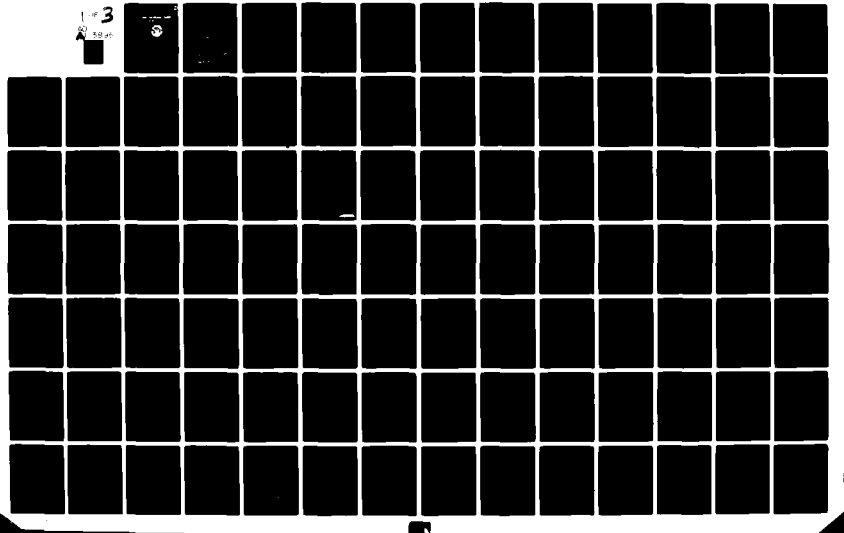
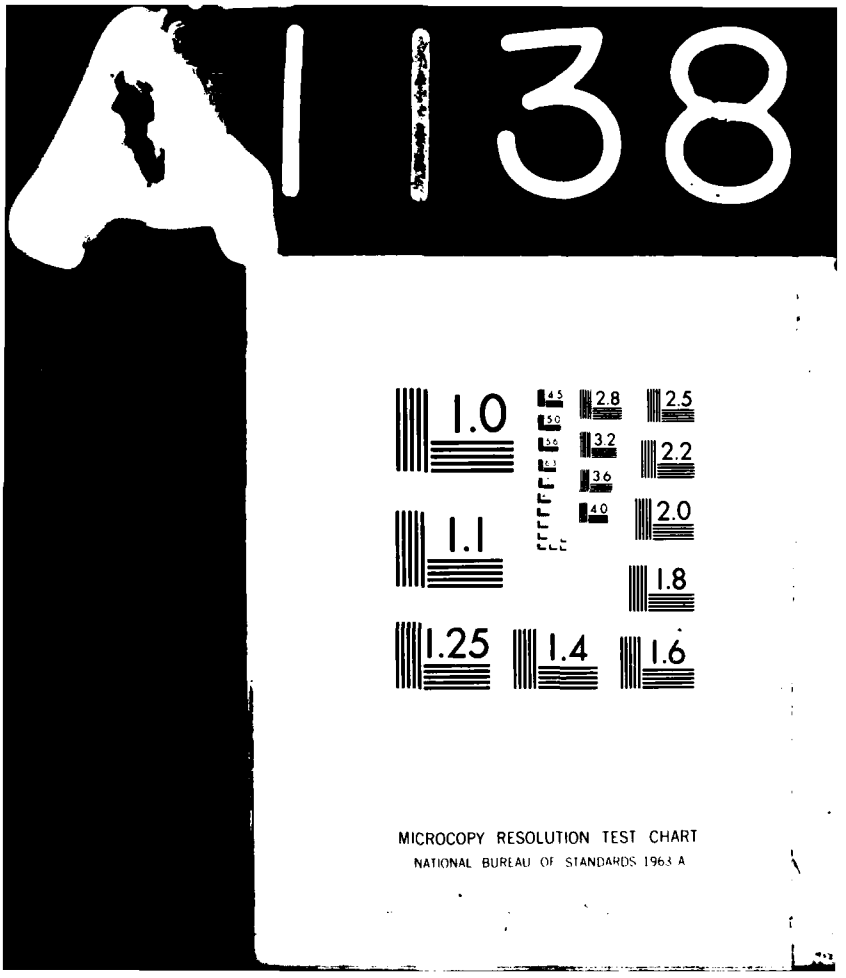
AD-A113 895 BDM CORP MONTEREY CA F/G 5/8
TRANSONIC COMPRESSOR: PROGRAM SYSTEM TXCO FOR DATA ACQUISITION --ETC(U)
OCT 80 H ZEBNER N00014-78-C-0204
UNCLASSIFIED NPS-67-80-02CR NL




1138
1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

NPS67-80-02CR

NAVAL POSTGRADUATE SCHOOL
Monterey, California

CONTRACTOR REPORT

TRANSONIC COMPRESSOR: PROGRAM SYSTEM TRCO FOR DATA ACQUISITION AND ON-LINE REDUCTION

H. ZEBNER
BDM CORPORATION
P.O. Box 2019
Monterey, CA 93940

October 1980

Interim Report for Period
October 1979 - September 1980

Approved for public release; distribution unlimited

Prepared for:
Naval Postgraduate School
Monterey, California 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California

Rear Admiral J. J. Ekelund
Superintendent

D. A. Schrady
Acting Provost

The work reported herein was carried out for the Naval Postgraduate School by the BDM Corporation under Work Order 426, under Contract Number N00014-78-C-0204. The work contributed to the program entitled Transonic Compressor Investigations funded in part by Work Request N00019-80-WR-00199 and under the cognizance of Dr. H. J. Mueller, and to the program entitled Axial Compressor Flow Fields funded in part by Work Request N00014-79-WR-90048 and under the cognizance of Mr. J. R. Patton, Jr.

The author is a candidate for the degree D. Ing under Prof. H. Gallus at Rhein.-Westf. Techn. Hochschule, 51 Aachen, W. Germany.

This report was prepared by:

BDM Corporation
P.O. Box 2019
Monterey, CA 93940

Publication of the report does not constitute approval of the sponsor for the findings or conclusions. It is published for information and for the exchange and stimulation of ideas.

Reviewed by:

R. P. SHREEVE, Director
Turbopropulsion Laboratory

M. F. PLATZER, Chairman
Department of Aeronautics

Released by:

W. M. TOLLES
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS67-80-02CR	2. GOVT ACCESSION NO. AD A113 895	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) TRANSONIC COMPRESSOR: PROGRAM SYSTEM TXCO FOR DATA ACQUISITION AND ON-LINE REDUCTION		5. TYPE OF REPORT & PERIOD COVERED Contractor Report October 1979 - September 1980 6. PERFORMING ORG. REPORT NUMBER XXXX
7. AUTHOR(s) H. Zebner		8. CONTRACT OR GRANT NUMBER(s) N00014-78-C-0204
9. PERFORMING ORGANIZATION NAME AND ADDRESS BDM Corporation P.O. Box 2019 Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Order 426
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE October 1980 13. NUMBER OF PAGES 279
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Interim Report: Principal Investigator, Dr. R. P. Shreeve, Director Turbopropulsion Laboratory, Naval Postgraduate School, Monterey, California 93940

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Data Acquisition Program

Transonic Compressor

Compressor Test

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A system of data acquisition and reduction programs, TXCO is described. The programs were written for the transonic compressor test facility at the NPS Turbopropulsion Laboratory which is served by an HP1000 series computer operating under RTE-IVB. However, the structure of the program system (strict separation of acquisition and reduction, store raw data as acquired, routines to verify the data system, etc.) is of more general interest, and allows the system to be applied to any test rig. The introduction of a program control →

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. cont'd Array accelerates execution and provides means for communication between programs, which otherwise execute individually.

ABSTRACT

A system of data acquisition and reduction programs, TXCO is described. The programs were written for the transonic compressor test facility at the NPS Turbopropulsion Laboratory which is served by an HP1000 series computer operating under RTE-IVB. However, the structure of the program system (strict separation of acquisition and reduction, store raw data as acquired, routines to verify the data system, etc.) is of more general interest, and allows the system to be applied to any test rig. The introduction of a "program control array" accelerates execution and provides means for communication between programs, which otherwise execute individually.

ACKNOWLEDGMENT

This study was made possible by Professor Ray Shreeve, Director of the Turbopropulsion Laboratory (TPL) of the Department of Aeronautics, and by Professor Dr. Ing Heinz Gallus, Institute for Aero Engines and Turbomachines of the Technical University Aachen, West Germany; to both I express my thanks. Jim Hammer, the TPL Laboratory Manager, contributed to this program system by letting me share his brilliant knowledge and vast experience in the areas of instrumentation and turbomachines.

TABLE OF CONTENTS

1. INTRODUCTION

This report describes a system of data acquisition and reduction programs, designed to acquire data from the Turbopropulsion Laboratory's transonic compressor test facility. The computer hardware consists of an HP21MX mini computer with various peripheral devices. Both steady-state and high speed data are required to be taken as scheduled during a compressor test. The entire hardware configuration is shown in Fig. 1.

At the outset, the system of computer programs was required to do the following:

(i) Control via the "Interface Bus", measurement devices such as Scanivalves (S/V's), Scanivalve controllers, scanners, digital voltmeters (DVM), digital counters, analog to digital converters, and the acquisition timing device called PACER.

(ii) Perform data acquisition as efficiently as possible, store data in disc files, and document the test conditions.

(iii) Provide a means to check the data system (e.g., SUBROUTINE CHECK, Section 6.2).

(iv) Provide a means to verify the raw data (e.g., SUBROUTINE PICTR, called from SUBROUTINE PACER, Section 4.5. PICTR uses the auxiliary terminal to display the acquired wave form).

(v) Provide a means for the operator to communicate interactively. Since the operator at the system console is usually the investigator or research engineer and not a computer specialist, the program flow and the programmed interactive messages were required to be clear, logical and easy to understand.

The demand to speed up the data acquisition conflicted with the requirement of keeping the dialogue between program and operator clear. Interactive programs necessarily have extensive input-output operations which slow down the execution of the program. A reasonable compromise between these two choices was the introduction of a "program control array", CNTRL, whose elements - once pre-assigned - relieved the operator from entering routine decisions (e.g., telling the subroutines FREER and PACER how many Kulite signals are to be recorded and where to locate them; see Appendix A3: CNTRL(23Ø) through CNTRL(246)). Additionally the control array provides accounting data (e.g., the sequential number for raw data files).

In the present report complete documentation is given of the program system "TXCO". The system consists of a "father" program, TXCOØ, which, in operation, calls on a series of "son" programs TXCO1, TXCO2 or TXCO3.

The father program, TXCOØ, offers the investigator a menu of program branches to be scheduled according to a single digital entry as follows:

1.	Survey using the type 'A' and the type 'B' Kulite semiconductor pressure probes (Ref 1 and 2).	ABSRV
2.	On-line calibration type 'A' and type 'B' probe.	CALIB
3.	Acquisition of high speed data through the fast A/D converter, which is operated in free run mode.	FREER
4.	Acquisition of high speed data through the fast A/D converter, which now is controlled by a timing device, the PACER (Ref 3).	PACER
5.	Radial flow survey using a temperature-pneumatic four hole COMBINATION PROBE.	COMB
6.	Acquisition of all steady state data.	STDY
7.	Check the instrumentation.	CHECK
8.	Change the program control array.	CHNGE
9.	Reduce high speed data from the 'A' - 'B' probe system. REDAB uses the data gathered by ABSRV.	REDAB
10.	Reduce flow data from the combination probe. REDCO uses the data gathered by COMB.	REDCO
11.	Reduce steady state data and add this operating point to the compressor performance map. REDST uses the data gathered by STDY.	REDST

The investigator selects the desired program module by entering the appropriate number between 1 and 11. Entering 12 halts the program. Subroutines ABSRV, CALIB, FREER and PACER - they handle the high speed data - are contained in PROGRAM TXC01 (Section 4). Subroutines COMB and STDY - they handle the steady state data - are contained in PROGRAM TXC02 (Section 5). Subroutines CHECK and CHNGE - they are used by the operator to control the program flow and verify the data system - are contained in PROGRAM TXC03 (Section 6). After the select code is entered, and verified either by entering an additional parameter or tapping the RETURN key, the "father" program suspends its operation while the desired "son" program (TXC01, TXC02 or TXC03) executes. The entire TXCO-system works interactively with the operator and displays as many informative messages as possible.

The program descriptions in the following sections explain, in user-manual form, how to handle each subroutine. The descriptions often resemble each other, which in the interests of utility was deliberately not avoided. A compressor failure prevented the author from using the programs for compressor test runs. The report is therefore presented with only a very short section of conclusions and recommendations. The program system is not considered to have been perfected, since little experience has been gained with its operation other than in "dry" runs.

2. GUIDE TO THE PROGRAM DESCRIPTIONS

Detailed descriptions of the programs are given in the following sections. First, in Section 3, a flow chart and listing are given for the father program TXCOØ. Then, the descriptions given in Sections 4 through 6 (of TXCO1 through TXCO3) are structured as follows:

PROGRAM XXXX (or SUBROUTINE XXXX): PURPOSE:
A brief description of the purpose of this particular program module is given, and its capabilities and restrictions are noted.

ARGUMENTS: If the program module is a subroutine, which is called with parameters, the parameter list is explained.

EXTERNALS: The externals of each program module are listed. This information is necessary when loading the relocatable binary version (indicated by the % sign as first character of the disc file name under the RTE-IV operating system).

COMMON BLOCKS: The members of the COMMON blocks and their length in 32-bit words are listed and explained.

MNEMONIC ABBREVIATIONS: The mnemonic acronyms which each program module uses are listed and explained.

ERROR MESSAGES: If a salvageable error occurs during the execution of a program module, an error message with suggestions for how to resolve the problem are described.

PROCEDURE: This subsection, which should always be used together with the flow chart, describes how to go through the program module. Hints for how to utilize all program features are given.

DATA FILE: The date file name is explained for all program modules, which save data. The first two characters are typical for the type of data which the file contains; for example,

Data File Name	Type of Data	Created By
T1rrss	'A'-'B' probe survey	ABSRV
T2rrss	free run sample	FREER
T3rrss	paced run sample	PACER
T4rrss	all raw steady state data	STDY
T5rrss	combination probe survey	COMB

rr — # of test run

ss — sequential # of data file type

The following modules are synchronized through the data file:

Data Reduction Program		File Name		Data Acquisition Program
REDAB	↔	T1rrss	↔	ABSRV
REDST	↔	T4rrss	↔	STDY
REDCO	↔	T5rrss	↔	COMB

VARIABLES: All variables, their type (REAL or INTEGER) and length (only if the variable is used as an array), together with a brief description, are listed.

The flow chart and a FORTRAN-IV compiler listing of the program module complete each description.

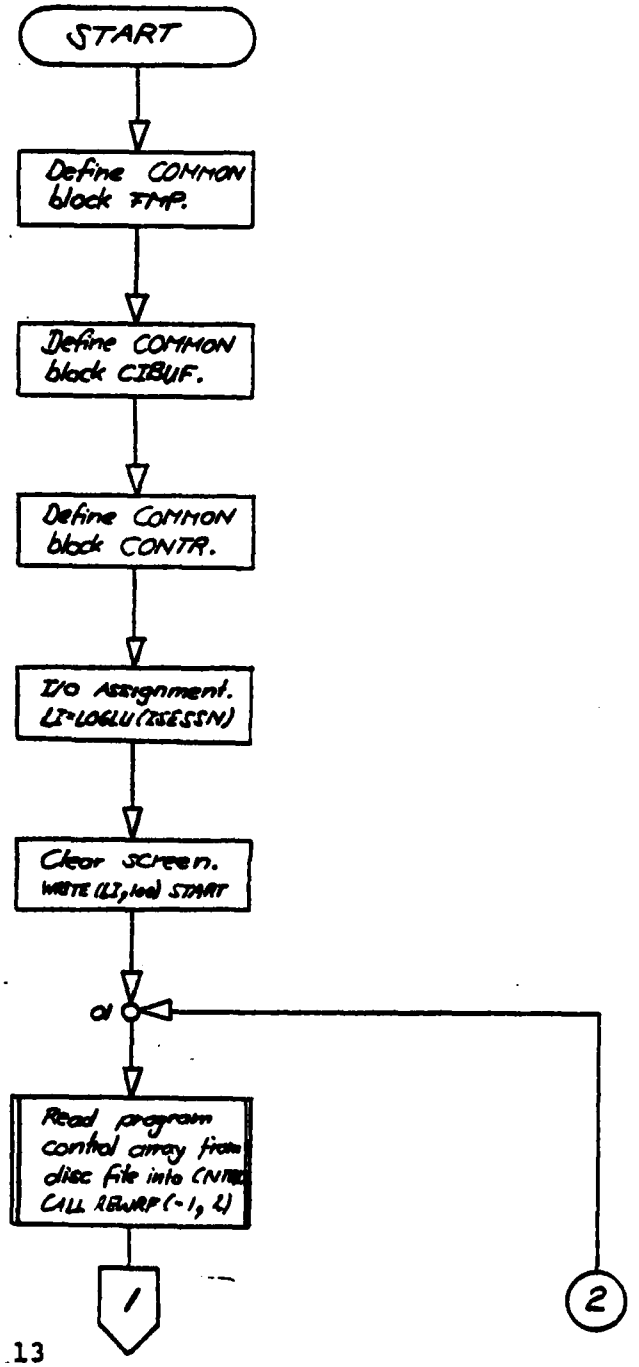
The source codes of programs TXCOØ, TXCO1, TXCO2 and TXCO3 are available in the disc files &TXCOØ, &TXCO1, &TXCO2 and &TXCO3. Since TXCO1, TXCO2 and TXCO3 use common subroutines and functions, the latter are grouped together in file &TXCOU, where the "U" indicates the following "utility" program modules:

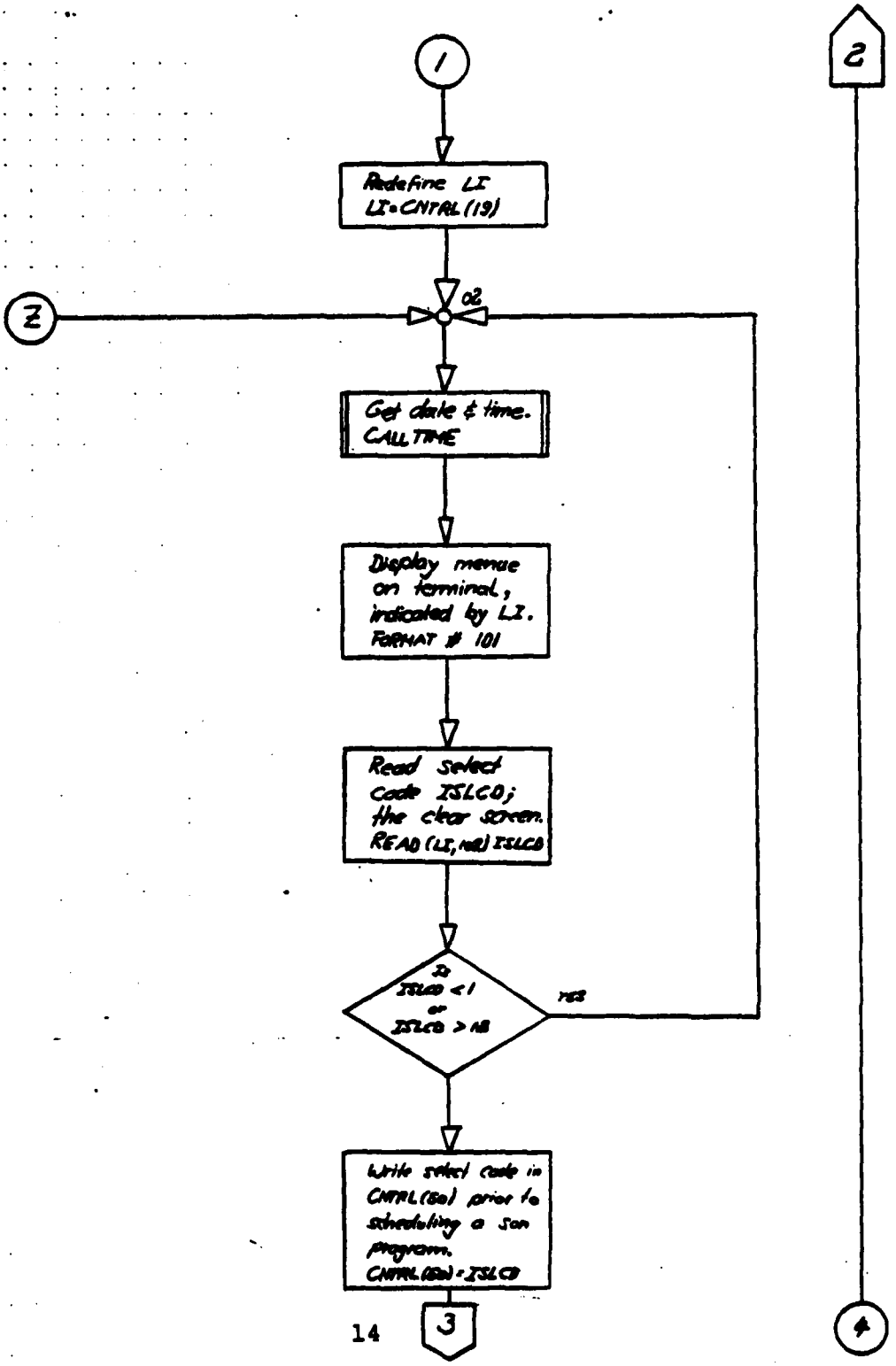
ACQN	Positions S/V and reads DVM.
CNTL	Closes scanner channel.
CURVE	Compute linear curve fit.
ICON	Converts two-digit INTEGER to ASCII-string.
IPORT	Interrogates S/V.
PICTR	Use CRT to display the acquired data.
REWRF	Data transfer disc ↔ array.
RPACE	Triggers A/D through PACER.
SCANR	Closes scanner channel and reads SVM, counter.
TIME	Gets date and time → ASCII-string.
WAIT	Causes a defined time delay.

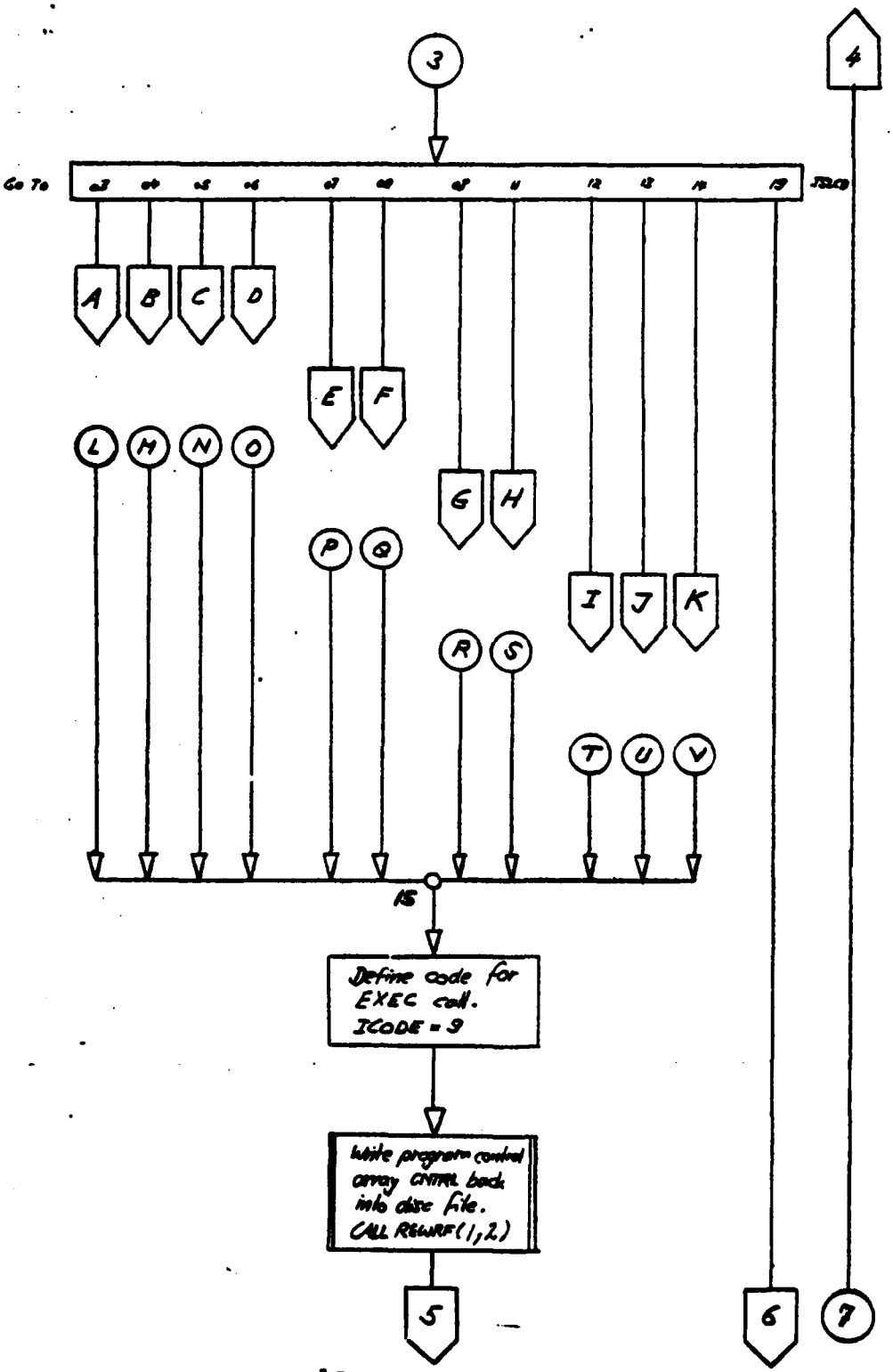
When loading %TXCO1, %TXCO2 or %TXCO3, the relocatable binary utility file %TXCOU must also be loaded in order to satisfy the externals. The modules of TXCOU are described in Section 7, but in less detail than the programs in Sections 4 through 6.

