
3.450	3.4	0	L
3.475	3.4	0	L
3.500	3.4	0	L
3.525	3.4	0	L

IFT = 2		ΔT= .05	
3.15	3.0	0	L
3.20	3.2	.2	H
3.25	3.2	0	L
3.30	3.2	0	L
3.35	3.2	0	L
3.40	3.4	.2	H
3.45	3.4	0	L
3.50	3.4	0	L
3.55	3.4	0	L
3.60	3.6	.2	H
3.65	3.6	0	L
3.70	3.6	0	L
3.75	3.6	0	L
3.80	3.8	.2	H
3.85	3.8	0	L

IFT = 3		ΔT=.075	
3.4	3.4	.2	H
3.475	3.4	0	L
3.550	3.4	0	L
3.625	3.6	.2	H
3.700	3.6	0	L
3.775	3.6	0	L
3.850	3.8	.2	H
3.925	3.8	0	L
4.000	4.0	.2	H
4.075	4.0	0	L
4.150	4.0	0	L
4.225	4.2	.2	H
4.300	4.2	0	L
4.375	4.2	0	L
4.450	4.4	.2	H

IFT = 4		ΔT=.1	
3.4	3.4	.2	H
3.5	3.4	0	L
3.6	3.6	.2	H
3.7	3.6	0	L
3.8	3.8	.2	H
3.9	3.8	0	L
4.0	4.0	.2	H
4.1	4.0	0	L
4.2	4.2	.2	H
4.3	4.2	0	L
4.4	4.4	.2	H
4.5	4.4	0	L
4.6	4.6	.2	H
4.7	4.6	0	L
4.8	4.8	.2	H

IFT = 5		ΔT=.125	
3.400	3.4	.2	H
3.525	3.4	0	L
3.650	3.6	.2	H
3.775	3.6	0	L
3.900	3.8	.2	H
4.025	4.0	.2	H
4.150	4.0	0	L
4.275	4.2	.2	H
4.400	4.4	.2	H
4.525	4.4	0	L
4.650	4.6	.2	H
4.775	4.6	0	L
4.900	4.8	.2	H
5.025	5.0	.2	H
5.150	5.0	0	L

IFT = 6		ΔT=.15	
3.4	3.4	.2	H
3.55	3.4	0	L
3.70	3.6	.2	H
3.85	3.8	.2	H
4.00	4.0	.2	H
4.15	4.0	0	L
4.30	4.2	.2	H
4.45	4.4	.2	H
4.60	4.6	.2	H
4.75	4.6	0	L
4.90	4.8	.2	H
5.05	5.0	.2	H
5.20	5.2	.2	H
5.35	5.2	0	L
5.50	5.4	.2	H

IFT = 7		ΔT =.175	
3.400	3.4	.2	H
3.575	3.4	0	L
3.750	3.6	.2	H
3.925	3.8	.2	H
4.100	4.0	.2	H
4.275	4.2	.2	H
4.450	4.4	.2	H
4.625	4.6	.2	H
4.800	4.8	.2	H
4.975	4.8	0	L
5.150	5.0	.2	H
5.325	5.2	.2	H
5.500	5.4	.2	H
5.675	5.6	.2	H
5.850	5.8	.2	H

IFT = 8		ΔT =.2	
3.4	3.4	.2	H
3.6	3.6	.2	H
3.8	3.8	.2	H
4.0	4.0	.2	H
4.2	4.2	.2	H
4.4	4.4	.2	H
4.6	4.6	.2	H
4.8	4.8	.2	H
5.0	5.0	.2	H
5.2	5.2	.2	H
5.4	5.4	.2	H
5.6	5.6	.2	H
5.8	5.8	.2	H
6.0	6.0	.2	H
6.2	6.2	.2	H

search the list of times to see whether T(I + 8) - T(I) is equal to T(I + 9) - T(I + 1). If not, it sets T(I) equal to zero and goes on to the next time. If it does, it is tentatively assumed that no change in time step on IFT occurs between the 8 points. The code computes the first differences of the 8 times and the second differences, storing them in S(J) and in RR(J). It then computes DT, the time between frames, from DT = (T(I + 8) - T(I))/8 and IFT from IFT = DT/.025. It computes DTS8 = $IFT_{Mod\ 8}$ to determine the type of pattern to look for. Next, it subtracts DT from the list of first differences S(J), the sign being "+" for a high and "-" for a low. It counts the number of highs and checks to see if it equals $IFT_{Mod\ 8}$. See Table II to verify that this is correct.

If DTS8 is not 5 or 3, it finds the first negative second difference RR(J). It then knows that T(I + J + 1) is a correct time. If DTS8 = 5 or 3, it places the signs of R(J) into the last 8 positions of the accumulator and compares them with a list PAT5 or PAT3 containing the bit pattern 01011011 or 00100101 rotated the 8 possible ways. Upon finding the PATX(J) that matches, it knows that T(I + J) is a good number.

Having found a good time, the code stores the first differences in PD(K) rotated so that PD(1) is the number added to the good number to yield the next uncorrected time. The code now proceeds through the list of times, 8 at a time, comparing the expected uncorrected time with the time in storage. If any times do not agree with the predicted time within 1 millisecond, that time is set negative. The code then goes through the list of 8 again, adding the correct time increment to the preceding corrected time and storing if the uncorrected time was positive. On reaching the next good time, that time is checked for correctness and is left alone if correct. This prevents the accumulation of round-off errors.

After correcting the good times, the code goes back through the list of 8 checking for negative times followed by good times. These presumably are spurious and are not the result of a change in IFT. Thus they are set equal to zero. The code then proceeds down the list of times in groups of 8, checking the predicted times and replacing with corrected times. If the backward search of the list of 8 reveals negative times not followed by good times, presumably there is a different IFT. They are reset positive and the index on the T's is set equal to the earliest time of this type. The code transfers back to the beginning to find a time such that $T(I + 8) - T(I) - T(I + 9) + T(I + 1)$ is approximately equal to zero and proceeds as previously described.

If at any time the list of 9600 times is exhausted, the list is reconverted to floating point and the code returns to the calling program. Figure 7 is a block diagram for the TIC subroutine.

e. Data Code Usage: TIC is called once per multiplexer by the CUST converting and unpacking subroutine. It replaces the uncorrected times in storage with a corrected list.

2. Source Deck (FAP) Listing

BLOCK DIAGRAM FOR TIC SUBROUTINE

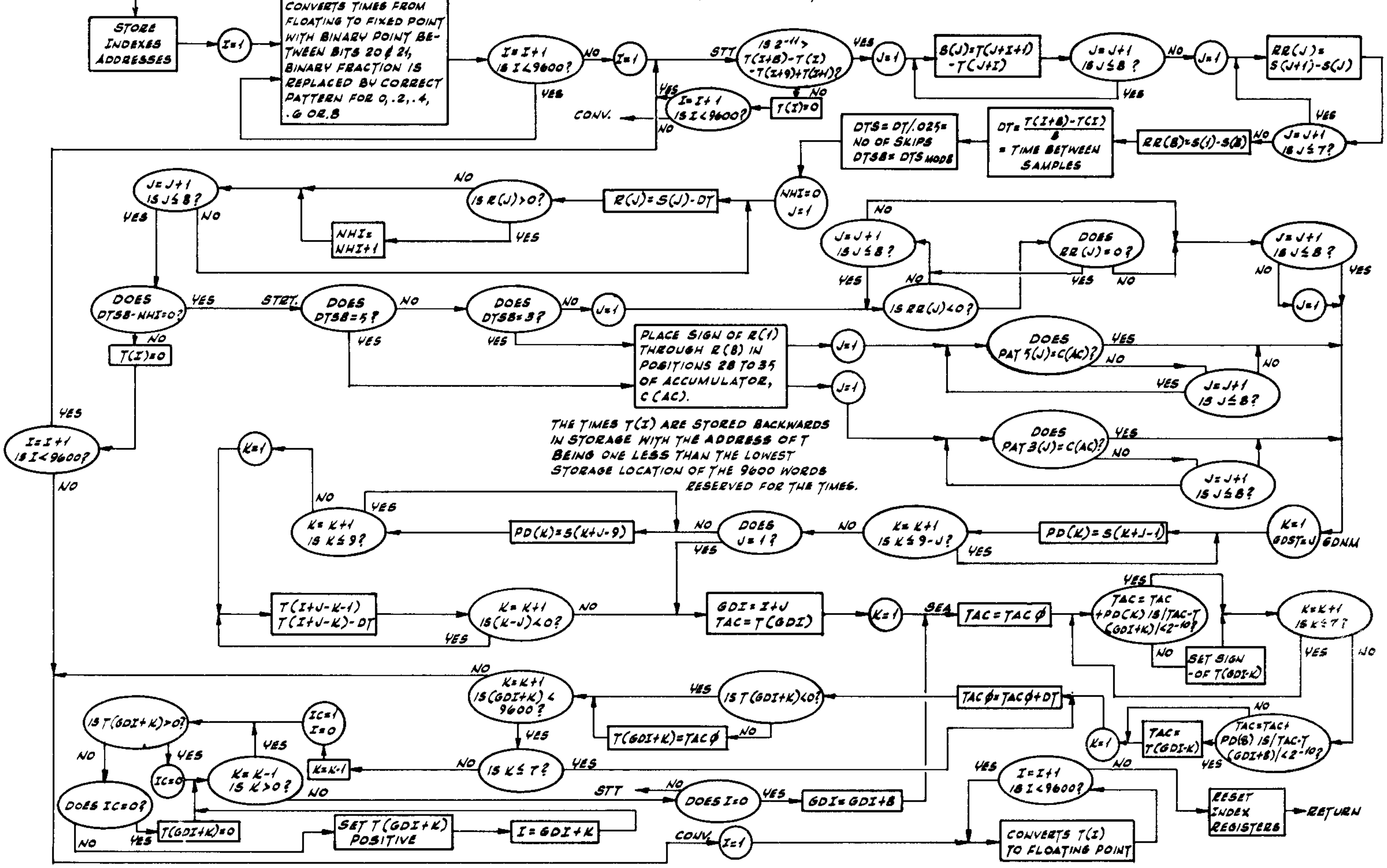


Figure 7

```
*       SPRINGER TOM   825260563TMRN0405F   4                             SPRINGER
*       LIST
*       LABEL
*       FAP
*TIC    SUBROUTINE TO CORRECT TIMES FOR RDH
        COUNT    425
        ENTRY    TIC
 TIC    SXD      XR1,1
        SXD      XR2,2
        SXD      XR4,4
        CLA      1,4     ADDRESSES ARE STORED.
        STA      BEG
        STA      T2
        STA      T3
        STA      T4
        STA      T5
        STA      T6
        STA      T8
        STA      T9
        STA      T11
        STA      T12
        STA      LP22
        STA      T13
        STA      BIG
        STA      T14
        STA      T15
        STA      T16
        STA      T17
        STA      T18
        STA      T19
        STA      T20
        STA      T21
        ADD      =1
        STA      TP1
        STA      TP2
        STA      TP3
        STA      TP4
        STA      TP5
        STA      TP6
        STA      TP7
        STA      TP8
        SUB      =2
        STA      TM1
        ADD      =9
        STA      STT
        STA      TP82
        ADD      =1
        STA      T9P
        LXD      X9600,2      CONVERTS FLOATING TO FIXED POINT
 BEG    CLA      ,2  T     WITH BINARY POINT BETWEEN BITS 20,21.
        TZE      EN
        CAS      X215
        TRA      RG
        TRA      RG
        UFA      X214
        ANA      =0777777777
        ADD      =16
 T2     STO      ,2  T
        TRA      SENT
 RG     SUB      X214
        ARS      27
        STA      *+3
```

```
T3     CLA    ,2  T
       ANA    =0000777777777
       ALS    0
       ADD    =16
T4     STO    ,2  T
SENT   ANA    =077400
       TZE    ZRO
       ALS    5
       PDX    ,1
       CLA    =0777777700000
T14    ANS    ,2  T
       CLA    FRA,1
T20    ORS    ,2  T
       TRA    EN
ZRO    CLA    =0777777700000
T21    ANS    ,2  T
EN     TIX    BEG,2,1
       LXD    X9600,2      SEARCHES FOR TWO ADJACENT TIMES HAVING THE SAME
STT    CLA    ,2  T+8      DIFFERENCES BETWEEN TWO TIMES 8 TIMES LATER.
       TZE    EN2
T5     SUB    ,2  T
TP1    ADD    ,2  T+1
T9P    SUB    ,2  T+9
       ARS    4
       TZE    ADV
T6     STZ    ,2  T
EN2    TIX    STT,2,1
       TRA    CONV
ADV    CLA    T6     COMPUTES FIRST DIFFERENCES OF FIRST GOOD BLOCK OF 8 TIMES.
       STZ    TEMP
       SXA    TEMP,2
       SUB    TEMP
       ADD    =8
       STA    TL1
       ADD    =1
       STA    TL2
       LXD    X8,4
TL2    CLA    ,4
TL1    SUB    ,4
       STO    S,4
       TIX    TL2,4,1
       LXD    X7,4      COMPUTES SECOND DIFFERENCES.
       CLA    S,4
       SUB    S-1,4
       ARS    4
       STO    RR-1,4
       TIX    *-4,4,1
       CLA    S-8
       SUB    S-1
       ARS    4
       STO    RR-1
TP82   CLA    ,2  T+8      COMPUTES DT, THE TIME BETWEEN FRAME SAMPLES,
T8     SUB    ,2  T       AND DTS, THE NUMBER OF SKIPS.  DETERMINES DTS MOD 8
       STO    DT
       XCA
       MPY    =5
       STQ    DTS
       XCA
       ADD    =16
       ANA    =0700000
       STO    DTS8
       CLA    DT
       ARS    3            WHICH  EQUALS DTS8.
```

```
       STO     DT
       STZ     NHI      COMPUTES HIGH AND LOW DIFFERENCES AND ADDS THEM.
       STZ     NLO
       LXD     X8,4
LP1    CLA     S,4
       SUB     DT
       STO     R,4
       ARS     4
       TMI     LP2
       TZE     LP2
HI     CLA     NHI
       ADD     ONEP
       STO     NHI
LP2    TIX     LP1,4,1
       CLA     DTS8     COMPARES FOR NO DROPOUTS.
       SUB     NHI
       TZE     STRT
T9     STZ     ,2  T
       TIX     STT,2,1
       TRA     CONV
STRT   CLA     DTS8
       SUB     =0500000
       TZE     SPC5
       CLA     DTS8
       SUB     =0300000
       TZE     SPC3
       LXD     X8,4
       CLA     RR,4
       TMI     CHECK
       TIX     *-2,4,1
       TRA     *+2
CHECK  TZE     *-2
       PXD     ,4
       SUB     ONE
       TNZ     *+2
       ADD     X8
       PDX     ,4
       TRA     GDNM
SPC5   CLM
       LXD     X8,4       FINDS GOOD TIMES FROM LLHLLHLH ETC. FOR 5 AND 3 CASES.
       LDQ     R,4
       LGL     1
       TIX     *-2,4,1
       LXD     X8,4
       LAS     PAT5,4
       TIX     *-1,4,1
       TRA     GDNM
       TIX     *-3,4,1
       TRA     GDNM
 SPC3  CLM
       LXD     X8,4
       LDQ     R,4
       LGL     1
       TIX     *-2,4,1
       LXD     X8,4
       LAS     PAT3,4
       TIX     *-1,4,1
       TRA     GDNM
       TIX     *-3,4,1
GDNM   STZ     GDST       SETS UP PATTERN IN PD WITH GOOD TIME FIRST.
       SXA     GDST,4       GDST CONTAINS INDEX OF FIRST GOOD POINT WRT XR2 AS 8.
       CLA     APD
       ADD     GDST
```

```
        SUB     =8
        STA     *+2
SAD     CLA     S,4
        STO     0,4
        TIX     *-2,4,1
        CLA     =8
        SUB     GDST
        TZE     NEX
        PAX     ,4
        ALS     18
        STO     SET
        CLA     SAD
        SUB     GDST
        STA     *+1
        CLA     0,4
APD     STO     PD,4
        TIX     *-2,4,1
        TRA     NEX1
NEX     SXD     GDI,2
        TRA     T12-1
NEX1    PXD     ,2
        SUB     ONE
        STD     EN5
        ADD     ONE
        SUB     SET
        PDX     ,4
        STD     GDI
T11     CLA     ,4  T     REPLACES TIMES FROM FIRST GOOD TIME BACKWARDS.
        SUB     DT
TM1     STO     ,4  T-1
        TXI     *+1,4,1
EN5     TXL     *-4,4,0
        LXD     GDI,2       SEARCHES GROUP OF 8 AND REPLACES BAD TIMES WITH 0.
T12     CLA     ,2  T
        STO     TAC
 SEA    LXD     X7,4
        LXD     GDI,2
        CLA     TAC
        STO     TACO
LP5     CLA     TAC
        ADD     PD-1,4
        STO     TAC
TP2     SUB     ,2  T+1
        ARS     5
        TZE     *+4
TP3     CLA     ,2  T+1
        SSM
TP8     STO     ,2  T+1
        TIX     *+1,2,1
        TIX     LP5,4,1
        CLA     TAC
        ADD     PD,4
        STO     TAC
TP6     SUB     ,2  T+1
        ARS     5
        TNZ     *+3
TP7     CLA     ,2  T+1
        STO     TAC
        LXD     GDI,2       REPLACES NEGATIVE TIMES WITH CORRECT TIME IN GROUPS
        LXD     X7,4                  OF EIGHT.
LP6     CLA     TACO
        ADD     DT
        STO     TACO
```

```
TP4    CLA    2  T+1
       TMI    *+3
       CLA    TACO
TP5    STO    ,2  T+1
       TIX    *+2,2,1
       TRA    CONV
       TIX    LP6,4,1
       STZ    TEMP
       CLA    ONE
       STO    IC
       CLA    GDI
       STD    EN8
T16    CLA    ,2   T REPLACES -FOLLOWED BY + TIMES WITH ZERO  OTHER MINUS
*                               TIMES ARE SET PLUS
       TPL    PLUG
       CLA    IC
       TNZ    *+3
T17    STZ    ,2  T
       TRA    EN8-1
T18    CLA    ,2  T
       SSP
T19    STO    ,2  T
       SXD    TEMP,2
       TRA    *+2
PLUG   STZ    IC
       TXI    *+1,2,1
EN8    TXL    T16,2,
       CLA    TEMP   IF TEMP=0 PROCEED. IF NOT SET XR2 AT FIRST GROUP
*            OF MINUS TIMES AND RETURN TO STT. (MINUS TIMES ARE NOW PLUS)
       TNZ    WAS
       CLA    GDI
       SUB    X8
       STO    GDI
       TRA    SEA
WAS    LXD    TEMP,2
       TRA    STT
CONV   LXD    X9600,2      SAME DIFFERENCE.
LP22   CLA    ,2  T      CONVERTS TIMES TO FLOATING POINT.
       CAS    =01000000000
       TRA    BIG
       TRA    BIG
       ORA    X214
       FAD    =0
T13    STO    ,2  T
       TRA    LP
BIG    CLA    ,2  T
       ARS    5
       ORA    =0221000000000
       FAD    =0
T15    STO    ,2  T
LP     TIX    LP22,2,1
OUT    LXD    XR1,1
       LXD    XR2,2
       LXD    XR4,4
       TRA    2,4
X9600  OCT    022600000000        DEFINITIONS
X215   OCT    215000000000
X214   OCT    214000000000
       OCT    63146
       OCT    46315
       OCT    31463
FRA    OCT    14632
XR1    BSS    1
```

```
XR2     BSS     1
XR4     BSS     1
S       BES     8
R       BES     8
DT      BSS     1
DTS     BSS     1
NC      BSS     1
NHI     BSS     1
NLO     BSS     1
        OCT     332
        OCT     155
        OCT     266
        OCT     133
        OCT     255
        OCT     326
        OCT     153
        OCT     265
PAT3    OCT     244
        OCT     122
        OCT     051
        OCT     224
        OCT     112
        OCT     045
        OCT     222
        OCT     111
PAT5    HTR     0
TEMP    BSS     1
RR      BES     8
PD      BES     8
X8      OCT     10000000
DTS8    BSS     1
IC      HTR     ,
ONEP    OCT     100000
ONE     OCT     1000000
X7      OCT     7000000
GDST    BSS     1
SET     BSS     1
GDI     BSS     1
TAC     BSS     1
TACO    BSS     1
TWO     OCT     2000000
        END
```

F. RDB1 SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To bring into core storage raw data and times stored on tape B1 by the CUS subroutine.

b. Calling Sequence: CALL RDB1 (NCH, NRC, NFSCH, NLSCH, STG)

NCH = Number of channels on tape B1

NRC = Number of records from tape A5 converted, unpacked and normalized by CUS and written on tape B1.