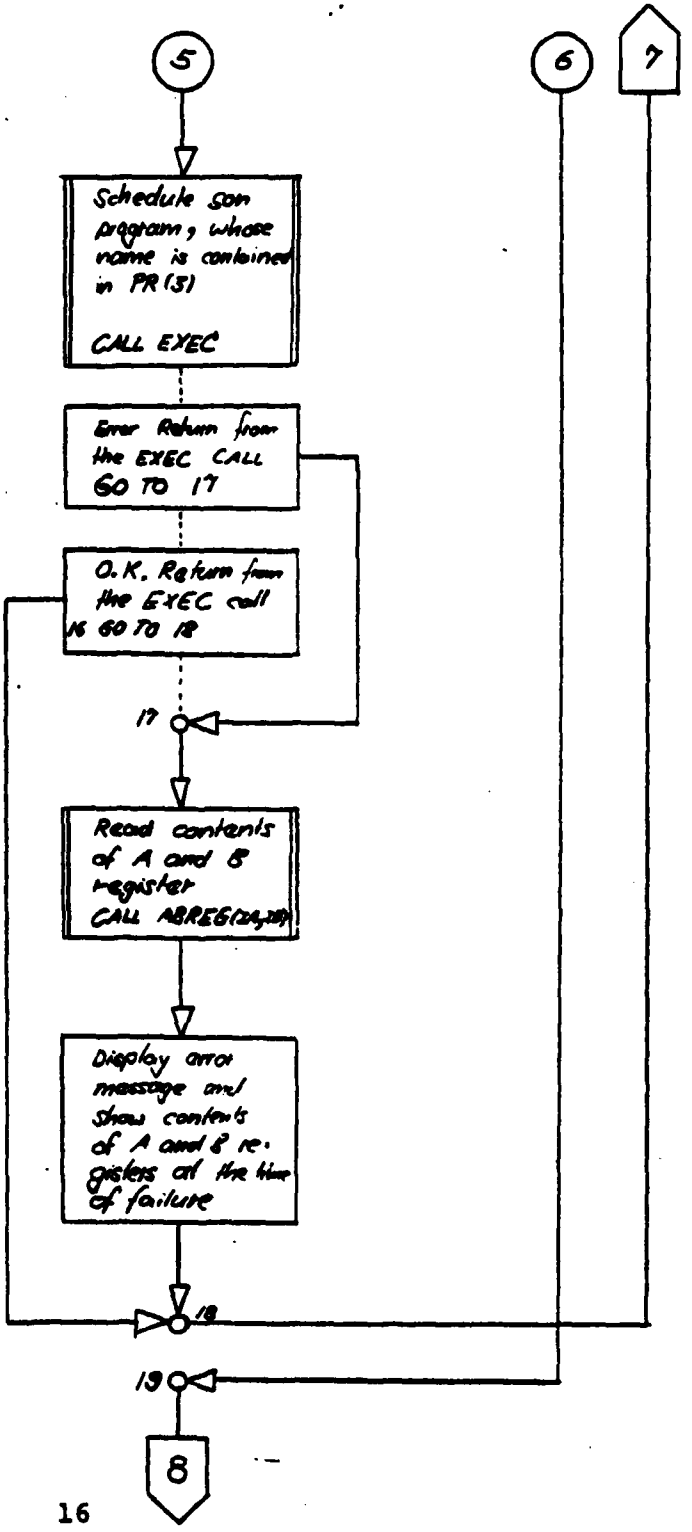
3. PROGRAM TXCOØ

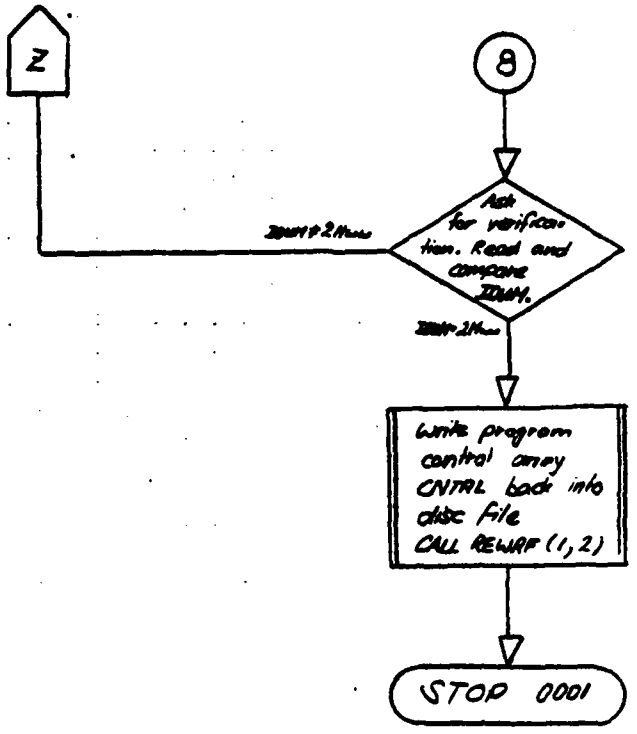
3.1. FLOW CHART PROGRAM TXCOØ:



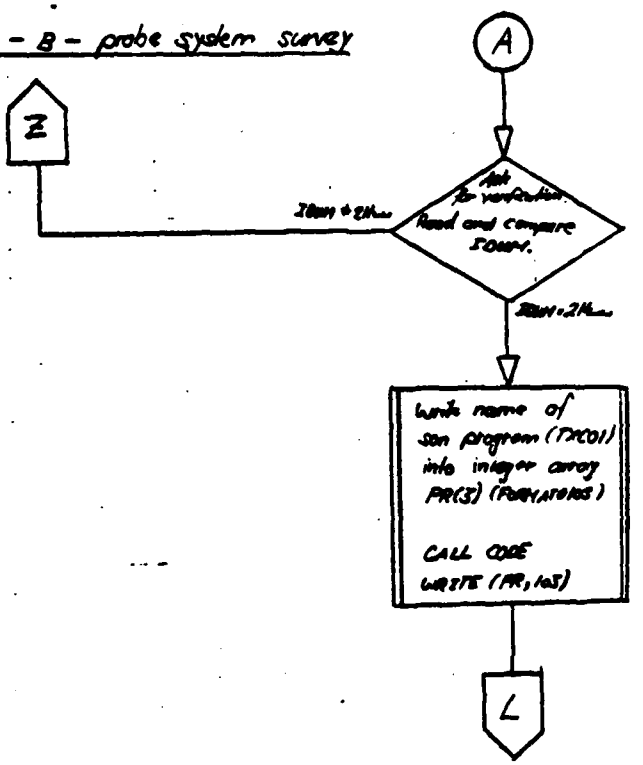

1
2
Redefine LI
LI=CNTRL(19)
02
2
Get date & time.
CALL TIME
Display menue
on terminal,
indicated by LI.
FORMAT # 101
Read select
code ISLCD;
the clear screen.
READ(LI,*) ISLCD
Is
ISLCD < 1
or
ISLCD > 18
yes
Write select code in
CNTRL(80) prior to
scheduling a son
program.
CNTRL(80)=ISLCD
3
4

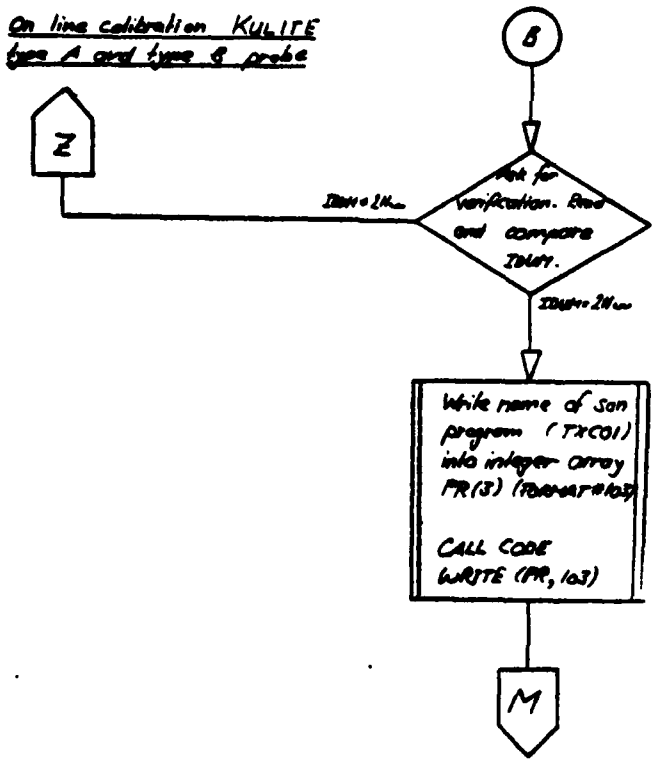

3
4
Go To
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
15
Define code for EXEC call. ICODE = 9
Write program control array CNTRL back into disc file. CALL REWRF(1,2)
5
6
7

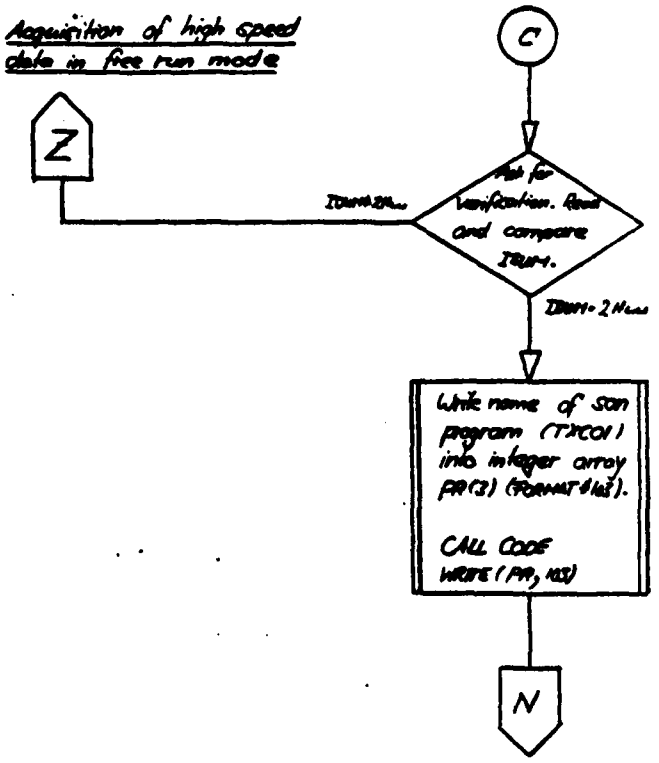

5
6
7
Schedule son program, whose name is contained in PR(5)
CALL EXEC
Error Return from the EXEC CALL GO TO 17
O.K. Return from the EXEC call IS GO TO 18
17
Read contents of A and B register
CALL ABREG(24,20)
Display error message and Show contents of A and B registers at the time of failure
18
19
8

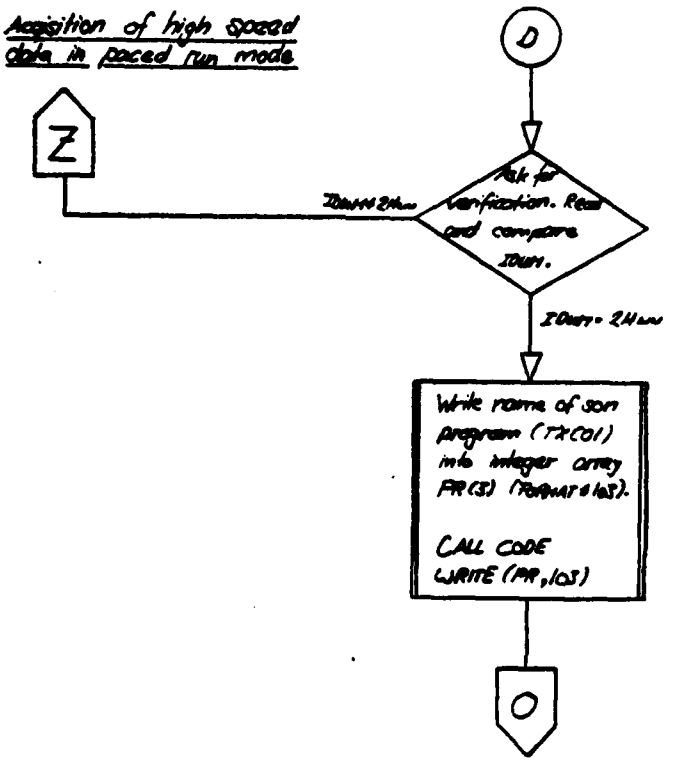

2
8
Ask for verification. Read and compare IDUM.
Write program control array CNTRL back into disc file
CALL REWAF (1, 2)
STOP 0001

A - B - probe system survey

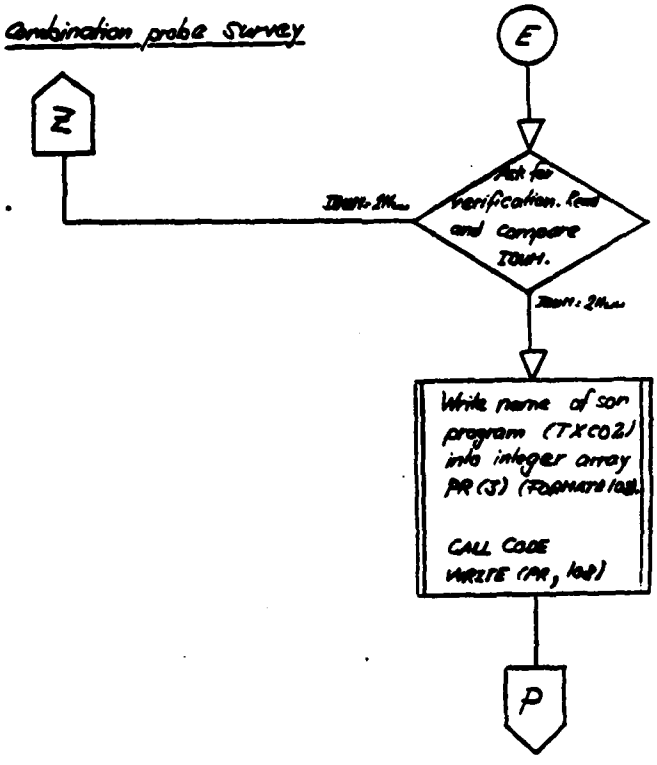


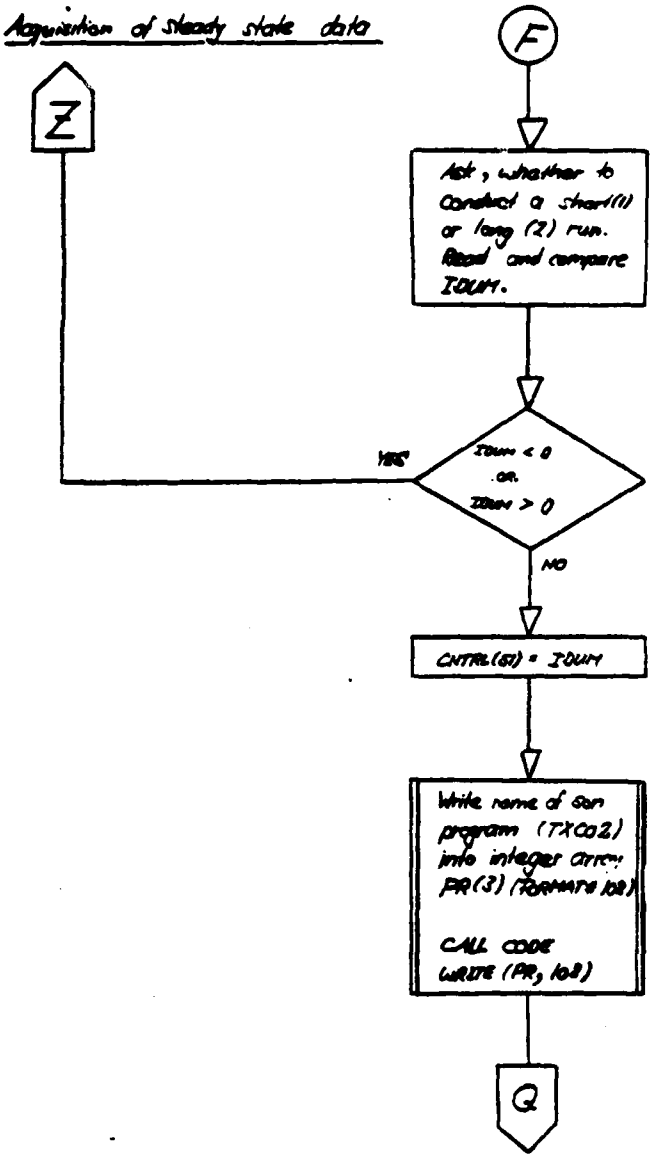

On line calibration KULITE
type A and type B probe
6
Z
Wait for verification. Read and compare IDATA.
While name of sen program (TXC01) into integer array PR(3) (
CALL CODE
WRITE (PR, 103)
M


Acquisition of high speed data in free run mode
C
Z
Ask for verification. Read and compare ITEM.
ITEM=ZERO
ITEM=2 NULL
Write name of scan program (TXCOM) into integer array PP(3) (FORMAT 3A2).
CALL CODE
WRITE (PP, 103)
N

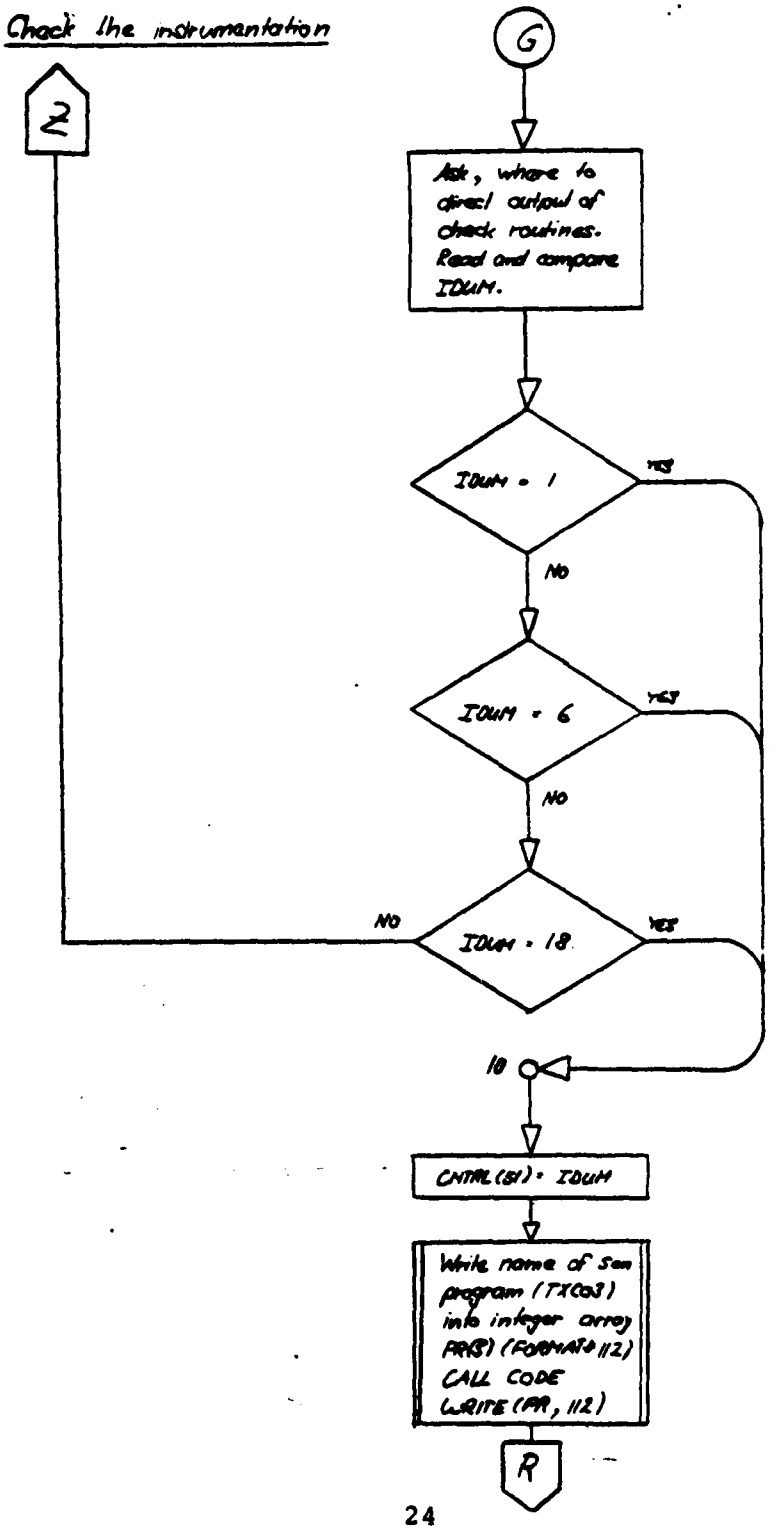

Acquition of high speed data in paced run mode
D
Z
Ask for verification. Read and compare IOstr.
IOstr = ZH
IOstr = ZH
Write name of son program (7*COL) into integer array PR(5) (Format 103).
CALL CODE
WRITE (PR, 103)
O

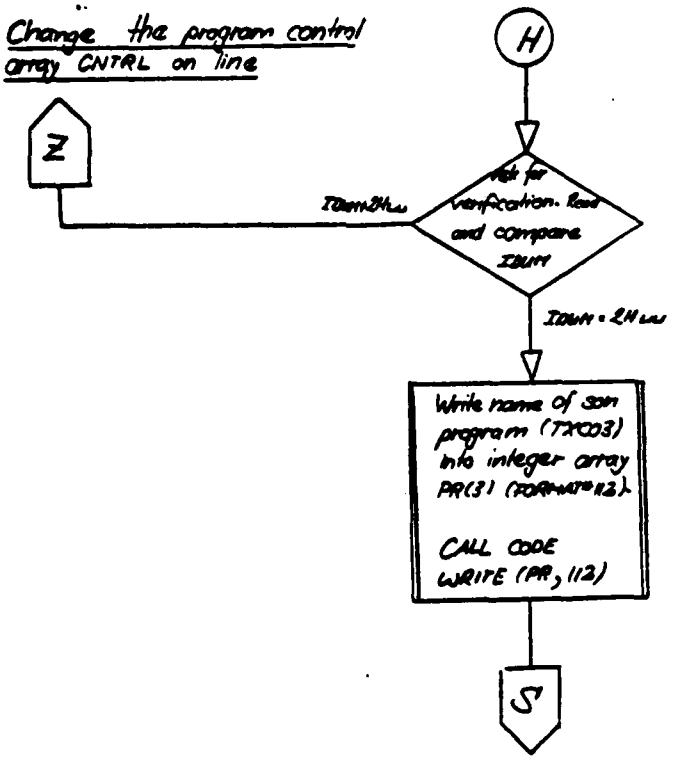
Combination probe survey

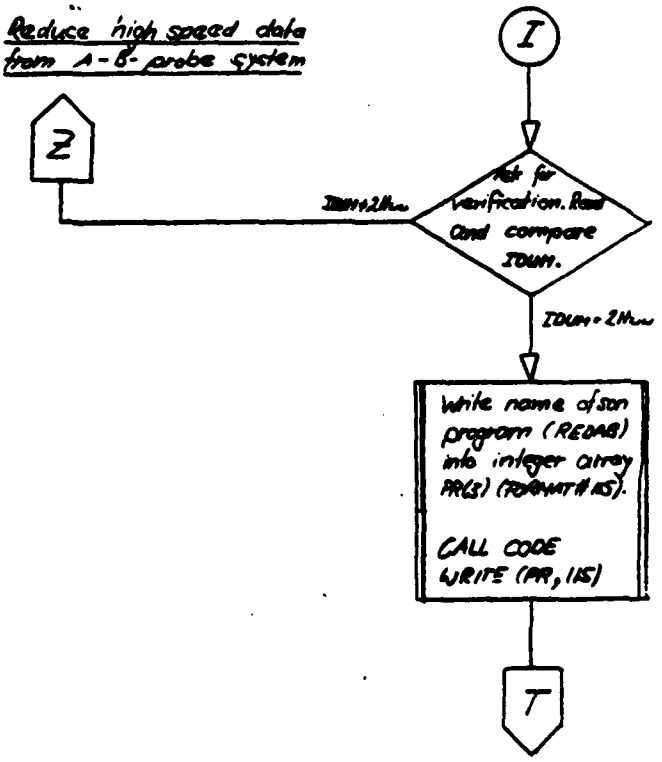


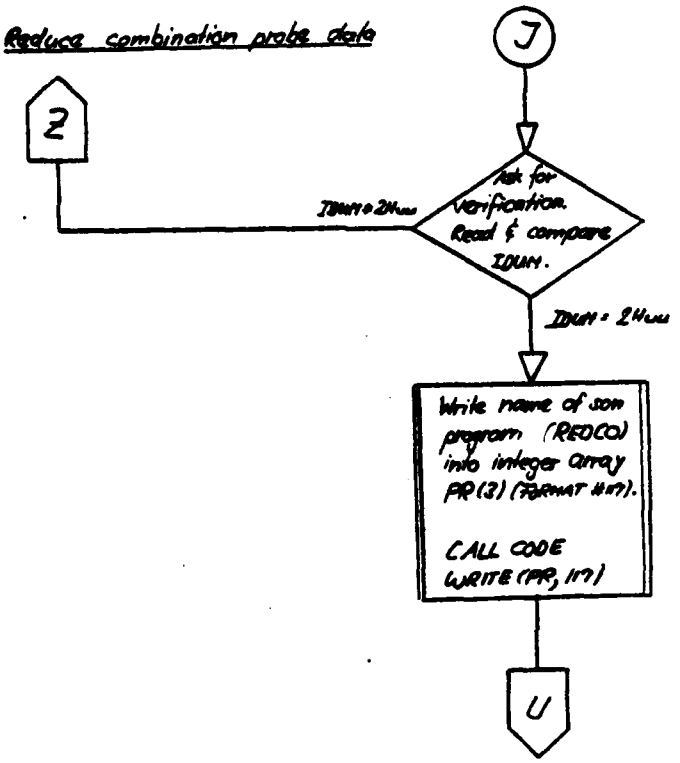

Acquisition of Steady state data
F
Z
Ask, whether to conduct a short(1) or long (2) run. Read and compare IDUM.
IDum < 0 .or. IDum > 0
YES
NO
CNTRL(61) = IDUM
Write name of scn program (TXCO2) into integer array PR(3) (FORMAT 108)
CALL CODE
WRITE (PR, 108)
Q

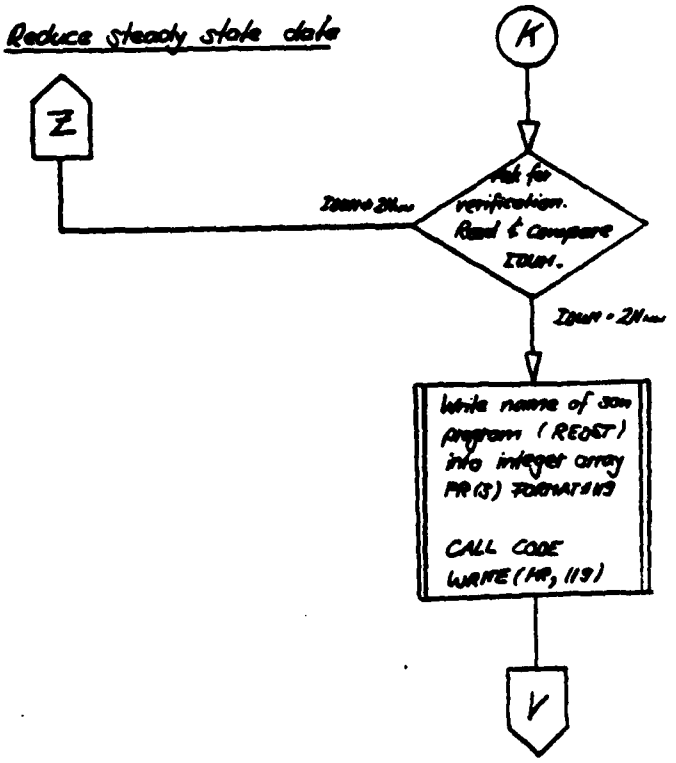
Check the instrumentation




Change the program control array CNTRL on line
H
Z
Ask for verification. Read and compare IDUM
IDUM=8Hxx
IDUM=2Hxx
Write name of son program (TXC03) into integer array PR(3) (FORMAT=12A).
CALL CODE
WRITE (PR, 112)
S


Reduce high speed data from A-B- probe system
I
2
Wait for verification. Read and compare IDUM.
IDUM=2Hxx
IDUM=2Hxx
Write name of son program (REDAB) into integer array PAR(3) (FORMAT #15).
CALL CODE
WRITE (PAR, 15)
T


Reduce combination probe data
J
Z
Ask for verification. Read & compare IQUH.
IQUH = 2Huu
IQUH = 2Huu
Write name of son program (REDCO) into integer array PR(3) (FORMAT 11H7).
CALL CODE
WRITE (PR, 117)
U


Reduce steady state data
K
Z
Ask for verification. Read & compare IOUT.
IOUT = 2H...
IOUT = 2H...
Write name of 3rd program (REQST) into integer array IP(3) FORMAT=119
CALL CODE
WRITE(IP, 119)
V

3.2. PROGRAM LISTING, TXCOØ (updated version, 20 September 1982)

```
 PAGE 0001  FTN.   9:32 AM  MON., 20  SEP., 1982

0001  FTN4,L
0002          BLOCK DATA
0003        *   / FMP   / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004          COMMON / FMP    / IDCB,IFILE,ISIZE,ISECU,ICR
0005          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006          END

  FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

  **  NO WARNINGS **  NO ERRORS **

 BLOCK COMMON FMP     SIZE =  00151
```

```
0007          BLOCK DATA
0008         *. / CIBUF / IFUF(1664)
0009          COMMON / CIBUF / IBUF
0010          INTEGER IBUF(1664)
0011          END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS **

BLOCK COMMON CIBUF SIZE = 01664

```
 PAGE 0003  FTN.   9:32 AM  MON., 20  SEP., 1982

0012          BLOCK DATA
0013         *, / CONTR / CNTRL(256)

0014          COMMON / CONTR / CNTRL
0015          INTEGER CNTRL(256)
0016          END

  FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)
  **  NO WARNINGS **  NO ERRORS **
BLOCK COMMON CONTR  SIZE =  00256
```

```
      PROGRAM TXC00 (3,99)
C     ..............................................................
C     .                                                            .
C     .  Data acquisition transsonic compressor.                   .
C     .                                                            .
C     ..............................................................
     *, Data acquisition transsonic compressor.

      COMMON / CONTR / CNTRL
      INTEGER CNTRL(256),IDCB(144),PR(3)
      INTEGER NOLF,NOCR(2),ICLR(3),START(3)

      DATA NOLF  /006537B/
      DATA NOCR  /000033B,040433B/
      DATA ICLR  /015524B,015515B,006537B/
      DATA START /015510B,015512B,015501B/


C     FORMATS  TXC00  START .
  100 FORMAT (3A2"  TXC00 : START")
  101 FORMAT (/" Transsonic Compressor Investigation: Test run #
     * "I3";   Test #"I2"; Point #"I2";"/38X"Date "A2"/"A2"/"A2";"1X"Tim
     *e "A2"."A2";"/" Select next step, please ..."///
     *"  1 ____ A - B - probe system survey "32X" ABSRV"
     */
     *"  2      On line calibration KULITE type A and type B pro
     *be "9X" CALIB"/
     *"  3      Acquisition of high speed data in free run mode
     *"12X" FREER"/
     *"  4      Acquisition of high speed data in paced run mode
     * "11X" PACER"//
     *"  5 ____ Combination probe survey "35X" COMB"/
     *"  6 ____ Acquisition of steady state data "27X" S
     *TDY"//
     *"  7 ____ Check the instrumentation "34X" CHECK"/
     *"  8      Change the program control array  CNTRL  on line
     * "11X" CHNGE"//
     *"  9      Reduce high speed data from A - B - probe system
     * "11X" REDAB"/
     *" 10      Reduce data from the combination probe "21X"
     * REDCO"/
     *" 11      Reduce steady state data "35X" REDST"//
     *" 12 "61X" STOP 0001"//
     *" Select desired program module!  Enter select code
     * "2A2)
  102 FORMAT (I2)

  103 FORMAT ("TXC01 ")
  104 FORMAT (" A - B - probe system survey?  Verify!"2A2)
  105 FORMAT (" On line calibration KULITE type A and type B probe?  Ve
     *rify!"2A2)
  106 FORMAT (" Acquisition of high speed data in free run mode.  PACER
     * all right?"2A2)
  107 FORMAT (" Acquisition of high speed data in paced run mode.  PACE
     *R all right?"2A2)

  108 FORMAT ("TXC02 ")
  109 FORMAT (" Combination probe survey?  Verify!"2A2)
  110 FORMAT (" Acquisition of steady state data.  Long (1) or short (0
     *) run?   "2A2)
  111 FORMAT (I1)

  112 FORMAT ("TXC03 ")
  113 FORMAT (" Check the instrumentation.  Output to LU
     *"2A2)
  114 FORMAT (" Change control array  CNTRL on line?  Verify!"2A2)

  115 FORMAT ("REDAB ")
  116 FORMAT (" Reduce high speed data from A - B - probe system.  Veri
     *fy!"2A2)
```

```
0086
0087     117 FORMAT ("REDCO ")
0088     118 FORMAT (" Reduce data from the combination probe?  Verify!"2A2)
0089
0090     119 FORMAT ("REDST ")
0091     120 FORMAT (" Reduce steady state data?  Verify!"2A2)
0092
0093     121 FORMAT ("  TXC00 : SCHEDULE "2A2,A1"   CNTRL(50) ="I2)
0094     122 FORMAT ("  TXC00 : FAILED TO SCHEDULE "2A2,A1".  A REGISTER IS"@7"
0095        *:  B REGISTER IS"@7,A2/10X"LOAD PROGRAM "2A2,A1"!")
0096     123 FORMAT (" STOP 0001 ? Verify!"2A2)
0097     124 FORMAT (9X""20X""A2)
0098     149 FORMAT ((3A2))
0099  C      FORMATS  TXC00  STOP  .
0100
0101         LI = LOGLU(ISESSN)
0102         CNTRL(19) = LI
0103         WRITE (LI,100) START
0104      01 CALL REWRF (-1,2)
0105         LI = CNTRL(19)
0106      02 CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0107         IF ( CNTRL(4) .GE. 100) IRUN = CNTRL(4) - 100
0108         WRITE (LI, 101) IRUN,CNTRL(5),CNTRL(6),IMON,IDAY,IYEAR,IHOUR,I
0109        *MIN,NOCR
0110         READ  (LI, 102) ISLCD
0111         WRITE (LI, 149) (ICLR,I=1,24)
0112         IF ( ISLCD  .LT.  1   .OR.   ISLCD  .GT.  12 ) GO TO 02
0113
0114         CNTRL(50) = ISLCD
0115         GO TO (03,04,05,06,07,08,09,11,12,13,14,19) ISLCD
0116
0117  C      .....................................................................
0118  C      .                                                                   .
0119  C      .  A - B - probe system survey.                                     .
0120  C      .                                                                   .
0121  C      .....................................................................
0122      03 WRITE (LI, 104) NOCR
0123         READ  (LI, 149) IDUM
0124         WRITE (LI, 149) ICLR
0125         IF ( IDUM  .NE.  2H   ) GO TO 02
0126         CALL CODE
0127         WRITE (PR, 103)
0128         GO TO 15
0129
0130  C      .....................................................................
0131  C      .                                                                   .
0132  C      .  On line calibration KULITE type A and type B probe.              .
0133  C      .                                                                   .
0134  C      .....................................................................
0135      04 WRITE (LI, 105) NOCR
0136         READ  (LI, 149) IDUM
0137         WRITE (LI, 149) ICLR
0138         IF ( IDUM  .NE.  2H   ) GO TO 02
0139         CALL CODE
0140         WRITE (PR, 103)
0141         GO TO 15
0142
0143  C      .....................................................................
0144  C      .                                                                   .
0145  C      .  Acquisition of high speed data in free run mode.                 .
0146  C      .                                                                   .
0147  C      .....................................................................
0148      05 WRITE (LI, 106) NOCR
0149         READ  (LI, 149) IDUM
0150         WRITE (LI, 149) ICLR
0151         IF ( IDUM  .NE.  2H   ) GO TO 02
0152         CALL CODE
0153         WRITE (PR, 103)
0154         GO TO 15
```

```
0155
0156  C     ..................................................................
0157  CC    .                                                                .
0158  CC    .  Acquisition of high speed data in paced run mode.             .
0159  CC    .                                                                .
0160  C     ..................................................................
0161        06 WRITE (LI, 107) NOCR
0162           READ  (LI, 149) IDUM
0163           WRITE (LI, 149) ICLR
0164           IF ( IDUM  .NE.  2H   ) GO TO 02
0165           CALL CODE
0166           WRITE (PR, 103)
0167           GO TO 15
0168
0169  C     ..................................................................
0170  CC    .                                                                .
0171  CC    .  Combination probe survey.                                     .
0172  CC    .                                                                .
0173  C     ..................................................................
0174        07 WRITE (LI, 109) NOCR
0175           READ  (LI, 149) IDUM
0176           WRITE (LI, 149) ICLR
0177           IF ( IDUM  .NE.  2H   ) GO TO 02
0178           CALL CODE
0179           WRITE (PR, 108)
0180           GO TO 15
0181
0182  C     ..................................................................
0183  CC    .                                                                .
0184  CC    .  Acquisition of steady state data.                             .
0185  CC    .                                                                .
0186  C     ..................................................................
0187        08 WRITE (LI, 110) NOCR
0188           READ  (LI, 111) IDUM
0189           WRITE (LI, 149) ICLR
0190           IF ( IDUM  .LT.  0   .OR.   IDUM  .GT.  1 ) GO TO 02
0191           CNTRL(51) = IDUM
0192           CALL CODE
0193           WRITE (PR, 108)
0194           GO TO 15
0195
0196  C     ..................................................................
0197  CC    .                                                                .
0198  CC    .  Check the instrumentation.                                    .
0199  CC    .                                                                .
0200  C     ..................................................................
0201        09 WRITE (LI, 113) NOCR
0202           READ  (LI, 102) IDUM
0203           WRITE (LI, 149) ICLR
0204           IF ( IDUM  .EQ.   1 ) GO TO 10
0205           IF ( IDUM  .EQ.   6 ) GO TO 10
0206           IF ( IDUM  .EQ.  18 ) GO TO 10
0207           GO TO 02
0208        10 CNTRL(51) = IDUM
0209           CALL CODE
0210           WRITE (PR, 112)
0211           GO TO 15
0212
0213  C     ..................................................................
0214  CC    .                                                                .
0215  CC    .  Change the program control array  CNTRL  on line.             .
0216  CC    .                                                                .
0217  C     ..................................................................
0218        11 WRITE (LI, 114) NOCR
0219           READ  (LI, 149) IDUM
0220           WRITE (LI, 149) ICLR
0221           IF ( IDUM  .NE.  2H   ) GO TO 02
0222           CALL CODE
0223           WRITE (PR, 112)
```

```
0224          GO TO 15
0225
0226    C     ..................................................................
0227    CC    .                                                                .
0228    C     .  Reduce high speed data from A - B - probe system.             .
0229    CC    .                                                                .
0230    C     ..................................................................
0231       12 WRITE (LI, 116) NOCR
0232          READ  (LI, 149) IDUM
0233          WRITE (LI, 149) ICLR
0234          IF ( IDUM  .NE.  2H   ) GO TO 02
0235          CALL CODE
0236          WRITE (PR, 115)
0237          GO TO 15
0238    C     ..................................................................
0239    CC    .                                                                .
0240    C     .  Reduce combination probe data.                                .
0241    CC    .                                                                .
0242    C     ..................................................................
0243       13 WRITE (LI, 118) NOCR
0244          READ  (LI, 149) IDUM
0245          WRITE (LI, 149) ICLR
0246          IF ( IDUM  .NE.  2H   ) GO TO 02
0247          CALL CODE
0248          WRITE (PR, 117)
0249          GO TO 15
0250
0251    C     ..................................................................
0252    CC    .                                                                .
0253    CC    .  Reduce steady state data.                                     .
0254    CC    .                                                                .
0255    C     ..................................................................
0256       14 WRITE (LI, 120) NOCR
0257          READ  (LI, 149) IDUM
0258          WRITE (LI, 149) ICLR
0259          IF ( IDUM  .NE.  2H   ) GO TO 02
0260          CALL CODE
0261          WRITE (PR, 119)
0262          GO TO 15
0263
0264    C     ..................................................................
0265    CC    .                                                                .
0266    C     .  Schedule desired son program.                                 .
0267    CC    .                                                                .
0268    C     ..................................................................
0269       15 WRITE (LI, 121) PR,CNTRL(50)
0270          ICODE = 9
0271          CALL REWRF (1,2)
0272          CALL EXEC (ICODE+100000B,PR,IDCB,IDCBS)
0273          GO TO 17
0274       16 GO TO 18
0275       17 CALL ABREG (IA,IB)
0276          WRITE (LI, 149) (ICLR,I=1,2)
0277          WRITE (LI, 122) PR,IA,IB,NOLF,PR
0278       18 GO TO 01
0279
0280    C     ..................................................................
0281    CC    .                                                                .
0282    C     .  STOP 0001.                                                    .
0283    CC    .                                                                .
0284    C     ..................................................................
0285       19 WRITE (LI, 123) NOCR
0286          READ  (LI, 149) IDUM
0287          WRITE (LI, 149) ICLR
0288          IF ( IDUM  .NE.  2H   ) GO TO 02
0289          CALL REWRF (1,2)
0290          WRITE ( 1, 124) NOLF
0291          STOP 0001
0292          END
```

```
PAGE 0008  TXC00  9:32 AM  MON., 20  SEP., 1982
 FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)
 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 02005       COMMON = 00000
```

```
0293          INTEGER FUNCTION ICON (I,N)
0294    C     .....................................................................
0295    C     :                                                                   :
0296    C     :  Converts integer numbers into ASCI string.                       :
0297    C     :  Author:    Robert N. Geopfarth                                   :
0298    C     :  Date:      January 31, 1979                                      :
0299    C     :  Because of the simplicity of the program the program             :
0300    C     :  description is included in this box.                             :
0301    C     :  I, N   ...   integer numbers to be added.                        :
0302    C     :  IC     ...   integer number to be converted into ASCII.          :
0303    C     :  ICON   ...   2 - character ASCII string to be returned          :
0304    C     :                                                                   :
0305    C     .....................................................................
0306         *, Converts integer to ASCII-string.
0307      100 FORMAT (I2)
0308
0309          IC = I+N
0310          IF ( IC  .LT.  10 ) GO TO 01
0311
0312          CALL CODE
0313          WRITE (ICON,100) IC
0314          RETURN
0315
0316       01 ICON = IC+30060B
0317          RETURN
0318          END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00036 COMMON = 00000

```
0319          SUBROUTINE REWRF (IREWR,IWHATA)
0320    C     .....................................................................
0321    C     .                                                                   .
0322    C     .  This subroutine reads (IREWR = +1) or writes (IREWR = -1) of     .
0323    C     .  of a array specified by IWHATA.                                  .
0324    C     .  Author:    Hans M. Zebner                                        .
0325    C     .  Date :     Febraury 08, 1980                                     .
0326    C     .  Detailed program description is available in TXCO log;  The      .
0327    C     .  Comment statements match to the flow chart explanations.         .
0328    C     .                                                                   .
0329    C     .....................................................................
0330         *, Data transfer disc    array.
0331          COMMON / CIBUF / IBUF
0332          COMMON / CONTR / CNTRL
0333    C     COMMON / CA    / A
0334          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
0335    C     REAL A(256)
0336          INTEGER IBUF(1664)
0337          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0338          INTEGER NOLF,NOCR(2),ICLR(3)
0339          DATA NOLF  /006537B/
0340          DATA NOCR  /000033B,040433B/
0341          DATA ICLR  /015524B,015515B,006537B/
0342    C     FORMATS REWRF START
0343       101 FORMAT ("  REWRF : ARRAY IBUF(1664)   DISC FILE IBUFF
0344         *:00:26")
0345       102 FORMAT ("  REWRF : DISC FILE IBUFF:00:26   ARRAY IBUF(1664)")
0346       103 FORMAT ("  REWRF : ARRAY CNTRL(256)   DISC FILE CNTRLF:00:26"
0347         *)
0348       104 FORMAT ("  REWRF : DISC FILE CNTRLF:00:26   ARRAY CNTRL(256)"
0349         *)
0350       105 FORMAT ("  REWRF : ARRAY A(256)   DISC FILE AF:00:26")
0351       106 FORMAT ("  REWRF : DISC FILE AF:00:26   ARRAY A(256)")
0352       107 FORMAT ("  REWRF : ERROR RETURN (IWHATA ="I3")")
0353       108 FORMAT ("IBUFF ")
0354       109 FORMAT ("CNTRLF")
0355       110 FORMAT ("AF    ")
0356       121 FORMAT (" CALL OPEN  (IDCB,IERR,"3A2","I2","I2","I2","I4"
0357         *)  failed; STOP"21X"")
0358       122 FORMAT (" CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),I
0359         *DUM,IDUM,ISIZE(2)) failed; STOP")
0360       123 FORMAT (" CALL RWNDF (IDCB,IERR)  failed; STOP"42X"")
0361       124 FORMAT (" CALL READF (IDCB,IERR,IBUF,"I3","I2","I2")  fai
0362         *led; STOP"27X"")
0363       125 FORMAT (" CALL WRITF (IDCB,IERR,IBUF,"I3","I2","I2")  fai
0364         *led; STOP"26X"")
0365       126 FORMAT (" CALL READF (IDCB,IERR,CNTRL,"I3","I2","I2")  fa
0366         *iled; STOP"27X"")
0367       127 FORMAT (" CALL WRITF (IDCB,IERR,CNTRL,"I3","I2","I2")  fc
0368         *iled; STOP"26X"")
0369       128 FORMAT (" CALL READF (IDCB,IERR,A,"I3","I2","I2")  failed
0370         *; STOP"27X"")
0371       129 FORMAT (" CALL WRITF (IDCB,IERR,A,"I3","I2","I2")  failed
0372         *; STOP"26X"")
0373       130 FORMAT (" CALL CLOSE (IDCB,IERR,0)  failed; STOP"40X""
0374         *)
0375          LI = LOGLU(ISESSN)
0376          ISECU = 0
0377          ICR = 26
0378          IF ( IWHATA  .LT.  1   .OR.   IWHATA  .GT.  2 ) GO TO 40
0379          GO TO (10,20) IWHATA
0380
0381
0382
0383    C     .....................................................................
0384    C     .                                                                   .
0385    C     .  Integer array IBUF being written back and forth.                 .
0386    C     .                                                                   .
0387    C     .....................................................................
```

```
0388      10 CALL CODE
0389         WRITE (IFILE,108)
0390         CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0391         IF ( IERR  .GE.  0 ) GO TO 11
0392         WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0393         STOP 1
0394      11 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0395         IF ( IERR  .GE.  0 ) GO TO 12
0396         WRITE (LI, 122)
0397         STOP 2
0398      12 CALL RWNDF (IDCB,IERR)
0399         IF ( IERR  .GE.  0 ) GO TO 13
0400         WRITE (LI, 123)
0401         STOP 3
0402      13 ISIZE(1) = ISIZE(1)/2
0403         IL          = ISIZE(1)*ISIZE(2)
0404         IF ( IREWR  .EQ.  -1 ) GO TO 14
0405         IF ( IREWR  .EQ.  +1 ) GO TO 15
0406      14 CALL READF (IDCB,IERR,IBUF,IL)
0407         IF ( IERR  .GE.  0 ) WRITE (LI, 102)
0408         IF ( IERR  .GE.  0 ) GO TO 16
0409         WRITE (LI, 124) IL,LEN,NUM
0410         STOP 4
0411      15 CALL WRITF (IDCB,IERR,IBUF,IL)
0412         IF ( IERR  .GE.  0 ) WRITE (LI, 101)
0413         IF ( IERR  .GE.  0 ) GO TO 16
0414         WRITE (LI, 125) IL,LEN,NUM
0415         STOP 5
0416      16 CALL CLOSE (IDCB,IERR,0)
0417         IF ( IERR  .GE.  0 ) GO TO 17
0418         WRITE (LI, 130)
0419         STOP 6
0420      17 RETURN
0421
0422
0423
0424  C     ..................................................................
0425  C     .                                                                .
0426  C     .  Integer array CNTRL being written back and forth.             .
0427  C     .                                                                .
0428  C     ..................................................................
0429      20 CALL CODE
0430         WRITE (IFILE,109)
0431         CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0432         IF ( IERR  .GE.  0 ) GO TO 21
0433         WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0434         STOP 11
0435      21 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0436         IF ( IERR  .GE.  0 ) GO TO 22
0437         WRITE (LI, 122)
0438         STOP 12
0439      22 CALL RWNDF (IDCB,IERR)
0440         IF ( IERR  .GE.  0 ) GO TO 23
0441         WRITE (LI, 123)
0442         STOP 13
0443      23 ISIZE(1) = ISIZE(1)/2
0444         IL          = ISIZE(1)*ISIZE(2)
0445         IF ( IREWR  .EQ.  -1 ) GO TO 24
0446         IF ( IREWR  .EQ.  +1 ) GO TO 25
0447      24 CALL READF (IDCB,IERR,CNTRL,IL)
0448         IF ( IERR  .GE.  0 ) WRITE (LI, 104)
0449         IF ( IERR  .GE.  0 ) GO TO 26
0450         WRITE (LI, 126) IL,LEN,NUM
0451         STOP 14
0452      25 CALL WRITF (IDCB,IERR,CNTRL,IL)
0453         IF ( IERR  .GE.  0 ) WRITE (LI, 103)
0454         IF ( IERR  .GE.  0 ) GO TO 26
0455         WRITE (LI, 127) IL,LEN,NUM
0456         STOP 15
```

```
0457        26 CALL CLOSE (IDCB,IERR,0)
0458           IF ( IERR  .GE.  0 ) GO TO 27
0459           WRITE (LI, 130)
0460           STOP 16
0461        27 RETURN
0462
0463
0464
0465   C       .....................................................................
0466   C       .                                                                   .
0467   C       .  Real array A being written back and forth.                       .
0468   C       .                                                                   .
0469   C       .....................................................................
0470   C    30 CALL CODE
0471   C       WRITE (IFILE,110)
0472   C       CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0473   C       IF ( IERR  .GE.  0 ) GO TO 31
0474   C       WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0475   C       STOP 21
0476   C    31 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0477   C       IF ( IERR  .GE.  0 ) GO TO 32
0478   C       WRITE (LI, 122)
0479   C       STOP 22
0480   C    32 CALL RWNDF (IDCB,IERR)
0481   C       IF ( IERR  .GE.  0 ) GO TO 33
0482   C       WRITE (LI, 123)
0483   C       STOP 23
0484   C    33 ISIZE(1) = ISIZE(1)/2
0485   C       IL          = ISIZE(1)*ISIZE(2)
0486   C       IF ( IREWR  .EQ.  -1 ) GO TO 34
0487   C       IF ( IREWR  .EQ.  +1 ) GO TO 35
0488   C    34 CALL READF (IDCB,IERR,A,IL)
0489   C       IF ( IERR  .GE.  0 ) WRITE (LI, 106)
0490   C       IF ( IERR  .GE.  0 ) GO TO 36
0491   C       WRITE (LI, 128) IL,LEN,NUM
0492   C       STOP 24
0493   C    35 CALL WRITF (IDCB,IERR,A,IL)
0494   C       IF ( IERR  .GE.  0 ) WRITE (LI, 105)
0495   C       IF ( IERR  .GE.  0 ) GO TO 36
0496   C       WRITE (LI, 129) IL,LEN,NUM
0497   C       STOP 25
0498   C    36 CALL CLOSE (IDCB,IERR,0)
0499   C       IF ( IERR  .GE.  0 ) GO TO 37
0500   C       WRITE (LI, 130)
0501   C       STOP 26
0502   C    37 RETURN
0503
0504
0505
0506   C       .....................................................................
0507   C       .                                                                   .
0508   C       .  Error;  IWHATA is not defined.                                   .
0509   C       .                                                                   .
0510   C       .....................................................................
0511        40 WRITE (LI, 107) IWHATA
0512           IWHATA = -IWHATA
0513           RETURN
0514           END
```

FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01146 COMMON = 00000

```
0515          SUBROUTINE TIME (IMON,IDAY,IYEAR,IHOUR,IMIN,ISEC)
0516    C     ..................................................................
0517    C     :                                                                :
0518    C     :  Get date and time and convert the variables to ASCII          :
0519    C     :                                                                :
0520    C     ..................................................................
0521          *  Gets date and time  ASCII string.
0522           COMMON / CONTR / CNTRL
0523           INTEGER ITIME(5)
0524           INTEGER CNTRL(256)
0525      901  FORMAT (" ERROR DETECTED IN PROGRAM TIME"/
0526          *          " CALL EXEC (11,ITIME)"/)
0527
0528           IMON = 2H$$
0529           IDAY = 2H$$
0530           IYEAR= 2H$$
0531           IHOUR= 2H$$
0532           IMIN = 2H$$
0533           ISEC = 2H$$
0534           CALL EXEC (11+100000B,ITIME)
0535           GO TO 02
0536        01 GO TO 03
0537        02 CALL ABREG (IA,IB)
0538           GO TO 04
0539        03 IMON  = ICON(CNTRL(1),0)
0540           IDAY  = ICON(CNTRL(2),0)
0541           IYEAR = ICON(CNTRL(3),0)
0542           IHOUR = ICON(ITIME(4),0)
0543           IMIN  = ICON(ITIME(3),0)
0544           ISEC  = ICON(ITIME(2),0)
0545           RETURN
0546        04 WRITE ( 6, 901) IA,IB
0547           RETURN
0548           END

 FTN4 COMPILER: HP92060-16092 REV. 2001 (791101)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00146      COMMON = 00000
```

```
FMP     26042 26270    / FMP    / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
CIBUF   26271 31470    / CIBUF  / IFUF(1664)
CONTR   31471 32070    / CONTR  / CNTRL(256)

TXCOO   32071 36015    Data acquisition transsonic compressor.

ICON    36016 36061    Convertes integer to ASCII-string.
REWRF   36062 40253    Data transfer disc    array.
TIME    40254 40475    Gets date and time    ASCII string.

LOGLU   40476 40553   92067-1X297 REV.2013 790228
READF   40554 41544   92067-16125 REV.2001 791015
OPEN    41545 42107   92067-16125 REV.2001 791018
CLOSE   42110 42323   92067-16125 REV.2001 791019
OVRD.   42324 42324   92067-16125 REV.1903 780526
$SMVE   42325 42417   92067-1X483 REV.2013 800129
LOCF    42420 42720   92067-16125 REV.1903 781110
ABREG   42721 42742   92068-1X013 REV.2013 750701
RWNDF   42743 43027   92067-16125 REV.1903 780724
LURQ    43030 43442   92067-1X270 REV.2013 791024
CLR10   43443 43451   24998-1X248 REV.2001 750701
FMT10   43452 44750   24998-1X230 REV.2001 790417
IFTTY   44751 45036   92067-1X295 REV.2013 790118
.DADS   45037 45146   24998-1X036 REV.2001 780818
.DMP    45147 45314   24998-1X045 REV.2001 780818
.DDI    45315 45615   24998-1X040 REV.2001 781021
SESSN   45616 45633   92067-16125 REV.1903 780413
R/W$    45634 45772   92067-16125 REV.1903 781214
P.PAS   45773 46021   92067-16125 REV.1903 740801
.DNG    46022 46031   24998-1X046 REV.2001 780818
PAUSE   46032 46132   24998-1X253 REV.2001 771122
$ALRN   46133 46250   92067-1X271 REV.2013 770715
.SBT    46251 46311   92068-1X011 REV.2013 770518
$OPEN   46312 46466   92067-16125 REV.1903 790103
RW$UB   46467 47034   92067-16125 REV.1903 781003
RWND$   47035 47157   92067-16125 REV.1903 780801
.DIN    47160 47165   24998-1X042 REV.2001 780818
.DDE    47166 47177   24998-1X039 REV.2001 780818
FRMTR   47200 52635   24998-1X231 REV.2001 790503
FMT.E   52636 52636   24998-1X232 REV.2001 781107
PAU.E   52637 52637   24998-1X254 REV.2001 750701
.CFER   52640 52701   24998-1X196 REV.2001 790523
$SETP   52702 52726   24998-1X013 REV.2001 781106
REIO    52727 53053   92067-1X275 REV.2013 790316
RMPAR   53054 53116   92068-1X025 REV.2013 781106
LUTRU   53117 53225   92067-1X308 REV.2013 790223
PNAME   53226 53273   92068-1X035 REV.2013 771121
.LBT    53274 53324   92068-1X008 REV.2013 770518

12 PAGES RELOCATED    12 PAGES REQ'D    NO PAGES EMA      NO PAGES MSEG
LINKS:BP    PROGRAM:BG    LOAD:TE    COMMON:NC
 /LOADR:TXCOO  READY AT  9:55 AM  MON., 20  SEPT, 1982

 /LOADR:$END
```

4. PROGRAM TXCO1

4.1. DESCRIPTION

TXCO1 is a son program of the father program TXCOØ, by which it is scheduled, if one of the following operations is desired:

1 - A - B - probe system survey

2 - On line calibration, KULITE type 'A' and 'B' probes

3 - Acquisition of high speed data in free run mode

4 - Acquisition of high speed data in paced run mode.