NFSCH = Number of the first channel on B1 to be brought into core storage.

NLSCH = Number of the last channel on B1 to be brought into core storage.

STG = Core storage location into which the information on B1 is to be loaded.

c. Length and Language: $213_8 = 139_{10}$; FAP language

d. Details: Assume that the following numbers are on tape B1 in high density, binary form:

39680.2	900.0	800.0	700.0	600.0	500.0
39680.4	901.0	801.0	701.0	601.0	501.0
39680.6	902.0	802.0	702.0	602.0	502.0
39680.8	903.0	803.0	703.0	603.0	503.0

Note that NCH = 5 and NRC = 4. If NFSCH = 2 and NLSCH = 4, then the following will be in core storage after RDB1 has been called:

STG(1) = 39680.2	STG(5) = 39680.4
STG(2) = 800.0	STG(6) = 801.0
STG(3) = 700.0	STG(7) = 701.0
STG(4) = 600.0	STG(8) = 601.0

STG(9) = 39680.6	STG(13) = 39680.8
STG(10) = 802.0	STG(14) = 803.0
STG(11) = 702.0	STG(15) = 703.0
STG(12) = 602.0	STG(16) = 603.0

For the normal case, tape B1 as written by CUS has one end-of-record gap per 200 original records on tape A5. RDB1 ignores end-of-record gaps. Note that it is required that

$1 \leqq \text{NCH} \leqq 47$

$1 \leqq \text{NFSCH} \leqq \text{NLSCH}$

$\text{NFSCH} \leqq \text{NLSCH} \leqq \text{NCH}$

Figures 8 and 9 are block diagrams of RDB1.

e. Data Code Usage: RDB1 is a companion subroutine to CUS and is used to get information from the intermediate tape B1, written by CUS, into the core for further processing.

2. Source Deck (FAP) Listing

BLOCK DIAGRAM OF RDB1 SUBROUTINE, ENTRY AND EXIT LOGIC

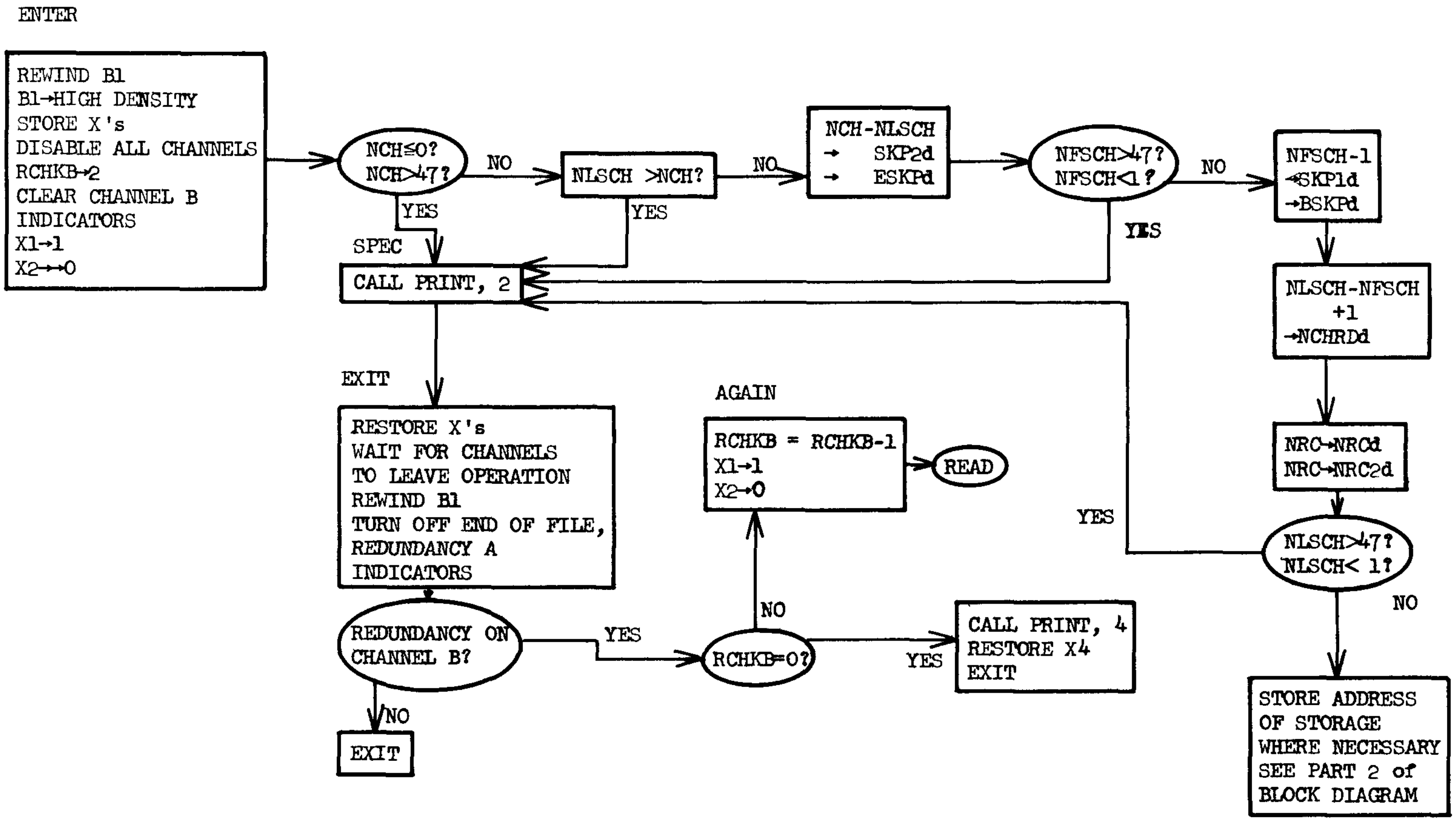


Figure 8

BLOCK DIAGRAM OF RDB1 SUBROUTINE
TAPE READING LOGIC

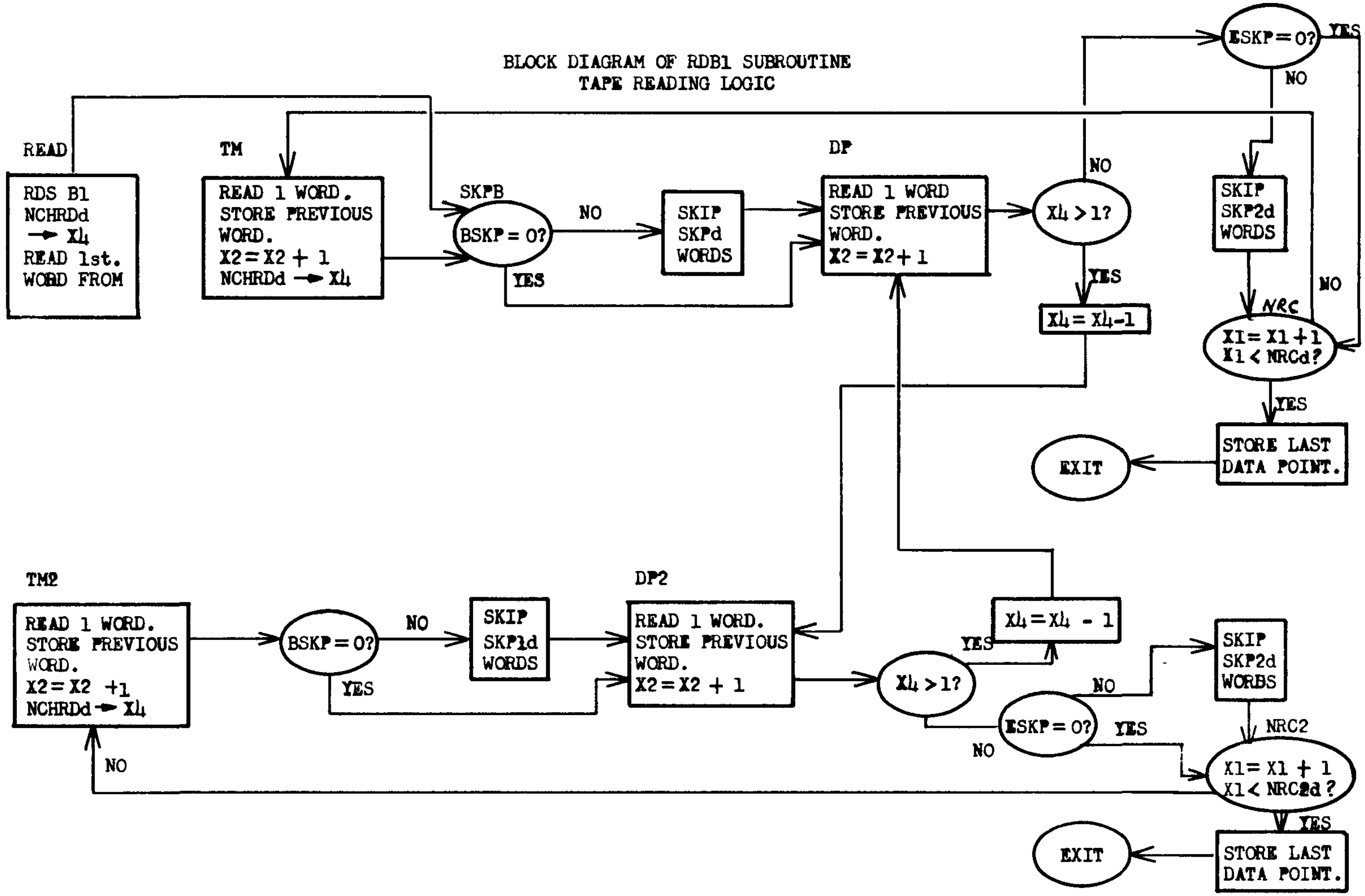


Figure 9

```
*RDB1
       COUNT   100
*CALLING SEQUENCE CALL RDB1(NCH,NRC,NFSCH,NLSCH,STG)
*XA=PSEUDO RECORD COUNT
*XB=STORAGE WORD COUNT
*XC=DATA WORD COUNT
       ENTRY   RDB1
RDB1   REWB    1
       SDHB    1
       SXA     EXIT,1
       SXA     EXIT+1,2
       SXA     EXIT+2,4
       SXA     SX4,4
       STZ     TEMP
       ENB     TEMP
       CLA     OCT2
       STO     RCHKB
       TRCB    *+1
       TEFB    *+1
       AXT     1,1
       AXT     0,2
       CLA*    1,4
       TZE     SPEC
       TMI     SPEC
       SUB     OCT47
       TPL     SPEC
       CLA*    1,4
       SUB*    4,4
       TMI     SPEC
       STD     SKP2
       STD     ESKP
       CLA*    3,4
       SUB     OCT47
       TPL     SPEC
       CLA*    3,4
       SUB     OCT1
       TMI     SPEC
       STD     SKP1
       STD     BSKP
       CLA*    4,4
       SUB*    3,4
       ADD     OCT1
       STD     NCHRD
       CLA*    2,4
       STD     NRC
       STD     NRC2
       CLA*    4,4
       TMI     SPEC
       TZE     SPEC
       SUB     OCT47
       TPL     SPEC
       CLA     5,4
       STA     ST01
       STA     ST02
       STA     ST03
       STA     ST04
       STA     ST05
       STA     ST06
       TCOB    *
READ   RTBB    1
       LXD     NCHRD,4
       RCHB    RD
```

```
         TRA      SKPB
TM       LCHB     RD
         CLA      TEMP2
ST01     STO      ,2
         TXI      *+1,2,1
         LXD      NCHRD,4
SKPB     NZT      BSKP
         TRA      *+2
         LCHB     SKP1
DP       LCHB     RD2
         CLA      TEMP
ST02     STO      ,2
         TXI      *+1,2,1
         TIX      DP2,4,1
         NZT      ESKP
         TRA      *+2
         LCHB     SKP2
         TXI      *+1,1,1
NRC      TXL      TM,1,
         TCOB     *
         CLA      TEMP2
ST03     STO      ,2
         TRA      EXIT
TM2      LCHB     RD2
         CLA      TEMP
ST04     STO      ,2
         TXI      *+1,2,1
         LXD      NCHRD,4
         NZT      BSKP
         TRA      *+2
         LCHB     SKP1
DP2      LCHB     RD
         CLA      TEMP2
ST05     STO      ,2
         TXI      *+1,2,1
         TIX      DP,4,1
         NZT      ESKP
         TRA      *+2
         LCHB     SKP2
         TXI      *+1,1,1
NRC2     TXL      TM2,1,
         TCOB     *
         CLA      TEMP
ST06     STO      ,2
         TRA      EXIT
SPLC     CALL     PRINT,OCT2
EXIT     AXT      0,1
         AXT      0,2
         AXT      0,4
         TCOA     *
         TCOB     *
         REWB     1
         TEFA     *+1
         TEFB     *+1
         TRCA     *+1
         TRCB     *+2
         TRA      6,4
         ZET      RCHKB
         TRA      AGAIN
         CALL     PRINT,OCT4
SX4      AXT      ,4
         TRA      6,4
AGAIN    CLA      RCHKB
```

```
         SUB     ØCT1
         STO     RCHKB
         AXT     1,1
         AXT     0,2
         TRA     READ
RD       IØCT    TEMP,0,1
RD2      IØCT    TEMP2,0,1
SKP1     IØCTN   ,,
SKP2     IØCTN   ,,
TEMP     HTR     ,
TEMP2    HTR     ,
NCHRD    HTR     ,
BSKP     HTR     ,
ESKP     HTR     ,
RCHKB    HTR     ,
ØCT1     ØCT     000001000000
ØCT2     ØCT     000002000000
ØCT4     ØCT     000004000000
ØCT47    ØCT     000057000001
         END
```

G. LSF SUBROUTINE, Written by M. Blaylock, EGG

1. Description

a. Purpose: To least-square-fit a set of up to 50 data points to a polynomial of up to the fifth power.

b. Calling Sequence: CALL LSF(N, M, X, Y, B, SIGMA)

N = Number of points to be fitted, ≤ 50

M = Power of fit, ≤ 5

X = Location of first x value

Y = Location of first y value

B = Location of the coefficients B_i in the equation

$$Y = \sum_{i=0}^{M} B_i X^i$$

SIGMA = Standard Deviation of the fit

c. Length and Language: $1013_8 = 523_{10}$; FORTRAN language.

d. Details: LSF is a short FORTRAN subroutine for doing a standard least-square-fit of a set of data points to a set of polynomial coefficients.

2. Source Deck (FORTRAN) Listing

```
CLSF7
      SUBROUTINE LSF (N,M,X,Y,B,SIGMA)
      DIMENSION X(50),Y(50),V(6),B(6),A(6,7),SUM(11)
      LS = 2*M + 1
      LB = M + 2
      LV = M + 1
      DO 500 J = 2,LS
  500 SUM(J) = 0.0
      SUM(1) = N
      DO 501 J = 1,LV
  501 V(J) = 0.0
      DO 510 I = 1,N
      P = 1.0
      V(1) = V(1) + Y(I)
      DO 505 J = 2,LV
      P = X(I)*P
      SUM(J) = SUM(J) + P
      V(J) = V(J) + Y(I) * P
  505 CONTINUE
      DO 510 J = LB,LS
      P = X(I) * P
      SUM(J) = SUM(J) + P
  510 CONTINUE
      DO 513 I = 1,LV
      DO 513 K = 1,LV
      J = K + I - 1
      A(K,I) = SUM(J)
  513 CONTINUE
      DO 515 K = 1,LV
      A(K,LB) = V(K)
  515 CONTINUE
      DO 530 L = 1,LV
      DIVA = A(L,L)
      DO 517 J = L,LB
  517 A(L,J) = A(L,J)/DIVA
      IA = L + 1
      IF (IA - LB) 520,535,535
  520 DO 530 I = IA,LV
      FMULTA = A(I,L)
      DO 530 J = L,LB
      A(I,J) = A(I,J) - A(L,J) * FMULTA
  530 CONTINUE
  535 B(LV) = A(LV,LB)
      I = LV
  536 DIFB = 0.0
      DO 540 J = I,LV
      IB = I - 1
      DIFB = DIFB + A(IB,J) * B(J)
  540 CONTINUE
      I = I - 1
      B(I) = A(I,LB) - DIFB
      IF (I - 1) 550,550,536
  550 DEVS = 0.0
      SY = 0.0
      DO 552 I = 1,N
      SY = SY + Y(I) **2.0
  552 CONTINUE
      DO 554 I = 1,LV
      DEVS = DEVS + B(I) * V(I)
  554 CONTINUE
      DEVS = SY - DEVS
      CN = N
```

```
      SIGMA = SQRTF(DEVS / CN)
      RETURN
      END
```

H. WRTB6 SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To write all processed data points and times on tape B6 for permanent storage. The contents of tape B6 may be written on tape A3 for BCD listing with the B6TØA3 code, or used as the input to various subcodes.

b. Calling Sequence: CALL WRTB6 (NCHB6, NPTS, TEMP, TITLE, STG, J, NWDRC)

NCHB6 = An identification number which is the first word in the channel record on B6; normally this is a serial number for the position of the record (1 channel = 1 record) on the tape.

NPTS = Number of data points in the tape B6 record. The number of words in the record is equal to 2 times NPTS plus 20, since there are two words (time word and data word) for each data point, plus twenty words of identification. NPTS is the second word in the channel record on B6.

TEMP = Twelve words (72 characters) of BCD information occupying words 3 to 14 on B6. This is normally the REC card information from the RDH-D code.

TITLE = Six words (36 characters) of BCD information occupying words 15 to 20 on B6. This is normally the TITLE card information from the RDH-D code.

STG = Location in the main program for a block of storage in which the time and data points are stored. For example, the following might be stored in the main program:

STG(1) = 39461.2 (First time)
STG(2) = 860.0 (First data point, first channel)
STG(3) = 951.0 (First data point, second channel)
STG(4) = 600.0 (First data point, third channel)
STG(5) = 39461.4 (Second time)
STG(6) = 861.0 (Second data point, first channel)
STG(7) = 952.0 (Second data point, second channel)
STG(8) = 601.0 (Second data point, third channel)

J = Serial number of the data channel in the storage block. For example, if J = 2 with the above STG block, then B6 will have the number 39461.2, 951.0, 39461.4, 952.0 for its twenty-first through twenty-fourth words.

NWDRC = Number of words in a storage record, that is the number of channels in the storage block plus 1. For the above example, NWDRC = 4.

c. Length and Language: $115_8 = 77_{10}$; FAP language.

d. Details: WRTB6 writes one high density binary record on tape B6 for each channel processed by the RDH-D code, and is called as a subroutine by that code. In "semi-FORTRAN" notation tape B6 is in the following order: NCHB6, NPTS, TEMP(12) to TEMP(1), TITLE(6) to TITLE(1),(TIME(I), DP(I), I = 1, NPTS). Figure 10 is a block diagram of the WRTB6 code.

e. Data Code Usage: WRTB6 is called once for each channel processed by RDH-D.