When scheduled by TXCOØ, which suspends operation while the son program TXCO1 executes, the program TXCO1, reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control and programs the Digital Voltmeter (DVM), the scanners and the counter. CNTRL (50) is the actual decision variable to select and call the subroutine, which performs the desired operation. When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note, that each stop coding ending on 77 indicates correct execution of a subroutine.

CNTRL (50)	Subroutine	STOP Code
1	ABSRV	TXCO1 : STOP 0177
2	CALIB	TXCO1 : STOP 0277
3	FREER	TXCO1 : STOP 0377
4	PACER	TXCO1 : STOP 0477

Any other STOP code indicates an error and utilizing a program list the operator can trace the problem. The first two digits of the STOP code are typical for the subroutines. An example: the program stops at STOP code 0304; the first two digits read 3 and this tells the operator that it was subroutine FREER which ran into trouble, because the ending two digits read 04, which is different from 77; a program list uncovers that the failure occurred while writing into a disc file using FMP (File Management Package) subroutine WRITF near line 1005. STOP codes are crucial for a complex program system in order to rapidly detect and salvage problems, even during a test run.

EXTERNALS: REWRF, ABRT, RMOTE, ABSRV, CALIB, FREER, PACER, CLEAR, LOCL

COMMON BLOCKS: CONTR, CIBUF, FMP

FORTRAN conventions for the HP21MX computer request COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

BLOCK DATA subroutine	arrays & variables	length in words
CONTR	CNTRL	400B = 256
CIBUF	IBUF	3200B = 1664
FMP	IDCB,IFILE,ISIZE,ISECU,ICR	227B = 151

The COMMON block CONTR allocates the space for the control array CNTRL. A key to decode the individual elements of CNTRL can be found in the Appendix.COMMON block CNTRL is designed to take the largest raw data array - IBUF (1664) in subroutine

FREER - even if other subroutines only partially use the space, allocated by the block CIBUF. The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. Since each individual subroutine saves the data prior to terminating, more than one subroutine or function may use the same buffer area.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL (50) is less than 1 or greater than 4, no subroutine can be selected and the program terminates, outputting an error message (FORMAT 102) to the terminal.

PROCEDURE: For more detailed information study the flow chart and the information given in the section PURPOSE.

DATA FILE: None

VARIABLES IN BLOCK DATA CONTR:

CNTRL (256)	integer	program control array.

VARIABLES IN BLOCK DATA CIBUF:

IBUF (1664)	integer	buffer array for the raw data.

VARIABLES IN BLOCK DATA FMP:

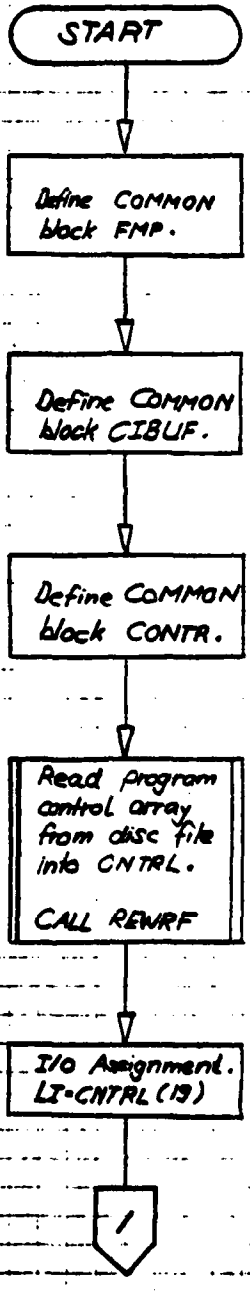
IDCB (144)	integer	data control block.
IFILE (3)	integer	array to contain file name.
ISIZE (2)	integer	array to contain # of records in the first and record length in 16-bit-words in the second word.

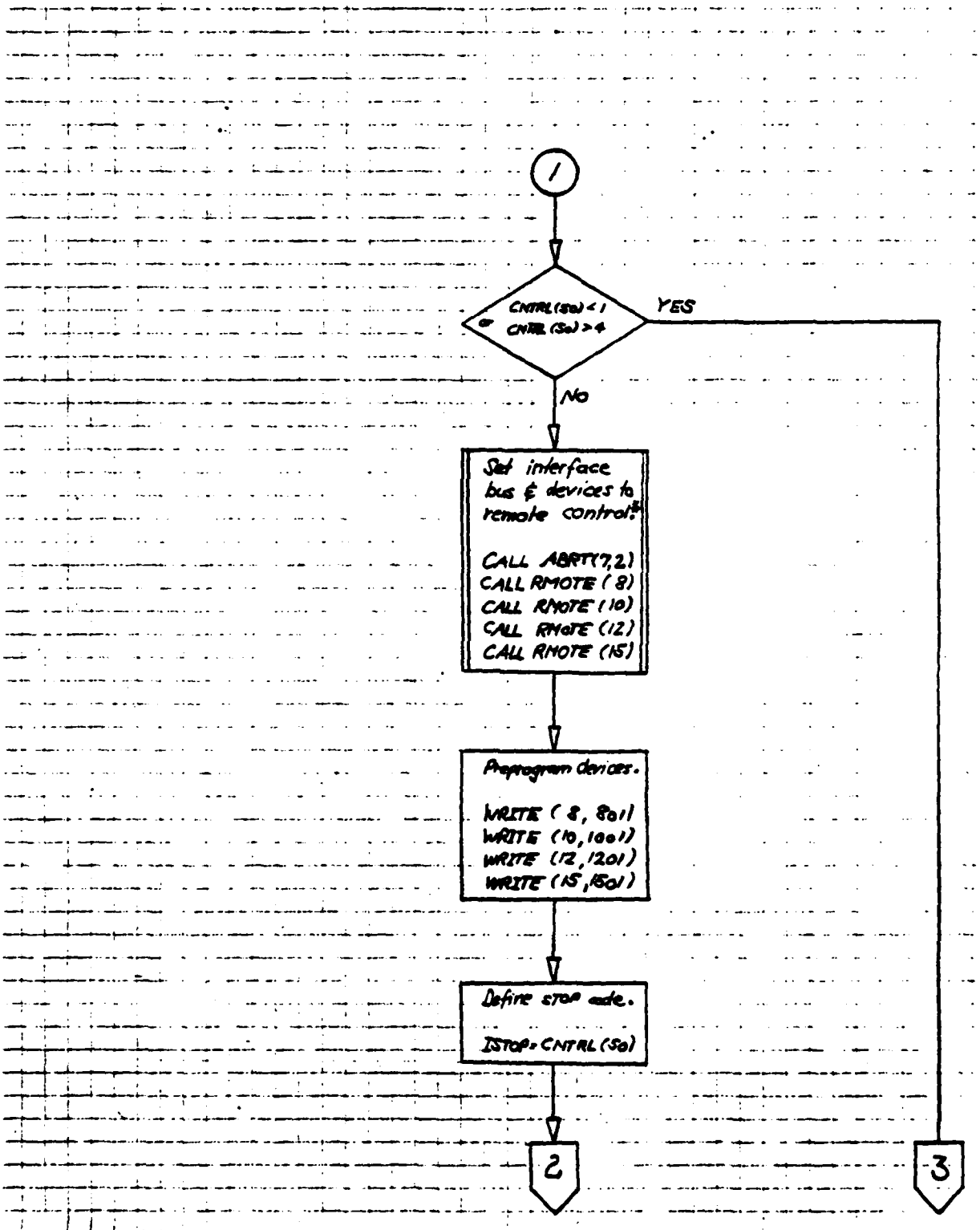
ISECU	integer	security code of data file.
ICR	integer	cartridge reference number, where data file is located.

VARIABLES IN PROGRAM TXC01:

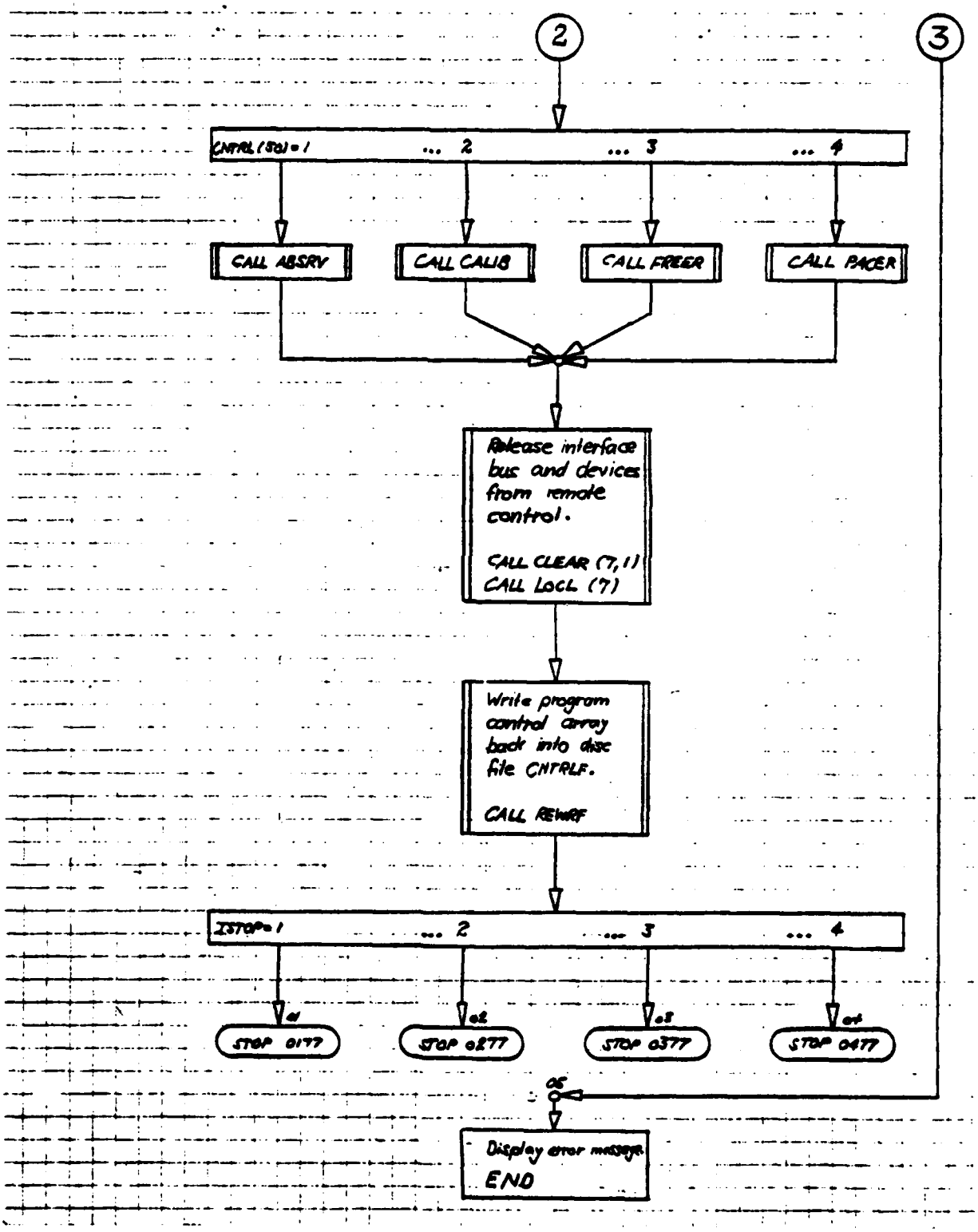
CNTRL (256)	integer	program control array.
NOLF	integer	suppresses line feed.
LI	integer	LU3 of standard input device (terminal).
ISTOP	integer	control variable to select STOP code.
X1	real	dummy variables.
X2	real	
X3	real	

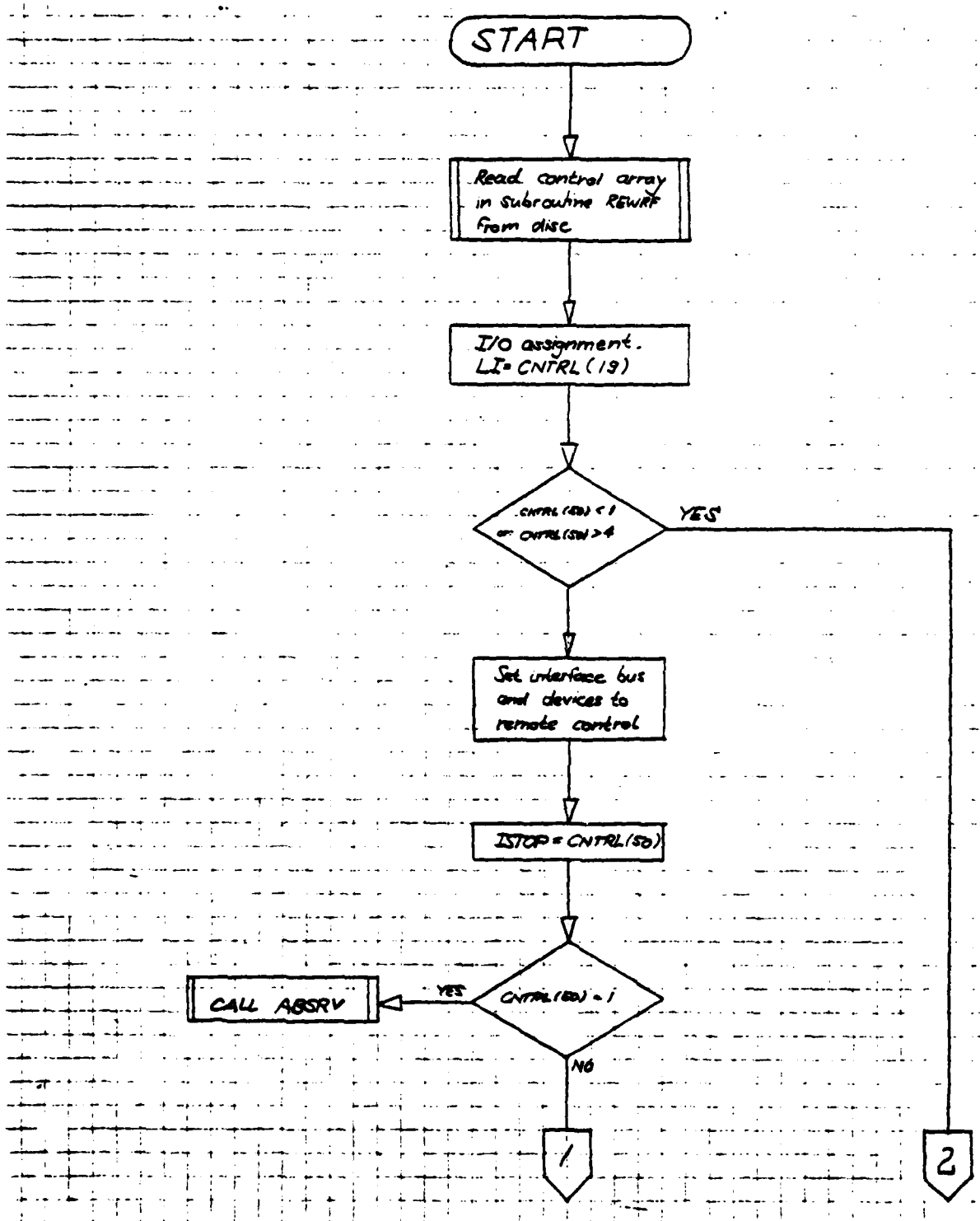
FLOW CHART PROGRAM TXC01 :

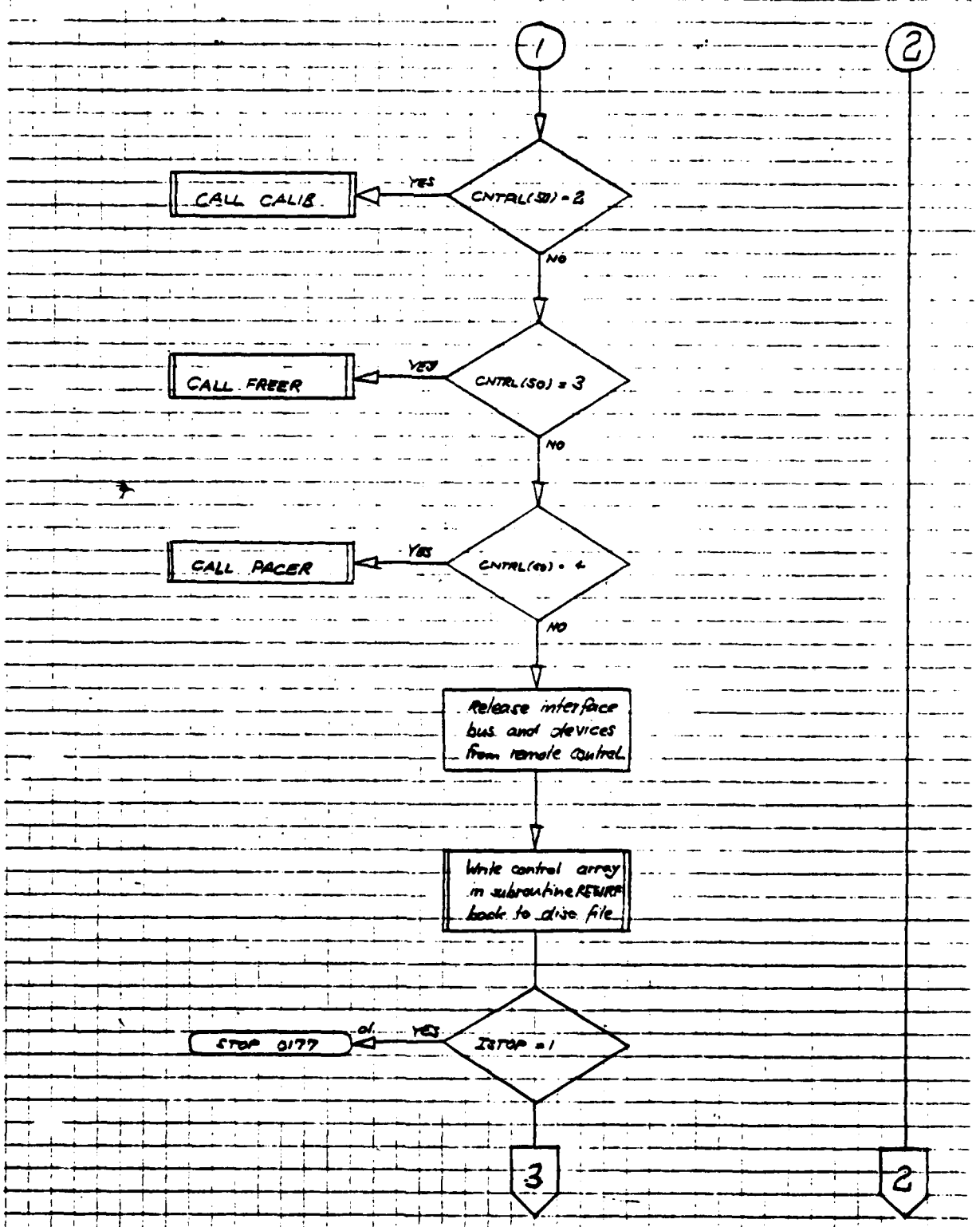


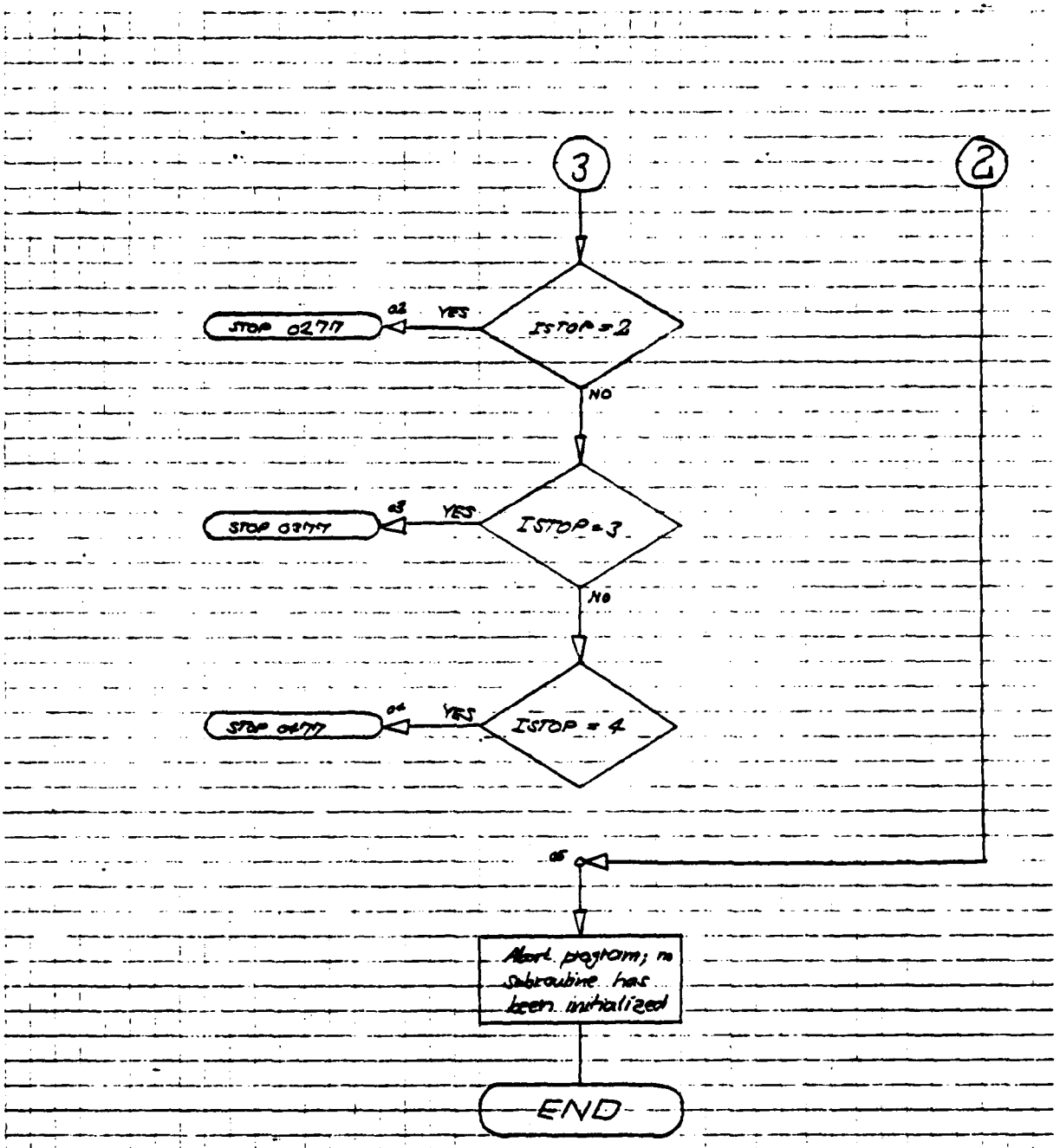


*) LU Assignments:
- 8 Scanner #1
- 10 Digital Voltmeter DVM
- 12 Counter
- 15 Scanner #2
- 7 HP-IB


2
3
CNTRL(50)=1
... 2
... 3
... 4
CALL ABSRV
CALL CALIB
CALL FREER
CALL PACER
Release interface bus and devices from remote control.
CALL CLEAR (7,1)
CALL LOCL (7)
Write program control array back into disc file CNTRLF.
CALL REWRF
ISTOP=1
... 2
... 3
... 4
STOP 0177
STOP 0277
STOP 0377
STOP 0477
05
Display error message
END


START
Read control array in subroutine REWRT From disc
I/O assignment. LI= CNTRL(19)
CNTRL(50) < 1 or CNTRL(50) > 4
YES
Set interface bus and devices to remote control
ISTOP = CNTRL(50)
CNTRL(50) = 1
YES
CALL ABSRV
NO
1
2


1
2
CALL CALIB.
YES
CNTRL(30) = 2
NO
CALL FREER
YES
CNTRL(50) = 3
NO
CALL PACER
YES
CNTRL(60) = 4
NO
Release interface bus and devices from remote control
Write control array in subroutine RETURN back to disc file
STOP 0177
YES
ISTOP = 1
3
2


3
2
ISTOP = 2
YES
02
STOP 02777
NO
ISTOP = 3
YES
03
STOP 03777
NO
ISTOP = 4
YES
04
STOP 04777
05
Abort program; no Subroutine has been initialized
END

4.2. SUBROUTINE ABSRV:

PURPOSE: Acquisition of high speed data from the 1-stage axial compressor using miniaturized probes equipped with KULITE semiconductor pressure transducers.