2. Source Deck (FAP) Listing

BLOCK DIAGRAM OF WRTB6 SUBROUTINE

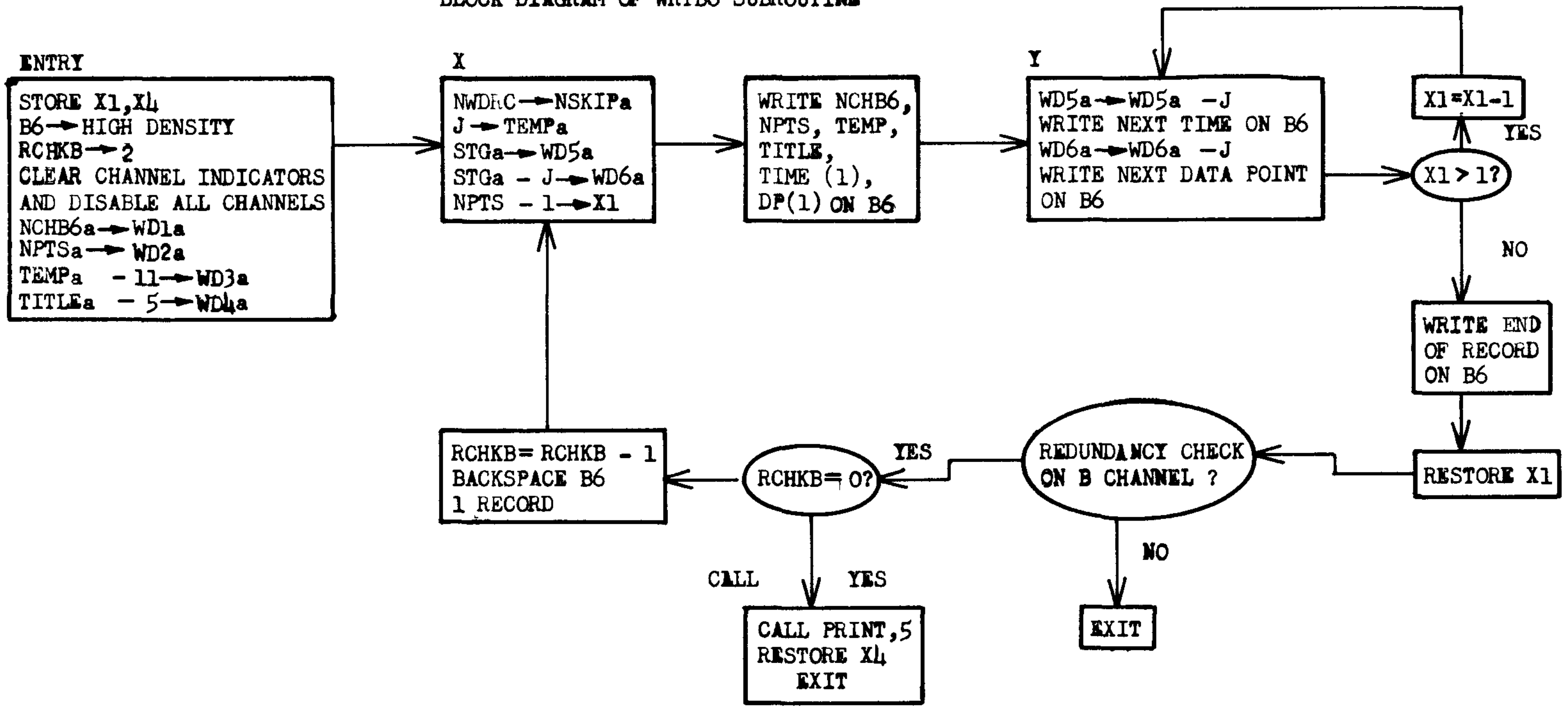


Figure 10

```
*WRTB6
         COUNT    80
*B6 IS IN THE FOLLOWING ORDER.NCHB3,NPTS,TEMP(12)TO TEMP(1),TITLE(6) TO
*TITLE(1),(TIME(I),DP(I),I=1,NPTS)
*CALLING SEQUENCE CALLWRTB6(NCHB6,NPTS,TEMP,TITLE,STG,J,NWDRC)
         ENTRY    WRTB6
  WRTB6  SXA      EXIT+3,1
         SXA      SX4,4
         SDHB     6
         CLA      OCT2
         STO      RCHKB
         STZ      TEMP
         ENB      TEMP
         TRCB     *+1
         TEFB     *+1
         CLA      1,4
         STA      WD1
         CLA      2,4
         STA      WD2
         CLA      3,4
         SUB      OCT13
         STA      WD3
         CLA      4,4
         SUB      OCT5A
         STA      WD4
  X      STZ      NSKIP
         CLA*     7,4
         ARS      18
         STA      NSKIP
         CLA*     6,4
         ARS      18
         STA      TEMP
         CLA      5,4
         STA      WD5
         SUB      TEMP
         STA      WD6
         CLA*     2,4
         SUB      OCT1
         PDX      ,1
         WTBB     6
         RCHB     WD1
         LCHB     WD5
         LCHB     WD6
  Y      CLA      WD5
         ADD      NSKIP
         STO      WD5
         LCHB     WD5
         CLA      WD6
         ADD      NSKIP
         STO      WD6
         LCHB     WD6
         TIX      Y,1,1
  EXIT   LCHB     ERG
         TCOB     *
         TEFB     *+1
         AXT      0,1
         TRCB     *+2
         TRA      8,4
         CLA      RCHKB
         TZE      CALL
         SUB      OCT1
         STO      RCHKB
```

```
          BSRB    6
          TRA     X
CALL      CALL    PRINT,OCT5
SX4       AXT     ,4
          TRA     8,4
ERG       IORTN   ,,0
WD1       IOCP    ,,1
WD2       IOCP    ,,1
WD3       IOCP    ,,12
WD4       IOCT    ,,6
WD5       IOCT    ,,1
WD6       IOCT    ,,1
OCT1      OCT     000001000000
OCT2      OCT     000002000000
OCT5      OCT     000005000000
OCT5A     OCT     000000000005
OCT13     OCT     000000000013
TEMP      HTR     ,
NSKIP     HTR     ,
RCHKB     HTR     ,
          END
```

I. RDB6 SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To bring into core storage information written on tape B6 by the WRTB6 subroutine.

b. Calling Sequence: CALL RDB6 (NCHB6, NPTS, TMIN, TMAX, TSTG, YSTG, TEMP, TITLE, IN)

NCHB6 = An identification number which is the first word in the channel record on B6. RDB6 will search for the channel with this number assuming that the channel numbering is serial, and increasing in increments of one; i.e., the first channel must be numbered 1 and the eighth channel 8, etc. Note that 1 channel of data = 1 record on tape B6.

NPTS = Number of data points in the tape B6 record. The number of words in the record is equal to 2 times NPTS plus 20, since there are two words (time word and data word) for each data point, plus twenty words of identification. NPTS is the second word in the channel record on B6. NPTS is an output number from RDB6.

TMIN = Minimum time for transmitted data points in the given tape B6 record. All time and data point pairs for which time is less than TMIN will not be brought into the core.

TMAX = Maximum time for transmitted data points in the given tape B6 record. All time and data point pairs for which time is greater than TMAX will not be brought into the core.

TSTG = Origin of FORTRAN dimensioned storage block into which the times from B6 are stored in the expected manner; i.e., the first time greater than or equal to TMIN goes into TSTG(1).

YSTG = Origin of data point storage block corresponding to TSTG.

TEMP = Twelve words (72 characters) of BCD information occupying words 3 to 14 on B6. This is normally the REC card information from the RDH-D code. RDB6 stores these words in the reverse of their normal FORTRAN dimensioned order, i.e., TEMP (1) from WRTB6 is stored in TEMP(12) by RDB6, etc.

TITLE = Six words (36 characters) of BCD information occupying words 15 to 20 on B6. This is normally the TITLE card information from the RDH-D code. The storage word order for TITLE is similar to that for TEMP; i.e., the reverse of the normal FORTRAN storage.

IN = Storage block increment. If IN = 1 the second time and data points are stored in TSTG(2), YSTG(2) respectively, the third in TSTG(3), YSTG(3), etc. If IN = 2 the second time and data points are stored in TSTG(3), YSTG(3), the third in TSTG(5), YSTG(5), etc.

Figure 11 is a block diagram of RDB6.

c. Length and Language: $206_8 = 134_{10}$; FAP language.

d. Data Code Usage: RDB6 is used for searching and retrieval of information by the B6TØA3 code, and by all other codes using tape B6 as a data source.

2. Source Deck (FAP) Listing

BLOCK DIAGRAM OF RDB6 SUBROUTINE

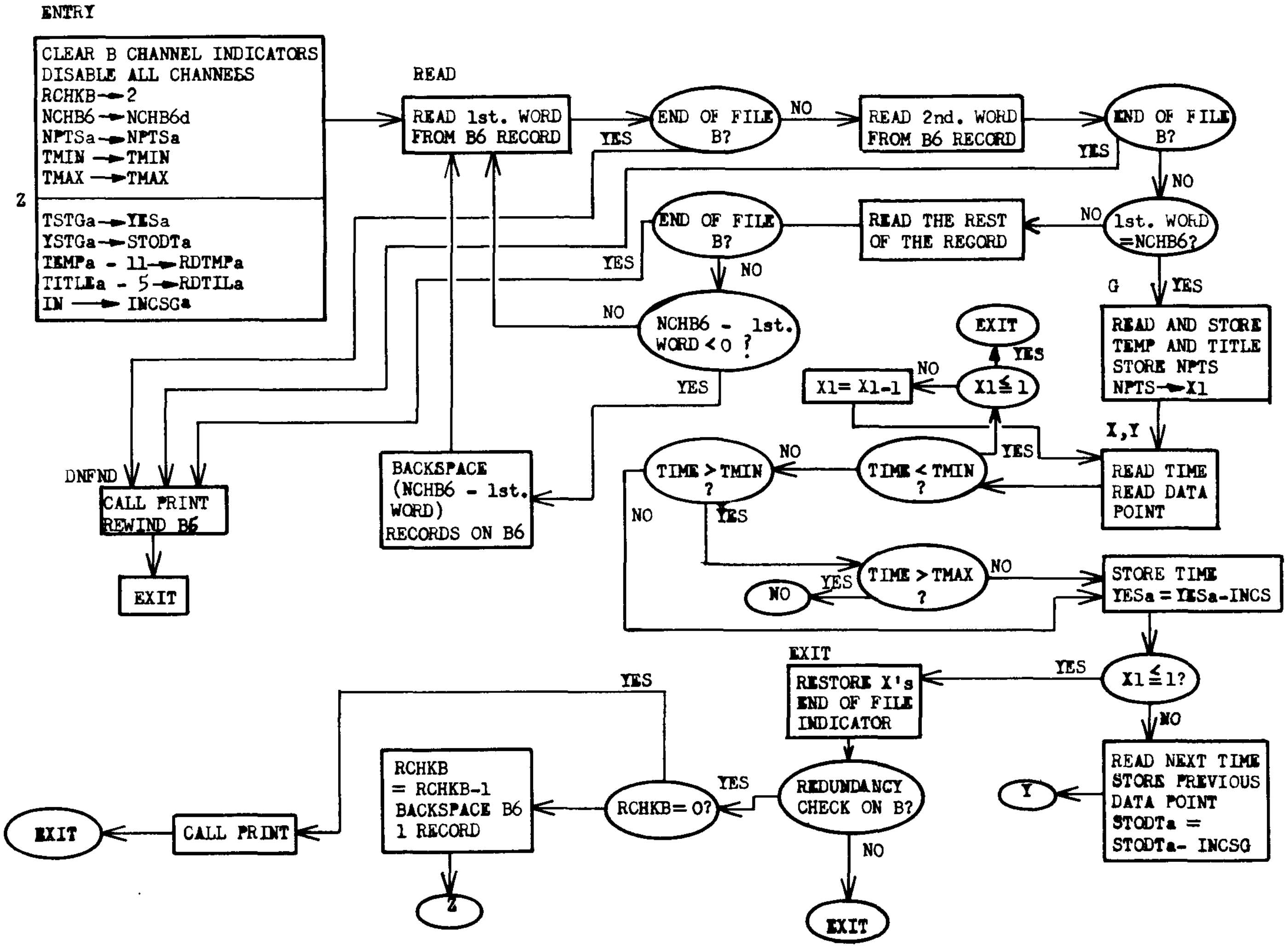


Figure 11

```
*RDB6
*CALLINGSEQUENCE CALL RDB6(NCHB6,NPTS,TMIN,TMAX,TSTG,YSTG,TEMP,TITLE,IN)
        COUNT   110
        ENTRY   RDB6
RDB6    TRCB    *+1
        TEFB    *+1
        STZ     TEMP
        ENB     TEMP
        STZ     LSTCH
        SDHB    6
        SXA     EXIT,1
        SXA     EXIT+1,4
        CLA     OCT2
        STO     RCHKB
        CLA*    1,4
        STD     NCHB6
        CLA     2,4
        STA     NPTS
        CLA*    3,4
        STO     TMIN
        CLA*    4,4
        STO     TMAX
Z       CLA     5,4
        STA     YES
        CLA     6,4
        STA     STODT
        CLA     7,4
        SUB     OCT13
        STA     RDTMP
        CLA     8,4
        SUB     OCT5
        STA     RDTIL
        CLA*    9,4
        ARS     18
        STA     INCSG
READ    RTDB    6
        RCHB    RD1
        TEFB    DNFND
        LCHB    RD2
        TEFB    DNFND
        CLA     TEMP
        SUB     NCHB6
        TZE     GO
        LCHB    RDN
        TCOB    *
        TEFB    DNFND
        TMI     READ
        ADD     OCT2
        PDX     ,1
        TNX     READ,1,1
        BSRB    6
        TCOB    *
        TRA     *-3
GO      LCHB    RDTMP
        CLA     TEMP+1
NPTS    STD     *
        PDX     ,1
X       LCHB    RD1
Y       LCHB    RD2
        CLA     TEMP
        CAS     TMIN
        TRA     MAYBE
```

```
          TRA     YES
          TRA     NO
 MAYBE    CAS     TMAX
          TRA     NO
          TRA     YES
 YES      STO     ,
          CLA     YES
          SUB     INCSG
          STO     YES
          TNX     LAST,1,1
          LCHB    RD1
          CLA     TEMP+1
 STODT    STO     ,
          CLA     STODT
          SUB     INCSG
          STO     STODT
          TRA     Y
 NO       TNX     EXIT,1,1
          TRA     X
 LAST     TCOB    *
          CLA     STODT
          STA     *+2
          CLA     TEMP+1
          STO     ,
 EXIT     AXT     ,1
          AXT     ,4
          TCOB    *
          TEFB    *+1
          TRCB    *+2
          TRA     10,4
          NZT     RCHKB
          TRA     OUT
          CLA     RCHKB
          SUB     OCT1
          BSRB    6
          TCOB    *
          TRA     Z
 OUT      CALL    PRINT,OCT10
          TRA     EXIT
 DNFND    IOT
          NOP
          NZT     LSTCH
          TRA     BACK
          REWB    6
          CALL    PRINT,OCT11
          TRA     EXIT
 BACK     CLA     OCT1
          STO     LSTCH
          BSFB    6
          BSRB    6
          TCOB    *
          TEFB    *+1
          TRCB    *+1
          TRA     READ
 RD1      IOCT    TEMP,,1
 RD2      IOCT    TEMP+1,,1
 RDN      IOCDN   TEMP,,1
 RDTMP    IOCP    ,,12
 RDTIL    IOCT    ,,6
 RDBCD    IOCT    ,,18
 INCSG    HTR     ,
 TMIN     HTR     ,
 TMAX     HTR     ,
```

```
NCHB6  HTR      ,
RCHKB  HTR      ,
LSTCH  HTR      ,
TEMP   BSS      2
OCT1   OCT      000001000000
OCT2   OCT      000002000000
OCT5   OCT      000000000005
OCT10  OCT      000010000000
OCT11  OCT      000011000000
OCT13  OCT      000000000013
       END
```

J. PØSB6 SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To position tape B6 for adding channels to a tape B6 with some channels already on it.

b. Calling Sequence: CALL PØSB6 (NCHB6)

c. Length and Language: $43_8 = 35_{10}$; FAP language

d. Details: PØSB6 advances tape B6 to the first end-of-file, then backspaces over the end-of-file mark and positions itself after the last record. The number of the last channel or tape B6 is put in NCHB6 in the main code.

e. Data Code Usage: PØSB6 is called at the beginning of an RDH-D computer run if the quantity IADVB6 (an input number) is not zero.

2. Source Deck (FAP) Listing

```
*POSB6
          COUNT   30
*CALLINGSEQUENCE CALLPOSB6(NCHB6)
          ENTRY   POSB6
 POSB6    TRCB    *+1
          SXA     SX4,4
          TEFB    *+1
          STZ     TEMP
          ENB     TEMP
          CLA     OCT2
          STO     RCHKB
 AGAIN    REWB    6
          SDHB    6
          RTBB    6
          TCOB    *
          TEFB    *+2
          TRA     *-3
          BSFB    6
          BSRB    6
          TCOB    *
          RTBB    6
          RCHB    WD1
          TCOB    *
          TEFB    *+1
          TRCB    *+4
          CLA     TEMP
          STO*    1,4
          TRA     2,4
          ZET     RCHKB
          TRA     AGAIN
          CALL    PRINT,OCT6
 SX4      AXT     ,4
          TRA     2,4
 WD1      IOCD    TEMP,,1
 RCHKB    HTR
 TEMP     HTR
 OCT2     OCT     000002000000
 OCT6     OCT     000006000000
          END
*         LIST
```

K. LSTB6 SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To write a list on tape A3 of the channel identification information for all of the channels on tape B6.

b. Calling Sequence: CALL LSTB6

c. Length and Language: $74_8 = 60_{10}$; FAP language

d. Details: LSTB6 with its subroutine LIST simply writes on tape A3 the first 20 words from each word on tape B6. Figure 12 is a block diagram of the LSTB6 code.

e. Data Code Usage: At the end of any computer run, LSTB6 is called automatically by RDH-D.

2. Source Deck (FAP) Listing

BLOCK DIAGRAM OF LSTB6 SUBROUTINE

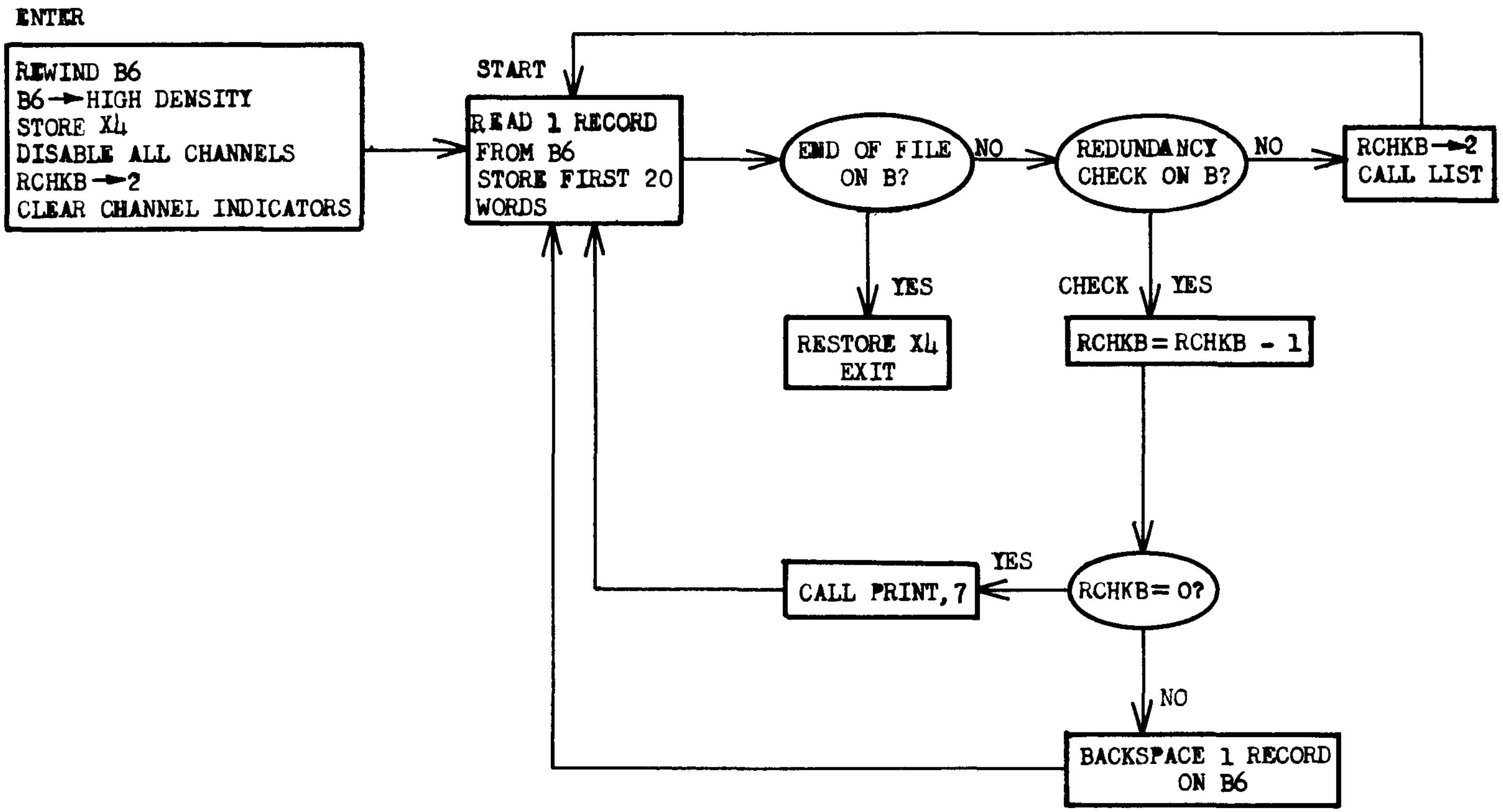


Figure 12

```
*       LABEL
*       FAP
*LSTB6
         COUNT   30
         ENTRY   LSTB6
 LSTB6   REWB    6
         SDHB    6
         SXA     EXIT+1,4
         STZ     RCHKB
         ENB     RCHKB
         CLA     OCT2
         STO     RCHKB
         TRCB    *+1
         TEFB    *+1
 START   RTBB    6
         RCHB    RD
         TCOB    *
         TEFB    EXIT
         TRCB    CHECK
         CLA     OCT2
         STO     RCHKB
         CALL    LIST,TEMP,TEMP+1,TEMP+13,TEMP+19
         TRA     START
 CHECK   CLA     RCHKB
         SUB     OCT1
         TZE     *+4
         BSRB    6
         STO     RCHKB
         TRA     START
         CALL    PRINT,OCT7
         TRA     START
 EXIT    TRCB    *+1
         AXT     0,4
         TRA     1,4
 RD      IOCD    TEMP,,20
 TEMP    BSS     20
 RCHKB   HTR     ,
 OCT2    OCT     000002000000
 OCT1    OCT     000001000000
 OCT7    OCT     000007000000
         END
```

L. LIST SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To write on tape A3 the identifying information from tape B6.

b. Calling Sequence: CALL LIST (NCHB6, NPTS, TEMP, TITLE)

NCHB6 = An identifying number which is the first word in the channel record on B6.

NPTS = Number of data points on the B6 record. This is the second word on the channel record on B6.

TEMP = Twelve words (72 characters) of BCD information occupying words 3 to 14 on B6. This is normally the REC card information from the RDH-D code.

TITLE = Six words (36 characters) of BCD information occupying words 15 to 20 on B6. This is normally the TITLE card information from the RDH-D code.

c. Length and Language: $62_8 = 50_{10}$, FORTRAN language

d. Details: This subroutine is called by the LSTB6 subroutine, and merely formats the information found by LSTB6.

2. Source Deck (FORTRAN) Listing

```
CLIST
      SUBROUTINELIST(NCHB6,NPTS,TEMP,TITLE)
      DIMENSIONTEMP(12),TITLE(6)
    1 FORMAT(2I5,9A6,9A6)
      WRITEOUTPUTTAPE9,1,NCHB6,NPTS,TEMP,TITLE
      RETURN
      END
```

M. PRINT SUBROUTINE, Written by F. Narin

1. Description

a. Purpose: To print and write on tape A3 information that redundancy checks or other difficulties were incurred by various subroutines.

b. Calling Sequence: CALL PRINT(N)

N = A number generated by the subcode telling PRINT which statement to print and write on tape A3.

c. Length and Language: $373_8 = 251_{10}$; FORTRAN language.

d. Details: When called, PRINT prints and writes on tape A3 the appropriate information.

e. Data Code Usage: PRINT may be called by the subroutines CUS, LSTB6, PØSB6, RDB1, RDB6 or WRTB6.

2. Source Deck (FORTRAN) Listing

```
CPRINT
      SUBROUTINEPRINT(N)
    1 FORMAT(55H0INCORRECT CALL. SEQ. INFO. IN CALL CUS (DELETED)      )
    2 FORMAT(55H0 CALL. SEQUENCE INFO. INCORRECT. CALLRDB1 DELETED     )
    3 FORMAT(55H0REDUNDENCY CHECK 3 TIMES IN CUS. CALL CUS DELETED     )
    4 FORMAT(55H0REDUNDENCY CHECK 3 TIMES IN READING B1.CALLRDB1DELETED)
    5 FORMAT(55H0REDUND. CHECK 3 TIMES IN WRIT.B6. CALLWRTB6 DELETED   )
    6 FORMAT(55H0REDUND.CHECK 3 TIMES IN ADVANCING B6.CALLPOSB6 DELETED)
   65 FORMAT(55H0 B6 HAS BEEN REWOUND AND UNLOADED. PUT ON NEW B6      )
    7 FORMAT(55H0REDUN. CK. 3 TIMES IN LSTB6. 1 CHANNEL TITLE OMITTED  )
    8 FORMAT(55H0REDUND. CHECK 3 TIMES IN READING B6.CALL RDB6 DELETED )
    9 FORMAT(55H0RDB6 AT END OF FILE IN SEARCHING. CALL RDB6 DELETED   )
   50 FORMAT(1H0)
      GOTO(101,102,103,104,105,106,107,108,109),N
  101 WRITEOUTPUTTAPE9,1
      PRINT1
      GOTO200
  102 WRITEOUTPUTTAPE9,2
      PRINT2
      GOTO200
  103 WRITEOUTPUTTAPE9,3
      PRINT3
      GOTO200
  104 WRITEOUTPUTTAPE9,4
       PRINT4
      GOTO200
  105 WRITEOUTPUTTAPE9,5
      PRINT5
      GOTO200
  106 WRITEOUTPUTTAPE9,6
      WRITEOUTPUTTAPE9,65
      PRINT6
      PRINT65
      GOTO200
  107 WRITEOUTPUTTAPE9,7
      PRINT7
      GOTO200
  108 WRITEOUTPUTTAPE9,8
      PRINT8
      GOTO200
  109 WRITEOUTPUTTAPE9,9
      PRINT9
  200 PRINT50
      PRINT50
      RETURN
      END
```

N. LBLFGD SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To label the grid drawn by the select grid subroutine SELGRD. For linear grids the labeling is fixed point, that is, all numbers are truncated to, and expressed as, integers.

b. Calling Sequence: CALL LBLFGD(I, J, XYBND)

I = Integer specifying option for labeling X boundary.

J = Integer specifying option for labeling Y boundary.

XYBND = Origin of array for which, in FORTRAN storage:

XYBND(1) = x-value at left boundary (X_{min})
XYBND(2) = x-value at right boundary(X_{max})
XYBND(3) = y-value at lower boundary(Y_{min})
XYBND(4) = y-value at upper boundary(Y_{max})

I or J may have the following values:

Linear Intervals	Cycle logarithmic intervals
1 = 1 interval	11 = 1 cycle
2 = 2	12 = 2
3 = 3	13 = 3
4 = 4	14 = 4
5 = 5	15 = 5
6 = 10	16 = 6
7 = 15	
8 = 20	
9 = 25	
10 = 50	

Note that for logarithmic labeling Y_{min} must be the exponent corresponding to Y_{min}. For example: if PTLGMX = 3.0 and NYLB = 12, then the ordinate of the plot will be labeled:

8
6
4
2
10 + 04
8
6
4
2
10 + 03

c. Length and Language: $616_8 = 398_{10}$; FAP language.

d. Details: On calling this subroutine, the appropriate information is written on tape B5.

e. Data Code Usage: LBLFGD is used by RDH-D for labeling the plots of data and calibration points.

2. Source Deck (FAP) Listing

```
*LBLFGD    FIX POINT LABEL GRID ROUTINE FOR 7090
           COUNT    300
*FIX PT LABELING VERSION OF LBLGRD,NOW LBLFGD
           ENTRY    LBLFGD
LBLFGD SXA      LBOUT,4
       SXA      LBOUT+1,2
       SXD LB21,1
       CLA 3,4
       STA LB01
       SUB ONE
       STA LB02
       CLA ONE
       STO FLAG
       CLA 1,4
       STA LB03
  LB01 CLA *
       STO MIN
  LB02 CLA *
       STO MAX
  LB03 LXD *,1
       TXH LB20,1,16
       TXH LOG,1,10
       TXL *+2,1,9
       LXD *-1,1
       TXL LB20,1,0
       CLA N+9,1
       STO OP
       ACL 0211
       FAD 0211
       SLW FOP
       CLA MAX
       FSB MIN
       FDP FOP
       STQ DELTA
       LXD ZERO,2
       LXD OP,1
       TXI *+1,1,1
       CLA MIN
  LB07 STO *+2
       TSX FIX,4
       NOP
       STO NUM,2
       CLA *-2
       FAD DELTA
       TXI *+1,2,-1
       TIX LB07,1,1
       TCOB     *
       OCT      77600002205
       CLA D900
       FDP FOP
       STQ COMMON
       LXD ZERO,1
       LXD OP,2
       CLA COMMON
       SSP
       UFA 0233
       RQL 8
       RND
       STZ DEL
       STA DEL
       LDQ FLAG
       TQP LB10
```

```
         CLS DEL
         STO DEL
         CLA      =8
         STO      COMMON
         CLA TSPZ
         STO TSP1
         TRA LB11
LB10     CLA DEL
         ALS 18
         STO DEL
         STZ      COMMON
         CLA TSPW
         STO TSP1
LB11     CLA XOYO
         STO COMMON+1
         TXI      *+1,2,1
         AXT      0,4
LB15     CLA      TSP1
         STO      BUFF,4
         TXL      LB21+1,4,0
         ADD      DEL
         STO      TSP1
         CLA      COMMON+1
         STO      BUFF+2,4
         ADD      DEL
         STO      COMMON+1
         CAL      NUM,1
         ALS      6
         ACL      =012
         SLW      BUFF+1,4
         TXI      *+1,1,-1
         TXI      *+1,4,-3
         TIX      LB15,2,1
LB16     PXA      ,4
         PAC      ,4
LB09     WTB      X
         SXD      IOC,4
         RCHX     IOC
LB20     CLA FLAG
         TMI LBOUT
         SSM
         STO FLAG
         LXA      LBOUT,4
         CLA 3,4
         SUB ONE
         SUB ONE
         STA LB01
         SUB ONE
         STA LB02
         CLA 2,4
         STA LB03
LB21     TXI LB01
         ADD      COMMON
         TRA      LB15+3
LBOUT    AXT      0,4
         AXT      0,2
         LXD LB21,1
         TOV *+1
         TRA 4,4
LOG      PXD 0,1
         SUB TEN
         PDX 0,1
         STO COMMON
```

```
        CLA     MIN
        UFA     0233
        ANA     0C77
        LLS     0
        ALS     18
        STO     TMP
LOG1    PXD 5
        SSM
        LDQ F900
        DVP COMMON
        STO COMMON
        LXA LOG1,2
LOG2    LDQ LOGN+5,2
        MPR COMMON
        STO COMMON+6,2
        TIX LOG2,2,1
        AXT     0,4
        CLA FLAG
        TMI LOG4
        AXT     6,2
LOG3    CLS COMMON+6,2
        ALS 18
        STO COMMON+6,2
        TIX LOG3,2,1
        CLA XPIP
        STO PIP
        TRA     LOG5-3
LOG4    CLA YPIP
        STO PIP
        CLA     INCY
        STO     INC
        CLA     XOYO
        TCOB    *
LOG5    STO COMMON+6
        LXA LOG1,2
LOG6    CLA COMMON+6
        ORA PMARK
        ADD COMMON+6,2
        STO     BUFF,4
        ANA MASK
        TXH     LOG7,2,4
        ADD PIP
        ORA NO+5,2
        STO     BUFF+1,4
        TXI     *+1,4,-2
LOG61   TIX LOG6,2,1
        CLA COMMON+6
        ADD COMMON
        TIX     LOG5,1,1
        TRA     LB16
LOG7    ADD     INC
        ORA     NJ
        STO     BUFF+1,4
        ADD     INC1
        ANA     MASK
        STO     BUFF+2,4
        ADD     INC2
        STO     BUFF+3,4
        CLA     TMP
        SSP
        CAS     TEN
        NOP
        ADD     066
```

```
              ØRA      SIP1
              STØ      BUFF+4,4
              CLA      TMP
              ADD      D1
              STØ      TMP
              TPL      *+3
              CAL      P40
              TRA      *+2
              CAL      P20
              ØRS      BUFF+4,4
              TXI      LØG61,4,-5
       FIX    SXA      FIX3,2
              CLA      1,4
              UFA      Ø233
              ANA      =Ø777777777
              LRS      35
              STZ      FIX9
              AXT      5,2
       FIX2   PXD      0,0
              DVP      FIX92+5,2
              XCA
              ACL      FIX9
              ALS      6
              SLW      FIX9
              TIX      FIX2,2,1
              LDQ      FIX9
              CLA      =Ø3760606060
              AXT      4,2
              TLQ      *+2
              TXI      *+3,2,1
              RQL      6
              TIX      *-3,2,1
              LGL      6
              TIX      *-1,2,1
       FIX3   AXT      0,2
              TRA      2,4
       FIX9   PZE
       FIX92  DEC      10000B35,1000B35,100B35,10B35,1B35
       TEN    PZE 0,,10
       Ø66    ØCT 66000000
       NJ     ØCT 10000,20000,40000,60000,100000
       LØGN   DEC 0,.30103000B0,.60205999B0,.77815125B0,.90308999B0
       ØNE    PZE 1
       FLAG   PZE
       MIN    PZE
       N      DEC 25B17,20B17,15B17,10B17,5B17,4B17,3B17,2B17,1B17
       ØP     PZE
       Ø211   ØCT 21100000000
       FØP    PZE
       MAX    PZE
       DELTA  PZE
       ZERØ   PZE
       D900   DEC 900.
       Ø233   ØCT 23300000000
       DEL    PZE
       TSPW   ØCT      200120601614
       TSPZ   ØCT      200113601575
       TSP1   PZE
       XJY0   ØCT 000173201604
       SIP1   ØCT 120000
       F900   PZE 0,,900
       ØC77   ØCT 77
       XPIP   PZE 16
```

```
  YPIP   MZE 0,,8
  INCY   MZE 0,,40
  INC    PZE
  INC1   PZE 0,,8
  INC2   ØCT 200000577770
  P40    MZE 0
  P20    PTW
  D1     PZE 0,,1
  PIP    PZE
  PMARK  ØCT 200000
  MASK   ØCT -377777007777
  TMP    BSS 2
  NUM    BSS 26
 CØMMØN  BSS 8
  IØC    IØCD     BUFF
  BUFF   BSS      80
  X      TAPENØ   B5
         END
```

O. LBLGRD SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To label the grid drawn by the select grid subroutine SELGRD. For linear grids the labeling is floating point.

b. Calling Sequence: CALL LBLGRD(I, J, XYBND)

I = Integer specifying option for labeling X boundary.

J = Integer specifying option for labeling Y boundary.

XYBND = Origin of array for which, in FORTRAN storage,

XYBND(1) = x-value at left boundary (X_{min})
XYBND(2) = x-value at right boundary(X_{max})
XYBND(3) = y-value at lower boundary(Y_{min})
XYBND(4) = y-value at upper boundary(Y_{max})

I or J may have the following values:

Linear Interval	Cycle logarithmic intervals
1 = 1 interval	11 = 1 cycle
2 = 2	12 = 2
3 = 3	13 = 3
4 = 4	14 = 4
5 = 5	15 = 5
6 = 10	16 = 6
7 = 15	
8 = 20	
9 = 25	
10 = 50	

Note that for logarithmic labeling Y_{min} must be the exponent corresponding to Y_{min}. For example, if PTLGMX = 3.0 and NYLB = 12, then the ordinate of the plot will be labeled:

8
6
4
2
10 + 04
8
6
4
2
10 + 03

c. Length and Language: $767_8 = 503_{10}$; FAP language.

d. Details: On calling this subroutine, the appropriate information is written on tape B5.

e. Data Code Usage: LBLGRD is used by the CAN-D code for labeling its plots of calibration points.