ARGUMENTS: None

EXTERNALS: CALIB, TIME, REAT, PURGE, OPEN, WRITE, POSNT, CLOSE, SCANR, PACER.

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed explanation refer to the TXCOl description.

MNENOMIC ABBREVIATIONS:

RE ... Repeat data acquisition of this yaw position.

NE ... Proceed to next yaw position.

EN ... End data acquisition at various yaw positions.

UP ... Update position readings of probes prior to data taking.

TA ... Initialisation command to take data.

PU ... Allow purge of an existing data file.

ERROR MESSAGES: If the number of yaw positions exceeds the previously defined number, the program terminates the subroutine correctly (in order to save the already acquired data) and displays an error message (FORMAT 118). The total # of possible yaw positions is input prior to creating the raw data file, so that latter can be created at the desired length.

PROCEDURE: For more detailed information, study the flow chart. After having read the accounting data, assigned the I/O references and preset the raw data array, ABSRV asks the operator, whether the 'A'-'B'- probe system has been calibrated on line. If the answer is NO, ABSRV calls the subroutine CALIB, which controls the calibration. Then the calibration results are entered and the operator is asked to input the number of different yaw positions. Based on this information a raw data file of the appropriate length will be created and positioned. If the file with the automatically determined name already exists, the operator either allows overwriting the existing file (Input : PU) or renames the current data file (Input : any alphabetic character other than T). Prior to taking data the position of the probes is scanned and displayed. This control loop can be repeated by keying UP. Inputting TA initializes the data acquisition by subroutine PACER. Upon completion of the scan the operator can repeat this scan (Input : RE), proceed to the next point (Input : NE) with a different yaw position of both 'A' and 'B' probe. If the operator accidentally has decided to proceed to a probe position beyond the previously specified number, ABSRV displays an error message and terminates the subroutine correctly, i.e. saves the data in file, closes the file and writes the accounting data back into the control array.

DATA FILE: For more detailed information, study the following flow chart. The default file name is Tlrrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).

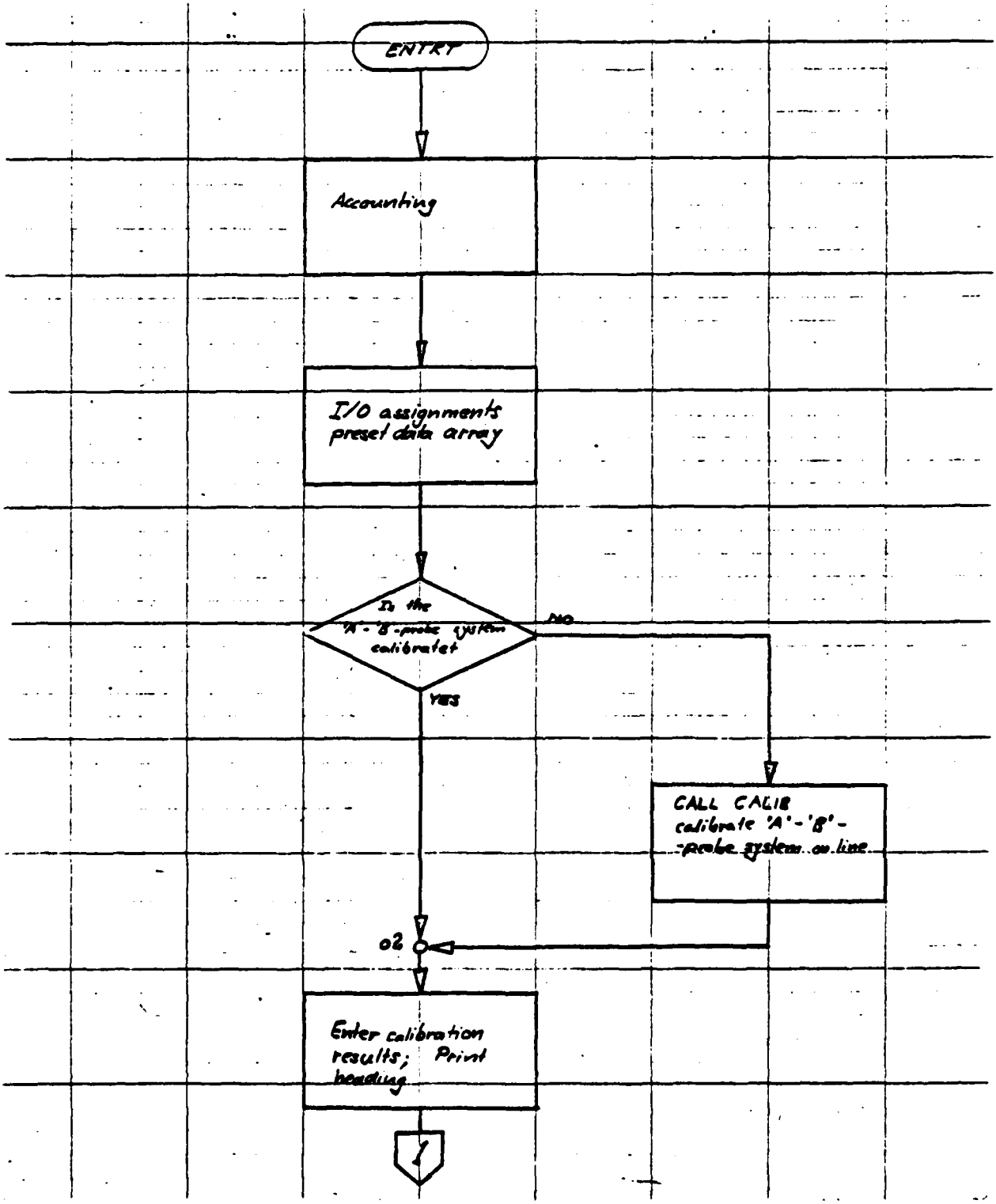
VARIABLES:

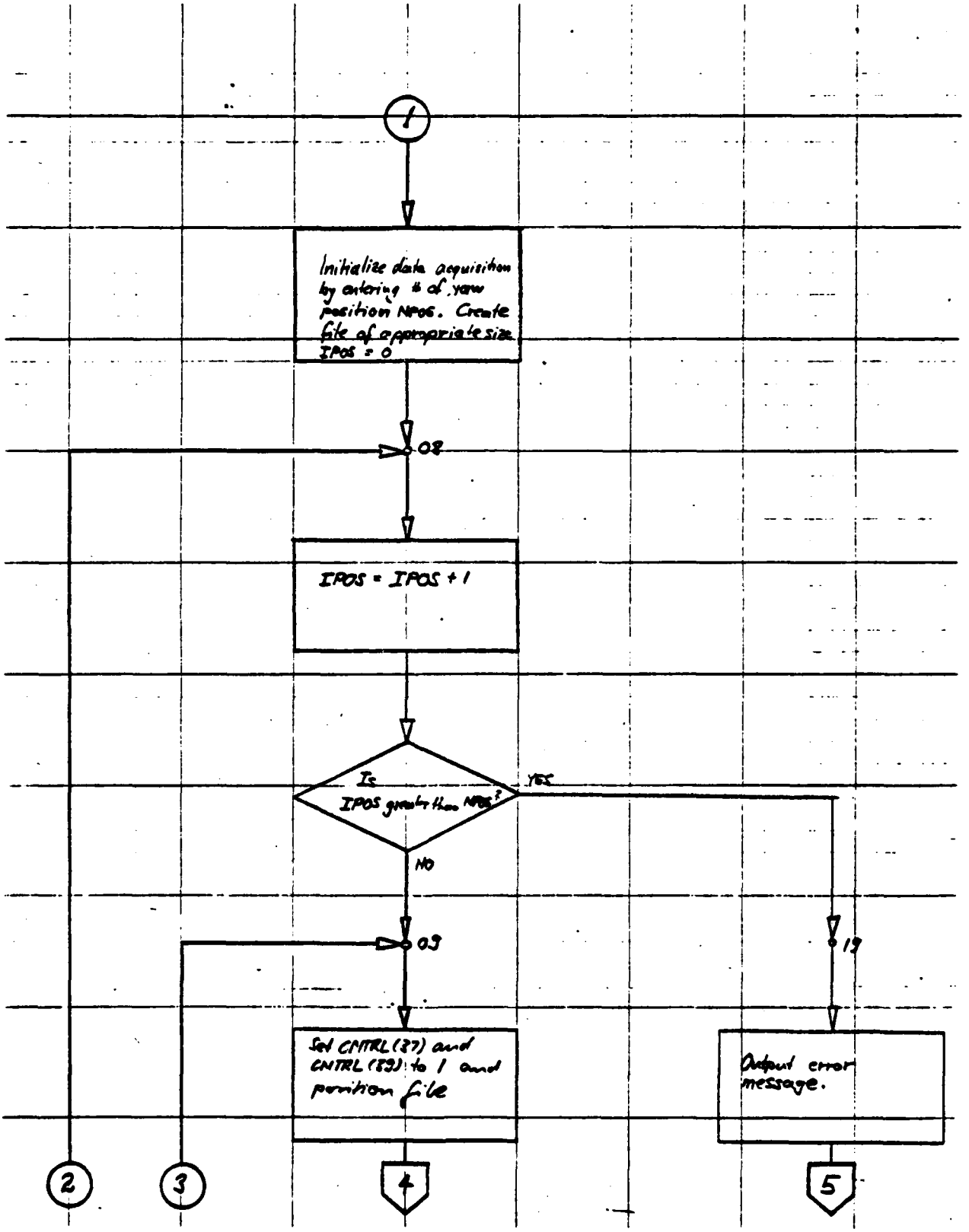
IBUF (1664)	integer	buffer array
CNTRL (256)	integer	control array
IDCB (144)	integer	data control block, used for FMP calls
IFILE (3)	integer	array to contain current file name calls
ISIZE (2)	integer	specifies # of records and record length
ISECU	integer	security code of data file
JSECU	integer	ASCll-converted security code
ICR	integer	cartridge reference number, when data file is located
JCR	integer	ASCII converted cartridge reference number
POS (7)	real	array to contain probe positions
RBUF (62)	real	data array, set equivalent to IBUF
NOLE	integer	suppresses line feed
NOCR (2)	integer	suppresses line feed and carriage return
ICLR (3)	integer	clear line above cursor
IDCBS	integer	length of data control block IDCB
IPAGE	integer	count of current page
IDOC	integer	count of current program run
IDOCF	integer	count of current data file sequential #
IL	integer	number of words to be transferred in FMP calls

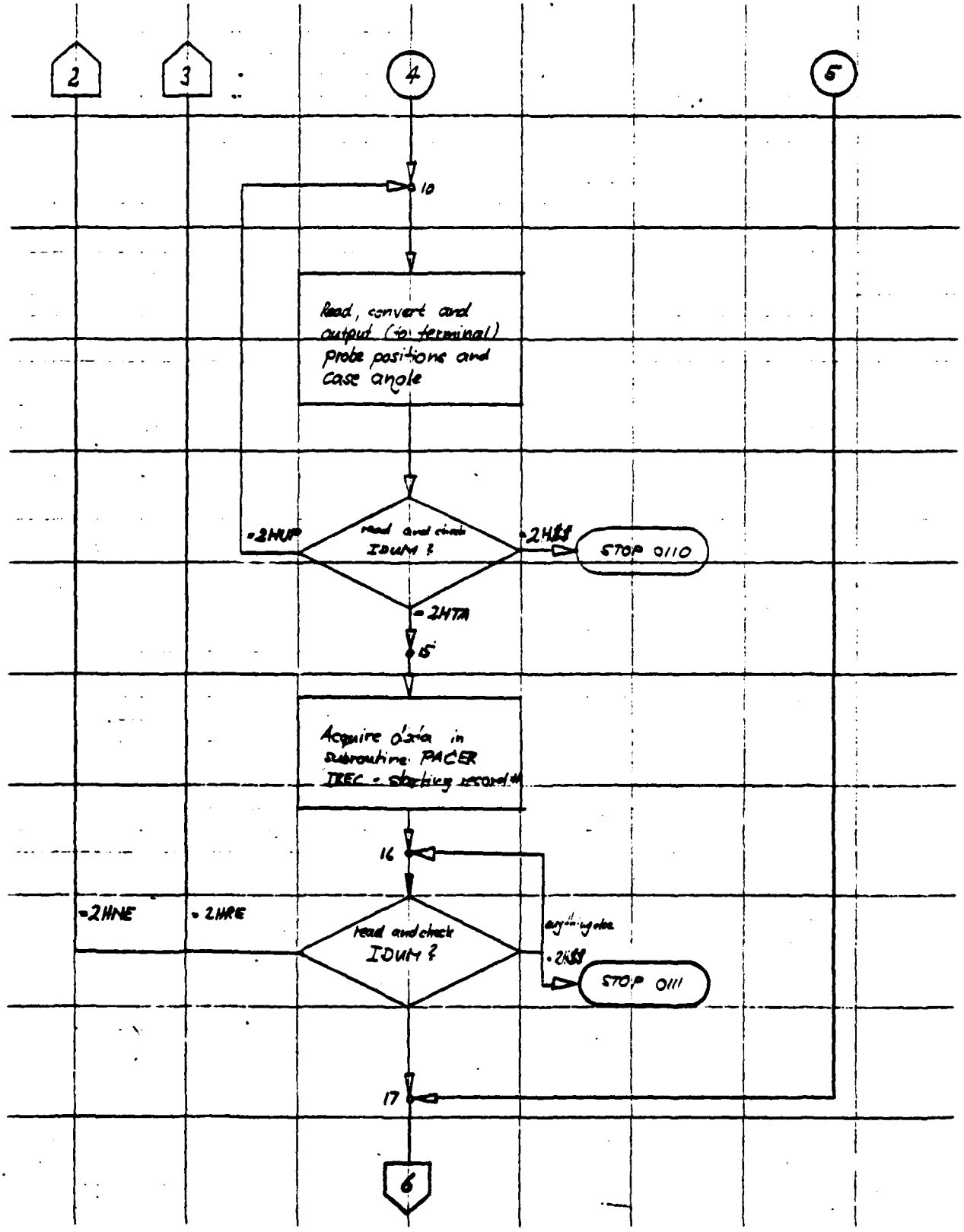
ITYPE	integer	type of data file
IFRST	integer	standard for the first two characters of file name
LI	integer	LU3 of standard input device (terminal)
LO	integer	LU# of standard output device (line position)
LS1	integer	LU# of scanner #1
LS2	integer	LU# of scanner #2
ICAL	integer	decision parameter
IDUM	integer	decision variable
SLOPEA	real	slope of linear curve fit for A probe calibration
SECONA	real	intercept of linear curve fit for A probe calibration
SLOPEB	real	slope of linear curve fit for B probe calibration
SECONB	real	intercept of linear curve fit for B probe calibration
AVRGEA	real	average voltage A probe, when aligned to flow
AVRGEB	real	average voltage B probe, when aligned to flow
PBARO	real	barometric pressure
NPOS	integer	number of different yaw positions 'A'-'B' survey
IERR	integer	error flag (FMP package)

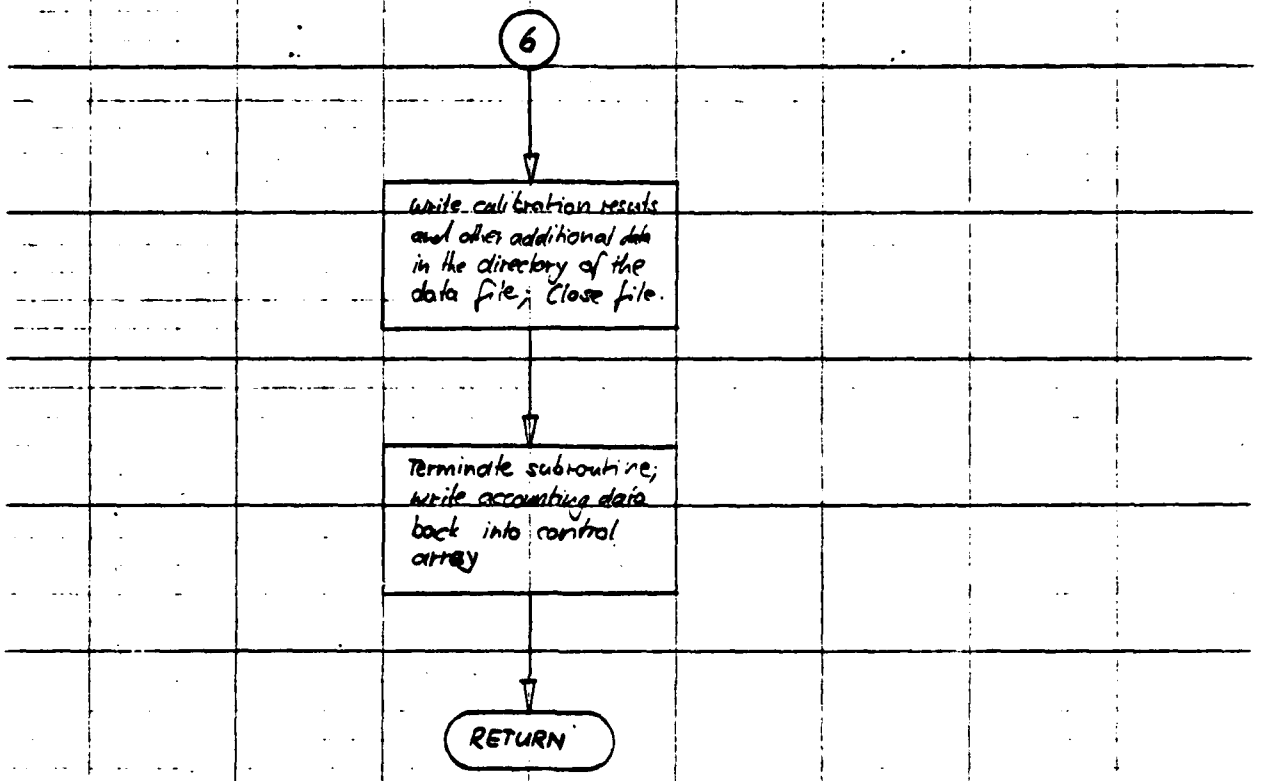
IMON	integer	ASCII converted month of the year
IDAY	integer	ASCII converted day of the month
IHOUR	integer	ASCII converted hour of the day (24 hr clock)
IMIN	integer	ASCII converted minute of the hour
IYEAR	integer	ASCII converted last two digits of current year
IFRST	integer	temporary buffer variable
NEW	integer	scratch variable for change of file name
IPOS	integer	current yaw position count
IREC	integer	record positioning variable

FLOW CHART SUBROUTINE ABSRV




1
Initialize data acquisition by entering # of yaw position NPOS. Create file of appropriate size. IPOS = 0
02
IPOS = IPOS + 1
Is IPOS greater than NPOS?
YES
NO
03
19
Set CNTRL(37) and CNTRL(39) to 1 and position file
Output error message.
2
3
4
5


2
3
4
5
10
Read, convert and output (to terminal) probe positions and case angle
read and check IDUM ?
=2HUP
=2HTA
STOP 0110
15
Acquire data in subroutine PACER
IREC = Starting record #
16
=2HNE
=2HRE
read and check IDUM ?
STOP 0111
17
6


6
write calibration results and other additional data in the directory of the data file; Close file.
Terminate subroutine; write accounting data back into control array
RETURN

4.3. SUBROUTINE CALIB

PURPOSE: Control the on-line calibration for the A-B- probe system. This includes data acquisition and storage as well as approximating the calibration results.

ARGUMENTS: None

EXTERNALS: TIME, FREER, PACER, CURVE

COMMON BLOCKS: CONTR. For detailed explanation refer to the TXCO1 description.

MNEMONIC ABBREVIATIONS:

RE ... Repeat this point

EN ... End the on-line probe calibrations

ERROR MESSAGES: If no calibration is performed, the subroutine outputs a warning (FORMAT 108) and terminates; this can happen, if at the first decision to be made the operator inputs EN.
If less than two points with different reference pressures are taken, the subroutine outputs an error message (FORMAT log) and terminates.
Both messages, if studied carefully, tell the operator how to avoid mistakes.

PROCEDURE: For more detailed information, study the flow chart. After having read the accounting data and assigned the I/O references, CALIB asks the operator to input a

number (which, when the program was debugged, was the digital multimeter read-out displaying the analog voltage of either 'A' or 'B' probe). This input initializes the data acquisition at the first reference pressure. Then the program reminds the operator to switch the pacer to free run mode. The operator responds by pressing the return key and the program calls subroutine FREER. Average voltage from both 'A' and 'B' probe, together with the KULITE reference pressure are written into the arrays AVOLT, BVOLT and RPRES, respectively. The operator then decides whether to repeat the measurements at this reference pressure (Input : RE), end the calibration (Input : EN) or proceed to the next point (Input : any numerical value). If the calibration is to be terminated, the operator is reminded to switch the pacer to paced run mode and, with the reference pressure unchanged, a paced scan is taken from both 'A' and 'B' probe (using PACER). Then subroutine CURVE computes an average linear curve fit through the data points (AVOLT vs. RPRES and BVOLT vs. RPRES respectively). In both cases slope and intercept are printed. Note, that the intercept is meaningless, but required in subroutine CURVE, which uses a least squares algorithm. CALIB then terminates and writes the accounting data back into the control array.

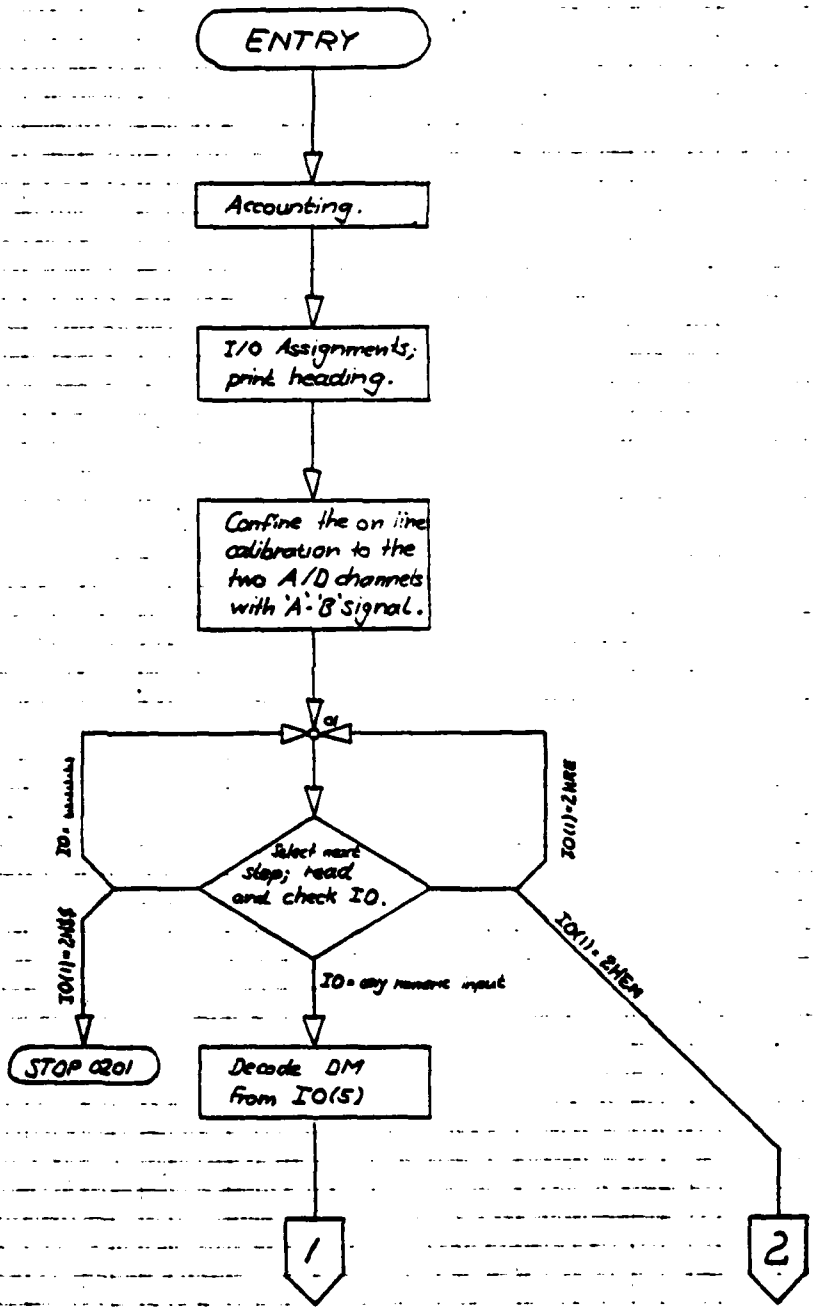
DATA FILE: Handled by subroutines FREER (Section 4.4) and PACER (Section 4.5).

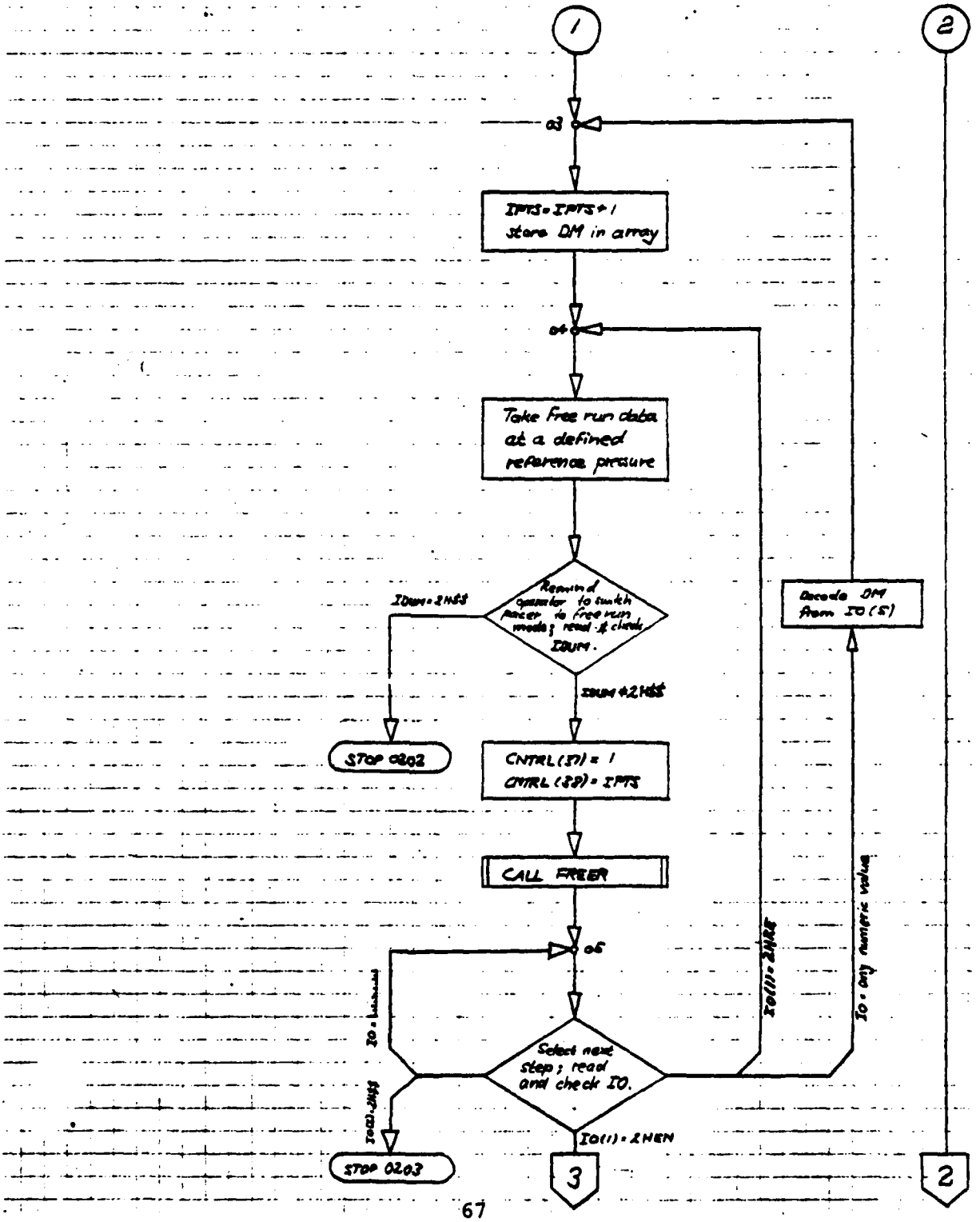
VARIABLES:

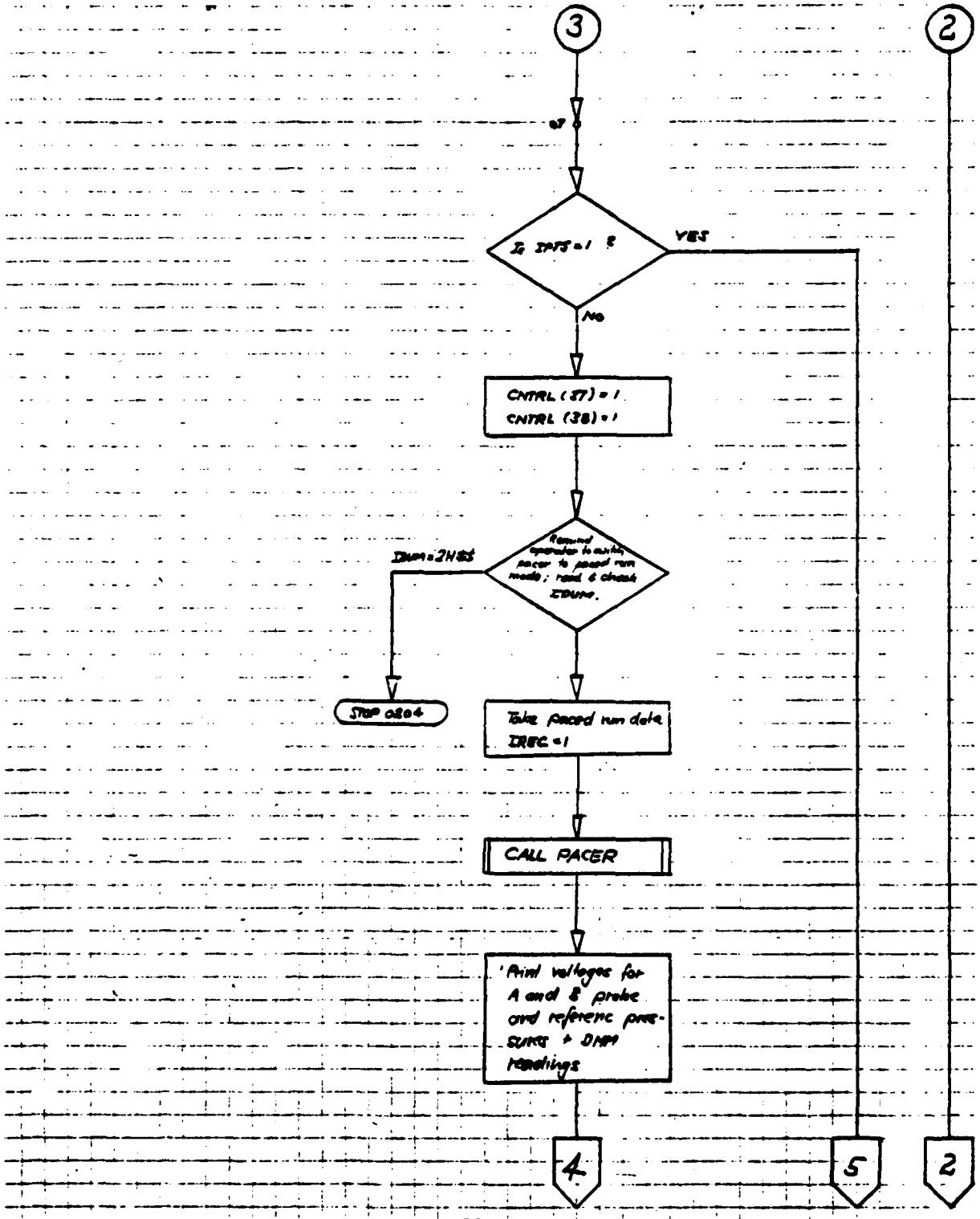
CNTRL (256)	integer	control array
AVOLT (10)	real	array to contain average voltages from A probe
BVOLT (10)	real	array to contain average voltages from B probe
RPRES (10)	real	array to contain reference pressures
DMM (10)	real	array to contain additional data (e.g.: DMM read outs)
NOLF	integer	suppresses line Feed
NOCR (2)	integer	suppresses line Feed and carriage return
ICLR (3)	integer	clears line above cursor
ITIME (5)	integer	array to contain ASCII converted date and time
IO (5)	integer	scratch array
IPAGE	integer	count of current page
LI	integer	LU# of standard input device (terminal)
LO	integer	LU# of standard output device (line printer)
IMON	integer	ASCII converted month of the year
IDAY	integer	ASCII converted day of the month
IYEAR	integer	ASCII converted last two digits
IHOUR	integer	ASCII converted hour of the day (24 hr clock)

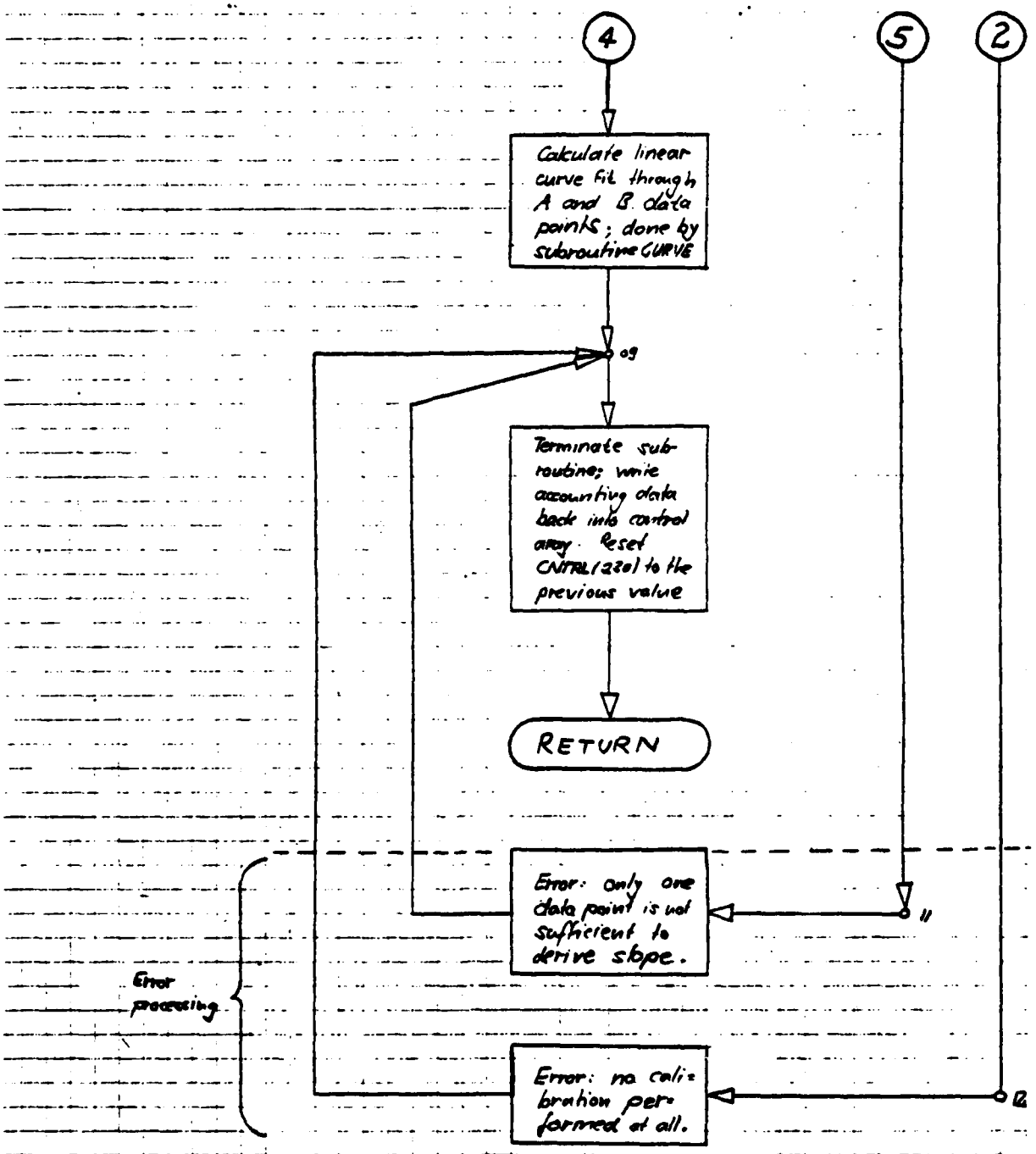
IMIN	integer	ASCII converted minute of the hour
IPTS	integer	variable count total # of calibration positions
IPREV	integer	variable temporary stores contents of CNTRL (219)
IDUM	integer	scratch variable
DM	real	variable used to decode value from array IO
IREC	integer	starting record for paced run-data array
SLOPE	real	slope of linear curve fits
SECON	real	intercept of linear curve fit (as from CURVE)

FLOW CHART SUBROUTINE CALIB




1
2
a3
IPTS = IPTS + 1
Store DM in array
a4
Take Free run data at a defined reference pressure
Remind operator to switch pacer to free run mode; read & check IDUM.
IDUM = 2H55
IDUM = 2HAA
STOP 0202
CNTRL(37) = 1
CNTRL(38) = IPTS
CALL FREER
a6
Select next step; read and check IO.
IO = 2H55
IO(1) = 2HEN
STOP 0203
3
IO(1) = 2HER
IO = any numeric value
Decode DM from IO(5)
2


3
2
Is IDFS = 1 ?
YES
No
CNTRL (37) = 1
CNTRL (38) = 1
IDum = 2H65
STOP 0804
Take paced run data
IREC = 1
CALL PACER
Print voltages for A and B probe and reference pressures + DMM readings
4
5
2


4
5
2
Calculate linear curve fit through A and B data points; done by subroutine CURVE
09
Terminate subroutine; write accounting data back into control array. Reset CNTRL(220) to the previous value
RETURN
Error: only one data point is not sufficient to derive slope.
11
Error processing
Error: no calibration performed at all.
12

4.4. SUBROUTINE FREER:

PURPOSE: Control data acquisition from HP 5610 A/D converter, store data in file and documentation, perform calculation of average voltage.

ARGUMENTS: AVOLT, BVOLT, PREFR

AVOLT	real	average voltage from 'A'- probe, based on NRPT3 points
BVOLT	real	average voltage from 'B'- probe, based on NRPT3 points
PREFR	real	reference pressure for KULITE transducers (raw data format)

EXTERNALS: TIME, ICON, SCANR, EXEC, ABREG, CREAT, OPEN, PURGE, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed description refer to the TXC01 description

MNEMONIC ABBREVIATIONS:

PU ... Allow purge of an existing data file

ERROR MESSAGES: If the EXEC call to read the voltages from the A/D converter is not executed correctly, an error return occurs as follows:

```
                                  ┌──▷ set the no-abort bit.
                                  ^
                            ┌─────┘ └─────┐
      CALL EXEC (1 + 100000B, 20, IBUF(1), NRPT2, ICHNL, 4)
      GO TO 11 ◁── error return:  perform error processing!
10    GO TO 12 ◁── good return:  proceed!
11    CALL ABREG (IA,IB) ◁── look, what's in the registers.
      GO TO 21
      .
      .
      .
C
C
C     Error returns ....
C
C
21   WRITE (6, 901) NRPT2, ICHNL   }──▷ output error message,
     WRITE (6, 902) IA, IB         }    then terminate
     GO TO 20                           subroutine
```

<u>PROCEDURE</u>: For more detailed information, study the flow chart. The subroutine reads the accounting data from the control array and defines FMP parameters (FMP : File Management Package, manipulates disc Files). Next the I/O references are assigned and all words of the raw data array are preset to be 177777B. If CNTRL (37) is set to 1, the heading for the Free Run documentation page is printed. If CNTRL (38) is set to one, a key to the printout is printed. Then the data acquisition loop starts and executes NRPT1 times (NRPT1 = CNTRL (230) = number of KULITE signals to be acquired; maximum is 16). Should the sequential number for the data file name become greater than 99, the first two characters of the file name are changed from T2 (default) to S2 and the count is reset to zero. Additional data is acquired and the probe positions are read and written into the variable IOX1M. Prior to the data acquisition all unused elements of the data array are set to zero. Utilizing the EXEC call NRP2 measurements are performed and the A/D digital

output is written into array IBUF, starting at address of word IBUF (1). ICHNL specifies the selected A/D analog input channel. The 4 in the parameter list causes the A/D converter to dump data into the CPU as fast as possible via DMA (Direct Memory Access). If an error occurs, its reason is enquired (see preceding segment ERROR MESSAGES). To calculate the average voltage, all words of IBUF must be anded with IMASK, because bits 0 through 5 are used to control the data transfer.

```
          bit  15   12     9     6     3     0
IBUF(J2)       1 1 1 1 0 0 1 0 1 1 1 0 0 0 1 0  =  171342B
IMASK          1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0  =  177700B = -64
IBUF(J2) = IAND (IBUF(J2), IMASK)
IBUF(J2) =     1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 0  =  171300B = -3392
```

To derive the voltage, IBUF(J2) must first be divided by the maximum value which can be transferred by a 16-bit word when the bits 0 through 5 do not contain data; this word is

```
               0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0  =  077700B = 32704
```

This bit configuration corresponds to the full scale voltage (FSVLT) of +1 Volt. When no bit is set, the voltage is 0 (ensured by calibration). Thus if the integer, IBUF(J2) is divided by 32 704 and multiplied by unity the voltage is obtained. Since the bits 0 through 5 are not used for data, the maximum voltage resolution of the A/D converter is

$$R = \frac{100B}{77700B} * FSVLT = \frac{64}{32\ 704} * FSVLT = \frac{1}{511} * FSVLT$$

$$R = .001\ 956\ 947\ Volt, \text{ if } FSVLT = +1.0\ V$$

The voltage associated with the bit configuration

1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 0 = 171300B = -3392

is -.103 718 Volt = $\frac{-3\ 392}{32\ 704}$ * 1.0 Volt. In the Subroutine, however, the division through 32 704 and the multiplication with FSVLT is executed after all the voltages from NRPT2 points are added in order to compute the average voltage. The average voltage then is written into the variable AVOLT or BVOLT, depending on which probe has been selected. The data then are saved in a file. If a file with the automatically determined name already exists, the operator either allows overwriting the existing file (Input : PU) or renames the current data file (Input : any alphabetic character other than T). This is the only interactive manipulation in the subroutine. The data acquisition loop terminates, printing the most important data. Accounting data are written back into the control array and the subroutine returns control to the calling program.

DATA FILE: The data file consist of 13 records with a length of 128 words each, so that 1664 = (128 * 13) words can be stored. The default file name is T2rrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).

VARIABLES:

IBUF (1664)	integer	buffer array for raw data
CNTRL (256)	integer	control array
IDCB (144)	integer	data control block, used for FMB calls

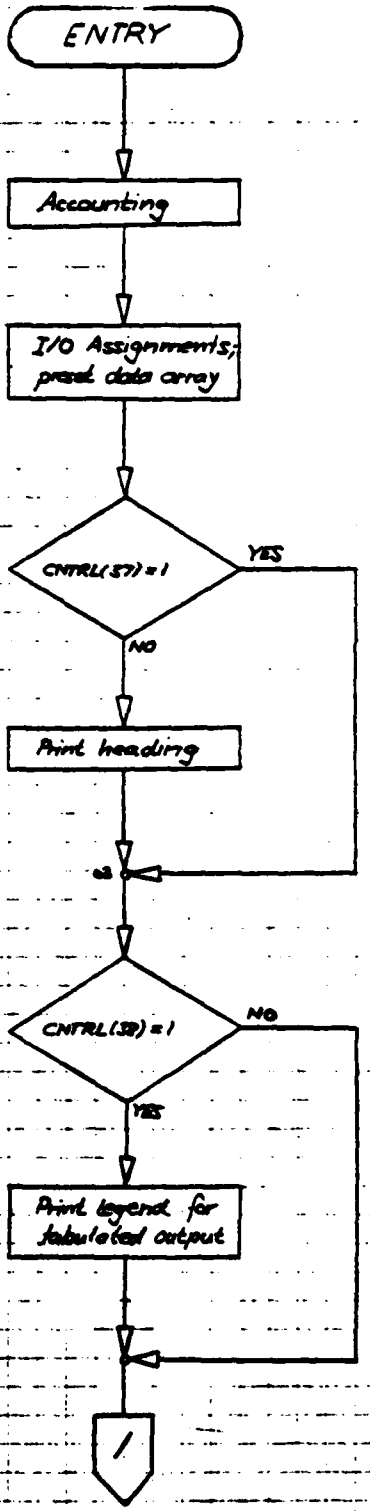
IFILE (3)	integer	array to contain current file name
ISIZE (2)	integer	specifies # of records and record length
ISECU	integer	security code of data file
ICR	integer	cartridge reference number, where data file is located
NOLF	integer	suppresses line feed
NOCR (2)	integer	suppresses line feed and carriage return
ICLR (3)	integer	clear line above cursor
IOXIM (9)	integer	array, where 'A' and 'B' probe positions are written into in ASCII code
FSVLT	real	full scale voltage of A/D converter
IDCBS	integer	length of data control block IDCB
IPAGE	integer	counts of current page
IDOC	integer	counts, how often this subroutine is called
IDOCF	integer	count of current data file sequential #
ITYPE	integer	type of data file
IFRST	integer	standard for the first two characters of file name
ISP	integer	decision variable, used to space the output

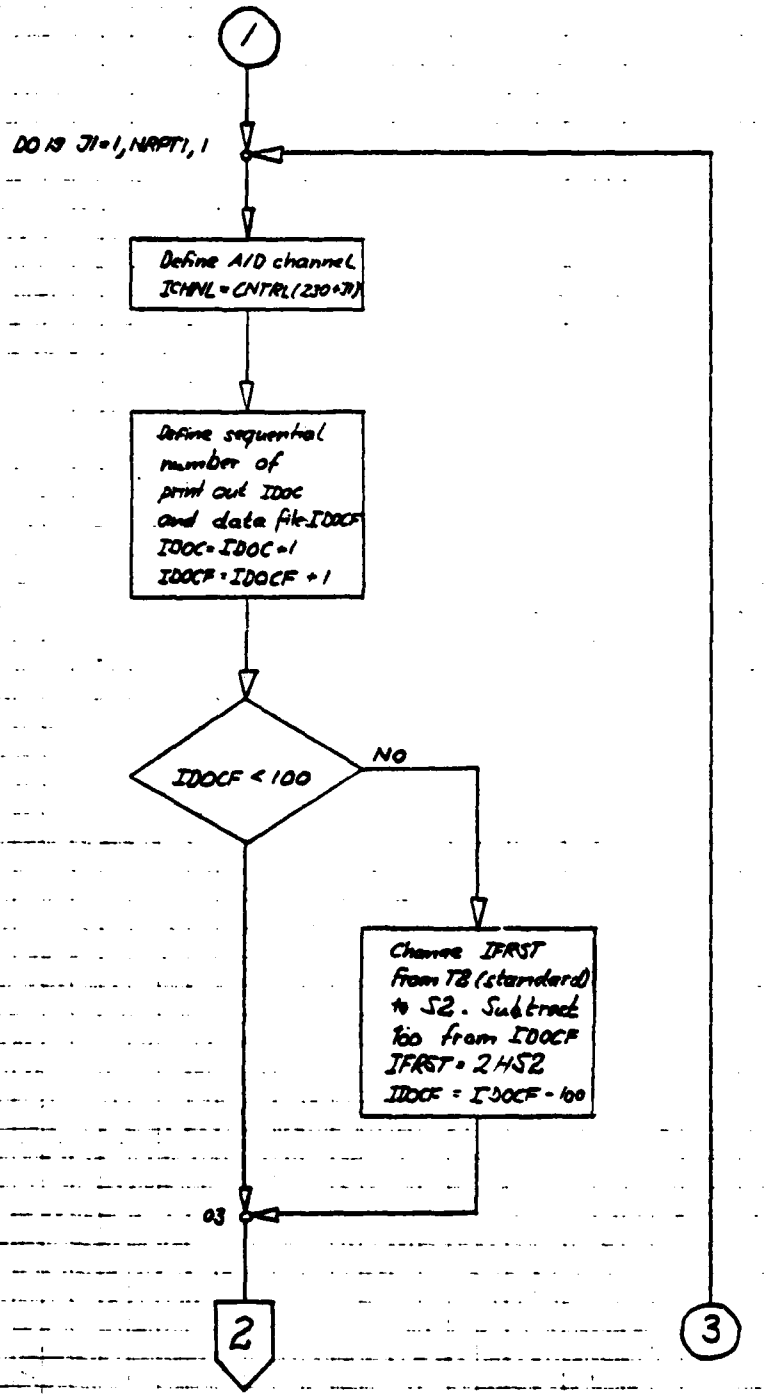
IL	integer	number of words to be transferred in FMP calls
LI	integer	LU# of standard input device (terminal)
LO	integer	LU# of standard output device (line printer)
LS1	integer	LU# of scanner #1
LS2	integer	LU# of scanner #2
ISV1	integer	code # of S/V controller #1
ISV4	integer	code # of S/V controller #2
NRPT1	integer	# of various KULITE signals to be acquired
NRPT2	integer	total # of point, taken from each KULITE signal
NRPT3	integer	= NRPT2+ : DO loop start address
IMASK	integer	masking variable
IW	integer	controls time delay between closing S/V port and reading voltage
IMON	integer	ASCII converted month of the year
IDAY	integer	ASCII converted day of the month
IHOUR	integer	ASCII converted hour of the day (24 hr clock)
IMIN	integer	ASCII converted minute of the hour
IYEAR	integer	ASCII converted last two digits of current year
ICHNL	integer	A/D input channel to be selected
FREQ	real	RPM of the transonic compressor

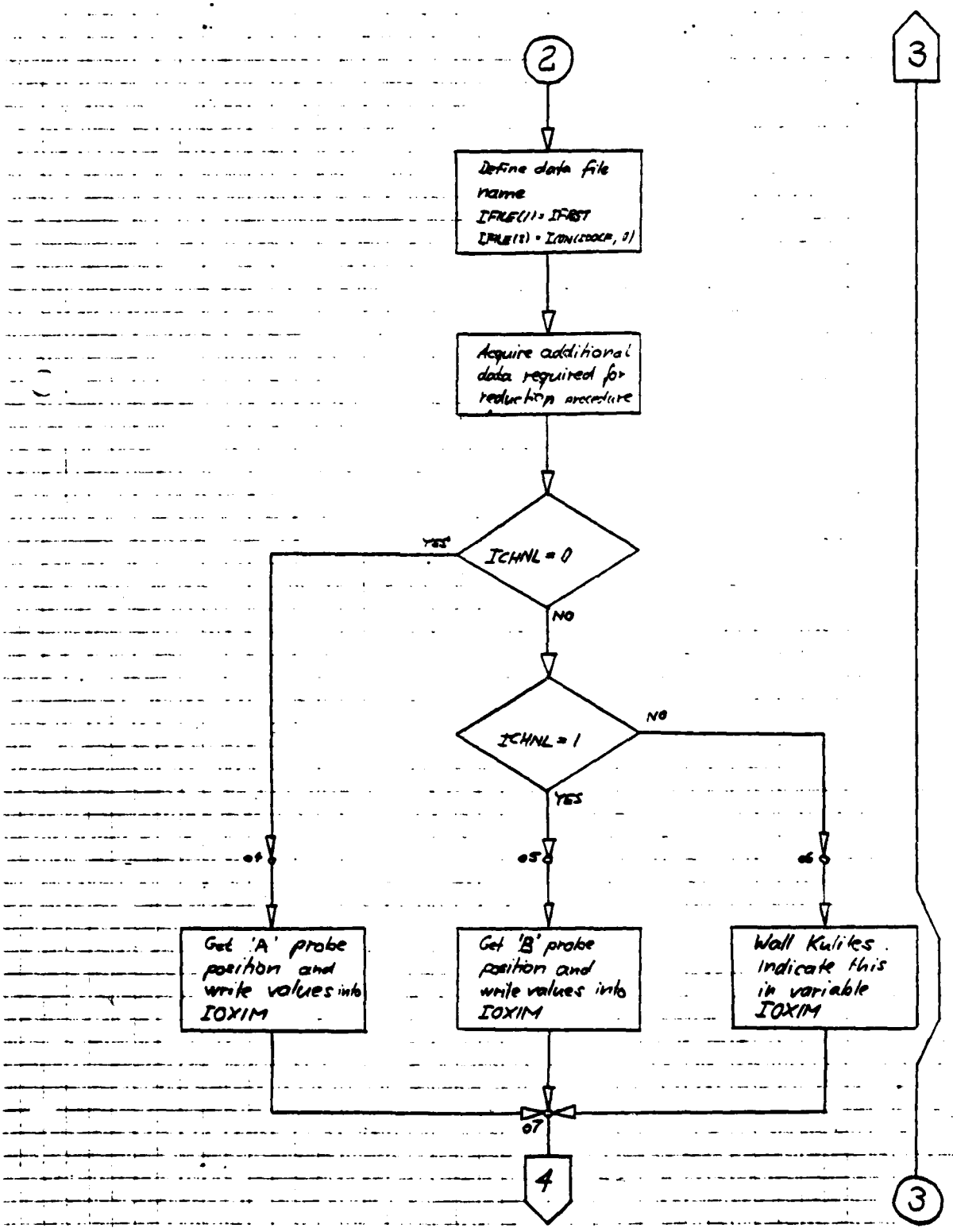
CIM	real	immersion of the combination probe
CYAW	real	yaw angle of the combination probe
PREF	real	KULITE reference pressure
PREFR	real	KULITE reference pressure as returned to the calling routine (either CALIB, ABSRV or TXCO1)
P1	real	pressures P_1, P_{23} & P_4 from calibration probe
P23	real	
P4	real	
E	real	Temperature reading from sensor ahead of rotor (in mV)
DE	real	Differential temperature reading from station ahead of rotor across rotor
XIM	real	Immersion of the KULITE probe
YAW	real	Yaw angle of the KULITE probe
IA	integer	Variable to contain contents of A register
IB	integer	Variable to contain contents of B register
AVRGE	real	KULITE output average voltage after amplification and A/D conversion
AVOLT	real	'A' probe output average voltage
BVOLT	real	'B' probe output average voltage
ISP	integer	control variable to space output
IDUM	integer	decision variable