2. Source Deck (FAP) Listing

```
                    00000         ENTRY   LBLGRD

00000 -0634 00 4 00043  LBLGRD SXD LB04,4
00001 -0634 00 2 00044         SXD LB04+1,2
00002 -0634 00 1 00176         SXD LB21,1
00003  0500 00 4 00003         CLA 3,4
00004  0621 00 0 00013         STA LB01
00005  0402 00 0 00525         SUB ONE
00006  0621 00 0 00015         STA LB02
00007  0500 00 0 00525         CLA ONE
00010  0601 00 0 00526         STO FLAG
00011  0500 00 4 00001         CLA 1,4
00012  0621 00 0 00017         STA LB03
00013  0500 00 0 00013  LB01   CLA *
00014  0601 00 0 00527         STO MIN
00015  0500 00 0 00015  LB02   CLA *
00016  0601 00 0 00544         STO MAX
00017 -0534 00 1 00017  LB03   LXD *,1
00020  3 00020 1 00161         TXH LB20,1,16
00021  3 00012 1 00204         TXH LOG,1,10
00022 -3 00011 1 00024         TXL *+2,1,9
00023 -0534 00 1 00022         LXD *-1,1
00024 -3 00000 1 00161         TXL LB20,1,0
00025  0500 00 1 00541         CLA N+9,1
00026  0601 00 0 00541         STO OP
00027  0361 00 0 00542         ACL O211
00030  0300 00 0 00542         FAD O211
00031  0602 00 0 00543         SLW FOP
00032  0500 00 0 00544         CLA MAX
00033  0302 00 0 00527         FSB MIN
00034  0241 00 0 00543         FDP FOP
00035 -0600 00 0 00545         STQ DELTA
00036  0500 00 0 00527         CLA MIN
00037  0602 00 0 00546         SLW ABMIN
00040  0500 00 0 00544         CLA MAX
00041  0760 00 0 00003         SSP
00042  0340 00 0 00546         CAS ABMIN
00043  1 00000 0 00046  LB04   TXI LB05
00044  1 00000 0 00046         TXI LB05
00045  0500 00 0 00546         CLA ABMIN
00046  0601 00 0 00050  LB05   STO LB06
00047  0074 00 4 00362         TSX FPT,4
00050  0761 00 0 00000  LB06   NOP
00051 -0534 00 2 00547         LXD ZERO,2
00052 -0534 00 1 00541         LXD OP,1
00053  1 00001 1 00054         TXI *+1,1,1
00054  0500 00 0 00527         CLA MIN
00055  0601 00 0 00057  LB07   STO *+2
00056  0074 00 4 00322         TSX FIX,4
00057  0761 00 0 00000         NOP
00060  0601 00 2 00604         STO NUM,2
00061  0500 00 0 00057         CLA *-2
00062  0300 00 0 00545         FAD DELTA
00063  1 77777 2 00064         TXI *+1,2,-1
00064  2 00001 1 00055         TIX LB07,1,1
00065  0061 00 0 00065         TCOB    *
00066 +077600002205            OCT     77600002205
```

```
00067  0500 00 0 00550         CLA D900
00070  0241 00 0 00543         FDP FOP
00071 -0600 00 0 00636         STQ COMMON
00072 -0534 00 1 00547         LXD ZERO,1
00073 -0534 00 2 00541         LXD OP,2
00074  0500 00 0 00636         CLA COMMON
00075  0760 00 0 00003         SSP
00076 -0300 00 0 00551         UFA 0233
00077 -0773 00 0 00010         RQL 8
00100  0760 00 0 00010         RND
00101  0600 00 0 00552         STZ DEL
00102  0621 00 0 00552         STA DEL
00103  0560 00 0 00526         LDQ FLAG
00104  0162 00 0 00113         TQP LB10
00105  0502 00 0 00552         CLS DEL
00106  0601 00 0 00552         STO DEL
00107  0500 00 0 00556         CLA TSPZ
00110  0601 00 0 00557         STO TSP1
00111  0500 00 0 00554         CLA TSPY
00112  0020 00 0 00121         TRA LB11
00113  0500 00 0 00552  LB10   CLA DEL
00114  0767 00 0 00022         ALS 18
00115  0601 00 0 00552         STO DEL
00116  0500 00 0 00555         CLA TSPW
00117  0601 00 0 00557         STO TSP1
00120  0500 00 0 00553         CLA TSPX
00121  0601 00 0 00560  LB11   STO TSP
00122  0601 00 0 00650         STO      BUFF
00123  0500 00 0 00561         CLA XOYO
00124  0601 00 0 00637         STO COMMON+1
00125  0601 00 0 00653         STO      BUFF+3
00126  0500 00 0 00604         CLA      NUM
00127  0361 00 0 00646         ACL      BB
00130  0602 00 0 00651         SLW      BUFF+1
00131 -0500 00 0 00603         CAL      TMP+1
00132  0361 00 0 00562         ACL      STP1
00133  0602 00 0 00652         SLW      BUFF+2
00134 -0774 00 4 00004         AXC      4,4
00135  0500 00 0 00557  LB15   CLA      TSP1
00136  0400 00 0 00552         ADD      DEL
00137  0601 00 0 00557         STO      TSP1
00140  0601 00 4 00650         STO      BUFF,4
00141  0500 00 0 00637         CLA      COMMON+1
00142  0400 00 0 00552         ADD      DEL
00143  0601 00 0 00637         STO      COMMON+1
00144  0601 00 4 00652         STO      BUFF+2,4
00145 -0500 00 1 00605         CAL      NUM+1,1
00146  0767 00 0 00006         ALS      6
00147  0361 00 0 00563         ACL      STP
00150  0602 00 4 00651         SLW      BUFF+1,4
00151  1 77777 1 00152         TXI      *+1,1,-1
00152  1 77775 4 00153         TXI      *+1,4,-3
00153  2 00001 2 00135         TIX      LB15,2,1
00154  0754 00 4 00000  LB16   PXA      ,4
00155  0737 00 4 00000         PAC      ,4
00156  0766 00 0 02225  LB09   WTB      X
00157 -0634 00 4 00647         SXD      IOC,4
00160 -0540 00 0 00647         RCHX     IOC
```

```
00161  0500 00 0 00526   LB20  CLA FLAG
00162 -0120 00 0 00177         TMI LBOUT
00163 -0760 00 0 00003         SSM
00164  0601 00 0 00526         STO FLAG
00165 -0534 00 4 00043         LXD LB04,4
00166  0500 00 4 00003         CLA 3,4
00167  0402 00 0 00525         SUB ONE
00170  0402 00 0 00525         SUB ONE
00171  0621 00 0 00013         STA LB01
00172  0402 00 0 00525         SUB ONE
00173  0621 00 0 00015         STA LB02
00174  0500 00 4 00002         CLA 2,4
00175  0621 00 0 00017         STA LB03
00176  1 00000 0 00013   LB21  TXI LB01
00177 -0534 00 4 00043   LBOUT LXD LB04,4
00200 -0534 00 2 00044         LXD LB04+1,2
00201 -0534 00 1 00176         LXD LB21,1
00202  0140 00 0 00203         TOV *+1
00203  0020 00 4 00004         TRA 4,4
00204 -0754 00 1 00000   LOG   PXD 0,1
00205  0402 00 0 00440         SUB TEN
00206 -0734 00 1 00000         PDX 0,1
00207  0601 00 0 00636         STO COMMON
00210  0500 00 0 00527         CLA     MIN
00211 -0300 00 0 00551         UFA     0233
00212 -0320 00 0 00565         ANA     0C77
00213  0763 00 0 00000         LLS     0
00214  0767 00 0 00022         ALS     18
00215  0601 00 0 00602         STO     TMP
00216 -0754 00 0 00005   LOG1  PXD 5
00217 -0760 00 0 00003         SSM
00220  0560 00 0 00564         LDQ F900
00221  0221 00 0 00636         DVP COMMON
00222 -0600 00 0 00636         STQ COMMON
00223  0534 00 2 00216         LXA LOG1,2
00224  0560 00 2 00525   LOG2  LDQ LOGN+5,2
00225 -0200 00 0 00636         MPR COMMON
00226  0601 00 2 00644         STO COMMON+6,2
00227  2 00001 2 00224         TIX LOG2,2,1
00230  0774 00 4 00000         AXT     0,4
00231  0500 00 0 00526         CLA FLAG
00232 -0120 00 0 00243         TMI LOG4
00233  0774 00 2 00006         AXT     6,2
00234  0502 00 2 00644   LOG3  CLS COMMON+6,2
00235  0767 00 0 00022         ALS 18
00236  0601 00 2 00644         STO COMMON+6,2
00237  2 00001 2 00234         TIX LOG3,2,1
00240  0500 00 0 00566         CLA XPIP
00241  0601 00 0 00577         STO PIP
00242  0020 00 0 00246         TRA     LOG5-3
00243  0500 00 0 00567   LOG4  CLA YPIP
00244  0601 00 0 00577         STO PIP
00245  0500 00 0 00570         CLA     INCY
00246  0601 00 0 00571         STO     INC
00247  0500 00 0 00561         CLA     XOYO
00250  0061 00 0 00250         TCOB    *
00251  0601 00 0 00644   LOG5  STO COMMON+6
00252  0534 00 2 00216         LXA LOG1,2
```

```
00253  0500 00 0 00644   LOG6  CLA COMMON+6
00254 -0501 00 0 00600         ORA PMARK
00255  0400 00 2 00644         ADD COMMON+6,2
00256  0601 00 4 00650         STO     BUFF,4
00257 -0320 00 0 00601         ANA MASK
00260  3 00004 2 00272         TXH     LOG7,2,4
00261  0400 00 0 00577         ADD PIP
00262 -0501 00 2 00517         ORA NO+5,2
00263  0601 00 4 00651         STO     BUFF+1,4
00264  1 77776 4 00265         TXI     *+1,4,-2
00265  2 00001 2 00253   LOG61 TIX LOG6,2,1
00266  0500 00 0 00644         CLA COMMON+6
00267  0400 00 0 00636         ADD COMMON
00270  2 00001 1 00251         TIX     LOG5,1,1
00271  0020 00 0 00154         TRA     LB16
00272  0400 00 0 00571   LOG7  ADD     INC
00273 -0501 00 0 00512         ORA     NO
00274  0601 00 4 00651         STO     BUFF+1,4
00275  0400 00 0 00572         ADD     INC1
00276 -0320 00 0 00601         ANA     MASK
00277  0601 00 4 00652         STO     BUFF+2,4
00300  0400 00 0 00573         ADD     INC2
00301  0601 00 4 00653         STO     BUFF+3,4
00302  0500 00 0 00602         CLA     TMP
00303  0760 00 0 00003         SSP
00304  0340 00 0 00440         CAS     TEN
00305  0761 00 0 00000         NOP
00306  0400 00 0 00441         ADD     066
00307 -0501 00 0 00562         ORA     STP1
00310  0601 00 4 00654         STO     BUFF+4,4
00311  0500 00 0 00602         CLA     TMP
00312  0400 00 0 00576         ADD     D1
00313  0601 00 0 00602         STO     TMP
00314  0120 00 0 00317         TPL     *+3
00315 -0500 00 0 00574         CAL     P40
00316  0020 00 0 00320         TRA     *+2
00317 -0500 00 0 00575         CAL     P20
00320 -0602 00 4 00654         ORS     BUFF+4,4
00321  1 77773 4 00265         TXI     LOG61,4,-5
00322 -0634 00 2 00345   FIX   SXD FI09,2
00323 -0534 00 2 00351         LXD FI11,2
00324  0500 00 4 00001         CLA 1,4
00325  0760 00 0 00003         SSP
00326  0765 00 0 00043         LRS 35
00327  0260 00 0 00602         FMP TMP
00330  0600 00 0 00355         STZ FI91
00331  0300 00 0 00356         FAD FI92
00332 -0300 00 0 00551         UFA 0233
00333  0763 00 0 00010         LLS 8
00334 -0600 00 0 00357   FI05  STQ FI93
00335  0500 00 0 00357         CLA FI93
00336  0771 00 0 00002         ARS 2
00337  0400 00 0 00357         ADD FI93
00340  0765 00 0 00040         LRS 32
00341  0400 00 0 00355         ADD FI91
00342 -2 00001 2 00346         TNX FI10,2,1
00343  0767 00 0 00006         ALS 6
00344  0601 00 0 00355         STO FI91
```

```
00345  1 00000 0 00334   FI09  TXI FI05,,
00346  0560 00 4 00001   FI10  LDQ 1,4
00347  0162 00 0 00352         TQP FI15
00350  0361 00 0 00360         ACL MINUS
00351  1 00003 0 00353   FI11  TXI FI16,,3
00352  0361 00 0 00361   FI15  ACL PLUS
00353 -0534 00 2 00345   FI16  LXD FI09,2
00354  0020 00 4 00002         TRA 2,4
00355  0 00000 0 00000   FI91  PZE
00356 +166406111564      FI92  DEC .0005
00357  0 00000 0 00000   FI93  PZE
00360  0 00040 0 00000   MINUS PZE 0,,32
00361  0 00020 0 00000   PLUS  PZE 0,,16
00362 -0634 00 2 00376   FPT   SXD FP06,2         SAVE X2
00363  0500 00 4 00001         CLA 1,4            ABS.VALUE OF X
00364  0765 00 0 00033         LRS 27             SHIFT CHARACTERICTIC
00365  0402 00 0 00435         SUB 0200           200 OCTAL
00366  0765 00 0 00043         LRS 35             POWER OF 2 TO MQ
00367  0601 00 0 00603         STO TMP+1
00370  0200 00 0 00521         MPY LOGN+1
00371  0734 00 2 00000         PAX 0,2
00372  0120 00 0 00422         TPL FP20
00373  0560 00 4 00001   FP05  LDQ 1,4            ABS. VALUE OF X
00374  0260 00 2 00511         FMP PTEN,2
00375  0340 00 0 00436         CAS TENTH
00376  1 00000 0 00401   FP06  TXI FP08           NBR. GREATER THAN 1/10
00377  1 00000 0 00401         TXI FP08           NBR. EQUAL 1/10
00400  1 00001 2 00373         TXI FP05,2,1       NBR. LESS THAN 1/10
00401  0500 00 2 00511   FP08  CLA PTEN,2
00402  0601 00 0 00602         STO TMP            TEN MULTIPLIER
00403 -0634 00 2 00603   FP09  SXD TMP+1,2
00404  0500 00 0 00603         CLA TMP+1
00405  0402 00 0 00437         SUB THREE
00406  0601 00 0 00603         STO TMP+1
00407  0760 00 0 00003         SSP
00410  0340 00 0 00440         CAS TEN
00411  0761 00 0 00000         NOP
00412  0400 00 0 00441         ADD 066
00413  0622 00 0 00603         STD TMP+1
00414  0500 00 0 00603         CLA TMP+1
00415 -0120 00 0 00417         TMI FP10
00416  0400 00 0 00442         ADD OCT1
00417  0601 00 0 00603   FP10  STO TMP+1
00420 -0534 00 2 00376         LXD FP06,2
00421  0020 00 4 00002         TRA 2,4
00422  0500 00 4 00001   FP20  CLA 1,4
00423  0241 00 2 00511         FDP PTEN,2
00424 -0600 00 0 00636         STQ COMMON
00425  0500 00 0 00636         CLA COMMON
00426  0340 00 0 00517         CAS D9995          .9995
00427  1 00001 2 00422         TXI FP20,2,1
00430  0761 00 0 00000         NOP
00431  0500 00 0 00511   FP25  CLA PTEN
00432  0241 00 2 00511         FDP PTEN,2
00433 -0600 00 0 00602         STQ TMP
00434  0020 00 0 00403         TRA FP09
00435 +000000000200      0200  OCT 200
00436 +175631312257      TENTH DEC .09995
```

```
00437  0 00003 0 00000   THREE PZE 0,,3
00440  0 00012 0 00000   TEN   PZE 0,,10
00441 +000066000000      066   OCT 66000000
00442 +200000000000      OCT1  OCT 200000000000
00443 +377454732313            DEC 1E38,1E37,1E36,1E35,1E34,1E33,1E32,1E31,1E30
00444 +373741367021
00445 +370601137164
00446 +365464114135
00447 +361755023373
00450 +356612334311
00451 +353473426555
00452 +347770675742
00453 +344623713116
00454 +341503074077            DEC 1E29,1E28,1E27,1E26,1E25,1E24,1E23,1E22,1E21,1E20
00455 +336402374714
00456 +332635456171
00457 +327512676456
00460 +324410545213
00461 +320647410336
00462 +315522640262
00463 +312417031702
00464 +306661534466
00465 +303532743536
00466 +300425434430            DEC 1E19,1E18,1E17,1E16,1E15,1E14,1E13,1E12,1E11,1E10
00467 +274674055532
00470 +271543212741
00471 +266434157116
00472 +262706576512
00473 +257553630410
00474 +254443023471
00475 +250721522451
00476 +245564416672
00477 +242452013710
00500 +236734654500            DEC 1E9,1E8,1E7,1E6,1E5,1E4,1E3,1E2,1E1
00501 +233575360400
00502 +230461132000
00503 +224750220000
00504 +221606500000
00505 +216470400000
00506 +212764000000
00507 +207620000000
00510 +204500000000
00511 +201400000000      PTEN  DEC 1.
00512 +000000010000      NO    OCT 10000,20000,40000,60000,100000
00513 +000000020000
00514 +000000040000
00515 +000000060000
00516 +000000100000
00517 +200777574733      D9995 DEC .9995
00520 +000000000000      LOGN  DEC 0,.30103000B0,.60205999B0,.77815125B0,.90308999B0
00521 +115040465250
00522 +232101151772
00523 +307151656320
00524 +347141637243
00525  0 00000 0 00001   ONE   PZE 1
00526  0 00000 0 00000   FLAG  PZE
00527  0 00000 0 00000   MIN   PZE
00530 +000031000000      N     DEC 25B17,20B17,15B17,10B17,5B17,4B17,3B17,2B17,1B17
```

```
00531 +000024000000
00532 +000017000000
00533 +000012000000
00534 +000005000000
00535 +000004000000
00536 +000003000000
00537 +000002000000
00540 +000001000000
00541  0 00000 0 00000   OP     PZE
00542 +211000000000      O211   OCT 211000000000
00543  0 00000 0 00000   FOP    PZE
00544  0 00000 0 00000   MAX    PZE
00545  0 00000 0 00000   DELTA  PZE
00546  0 00000 0 00000   ABMIN  PZE
00547  0 00000 0 00000   ZERO   PZE
00550 +212702000000      D900   DEC 900.
00551 +233000000000      O233   OCT 233000000000
00552  0 00000 0 00000   DEL    PZE
00553 +200053601614      TSPX   OCT     200053601614
00554 +200053601574      TSPY   OCT     200053601574
00555 +200117601614      TSPW   OCT     200117601614
00556 +200113601605      TSPZ   OCT     200113601605
00557  0 00000 0 00000   TSP1   PZE
00560  0 00000 0 00000   TSP    PZE
00561 +000173201604      XOYO   OCT 000173201604
00562 +000000120000      STP1   OCT 120000
00563 -200000000012      STP    OCT     -200000000012
00564  0 01604 0 00000   F900   PZE 0,,900
00565 +000000000077      OC77   OCT 77
00566  0 00000 0 00020   XPIP   PZE 16
00567 -0 00010 0 00000   YPIP   MZE 0,,8
00570 -0 00050 0 00000   INCY   MZE 0,,40
00571  0 00000 0 00000   INC    PZE
00572  0 00010 0 00000   INC1   PZE 0,,8
00573 +200000577770      INC2   OCT 200000577770
00574 -0 00000 0 00000   P40    MZE 0
00575  2 00000 0 00000   P20    PTW
00576  0 00001 0 00000   D1     PZE 0,,1
00577  0 00000 0 00000   PIP    PZE
00600 +000000200000      PMARK  OCT 200000
00601 -377777007777      MASK   OCT -377777007777
00602                    TMP    BSS 2
00604                    NUM    BSS 26
00636                   COMMON  BSS 8
00646  606000000000      BB     BCD 1  0000
00647  0 00000 0 00650   IOC    IOCD    BUFF
00650                    BUFF   BSS     80
               02205     X      TAPENO  B5
                                END

00770 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

   NO ERROR IN ABOVE ASSEMBLY
```

P. ADVFLM SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To write information on tape B5 commanding the 40/20 to advance its film n frames.

b. Calling Sequence: CALL ADVFLM(I, J)

I = 0 (Hard copy not requested)

I = 1 (Hard copy requested)

J = Number of additional frames to advance.

c. Length and Language $41_8 = 33_{10}$; FAP language.

d. Details: Upon entry to this subroutine, the film is advanced one frame. If a hard copy is requested, this frame is marked with a series of vertical lines and the film is advanced IF additional frames.

e. Data Code Usage: This subroutine is normally used with the following calling sequence: CALL ADVFLM (0, 2). This leaves enough room between data frames for mounting on cards for input to the Xerox printer.

2. Source Deck (FAP) Listing

LAADVF   ADVANCE FILM ROUTINE   5/24/61

```
                  00000            ENTRY    ADVFLM

00000  0061 00 0 00000  ADVFLM TCOB     *
00001 +077600002205            OCT      7760002205
00002  0766 00 0 02225         WTBX
00003 -0540 00 0 00037         RCHX     AD93
00004  0500 60 4 00001         CLA*     1,4
00005 -0634 00 2 00034         SXD      AD90,2
00006  0100 00 0 00022         TZE      AD03
00007  0500 00 0 00033         CLA      GYA
00010  0601 00 0 00035         STO      AD91
00011  0774 00 2 00013         AXT      11,2
00012  0766 00 0 02225  AD01   WTBX
00013  0500 00 0 00035         CLA      AD91
00014  0400 00 0 00036         ADD      AD92
00015  0601 00 0 00035         STO      AD91
00016 -0540 00 0 00040         RCHX     AD94
00017  2 00001 2 00012         TIX      AD01,2,1
00020  0766 00 0 02225         WTBX
00021 -0540 00 0 00037         RCHX     AD93
00022  0500 60 4 00002  AD03   CLA*     2,4
00023  0100 00 0 00030         TZE      AD05
00024 -0734 00 2 00000         PDX      0,2
00025  0766 00 0 02225  AD04   WTBX
00026 -0540 00 0 00037         RCHX     AD93
00027  2 00001 2 00025         TIX      AD04,2,1
00030 -0534 00 2 00034  AD05   LXD      AD90,2
00031  0020 00 4 00003         TRA      3,4
00032 -160000000000     RST    OCT      -160000000000
00033 +320704001777     GYA    OCT      320704001777
00034  0 00000 0 00000  AD90   PZE
00035  0 00000 0 00000  AD91   PZE
00036  0 00012 0 00000  AD92   PZE      0,,10
00037  0 00001 0 00032  AD93   IOCD     RST,,1
00040  0 00001 0 00035  AD94   IOCD     AD91,,1
                  02205 X      TAPENO   B5
                               END

00041 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

   NO ERROR IN ABOVE ASSEMBLY
```

Q. CLAS SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To write information on tape B5 commanding the 40/20 to write "CONFIDENTIAL RESTRICTED DATA" on its output film

b. Calling Sequence: CALL CLAS

c. Length and Language, $232_8 = 154_{10}$; FAP language

d. Details: On calling this subroutine, the appropriate information is written on tape B5.

e. Data Code Usage: This is used by RDH-D for each data frame (but not on the calibration frames) whenever the quantity ICNFRD is non-zero. ICNFRD is an input number punched once for each multiplexer.

2. Source Deck (FAP) Listing

```
       ENTRY   CLAS
*LACLAS   WRITE LARGE CONFIDENTIAL OF FILM
 CLAS  WTB 7
       SXD CL90,4
       LXA CL90,4
       CPY BUFF+147,4
       CPY BUFF+147,4
       TIX *-2,4,1
       LXD CL90,4
       TRA 1,4
 CL90  PZE 147
 BUFF  OCT 600404160004,660404400004,660404400074
       OCT 600434020004,600434220074
       OCT 600444120014,620444420064,620454400074
       OCT 620464620074,600474320064,620474220014
       OCT 620464000004,620454020004
       OCT 600504360074,660504560004,600534360074
       OCT 600544360074,660544400004,650544400040
       OCT 600620360074,620614400004,620614400074
       OCT 600644160004,630644400074,630660630074
       OCT 600674300060,630674230020,630660000004
       OCT 600704160004,660704400004,650704400040
       OCT 660704400074
       OCT 600744360074,660744560004,600774360074
       OCT 601020160004,661004400004
       OCT 601060360074,621054400004,621054400074
       OCT 631120160004,631120560004,631112400040
       OCT 601144360074,661144400074
       OCT 600104161704,660104401704,660104401774
       OCT 600134021704,600134221774
       OCT 600144121714,620144421764,620154401774
       OCT 620164621774,600174321764,620174221714
       OCT 620164001704,620154021704
       OCT 600204361774,660204561704,600234361774
       OCT 600244361774,660244401704,650244401740
       OCT 600320361774,620314401704,620314401774
       OCT 600344161704,630344401774,630360631774
       OCT 600374301760,630374231720,630360001704
       OCT 600404161704,660404401704,650404401740
       OCT 660404401774
       OCT 600444361774,660444561704,600474361774
       OCT 600520161704,660504401704
       OCT 600560361774,620554401704,620554401774
       OCT 630620161704,630620561704,630612401740
       OCT 600644361774,660644401774
       OCT 601044361774,661044401704,661044401734
       OCT 601074061704,631060501734
       OCT 601104161704,661104401704,651104401740
       OCT 661104401774
       OCT 601174221714,661174001704,601144071704
       OCT 661144401740,601174071740,661174001774
       OCT 601144221774
       OCT 601220361774,661204401704
       OCT 601244361774,661244401704,661244401734
       OCT 601274061704,631260501734
       OCT 601320361774,621314401704,621314401774
       OCT 601344161704,661344401704,661344401774
       OCT 601374021704,601374221774
       OCT 601420161704,661404401704
       OCT 601444161704,661444401704,651444401740,661444401774
       OCT 601504161704,631504401774,631520631774
       OCT 601534301760,631534231720,631520001704
```

```
      OCT 601604161704,631604401774,631620631774
      OCT 601634301760,631634231720,631620001704
      OCT 631660161704,631660561704,631667001740
      OCT 601720161704,661704401704
      OCT 631760161704,631760561704,631767001740
      END
```

R. LINCNT SUBROUTINE

Purpose: To set a line count for writing BCD information on the 40/20 with FØRTRAN WRITE ØUTPUT TAPE 12 statements.

Calling Sequence: CALL LINCNT(N)

N = an integer, $1 \leqq N \leqq 64$, indicating the next line to be written.

Data Code Usage: LINCNT is used by CAN-D and RDH-D. This usage is completely automatic.

S. PLFILM SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To plot linearly a set of points with a specified symbol, with an option of connecting successive points with a straight line.

b. Calling Sequence: CALL PLFILM (N, NWORDS, X, IX, Y, IY, XYBND, L, IOP)

N = Number of points.

NWORDS = Location of N words (stored backwards) that may be used for temporary storage by the subroutine.

X = Origin of table of abscissa.

IX = Memory spacing between members of the X table (-1 for single dimensioned FORTRAN storage).

Y = Origin of table of ordinates.

IY = Memory spacing between members of the y table (-1 for single dimensioned FORTRAN storage).

XYBND = Origin of an array of 4 values defining the boundaries of the graph

XYBND(1) = x-value at left boundary (X_{min})
XYBND(2) = x-value at right boundary (X_{max})
XYBND(3) = y-value at lower boundary (Y_{min})
XYBND(4) = y-value at upper boundary (Y_{max})

L = Integer value of the BCD code for the plot symbol.

If superposition of two symbols is desired calculate

L = 64 x symbol 2 + symbol 1

IOP = 0 - don't connect points
      1 - connect points

N, IX, IY, L, and IOP are fixed point integers. The following symbols may be used on the 40/20:

Symbol	Decimal	Octal	Symbol	Decimal	Octal	Symbol	Decimal	Octal
0	0	00	F	22	26	*	44	54
1	1	01	G	23	27	γ	45	55
2	2	02	H	24	30	~	46	56*
3	3	03	I	25	31	d	47	57
4	4	04	π	26	32		48	60
5	5	05	°	27	33	/	49	61
6	6	06	)	28	34	S	50	62
7	7	07	β	29	35	T	51	63
8	8	10	"	30	36	U	52	64
9	9	11	?	31	37	V	53	65
∂	10	12*	-	32	40	W	54	66
=	11	13	J	33	41	X	55	67
±	12	14	K	34	42	Y	56	70
'	13	15	L	35	43	Z	57	71
δ	14	16	M	36	44	o	58	72
α	15	17	N	37	45	,	59	73
+	16	20	O	38	46	(	60	74
A	17	21	P	39	47	$\int$	61	75
B	18	22	Q	40	50	Σ	62	76
C	19	23	R	41	51	□	63	77
D	20	24	.	42	52*			
E	21	25	$	43	53			

* The octal codes 12, 52 and 56 cannot be used on the 40/20 ID card and when writing horizontally on the film (M-093-F)

c. Length and Language: $461_8 = 305_{10}$; FAP language.

d. Details: A frame is considered to be divided into 64 rows and 128 character positions (columns). However, both R and C may be multiples of 1/4 ($3/4 \leq R \leq 64\text{-}1/4$; $3/4 \leq C \leq 128\text{-}1/4$) to allow more exact positioning, superscripts, or subscripts.

When writing vertically, the first character is positioned at row R and column C and succeeding characters are each spaced down one row.

No more characters will be printed after row 64 is reached. When writing horizontally, the first character is positioned as before and succeeding characters are each spaced one column to the right. After column 128 is reached, the next character (if any), will be positioned one row down and at column 1.

e. Data Code Usage: This subroutine is used by RDH-D and CAN-D to write data and calibration points on tape B5 for input to the 40/20.

2. Source Deck (FAP) Listing

```
                       00000          ENTRY   PLFILM

00000 -0634 00 4 00073  PLFILM SXD PL31,4
00001 -0634 00 2 00074         SXD PL31+1,2
00002 -0634 00 1 00111         SXD PL35+1,1
00003  0500 00 4 00007         CLA 7,4
00004  0621 00 0 00056         STA PL20+1              L(XMIN)
00005  0621 00 0 00075         STA PL32
00006  0402 00 0 00355         SUB PL90                1
00007  0621 00 0 00055         STA PL20                L(XMAX)
00010  0621 00 0 00072         STA PL30+1
00011  0402 00 0 00355         SUB PL90                1
00012  0621 00 0 00064         STA PL21+1              L(YMIN)
00013  0621 00 0 00110         STA PL35
00014  0402 00 0 00355         SUB PL90                1
00015  0621 00 0 00063         STA PL21                L(YMAX)
00016  0621 00 0 00115         STA PL38
00017  0500 00 4 00003         CLA 3,4                 L(X)
00020  0621 00 0 00071         STA PL30
00021  0500 60 4 00004         CLA*    4,4
00022 -0120 00 0 00025         TMI PL015
00023  0760 00 0 00006         COM
00024  0361 00 0 00372         ACL PL99
00025  0622 00 0 00127  PL015  STD PL40
00026  0500 00 4 00005         CLA 5,4                 L(Y)
00027  0621 00 0 00107         STA PL34
00030  0500 60 4 00006         CLA*    6,4
00031 -0120 00 0 00034         TMI PL025
00032  0760 00 0 00006         COM
00033  0361 00 0 00372         ACL PL99
00034  0622 00 0 00130  PL025  STD PL41
00035  0500 00 4 00002         CLA 2,4                 L(NWORDS)
00036  0621 00 0 00106         STA PL33
00037  0621 00 0 00126         STA PL39
00040  0621 00 0 00157         STA PL50
00041  0621 00 0 00232         STA PL81
00042  0621 00 0 00240         STA PL82
00043  0621 00 0 00331         STA PL87
00044  0621 00 0 00336         STA PL89
00045  0500 00 4 00001         CLA 1,4                 L(N)
00046  0621 00 0 00152         STA PL44
00047  0621 00 0 00050         STA PL03
00050 -0534 00 1 00050  PL03   LXD *,1
00051 -3 00000 1 00207         TXL PL56,1,0
00052  1 77777 1 00053         TXI PL04,1,-1
00053 -0634 00 1 00132  PL04   SXD PL43,1
00054 -0534 00 7 00051         LXD PL03+1,7
00055  0500 00 0 00055  PL20   CLA *                   XMAX
00056  0302 00 0 00056         FSB *                   XMIN
00057  0601 00 0 00370         STO TEMP
00060  0500 00 0 00365         CLA P900
00061  0241 00 0 00370         FDP TEMP
00062 -0600 00 0 00366         STQ XMC
00063  0500 00 0 00063  PL21   CLA *                   YMAX
00064  0302 00 0 00064         FSB *                   YMIN
00065  0601 00 0 00370         STO TEMP
00066  0500 00 0 00365         CLA P900
```

```
00067  0241 00 0 00370          FDP TEMP
00070 -0600 00 0 00367          STQ YMC
00071  0500 00 1 00071   PL30   CLA *,1              X
00072  0340 00 0 00072          CAS *                XMAX
00073  1 00000 0 00212   PL31   TXI PL70
00074  1 00000 0 00212          TXI PL70
00075  0302 00 0 00075   PL32   FSB *                XMIN
00076 -0120 00 0 00214          TMI PL71
00077  0601 00 0 00370          STO TEMP             X-XMIN
00100  0560 00 0 00370          LDQ TEMP
00101  0260 00 0 00366          FMP XMC
00102 -0300 00 0 00356          UFA PL91             23200...
00103  0400 00 0 00357          ADD PL92             247
00104 -0320 00 0 00360          ANA PL93             77776
00105  0767 00 0 00021          ALS 17
00106  0602 00 4 00106   PL33   SLW *,4              NWORDS
00107  0500 00 2 00107   PL34   CLA *,2              Y
00110  0340 00 0 00110   PL35   CAS *                YMIN
00111  1 00000 0 00114          TXI PL38-1
00112  1 00000 0 00216          TXI PL72
00113  1 00000 0 00216          TXI PL72
00114  0760 00 0 00002          CHS
00115  0300 00 0 00115   PL38   FAD *                YMAX
00116 -0120 00 0 00220          TMI PL73
00117  0601 00 0 00370          STO TEMP
00120  0560 00 0 00370          LDQ TEMP
00121  0260 00 0 00367          FMP YMC
00122 -0300 00 0 00356          UFA PL91             23200...
00123  0400 00 0 00355          ADD PL90             1
00124 -0320 00 0 00360          ANA PL93             77776
00125  0771 00 0 00001          ARS 1
00126  0621 00 4 00126   PL39   STA *,4              NWORDS
00127  1 00127 1 00130   PL40   TXI PL41,1,*         IX
00130  1 00130 2 00131   PL41   TXI PL42,2,*         IY
00131  1 00001 4 00132   PL42   TXI PL43,4,1
00132 -3 00132 4 00071   PL43   TXL PL30,4,*         M-1
00133 -0534 00 4 00073          LXD     PL31,4
00134  0500 60 4 00010          CLA*    8,4
00135  0771 00 0 00006          ARS     6
00136  0560 00 0 00361          LDQ     PL95
00137  0040 00 0 00142          TLQ     PL435
00140  0601 00 0 00374          STO     K1
00141  0020 00 0 00151          TRA     PL436
00142 -0320 00 0 00361   PL435  ANA     PL95
00143  0601 00 0 00374          STO     K1
00144  0500 60 4 00010          CLA*    8,4
00145  0771 00 0 00014          ARS     12
00146 -0320 00 0 00361          ANA     PL95
00147  0601 00 0 00375          STO     K2
00150  0020 00 0 00152          TRA     *+2
00151  0560 00 0 00132   PL436  LDQ     PL43
00152 -0534 00 4 00000   PL44   LXD     0,4
00153  0774 00 2 00000          AXT     0,2
00154  0774 00 1 00062   PL45   AXT     50,1
00155  0061 00 0 00155          TCOB    *
00156 +07760000220 5            OCT     77600002205
00157  0500 00 2 00157   PL50   CLA     *,2
00160 -0501 00 0 00374          ORA     K1
```

```
00161  0601 00 1 00461          STO      BUFF+50,1
00162 -2 00001 1 00177          TNX      PL52,1,1
00163  0162 00 0 00172          TQP      PL51
00164  1 00001 2 00165          TXI      *+1,2,1
00165  2 00001 4 00157   PL505  TIX      PL50,4,1
00166  1 77716 1 00167          TXI      *+1,1,-50
00167  0754 00 1 00000          PXA      0,1
00170  0737 00 1 00000          PAC      0,1
00171  0020 00 0 00200          TRA      PL52+1
00172  0522 00 0 00157   PL51   XEC      PL50
00173 -0501 00 0 00375          ORA      K2
00174  0601 00 1 00461          STO      BUFF+50,1
00175 -2 00001 1 00177          TNX      PL52,1,1
00176  1 00001 2 00165          TXI      PL505,2,1
00177  0534 00 1 00154   PL52   LXA      PL45,1
00200  0766 00 0 02225          WTB      X
00201 -0634 00 1 00376          SXD      IOC,1
00202 -0540 00 0 00376          RCHX     IOC
00203  2 00001 4 00154          TIX      PL45,4,1
00204 -0534 00 4 00073          LXD PL31,4
00205  0500 60 4 00011          CLA*     9,4
00206 -0100 00 0 00222          TNZ PL80
00207 -0534 00 2 00074   PL56   LXD PL31+1,2
00210 -0534 00 1 00111          LXD PL35+1,1
00211  0020 00 4 00012          TRA 10,4
00212 -0500 00 0 00362   PL70   CAL PL96                  1023
00213  0020 00 0 00106          TRA PL33
00214 -0500 00 0 00363   PL71   CAL PL97                  123
00215  0020 00 0 00106          TRA PL33
00216  0500 00 0 00364   PL72   CLA PL98                  900
00217  0020 00 0 00126          TRA PL39
00220  0760 00 0 00000   PL73   CLM
00221  0020 00 0 00126          TRA PL39
00222 -0534 00 1 00132   PL80   LXD PL43,1
00223 -3 00000 1 00207          TXL PL56,1,0
00224  1 77777 1 00225          TXI *+1,1,-1
00225 -0634 00 1 00277          SXD PL84,1
00226 -0634 00 1 00310          SXD      PL846,1
00227 -0534 00 1 00223          LXD PL80+1,1
00230  0534 00 4 00154   PL805  LXA      PL45,4
00231  0061 00 0 00231          TCOB     *
00232  0500 00 1 00232   PL81   CLA *,1
00233  0601 00 0 00370          STO TEMP
00234  0621 00 0 00345          STA Y1
00235  0771 00 0 00022          ARS 18
00236  0621 00 0 00344          STA X1
00237  1 00001 1 00240          TXI *+1,1,1
00240  0500 00 1 00240   PL82   CLA *,1
00241  0621 00 0 00346          STA Y2
00242  0771 00 0 00022          ARS 18
00243  0402 00 0 00344          SUB X1
00244  0601 00 0 00347          STO VX
00245  0602 00 0 00350          SLW ABVX
00246  0500 00 0 00345          CLA Y1
00247  0402 00 0 00346          SUB Y2
00250  0601 00 0 00351          STO VY
00251  0760 00 0 00003          SSP
00252  0340 00 0 00350          CAS ABVX
```

```
00253  0020 00 0 00256          TRA PL83+1
00254  0020 00 0 00256          TRA PL83+1
00255  0500 00 0 00350   PL83   CLA ABVX
00256  0340 00 0 00373          CAS PL995
00257  0020 00 0 00313          TRA PL85
00260  0020 00 0 00313          TRA PL85
00261  0500 00 0 00347   PL835  CLA VX
00262  0767 00 0 00034          ALS 28
00263 -0120 00 0 00265          TMI *+2
00264  0361 00 0 00353          ACL XSIGN
00265  0602 00 0 00371          SLW TEMP+1
00266  0500 00 0 00351          CLA VY
00267  0767 00 0 00012          ALS 10
00270 -0120 00 0 00272          TMI *+2
00271  0361 00 0 00354          ACL YSIGN
00272  0361 00 0 00371          ACL TEMP+1
00273  0361 00 0 00352          ACL DRV
00274  0361 00 0 00370          ACL TEMP
00275  0602 00 4 00461          SLW      BUFF+50,4
00276 -2 00001 4 00304          TNX      PL845,4,1
00277 -3 00277 1 00232   PL84   TXL PL81,1,*
00300  1 77716 4 00301          TXI      *+1,4,-50
00301  0754 00 4 00000          PXA      0,4
00302  0737 00 4 00000          PAC      0,4
00303  0020 00 0 00305          TRA      *+2
00304  0774 00 4 00062   PL845  AXT      50,4
00305  0766 00 0 02225          WTB      X
00306 -0634 00 4 00376          SXD      IOC,4
00307 -0540 00 0 00376          RCHX     IOC
00310 -3 00310 1 00230   PL846  TXL      PL805,1,*
00311 -0534 00 4 00073          LXD      PL31,4
00312  0020 00 0 00207          TRA      PL56
00313 -0534 00 2 00223   PL85   LXD PL80+1,2
00314  0771 00 0 00001          ARS 1
00315  0340 00 0 00373          CAS PL995
00316  1 77777 2 00314          TXI PL85+1,2,-1
00317  1 77777 2 00314          TXI PL85+1,2,-1
00320  0500 00 2 00340          CLA ARS,2
00321  0601 00 0 00324          STO PL86
00322  0601 00 0 00333          STO PL88
00323  0500 00 0 00347          CLA VX
00324  0771 00 0 00001   PL86   ARS 1
00325  0601 00 0 00347          STO VX
00326  0400 00 0 00344          ADD X1
00327  0767 00 0 00022          ALS 18
00330  1 77777 1 00331          TXI *+1,1,-1
00331  0622 00 1 00331   PL87   STD *,1
00332  0502 00 0 00351          CLS VY
00333  0771 00 0 00001   PL88   ARS 1
00334  0621 00 0 00351          STA VY
00335  0400 00 0 00345          ADD Y1
00336  0621 00 1 00336   PL89   STA *,1
00337  0020 00 0 00261          TRA PL835
00340  0771 00 0 00001   ARS    ARS 1
00341  0771 00 0 00002          ARS 2
00342  0771 00 0 00003          ARS 3
00343  0771 00 0 00004          ARS 4
00344  0 00000 0 00000   X1     PZE
```

```
00345   0 00000 0 00000    Y1     PZE
00346   0 00000 0 00000    Y2     PZE
00347   0 00000 0 00000    VX     PZE
00350   0 00000 0 00000    ABVX   PZE
00351   0 00000 0 00000    VY     PZE
00352   600000000000       DRV    BCD 1 00000
00353   0 00000 4 00000    XSIGN  PZE 0,4,0
00354   0 00000 2 00000    YSIGN  PZE 0,2,0
00355   0 00000 0 00001    PL90   PZE 1
00356 +232000000000        PL91   OCT 232000000000
00357   0 00000 0 00367    PL92   PZE 247
00360 +000000003776        PL93   OCT      3776
00361 +000000770000        PL95   OCT 770000
00362   0 01777 0 00000    PL96   PZE 0,,1023
00363   0 00173 0 00000    PL97   PZE 0,,123
00364   0 00000 0 01604    PL98   PZE 900
00365 +212702000000        P900   DEC 900.0
00366   0 00000 0 00000    XMC    PZE
00367   0 00000 0 00000    YMC    PZE
00370                      TEMP   BSS 2
00372   0 00001 0 00000    PL99   PZE 0,,1
00373   0 00000 0 00100    PL995  PZE 64
00374   0 00000 0 00000    K1     PZE
00375   0 00000 0 00000    K2     PZE
00376   0 00000 0 00377    IOC    IOCD      BUFF
00377                      BUFF   BSS       50
                 02205     X      TAPENO    B5
                                  END

00461 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

   NO ERROR IN ABOVE ASSEMBLY
```