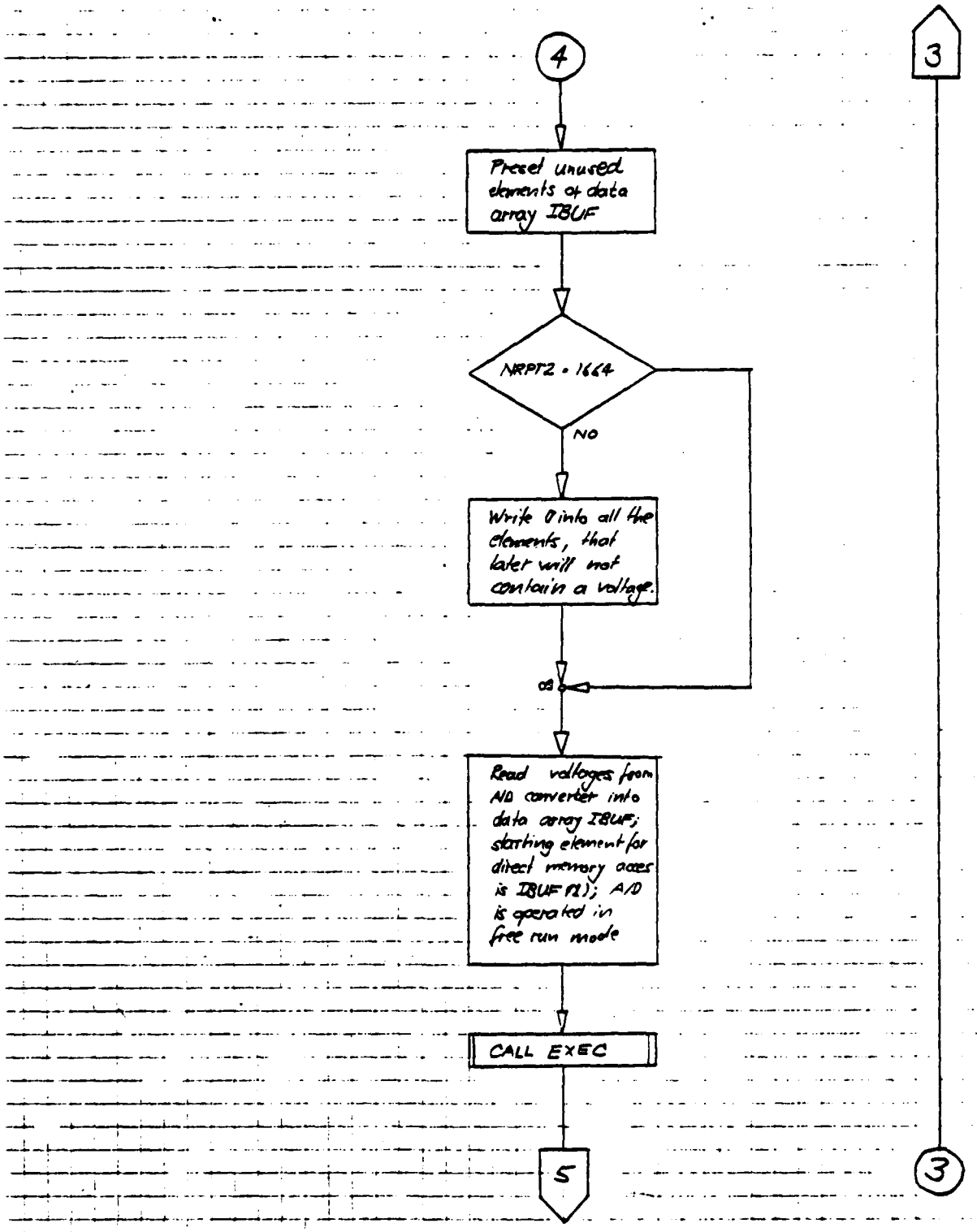
IERR	integer	error flag returned from FMP calls
JSECU	integer	ASCII converted security code
JCR	integer	ASCII converted cartridge reference number
NEW	integer	variable to contain changed first two characters of raw data file name

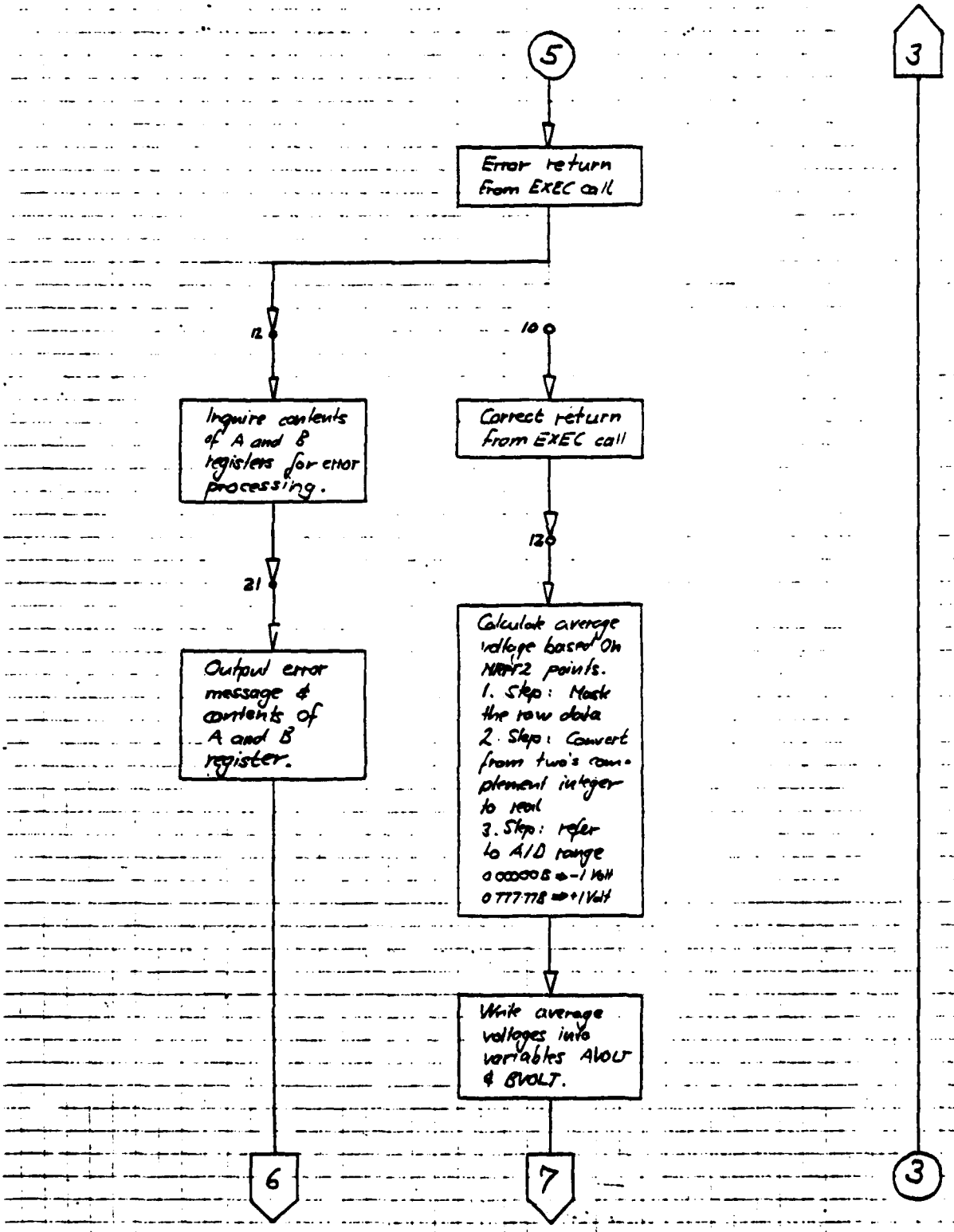
FLOW CHART SUBROUTINE FREER

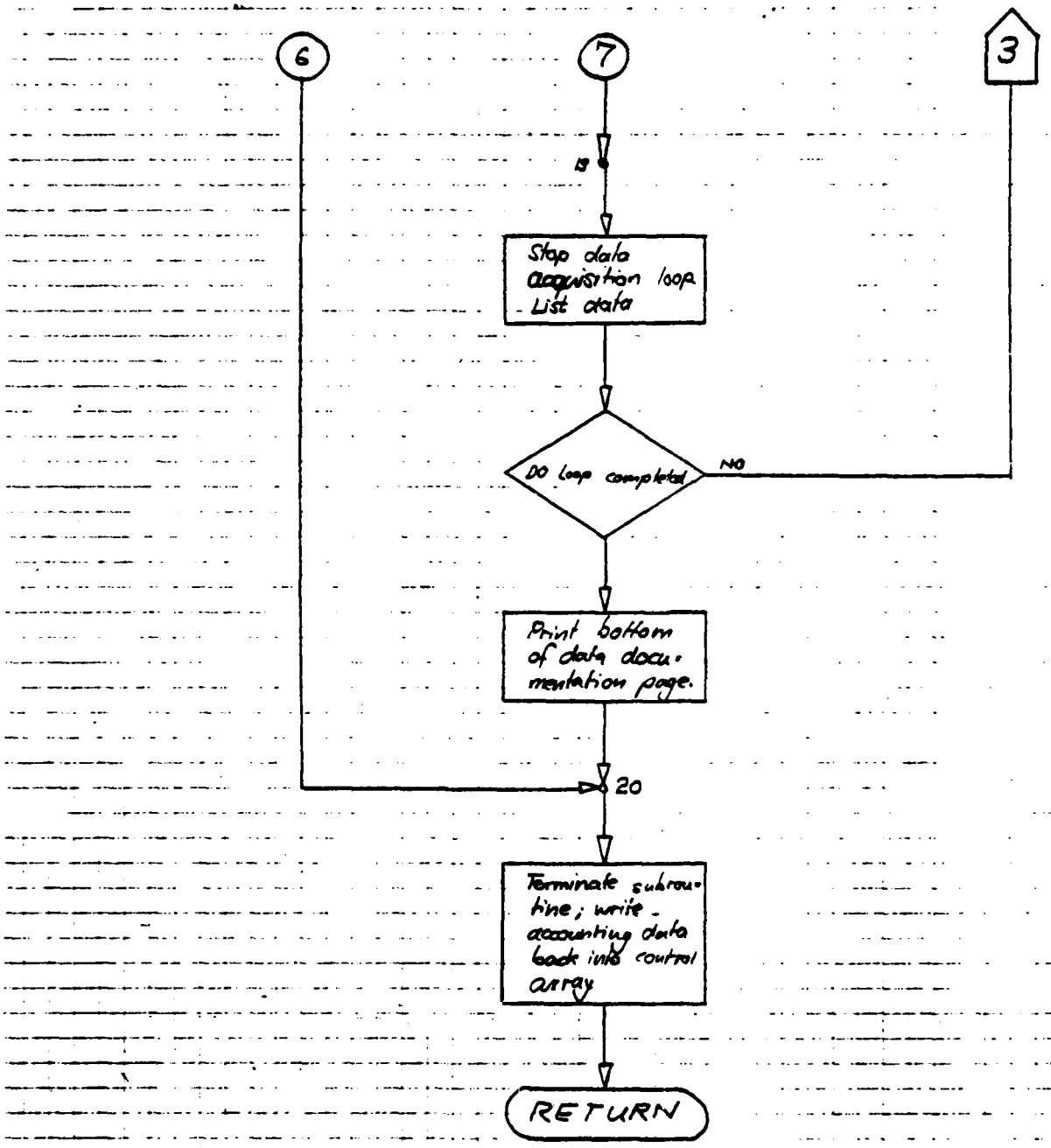



1
DO 19 JI=1,NAPTI,1
Define A/D channel
ICHNL=CNTRL(230+JI)
Define sequential number of print out IDOC and data file-IDOCF
IDOC=IDOC+1
IDOCF=IDOCF+1
IDOCF < 100
NO
Change IFRST from TB (standard) to S2. Subtract 100 from IDOCF
IFRST=2HS2
IDOCF=IDOCF-100
03
2
3


2
3
Define data file name
IFILE(1) = IFIRST
IFILE(2) = ICONCODE, 0)
Acquire additional data required for reduction procedure
ICHNL = 0
YES
NO
ICHNL = 1
NO
YES
05
06
Get 'A' probe position and write values into IOXIM
Get 'B' probe position and write values into IOXIM
Wall Kulites. Indicate this in variable IOXIM
07
4
3


4
3
Preset unused elements of data array IBUF
NRPT2 = 1664
NO
Write 0 into all the elements, that later will not contain a voltage.
Read voltages from A/D converter into data array IBUF; starting element for direct memory acces is IBUF(1); A/D is operated in free run mode
CALL EXEC
5
3


5
3
Error return from EXEC call
12
10
Inquire contents of A and B registers for error processing.
Correct return from EXEC call
21
120
Output error message & contents of A and B register.
Calculate average voltage based on NREP2 points.
1. Step: Mask the raw data
2. Step: Convert from two's complement integer to real
3. Step: refer to A/D range
000000B ⇒ -1 Volt
077777B ⇒ +1 Volt
Write average voltages into variables AVOLT & BVOLT.
6
7
3


6
7
3
IF
Stop data acquisition loop. List data
DO loop completed
NO
Print bottom of data documentation page.
20
Terminate subroutine; write accounting data back into control array
RETURN

4.5. SUBROUTINE PACER:

PURPOSE: Control data acquisition from HP 5640 A/D converter if this device is triggered by the pacer, store data in file and document all steps.

ARGUMENTS: IREC

IREC integer starting record #, where raw KULITE and additional data are written

EXTERNALS: TIME, ICON, SCANR, ACQN, RPACE, PICTR, CREAT, OPEN, PURGE, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed description refer to the TXC01 description.

MNEMONIC ABBREVIATIONS:

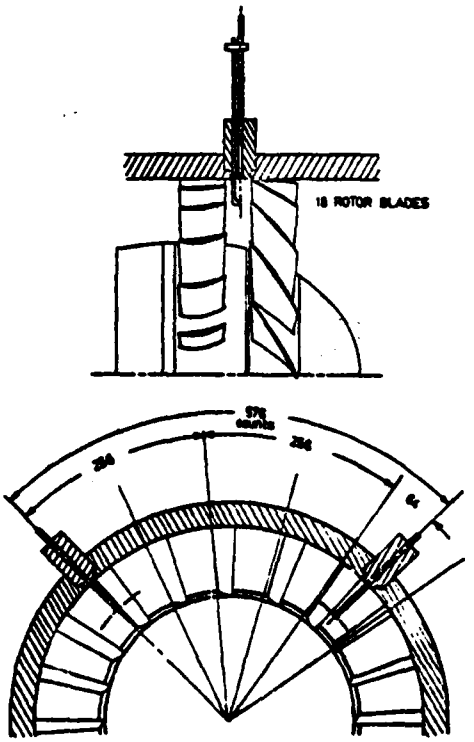
PU ... Allow purge of an existing data file

RE ... Repeat data acquisition

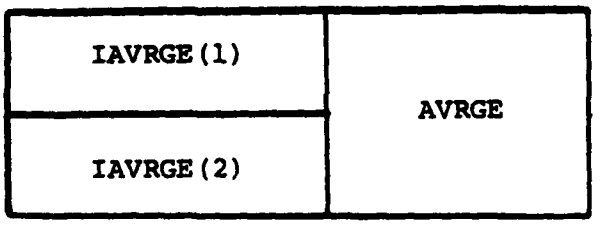
ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart! The subroutine reads the accounting data from the control array and defines FMP parameters (FMP: file management package, manipulates disc files; refer to the HP manuals for more information). Next the I/O references are assigned and the words of the data array are present. If CNTRL (37) is set to 1, the heading for the paced run documentation page is printed. If CNTRL (38) is set to 1, a key to the print

out is printed. Then the data acquisition loop starts and executes NRPT1 times (NRPT1 = CNTRL (230) = number of KULITE signals to be acquired; maximum is 16). If CNTRL (39) is set to 1 (i.e.: subroutine PACER is called from subroutine ABSRV, which takes care of creating/opening, positioning and closing files), the accounting of the data file names is skipped. If otherwise, i.e. CNTRL (39) is not equal to 1, the sequential number for the data file name exceeds 99, the first two characters of the file name are changed from T3 (default) to S3 and the count is reset to zero. Additional data is acquired and the probe positions are read and written into the variable IOX1M. Since the KULITE probes are mounted in physically different positions (the phase angle is 90°, i.e. $2\frac{1}{4}$ times 40°, where 40° is double the rotor inter-blade angle), and the signals must be converted from the same point in the rotor blade wake, the IBLADE for the 'B' probe has to be increased by the appropriate amount, which is 576 (see sketch).



The operator is then informed that the system is ready for the next data scan. Depressing the RETURN key starts the data acquisition. Pacer mode (1 or 2), selected blade pair, increment to step through the 256 blade passage locations and the number of measurement repetitions at each location (i.e. at each IBLADE) are read from the control array. If the pacer is operated in mode 2 (i.e. selects a specific blade pair), the bit 15 is set by adding IADD = 100000B to the start and stop address. Refer to the RPACE description for details concerning how the data acquisition is performed. Not only the voltages, through subroutine RPACE, but also additional data are written into the raw data array. Some of the data are multiplied by 1,000,000 in order to be able to store all valid digits in integer constants and the average voltage AVRBE is set equivalent to the array IAVERGE(2) by an EQUIVALENCE statement.



Date and time are written into the raw data array also. If CNTRL(40) is set to 1, the wave as acquired is displayed on the terminal, which is selected by its logical unit number LA. Refer to the detailed description of subroutine PICTR for further information on how this is achieved; i.e. to use a non-graphics

terminal for plotting. The resolution of the terminal plot is is very limited. The option to display the just-acquired periodic high speed signal is designed to give the operator an opportunity to immediately verify the correctness of the data acquisition. Connecting a lead from KULITE amplifier output to an oscilloscope gives the investigator the chance to check digitized data against original analog data. If an error is encountered, the data scan may be repeated (Input : ... RE). Depressing the RETURN key causes the subroutine to proceed to the next task, the storing of the data. File name, ASCII converted security code and ASCII converted cartridge reference number are written into the raw data array. The raw data file is either created/opened and closed by subroutine PACER (CNTRL (39) is not equal 1) or this subroutine is called from subroutine ABSRV, which already has created/opened and positioned the raw data file and will close it (CNTRL (39) is set to 1). If, in the first mode, the automatically determined file name already exists, the operator either allows overwriting the existing file (Input: PU) or renames the current data file name (Input: any alphabetic character other than T). The starting record number is also written in the data array. If CNTRL (39) is not equal 1, the raw data file is closed and the data acquisition loop stops printing all the additional data on the documentation page. The accounting data are written back into the control array and the subroutine terminates.

DATA FILE: For more detailed information, study the key to the raw data file following this description. The default file name is T3rrss (rr ... ASCII converted run #, ss ... ASCII converted sequential #).

VARIABLES:

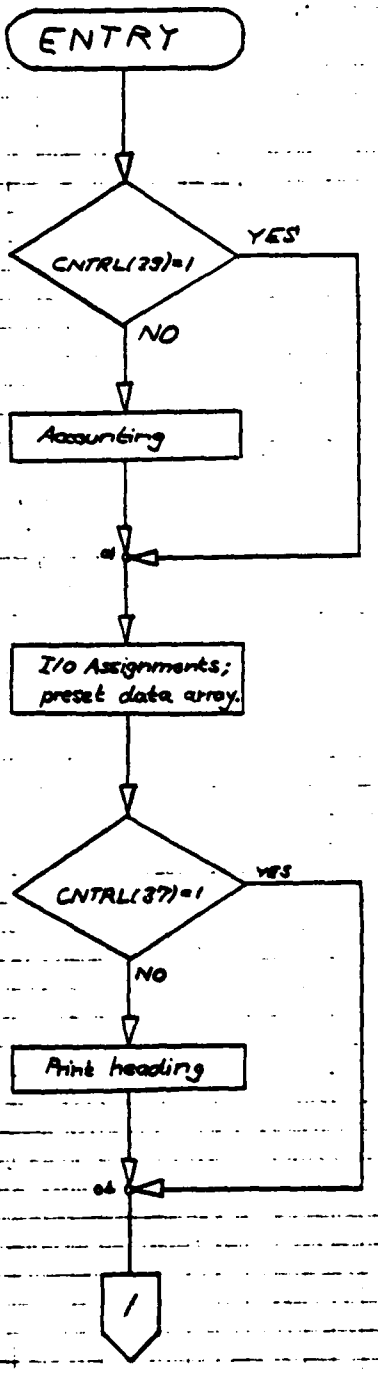
IBUF (1664)	integer	buffer array
CNTRL (256)	integer	control array
IDCB (144)	integer	data control block; used for FMP calls
IFILE (3)	integer	array to contain current file name
ISIZE (2)	integer	specifies # of records and record length in words
ISECU	integer	security code of data file
JSECU	integer	ASCII converted security code
ICR	integer	cartridge reference number, where data file is located
JCR	integer	ASCII converted cartridge reference number
NOLF	integer	suppresses line feed on terminal
NOCR (2)	integer	suppresses line feed and carriage return on terminal
ICLR (3)	integer	clear line above cursor
IBUF1 (384)	integer	raw data array, set equivalent to IBUF
IOXIM (9)	integer	array to contain probe positions in ASCII code

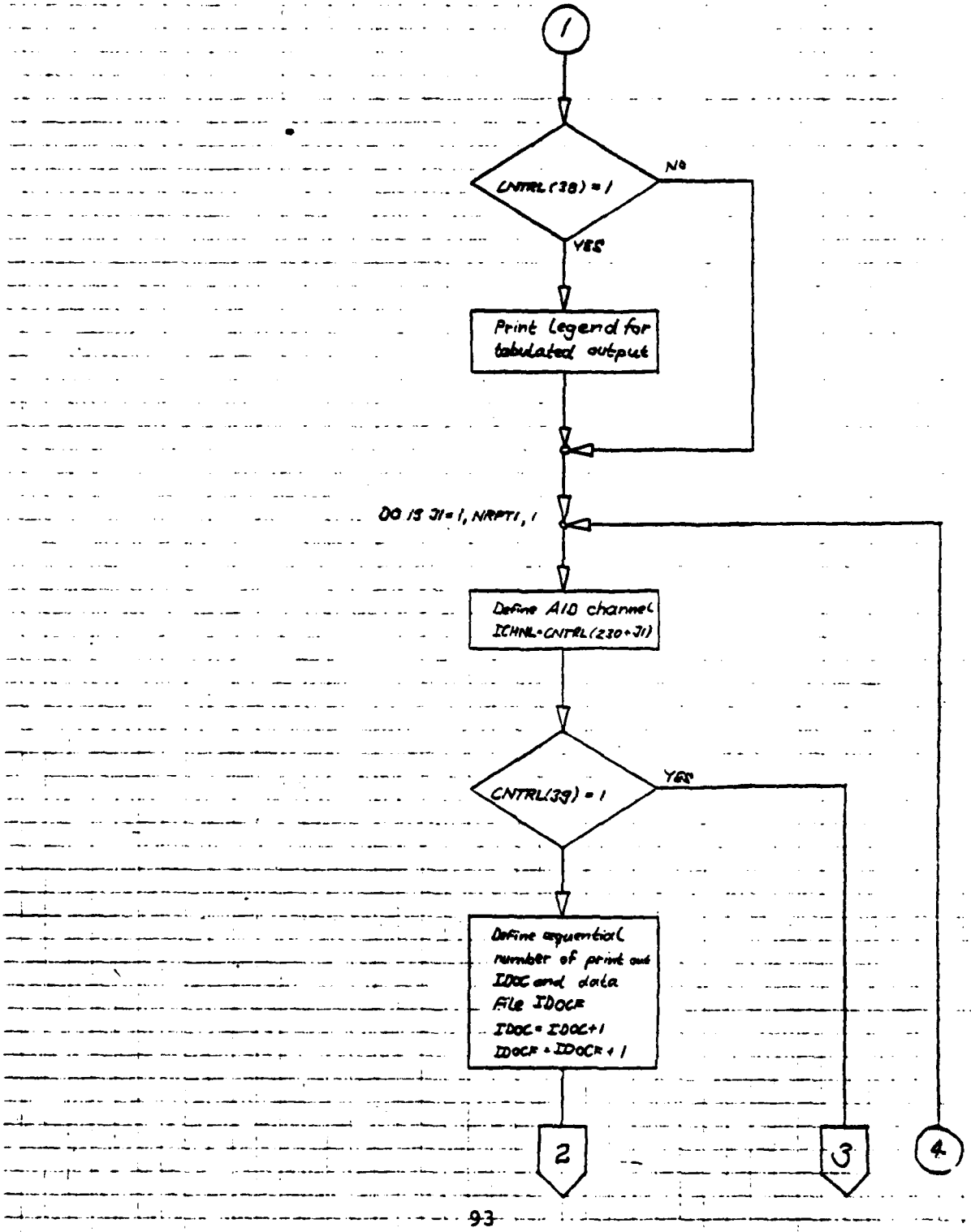
IAVRGE (2)	integer	array to contain average voltage, set equivalent to AVRGE
IDCBS	integer	length of data control block IDCB in words (here : 144)
IPAGE	integer	count of current page
IDOC	integer	count of current program run
IDOCF	integer	count of current data file sequential #
IL	integer	number of words to be transferred in FMP calls
ITYPE	integer	type of data file (here: 1)
IFRST	integer	standard for the first two characters of data file name
ISP	integer	control variable, used to space the output
LI	integer	LU# of standard input device (system console)
LO	integer	LU# of standard output device (line printer)
LA	integer	LU# of auxiliary output device (auxiliary terminal)
LS1	integer	LU# of scanner 1
LS2	integer	LU# of scanner 2
ISV1	integer	number of S/V controller 1
ISV4	integer	number of S/V controller 2
NRPT1	integer	number of KULITE measurements ('A', 'B', case KULITES)