T. SELGRD SUBROUTINE, Written by L. Egbert

1. Description

a. Purpose: To write information on tape B5 commanding the 40/20 to select various grids.

b. Calling Sequence: CALL SELGRD(I, J)

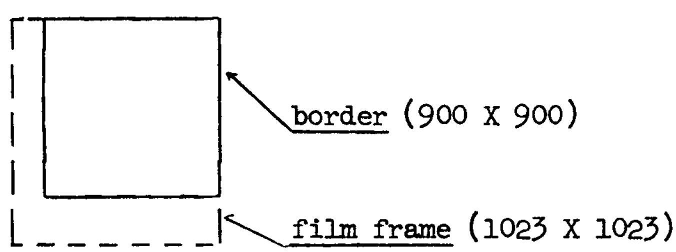
I = Horizontal option

J = Vertical option

Option	Number of Intervals	Option	Number of Intervals
≦ 1	1 (border only)	10	50
2	2	11	1 cycle logarithmic
3	3	12	2 " "
4	4	13	3 " "
5	5	14	4 " "
6	10	15	5 " "
7	15	16	6 " "
8	20	≧17	Border Only
9	25		

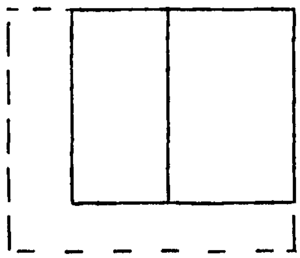
c. Length and Language: $540_8 = 352_{10}$; FAP language

d. Details: Whenever this routine is used a border is drawn as follows:

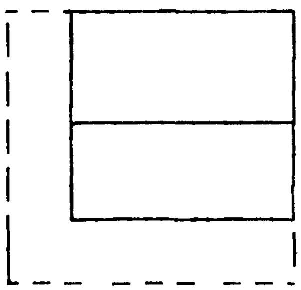


The grids are always drawn inside this border leaving additional room on the left and bottom edges of each frame for labeling purposes. (The dashed lines are not drawn on film.)

For example, with I = 2 and J = 1 the following would result:



I = 1, J = 2



e. Data Code Usage: This subroutine is used by RDH-D and CAN-D in writing grid lines for the 40/20 plotter.

2. Source Deck (FAP) Listing

```
                           *       SELGRD  WRITES ON TAPE B5 IN LOW DENSITY        (J5 LA SGRD)
                   00000           ENTRY   SELGRD

00000 -0634 00 1 00516   SELGRD SXD IR1,1                   SAVE INDEX REG
00001 -0634 00 2 00517          SXD IR2,2
00002  0500 60 4 00001          CLA*     1,4
00003  0601 00 0 00520          STO I
00004  0500 60 4 00002          CLA*     2,4
00005  0601 00 0 00521          STO J
00006  0 77600 0 02205          PZE      Z,,32640            SET LOW DENSITY
00007  0766 00 0 02225          WTBZ                         STOP TYPE INST TO 4020
00010 -0540 00 0 00532          RCHZ     CW1
00011  0766 00 0 02225   BORD   WTBZ
00012 -0540 00 0 00533          RCHZ     CW2                 GENERATE BORDER INST FOR 4020
00013 -0534 00 1 00520          LXD I,1
00014 -3 00020 1 00016          TXL TSTI2,1,16              I IS BETWEEN 1 AND 16
00015  0020 00 0 00064          TRA TSTJ1
00016 -3 00001 1 00064   TSTI2  TXL TSTJ1,1,1
00017  0500 00 1 00504          CLA NS+2,1
00020  0601 00 0 00522          STO N
00021 -0754 00 0 00000          PXD
00022  0560 00 0 00505          LDQ AD900
00023  0221 00 0 00522          DVP N
00024 -0600 00 0 00524          STQ INCR
00025 -3 00012 1 00036          TXL LINY,1,10
00026  0534 00 1 00522          LXA N,1                     LOG SET UP FOR Y
00027  0500 00 1 00464          CLA ADLG1+1,1
00030  0621 00 0 00046          STA RITEY+1
00031 -0734 00 2 00000          PDX ,2
00032  0622 00 0 00523          STD M
00033  0500 00 0 00504          CLA AD123
00034  0601 00 0 00525          STO INCRE
00035  0020 00 0 00045          TRA RITEY
00036 -0600 00 0 00525   LINY   STQ INCRE                   LINEAR SET UP FOR Y
00037  0500 00 0 00506          CLA A123
00040  0621 00 0 00046          STA RITEY+1
00041  0534 00 2 00503          LXA ZERO,2
00042  0534 00 1 00522          LXA N,1
00043  0600 00 0 00523          STZ M
00044  1 77777 1 00045          TXI RITEY,1,-1
00045  0766 00 0 02225   RITEY  WTBZ
00046  0500 00 2 00000          CLA **,2
00047  0400 00 0 00525          ADD INCRE
00050  0767 00 0 00022          ALS 18
00051  0400 00 0 00530          ADD FIXIT
00052  0622 00 0 00511          STD GENY
00053  0500 00 0 00511          CLA      GENY
00054  0601 00 0 00540          STO      OTPUT               PREPARE OUTPUT REGISTER
00055 -0540 00 0 00537          RCHZ     CW3                 WRITE Y AXIS
00056  2 00001 2 00045          TIX RITEY,2,1
00057  0500 00 0 00525          CLA INCRE
00060  0400 00 0 00524          ADD INCR
00061  0601 00 0 00525          STO INCRE
00062 -0534 00 2 00523          LXD M,2
00063  2 00001 1 00045          TIX RITEY,1,1               TRANSFER FOR CYCLES
00064 -0534 00 1 00521   TSTJ1  LXD J,1
00065 -3 00020 1 00067          TXL TSTJ2,1,16              TEST TO SEE IF
```

```
00066  0020 00 0 00132         TRA EXIT               J IS BETWEEN 1 AND 1L
00067 -3 00001 1 00132   TSTJ2 TXL EXIT,1,1
00070  0500 00 0 00505         CLA AD900
00071  0601 00 0 00525         STO INCRE
00072  0500 00 1 00504         CLA NS+2,1
00073  0601 00 0 00522         STO N
00074 -0754 00 0 00000         PXD
00075  0560 00 0 00505         LDQ AD900
00076  0221 00 0 00522         DVP N
00077 -0600 00 0 00524         STQ INCR
00100 -3 00012 1 00107         TXL LINX,1,10
00101  0534 00 1 00522         LXA N,1                LOG SET UP FOR X
00102  0500 00 1 00464         CLA ADLG1+1,1
00103  0621 00 0 00117         STA RITEX+2
00104 -0734 00 2 00000         PDX ,2
00105  0622 00 0 00523         STD M
00106  0020 00 0 00115         TRA RITEX
00107  0500 00 0 00527   LINX  CLA ADINCR
00110  0621 00 0 00117         STA RITEX+2
00111  0534 00 2 00503         LXA ZERO,2
00112  0534 00 1 00522         LXA N,1
00113  0600 00 0 00523         STZ M
00114  1 77777 1 00115         TXI RITEX,1,-1
00115  0766 00 0 02225   RITEX WTBZ
00116  0500 00 0 00525         CLA INCRE
00117  0402 00 2 00000         SUB **,2
00120  0621 00 0 00510         STA GENX
00121  0500 00 0 00510         CLA     GENX
00122  0601 00 0 00540         STO     OTPUT           PREPARE OUTPUT REGISTER
00123 -0540 00 0 00537         RCHZ    CW3             WRITE X AXIS
00124  2 00001 2 00115         TIX RITEX,2,1          CONTINUE FOR CYCLES
00125  0500 00 0 00525         CLA INCRE
00126  0402 00 0 00524         SUB INCR
00127  0601 00 0 00525         STO INCRE
00130 -0534 00 2 00523         LXD M,2
00131  2 00001 1 00115         TIX RITEX,1,1          TRANSFER FOR CYCLES
00132 -0534 00 1 00516   EXIT  LXD IR1,1
00133 -0534 00 2 00517         LXD IR2,2
00134  0020 00 4 00003         TRA 3,4                RETURN TO CALLING SEQ
```

```
      SELGRD  FOR 7090   SELECTS VARIOUS GRIDS FOR THE S-C 4020           5/23/61          PAGE 3
                                                                         (LA SGRD)
                                          1 CYCLE LOGARITHM TABLE
00135 +000000001560                DEC 880,859,859,836,813,813,788,761,761,732,700,700
00136 +000000001533
00137 +000000001533
00140 +000000001504
00141 +000000001455
00142 +000000001455
00143 +000000001424
00144 +000000001371
00145 +000000001371
00146 +000000001334
00147 +000000001274
00150 +000000001274
00151 +000000001232                DEC 666,629,629,629,629,621,613,605,597,588,588,579
00152 +000000001165
00153 +000000001165
00154 +000000001165
00155 +000000001165
00156 +000000001155
00157 +000000001145
00160 +000000001135
00161 +000000001125
00162 +000000001114
00163 +000000001114
00164 +000000001103
00165 +000000001072                DEC 570,561,552,542,542,542,542,532,522,511,501,490
00166 +000000001061
00167 +000000001050
00170 +000000001036
00171 +000000001036
00172 +000000001036
00173 +000000001036
00174 +000000001024
00175 +000000001012
00176 +000000000777
00177 +000000000765
00200 +000000000752
00201 +000000000752                DEC 490,478,467,455,442,429,429,429,429,416,402,388
00202 +000000000736
00203 +000000000723
00204 +000000000707
00205 +000000000672
00206 +000000000655
00207 +000000000655
00210 +000000000655
00211 +000000000655
00212 +000000000640
00213 +000000000622
00214 +000000000604
00215 +000000000566                DEC 374,358,358,342,326,308,290,271,271,271,271,251
00216 +000000000546
00217 +000000000546
00220 +000000000526
00221 +000000000506
00222 +000000000464
00223 +000000000442
00224 +000000000417
00225 +000000000417
```

```
00226 +C0000CC00417
CC227 +000C0CCCC417
CC23C +COCOCCOOC373
C0231 +CCCCCCOOC346          CEC 230,207,183,158,158,131,1C3,71,37,0
00232 +C0000CC0C317
C0233 +C0000CCCC267
00234 +C0C0CCCC0236
0C235 +000C0C00C0236
00236 +0C000CC0C2C3
0C237 +CCCCCCC00147
0024C +C0000CCC0IC7
CC241 +00000CCC0CC45
00242 +CCCCCC0CCCC0
00243 +CCCCCCCCCCCC     LCC1  CEC C
```

```
                                   2 CYCLE LOGARITHM TABLE
00244 +000000000670          DEC 440,429,429,418,406,406,394,380,380,366,350,350
00245 +000000000655
00246 +000000000655
00247 +000000000642
00250 +000000000626
00251 +000000000626
00252 +000000000612
00253 +000000000574
00254 +000000000574
00255 +000000000556
00256 +000000000536
00257 +000000000536
00260 +000000000515          DEC 333,315,315,294,271,271,245,215,215,179,135,135
00261 +000000000473
00262 +000000000473
00263 +000000000446
00264 +000000000417
00265 +000000000417
00266 +000000000365
00267 +000000000327
00270 +000000000327
00271 +000000000263
00272 +000000000207
00273 +000000000207
00274 +000000000207          DEC 135,135,125,115,104,92,79,79,66,51,36,19
00275 +000000000207
00276 +000000000175
00277 +000000000163
00300 +000000000150
00301 +000000000134
00302 +000000000117
00303 +000000000117
00304 +000000000102
00305 +000000000063
00306 +000000000044
00307 +000000000023
00310 +000000000000          DEC 0,0,0
00311 +000000000000
00312 +000000000000
00313 +000000000000     LOG2 DEC 0
```

```
                                          3 CYCLE LOGARITHM TABLE
00314 +000000000445          DEC 293,286,286,279,271,271,263,254,254,244,233,233
00315 +000000000436
00316 +000000000436
00317 +000000000427
00320 +000000000417
00321 +000000000417
00322 +000000000407
00323 +000000000376
00324 +000000000376
00325 +000000000364
00326 +000000000351
00327 +000000000351
00330 +000000000336          DEC 222,210,210,196,181,181,163,143,143,119,90,90
00331 +000000000322
00332 +000000000322
00333 +000000000304
00334 +000000000265
00335 +000000000265
00336 +000000000243
00337 +000000000217
00340 +000000000217
00341 +000000000167
00342 +000000000132
00343 +000000000132
00344 +000000000064          DEC 52,0,0,0
00345 +000000000000
00346 +000000000000
00347 +000000000000
00350 +000000000000     LOG3 DEC 0
```

```
                                       4 CYCLE LOGARITHM TABLE
C0351 +CCCOCCCCC334               DEC 22C,215,215,209,203,203,197,19C,19C,183,175,175
C0352 +CCCOCCCCC327
CC353 +CCCOCCCCC327
C0354 +COCOCCC00321
CC355 +CCCCCCCCC313
C0356 +0000CCC00313
CC357 +CCCOCCCCC3C5
0C360 +0C0C0CC0C276
0C361 +CCCOCCC0C276
CC362 +0000CCC00267
CC363 +CCCCCCCCC257
C0364 +0CCCCCCCC257
C0365 +C0COCCCCC247               DEC 167,157,157,147,135,135,122,1C7,107,90,68,68
0C366 +C0CCCCCCC235
CC367 +COCCCCCCC235
C037C +0C0CCC00C223
C0371 +0CCCCCCCC2C7
C0372 +COCOCCOCC2C7
CC373 +CCCOCC000172
C0374 +0CCOCCCCC153
C0375 +0CCOCCCCC153
C0376 +C0COCCCCC132
CC377 +00COCCC001C4
C04CC +0CCOCCCCC1C4
C04C1 +COCOCCCCCC5C               DEC 4C,C,0,0
C04C2 +CCCCCCCCCCCC
CC4C3 +0CCOCCCCCCC0
C0404 +CCCOCCCCCCC0
C0405 +CCCOCCOOCCCC          LOG4 DEC C
```

```
                                     5 CYCLE LOGARITHM TABLE
00406 +000000000254            DEC 172,163,152,140,126,126,118,108,108,98,86,86
00407 +000000000243
00410 +000000000230
00411 +000000000214
00412 +000000000176
00413 +000000000176
00414 +000000000166
00415 +000000000154
00416 +000000000154
00417 +000000000142
00420 +000000000126
00421 +000000000126
00422 +000000000110            DEC 72,54,54,32,0,0,0
00423 +000000000066
00424 +000000000066
00425 +000000000040
00426 +000000000000
00427 +000000000000
00430 +000000000000
00431 +000000000000       LOC5  DEC 0
```

```
                                   6 CYCLE LOGARITHM TABLE
00432 +C0000CCC0C217          DEC 143,135,127,117,105,105,98,90,90,82,72,72
00433 +0CCCCCCCC0207
00434 +00CCCCCCC177
00435 +0000CCC000165
00436 +C0C0CCCCC151
00437 +00C0CCC000151
0044C +C0CCCC000C142
00441 +0CC0CCCCC132
00442 +0CC0CCC00132
00443 +C0C0CC000C122
00444 +C0000CCCC110
00445 +CCC0CCC00C11C
00446 +0CC00C000C74          DEC 60,45,45,26,0,C,C
00447 +C000CCC0C055
0045C +CCC0CCCCC055
00451 +00CCCCCCC032
00452 +C000CC0000CCC
00453 +C00CCCC00CCC
00454 +C0CCCC000CCC
00455 +C000CCC00CCCC     LOG6  DEC C
```

```
                                   CODE CONSTANTS TABLE
00456  0 00024 0 00456             LOG6+1,,20
00457  0 00024 0 00432             LOG5+1,,20
00460  0 00035 0 00406             LOG4+1,,29
00461  0 00035 0 00351             LOG3+1,,29
00462  0 00050 0 00314             LOG2+1,,40
00463  0 00107 0 00244   ADLC1     LOG1+1,,71
00464 +000000000006            DEC 6,5,4,3,2,1,50,25,20,15,10,5,4,3
00465 +000000000005
00466 +000000000004
00467 +000000000003
00470 +000000000002
00471 +000000000001
00472 +000000000062
00473 +000000000031
00474 +000000000024
00475 +000000000017
00476 +000000000012
00477 +000000000005
00500 +000000000004
00501 +000000000003
00502 +000000000002      NS     DEC 2
00503 +000000000000      ZERO   DEC 0
00504  0 00000 0 00173   AD123      123
00505  0 00000 0 01604   AD900      900
00506  0 00000 0 00504   A123       AD123
00507  0 00000 0 00505   A900       AD900
00510 +300173000000      GENX   OCT 300173000000
00511 +320000001604      GENY   OCT 320000001604
00512 +300173001604      LOWX   OCT 300173001604
00513 +300173000000      HIGHX  OCT 300173000000
00514 +320173001604      LEFTY  OCT 320173001604
00515 +321777001604      RITY   OCT 321777001604
00516  0 00000 0 00000   IR1
00517  0 00000 0 00000   IR2
00520  0 00000 0 00000   I
00521  0 00000 0 00000   J
00522  0 00000 0 00000   N
00523  0 00000 0 00000   M
00524  0 00000 0 00000   INCR
00525  0 00000 0 00000   INCRE
00526 +000000000227      TAPE   OCT 000000000227
00527  0 00000 0 00524   ADINCR     INCR
00530 +020000000000      FIXIT  OCT 020000000000
              02205      Z      TAPENO  B5
00531 +120000000000      STPTP  OCT     120000000000     STOP TYPE
00532  0 00001 0 00531   CW1    IOCD    STPTP,,1         CONTROL WORD FOR STOP TYPE
00533  2 00001 0 00512   CW2    IORP    LOWX,,1          CONTROL WORDS FOR BORDER
00534  2 00001 0 00513          IORP    HIGHX,,1
00535  2 00001 0 00514          IORP    LEFTY,,1
00536  0 00001 0 00515          IOCD    RITY,,1
00537  0 00001 0 00540   CW3    IOCD    OTPUT,,1         CONTROL WORD FOR GRIDS
00540  0 00000 0 00000   OTPUT                           GRID OUTPUT REGISTER
                                END
```

00541 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM.

U. WFILM SUBROUTINE, Written by K. Crandall

1. Description

a. Purpose: To write BCD information either horizontally or vertically on film.

b. Calling Sequence: CALL WFILM (K, N, BCD, IBCD, R, C)

K = 0 - write horizontally
    1 - write vertically

N = Number of BCD characters to write

BCD = Location of first of N consecutively stored BCD characters (6 characters per word).

IBCD = 0 - BCD characters are stored backwards
       1 - BCD characters are stored forwards

R = Row number at which to start writing.

C = Character position at which to start writing.

K and N are fixed point integers

R and C are floating point numbers

c. Details: A frame is considered to be divided into 64 rows and 128 character positions (columns). However, both R and C may be multiples of 1/4 ($3/4 \leq R \leq 64-1/4$; $3/4 \leq C \leq 128-1/4$) to allow more exact positioning, superscripts, or subscripts.

When writing vertically, the first character is positioned at row R and column C and succeeding characters are each spaced down one row. No more characters will be printed after row 64 is reached.

d. Length and Language: $275_8 = 189_{10}$; FAP language.

e. Data Code Usage: This subroutine is used by RDH-D and CAN-D in writing grid lines for the 40/20 plotter.

2. Source Deck (FAP) Listing

```
                    00000          ENTRY    WFILM

00000  0634 00 4 00107   WFILM SXA      WF26,4
00001  0634 00 2 00110         SXA      WF26+1,2
00002  0634 00 1 00111         SXA      WF26+2,1
00003  0500 60 4 00005         CLA*     5,4
00004 -0300 00 0 00145         UFA      WF83
00005 -0773 00 0 00010         RQL      8
00006  0760 00 0 00010         RND
00007 -0320 00 0 00146         ANA      WF84
00010  0767 00 0 00002         ALS      2
00011  0402 00 0 00147         SUB      WF86
00012  0120 00 0 00014         TPL      WF02
00013  0400 00 0 00150         ADD      WF87
00014  0601 00 0 00173   WF02  STO      WF94
00015  0500 60 4 00006         CLA*     6,4
00016 -0300 00 0 00145         UFA      WF83
00017 -0773 00 0 00010         RQL      8
00020  0760 00 0 00010         RND
00021 -0320 00 0 00146         ANA      WF84
00022  0767 00 0 00001         ALS      1
00023  0402 00 0 00151         SUB      WF88
00024  0120 00 0 00026         TPL      WF04
00025  0400 00 0 00147         ADD      WF86
00026  0767 00 0 00022   WF04  ALS      18
00027 -0501 00 0 00173         ORA      WF94
00030 -0320 00 0 00152         ANA      WF89
00031  0601 00 0 00173         STO      WF94
00032  0500 60 4 00004         CLA*     4,4
00033  0774 00 1 00001         AXT      1,1
00034  0100 00 0 00036         TZE      WF06
00035 -0774 00 1 00001         AXC      1,1
00036 -0634 00 1 00072   WF06  SXD      WF13,1
00037 -0634 00 1 00130         SXD      WF20,1
00040  0500 00 4 00003         CLA      3,4
00041  0737 00 1 00000         PAC      ,1
00042  0500 60 4 00002         CLA*     2,4
00043 -0734 00 2 00000         PDX      ,2
00044 -3 00000 2 00107         TXL      WF26,2,0
00045  0061 00 0 00045         TCOB     *
00046 +077600002205            OCT      77600002205
00047  0500 60 4 00001         CLA*     1,4
00050 -0100 00 0 00113         TNZ      WF17
00051 -3 00563 2 00053         TXL      *+2,2,371
00052 -0534 00 2 00051         LXD      *-1,2
00053 -0500 00 1 00000         CAL      0,1
00054  0771 00 0 00022         ARS      18
00055 -0320 00 0 00153         ANA      WF90
00056 -0501 00 0 00173         ORA      WF94
00057  0601 00 0 00175         STO      BUFF
00060 -0320 00 0 00161         ANA      WF93+2
00061 -0501 00 0 00154         ORA      BLANK
00062  0400 00 0 00155         ADD      D8
00063  0601 00 0 00176         STO      BUFF+1
00064 -0774 00 4 00002         AXC      2,4
00065 -0500 00 1 00000         CAL      0,1
00066 -0320 00 0 00164         ANA      WF93+5
```

```
00067 -0501 00 0 00156          ORA    TCP
00070 -3 00005 2 00076   WF12   TXL    WF14,2,5
00071  0602 00 4 00175          SLW    BUFF,4
00072  1 00000 1 00073   WF13   TXI    *+1,1,0
00073 -0500 00 1 00000          CAL    0,1
00074  1 77777 4 00075          TXI    *+1,4,-1
00075  1 77772 2 00070          TXI    WF12,2,-6
00076 -0320 00 2 00164   WF14   ANA    WF93+5,2
00077 -0501 00 2 00172          ORA    STT+5,2
00100  0602 00 4 00175          SLW    BUFF,4
00101  1 77777 4 00102          TXI    *+1,4,-1
00102  0754 00 4 00000          PXA    0,4
00103  0737 00 4 00000          PAC    0,4
00104 -0634 00 4 00174          SXD    IOC,4
00105  0766 00 0 02225   WB15   WTBX
00106 -0540 00 0 00174          RCHX   IOC
00107  0774 00 4 00000   WF26   AXT    0,4
00110  0774 00 2 00000          AXT    0,2
00111  0774 00 1 00000          AXT    0,1
00112  0020 00 4 00007          TRA    7,4
00113 -3 00100 2 00115   WF17   TXL    *+2,2,64
00114 -0534 00 2 00113          LXD    *-1,2
00115 -0634 00 2 00174          SXD    IOC,2
00116  1 00175 2 00117          TXI    *+1,2,BUFF
00117  0634 00 2 00125          SXA    WF19+1,2
00120 -0534 00 4 00174          LXD    IOC,4
00121  0774 00 2 00006   WF18   AXT    6,2
00122  0560 00 1 00000          LDQ    0,1
00123 -0773 00 0 00014          RQL    12
00124 -0773 00 0 00006   WF19   RQL    6
00125 -0600 00 4 00125          STQ    *,4
00126 -2 00001 4 00131          TNX    *+3,4,1
00127  2 00001 2 00124          TIX    WF19,2,1
00130  1 00000 1 00121   WF20   TXI    WF18,1,0
00131 -0534 00 4 00174          LXD    IOC,4
00132  0774 00 2 00000          AXT    0,2
00133 -0500 00 2 00175   WF21   CAL    BUFF,2
00134 -0320 00 0 00153          ANA    WF90
00135  0361 00 0 00173          ACL    WF94
00136  0601 00 2 00175          STO    BUFF,2
00137 -0320 00 0 00161          ANA    WF93+2
00140  0400 00 0 00150          ADD    WF87
00141  0601 00 0 00173          STO    WF94
00142  1 77777 2 00143          TXI    *+1,2,-1
00143  2 00001 4 00133          TIX    WF21,4,1
00144  0020 00 0 00105          TRA    WB15
00145 +231000000000      WF83   OCT    231000000000
00146 +000000077777      WF84   OCT    77777
00147  0 00000 0 00010   WF86   PZE    8
00150  0 00000 0 00020   WF87   PZE    16
00151  0 00000 0 00004   WF88   PZE    4
00152 +001777001777      WF89   OCT    1777001777
00153 +000000770000      WF90   OCT    770000
00154 +000000600000      BLANK  OCT    600000
00155  0 00010 0 00000   D8     PZE    0,,8
00156 +220000000000      TCP    OCT    220000000000
00157 -377777777700      WF93   OCT    -377777777700,-377777770077,-377777007777
00160 -377777770077
```

```
00161 -377777007777
00162 -377700777777         OCT     -377700777777,-370077777777,7777777777                WFLM117
00163 -370077777777
00164 +007777777777
00165 +000000000012   STT   OCT     12,1200,120000,12000000,1200000000,120000000000       WFLM118
00166 +000000001200
00167 +000000120000
00170 +000012000000
00171 +001200000000
00172 +120000000000
00173  0 00000 0 00000   WF94  PZE                                                          WFLM119
00174  0 00000 0 00175   IOC   IOCD    BUFF                                                 WFLM120
00175                    BUFF  BSS     64                                                   WFLM121
                 02205   X     TAPENO  B5                                                   WFLM122
                               END                                                          WFLM123

00275 IS THE FIRST LOCATION NOT USED BY THIS PROGRAM

  NO ERROR IN ABOVE ASSEMBLY
```

V. PØLEV SUBROUTINE, Written by T. Jordan

1. Description

a. Purpose: To evaluate a polynomial p(Z) with real coefficients for arbitrary argument X and produce coefficients of Q(Z) where P(Z) = Q(Z)·(Z - X) + P(X).

b. Calling Sequence: Y = PØLEV(X, N, A, B) where

X = The argument

N = The number of coefficients of P(X) or the degree + 1.

A = The vector of dimension N containing components defined by

$$P(Z) = A_1 + A_2 Z + \;.\;.\;.\; A_N Z^{N-1}$$

B = A vector of dimension N - 1 containing components defined by

$$Q(Z) = B_1 + B_2 Z + \;.\;.\;.\; B_{N-1} Z^{N-2}$$

c. Length and Language: $34_8 = 28_{10}$; FAP language.

d. Data Code Usage: PØLEV is used in the RDH-D code for transforming each raw data point into a reduced data point after a final set of least square fit coefficients has been determined.

2. Source Deck (FAP) Listing

```
                    00000        ENTRY   POLEV                                        POLE006

00000 -0634 00 2 00025   POLEV SXD     T,2              SAVE INDEX 2                POLE007
00001  0500 60 4 00002         CLA*    2,4              PICKUP N                    POLE008
00002  0100 00 0 00023         TZE     NZERO            TEST FOR N=0                POLE009
00003 -0734 00 2 00000         PCX     0,2              N TO INDEX 2                POLE010
00004  0500 00 4 00003         CLA     3,4              A TO ACCUMULATOR            POLE011
00005  0400 00 0 00026         ADD     =1               A+1                         POLE012
00006  0621 00 0 00017         STA     A                SET ADDRESS                 POLE013
00007  0500 00 4 00004         CLA     4,4              B TO ACCUMULATOR            POLE014
00010  0400 00 0 00026         ADD     =1               B+1                         POLE015
00011  0621 00 0 00015         STA     LOOP+1           SET ADDRESS                 POLE016
00012 -0754 00 0 00000         ZAC                      0 TO ACCUMULATOR            POLE017
00013  0020 00 0 00017         TRA     A                                            POLE018
00014  0131 00 0 00000   LOOP  XCA                      PARTIAL SUM TO MQ           POLE019
00015 -0600 00 2 00015         STQ     *,2              SAVE IN B(I)                POLE0191
00016  0260 60 4 00001         FMP*    1,4              MULTIPLY BY X               POLE020
00017  0300 00 2 00017   A     FAD     *,2              ADD A(I)                    POLE022
00020  2 00001 2 00014         TIX     LOOP,2,1                                     POLE023
00021 -0534 00 2 00025   EXIT  LXD     T,2              RESTORE INDEX 2             POLE024
00022  0020 00 4 00005         TRA     5,4                                          POLE025
00023  0500 00 0 00027   NZERO CLA     =0377777777777                               POLE026
00024  0020 00 0 00021         TRA     EXIT                                         POLE027
00025  0 00000 0 00000   T     PZE     0                                            POLE028
                               END                                                  POLE029

  LITERALS
00026  000000000001
00027  377777777777
```

W. MAX3 SUBROUTINE, Written by B. Bacon

1. Description

a. Purpose: To determine the algebraic maximum from an array of floating or fixed point variables.

b. Calling Sequence: CALL MAX3 (X, M, N, I, Y).

X = Origin of the array of numbers.

M = Memory spacing between the values of the array. A negative M implies FORTRAN-type storage.

N = Number of words in the array.

I = The index location of the maximum value. In FORTRAN, the maximum value is located at X(I).

Y = The maximum value.

c. Length and Language: $66_8 = 54_{10}$; FAP language.

d. Data Code Usage: This subroutine is used by RDH-D and CAN-D. This usage is completely automatic.

2. Source Deck (FAP) Listing

```
                         00000          ENTRY   MAX3

00000   0500 00 4 00001   MAX3   CLA    1,4
00001   0621 00 0 00034          STA    X1
00002   0621 00 0 00037          STA    X2
00003   0500 00 4 00002          CLA    2,4
00004   0621 00 0 00005          STA    *+1
00005   0500 00 0 00000          CLA    **
00006   0120 00 0 00013          TPL    *+5
00007   0622 00 0 00046          STD    M
00010   0500 00 0 00014          CLA    LONE
00011   0601 00 0 00054          STO    ADD
00012   1 00000 0 00020   X0     TXI    *+6,0,0
00013   0760 00 0 00006          COM
00014   0400 00 0 00062   LONE   ADD    ONE
00015   0622 00 0 00046          STD    M
00016   0500 00 0 00063          CLA    XB
00017   0621 00 0 00054          STA    ADD
00020   0500 00 4 00003          CLA    3,4
00021   0621 00 0 00022          STA    *+1
00022   0500 00 0 00000          CLA    **
00023   0622 00 0 00050          STD    N
00024   0500 00 4 00004          CLA    4,4
00025   0621 00 0 00055          STA    I1
00026   0500 00 4 00005          CLA    5,4
00027   0621 00 0 00052          STA    XMAX1
00030  -0634 00 2 00063          SXD    XB,2
00031  -0634 00 4 00064          SXD    XC,4
00032  -0534 00 2 00047          LXD    M+1,2
00033  -0534 00 4 00012          LXD    X0,4
00034   0500 00 0 00000   X1     CLA    **
00035   0601 00 0 00065          STO    XMAX
00036   0600 00 0 00066          STZ    I
00037   0500 00 4 00000   X2     CLA    **,4
00040   0340 00 0 00065          CAS    XMAX
00041   0020 00 0 00044          TRA    *+3
00042   0020 00 0 00046          TRA    M
00043   0020 00 0 00046          TRA    M
00044   0601 00 0 00065          STO    XMAX
00045  -0634 00 4 00066          SXD    I,4
00046   1 00000 4 00047   M      TXI    *+1,4,**
00047   1 00001 2 00050          TXI    *+1,2,1
00050  -3 00000 2 00037   N      TXL    X2,2,**
00051   0500 00 0 00065          CLA    XMAX
00052   0601 00 0 00000   XMAX1  STO    **
00053   0500 00 0 00066          CLA    I
00054   0400 00 0 00000   ADD    ADD    **
00055   0601 00 0 00000   I1     STO    **
00056  -0534 00 2 00063          LXD    XB,2
00057  -0534 00 4 00064          LXD    XC,4
00060   0020 00 4 00006          TRA    6,4
00061   0 00000 0 00000   ZERO   PZE
00062   0 00001 0 00000   ONE    PZE    ,,1
00063   0 00000 0 00061   XB     PZE    ZERO
00064   0 00000 0 00000   XC     PZE
00065   0 00000 0 00000   XMAX   PZE
00066   0 00000 0 00000   I      PZE
                                 END
```

X. MIN3 Subroutine, Written by B. Bacon

1. Description

a. Purpose: To determine the algebraic minimum from an array of floating or fixed point variables.

b. Calling Sequence: CALL MIN3 (X, M, N, I, Y)

X = Origin of the array of numbers

M = Memory spacing between the values of the array. A negative M implies FORTRAN-type storage.

N = Number of words in the array.

I = The index location of the minimum value. In FORTRAN, the minimum value is located at X(I).

Y = The minimum value.

c. Length and Language: $65_8 = 53_{10}$; FAP language

d. Data Code Usage: This subroutine is used by RDH-D and CAN-D. This usage is completely automatic.

2. Source Deck (FAP) Listing

```
                        00000         ENTRY   MIN3

00000   0500 00 4 00001   MIN3    CLA     I,4
00001   0621 00 0 00034           STA     X1
00002   0621 00 0 00037           STA     X2
00003   0500 00 4 00002           CLA     2,4
00004   0621 00 0 00005           STA     *+1
00005   0500 00 0 00000           CLA     **
00006   0120 00 0 00013           TPL     *+5
00007   0622 00 0 00045           STD     M
00010   0500 00 0 00014           CLA     LONE
00011   0601 00 0 00053           STO     ADD
00012   1 00000 0 00020   X0      TXI     *+6,0,0
00013   0760 00 0 00006           COM
00014   0400 00 0 00061   LONE    ADD     ONE
00015   0622 00 0 00045           STD     M
00016   0500 00 0 00062           CLA     XB
00017   0621 00 0 00053           STA     ADD
00020   0500 00 4 00003           CLA     3,4
00021   0621 00 0 00022           STA     *+1
00022   0500 00 0 00000           CLA     **
00023   0622 00 0 00047           STD     N
00024   0500 00 4 00004           CLA     4,4
00025   0621 00 0 00054           STA     I1
00026   0500 00 4 00005           CLA     5,4
00027   0621 00 0 00051           STA     XMIN1
00030  -0634 00 2 00062           SXD     XB,2
00031  -0634 00 4 00063           SXD     XC,4
00032  -0534 00 2 00046           LXD     M+1,2
00033  -0534 00 4 00012           LXD     X0,4
00034   0500 00 0 00000   X1      CLA     **
00035   0601 00 0 00064           STO     XMIN
00036   0600 00 0 00065           STZ     I
00037   0500 00 4 00000   X2      CLA     **,4
00040   0340 00 0 00064           CAS     XMIN
00041   0020 00 0 00045           TRA     M
00042   0020 00 0 00045           TRA     M
00043   0601 00 0 00064           STO     XMIN
00044  -0634 00 4 00065           SXD     I,4
00045   1 00000 4 00046   M       TXI     *+1,4,**
00046   1 00001 2 00047           TXI     *+1,2,1
00047  -3 00000 2 00037   N       TXL     X2,2,**
00050   0500 00 0 00064           CLA     XMIN
00051   0601 00 0 00000   XMIN1   STO     **
00052   0500 00 0 00065           CLA     I
00053   0400 00 0 00000   ADD     ADD     **
00054   0601 00 0 00000   I1      STO     **
00055  -0534 00 2 00062           LXD     XB,2
00056  -0534 00 4 00063           LXD     XC,4
00057   0020 00 4 00006           TRA     6,4
00060   0 00000 0 00000   ZERO    PZE
00061   0 00001 0 00000   ONE     PZE     ,,1
00062   0 00000 0 00060   XB      PZE     ZERO
00063   0 00000 0 00000   XC      PZE
00064   0 00000 0 00000   XMIN    PZE
00065   0 00000 0 00000   I       PZE

LA MIN3   SUBROUTINE TO FIND MINIMUM OF AN ARRAY

                  END
```