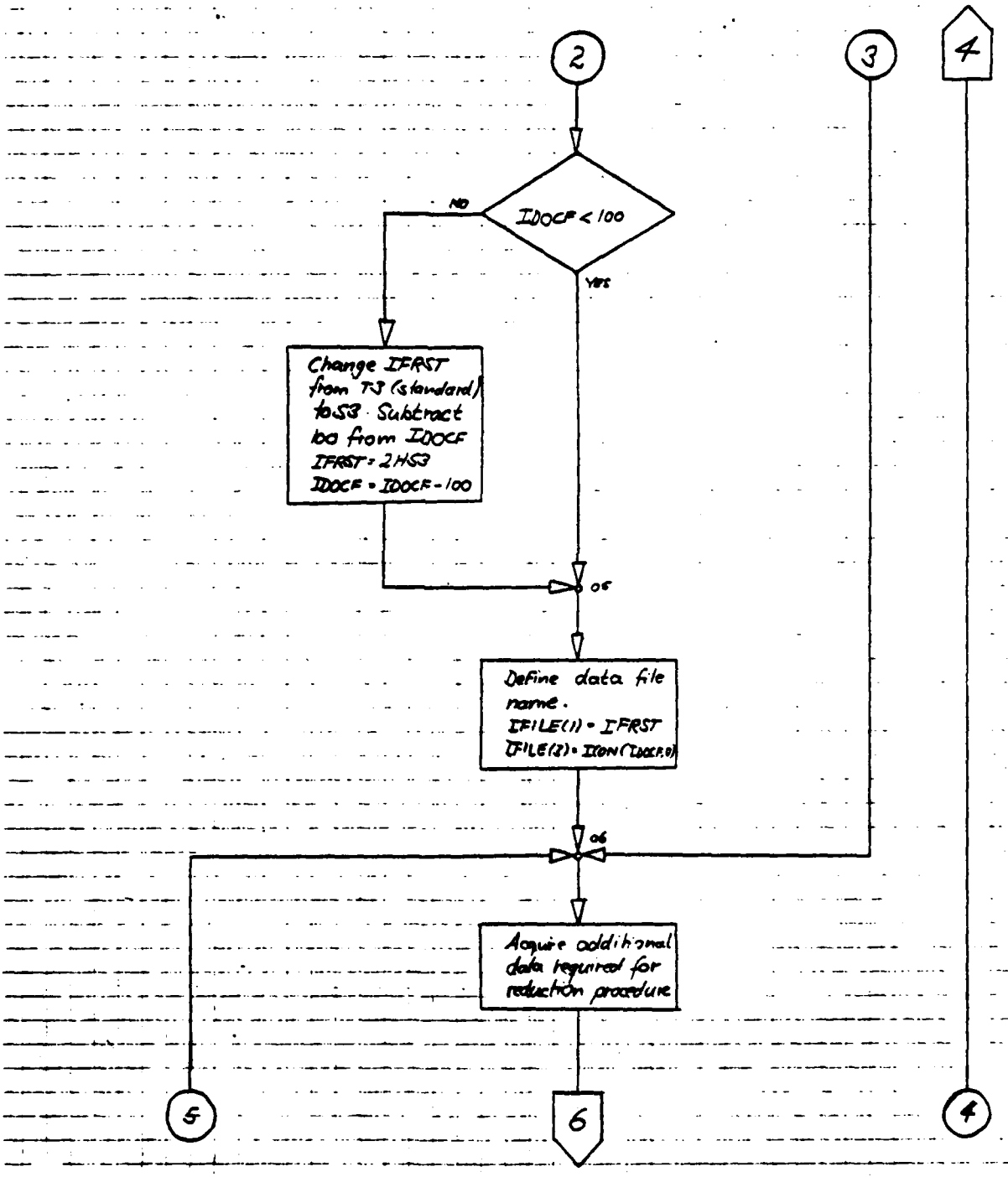
IW	integer	time delay between closing S/V part and reading transducer voltage From DVM
IMON	integer	ASCII converted month of the year
IDAY	integer	ASCII converted day of the month
IYEAR	integer	ASCII converted last two digits of current year
FREQ	real	RPM of the compressor
CIM	real	immersion of the combination probe
CYAW	real	yaw angle of the combination probe
PREF	real	reference pressure for the KULITE probes
P1	real	pressure P_1 from the combination probe
P23	real	pressure P_{23} from the combination probe
P4	real	pressure P_4 from the combination probe
E	real	thermocouple output, Station '0'
DE	real	thermocouple differential output from '0' across rotor
XIM	real	immersion of either 'A'- or 'B' probe
YAW	real	yaw angle of either 'A'- or 'B' probe
IADD	integer	variable to be added to start and stop address for paced run to

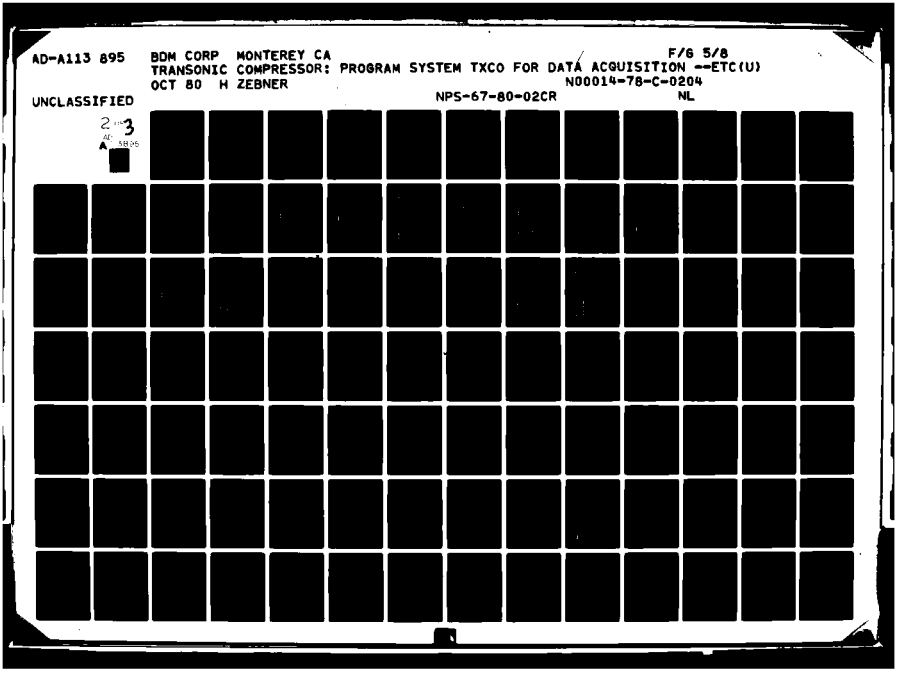
		compensate phase angle between these probes
IDUM	integer	decision variable
IPAMO	integer	pacer mode (1 or 2)
IPAIR	integer	selected blade pair
ISTART	integer	start address for paced run
ISTOP	integer	stop address for paced run
INCR	integer	increment for paced run
IRPT	integer	number of repetitions at each IBLADE
J111	integer	dummy variable
J222	integer	dummy variable
DUM	real	dummy variable
IERR	integer	error flag used by FMP calls
NEW	integer	scratch variable used to change file name

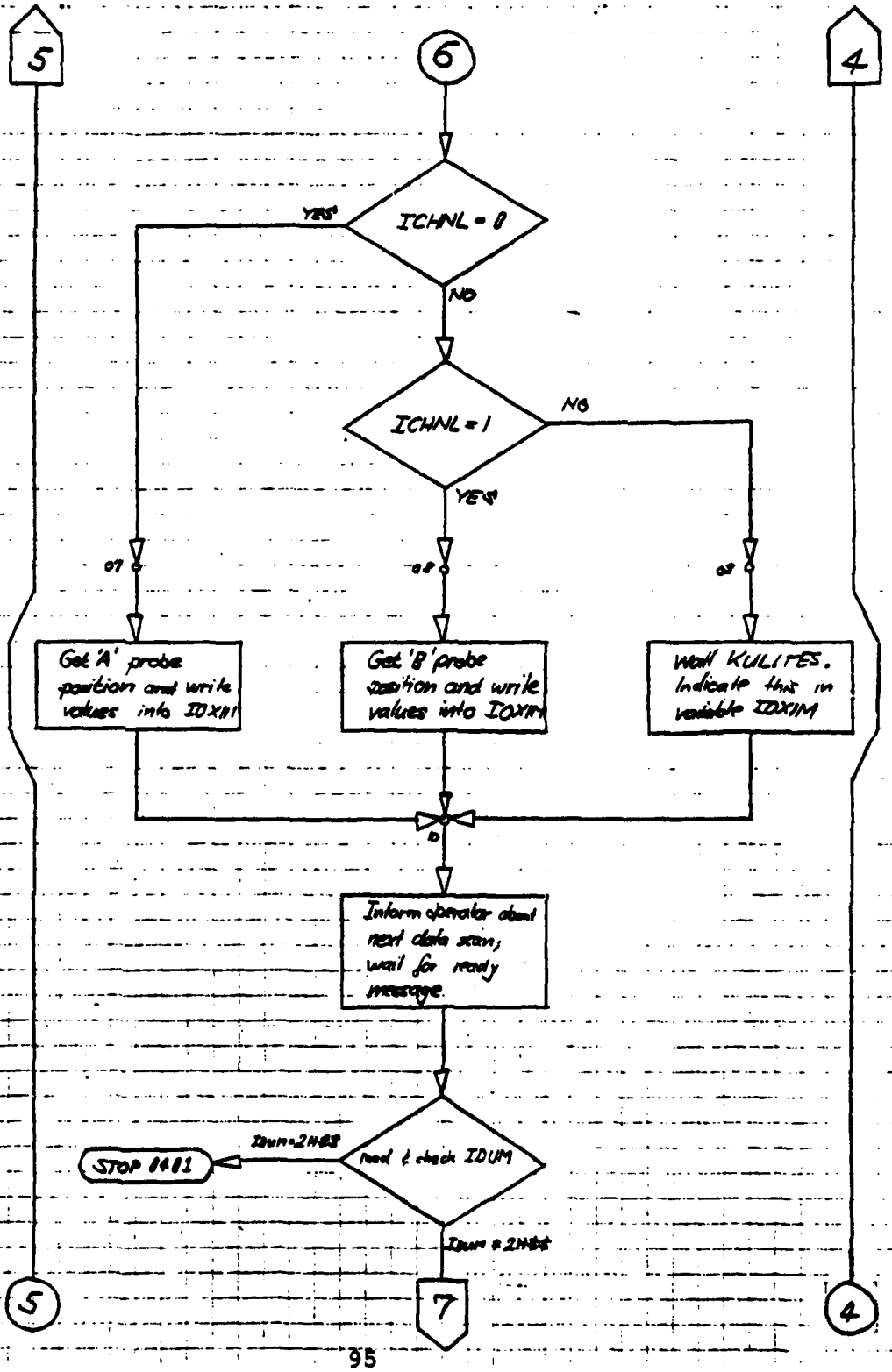
FLOW CHART SUBROUTINE PACER

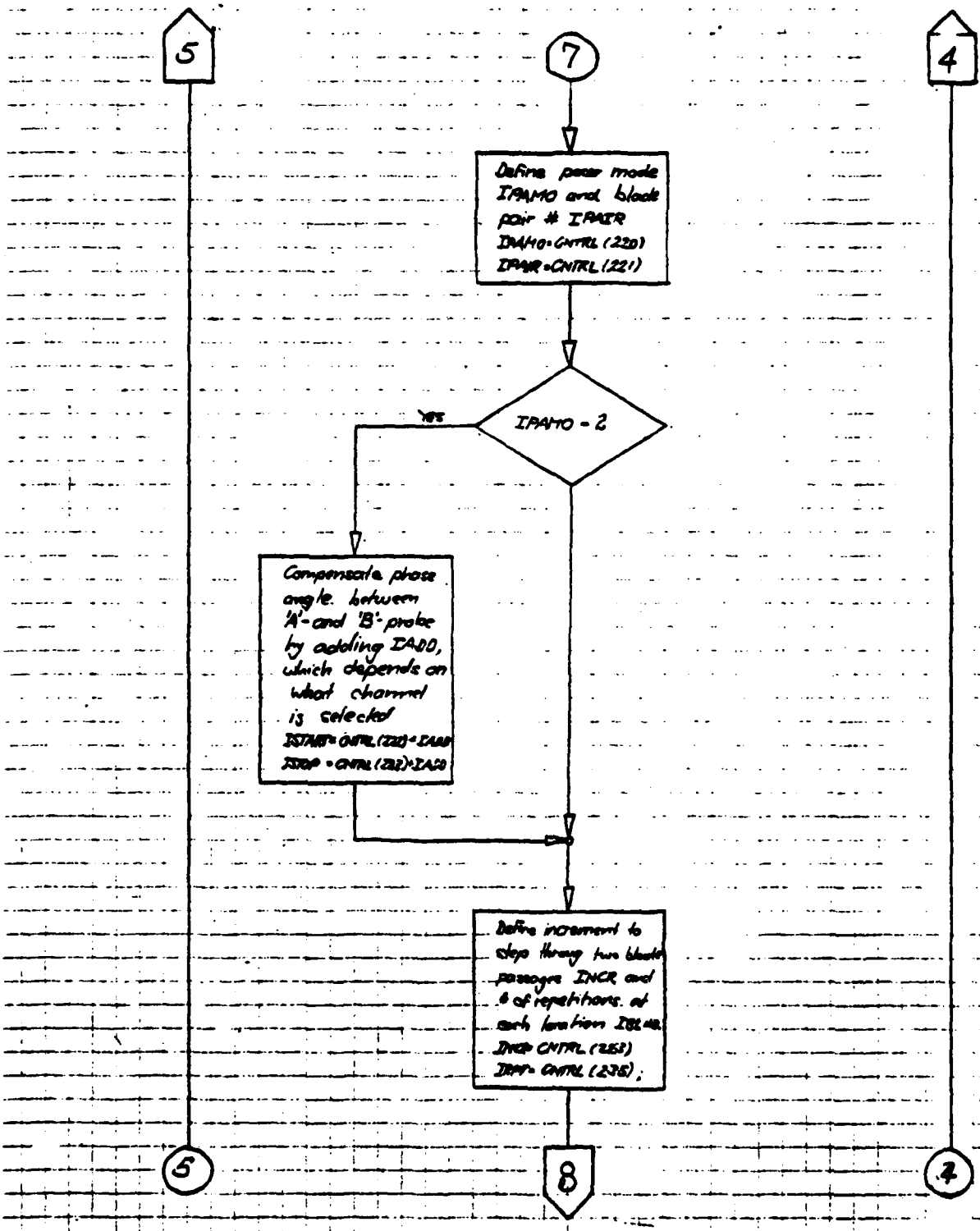


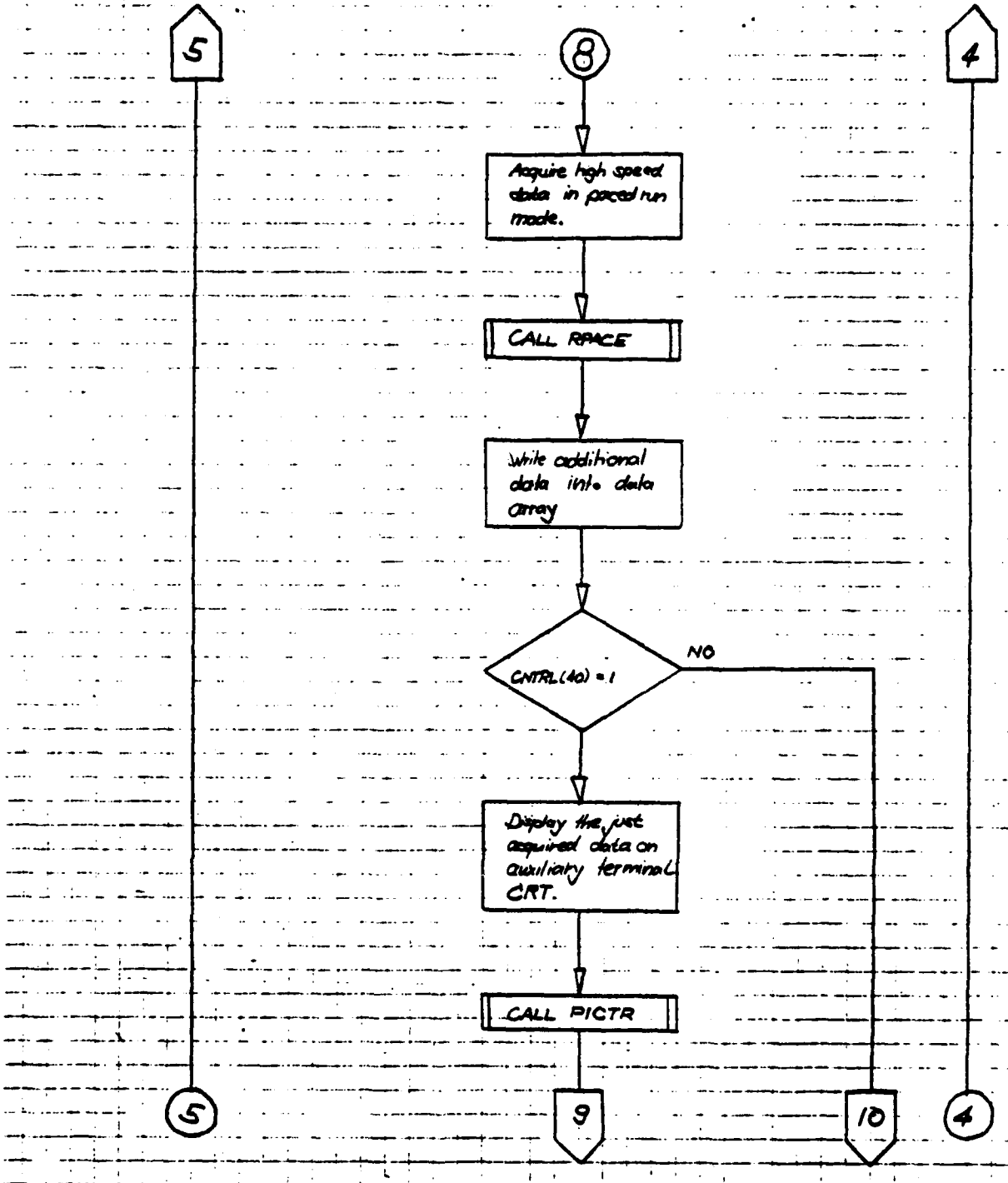

1
CNTRL(38) = 1
NO
YES
Print legend for tabulated output
DO 15 JI = 1, NRPTI, 1
Define A/D channel ICHNL = CNTRL(230+JI)
CNTRL(39) = 1
YES
Define sequential number of print out IDOC and data file IDOCR
IDOC = IDOC+1
IDOCR = IDOCR+1
2
3
4

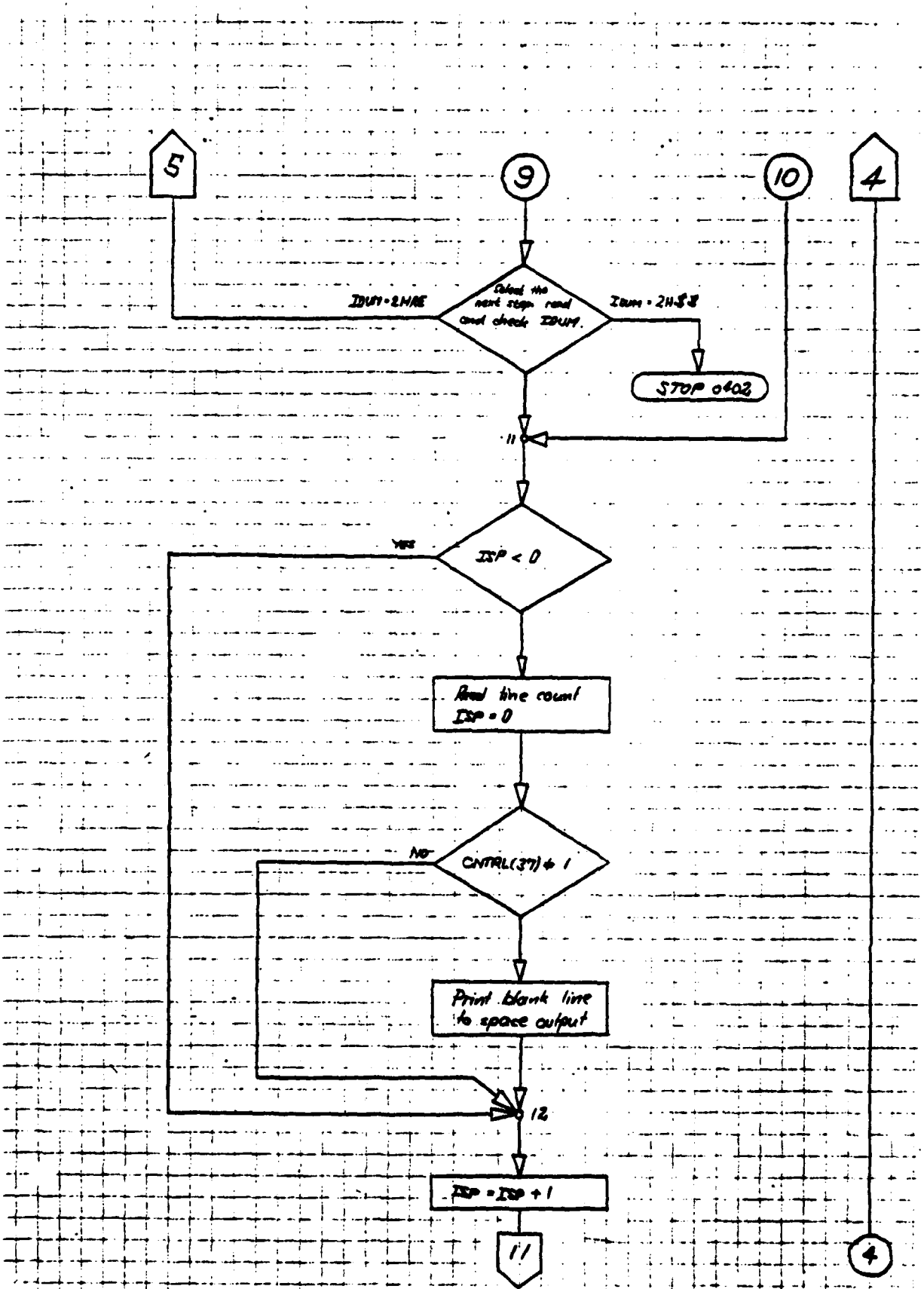

2
3
4
IDOCF < 100
NO
YES
Change IFRST from TJ (standard) to S3. Subtract 100 from IDOCF
IFRST = 2HS3
IDOCF = IDOCF - 100
Define data file name.
IFILE(1) = IFRST
IFILE(2) = ICON(IDOCF,0)
Acquire additional data required for reduction procedure
5
6
4

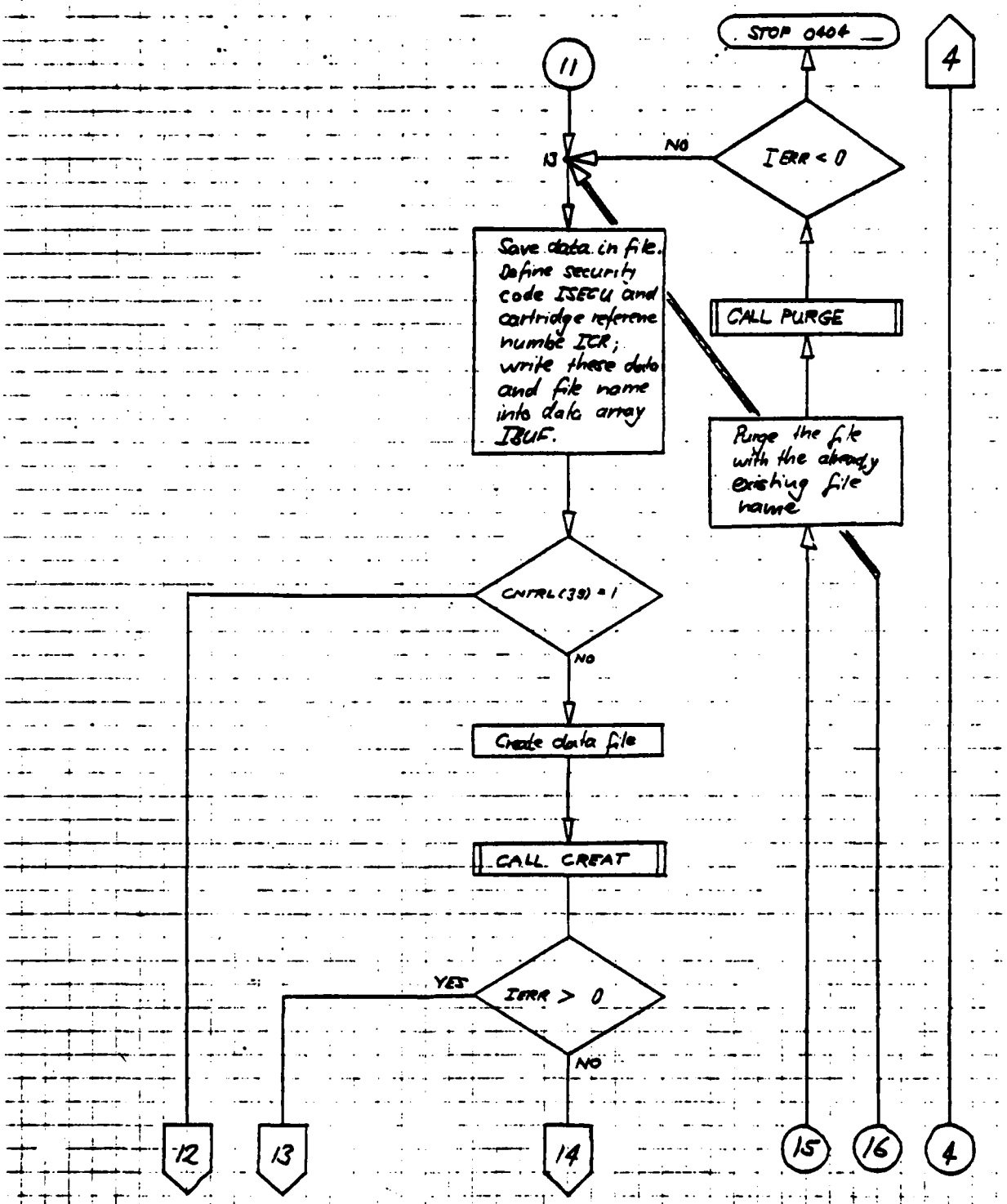

AD-A113 895
BDM CORP MONTEREY CA
TRANSONIC COMPRESSOR: PROGRAM SYSTEM TXCO FOR DATA ACQUISITION --ETC(U)
OCT 80 H ZEBNER
F/6 5/8
N00014-78-C-0204
NPS-67-80-02CR
NL
UNCLASSIFIED
2 OF 3
AD A113895

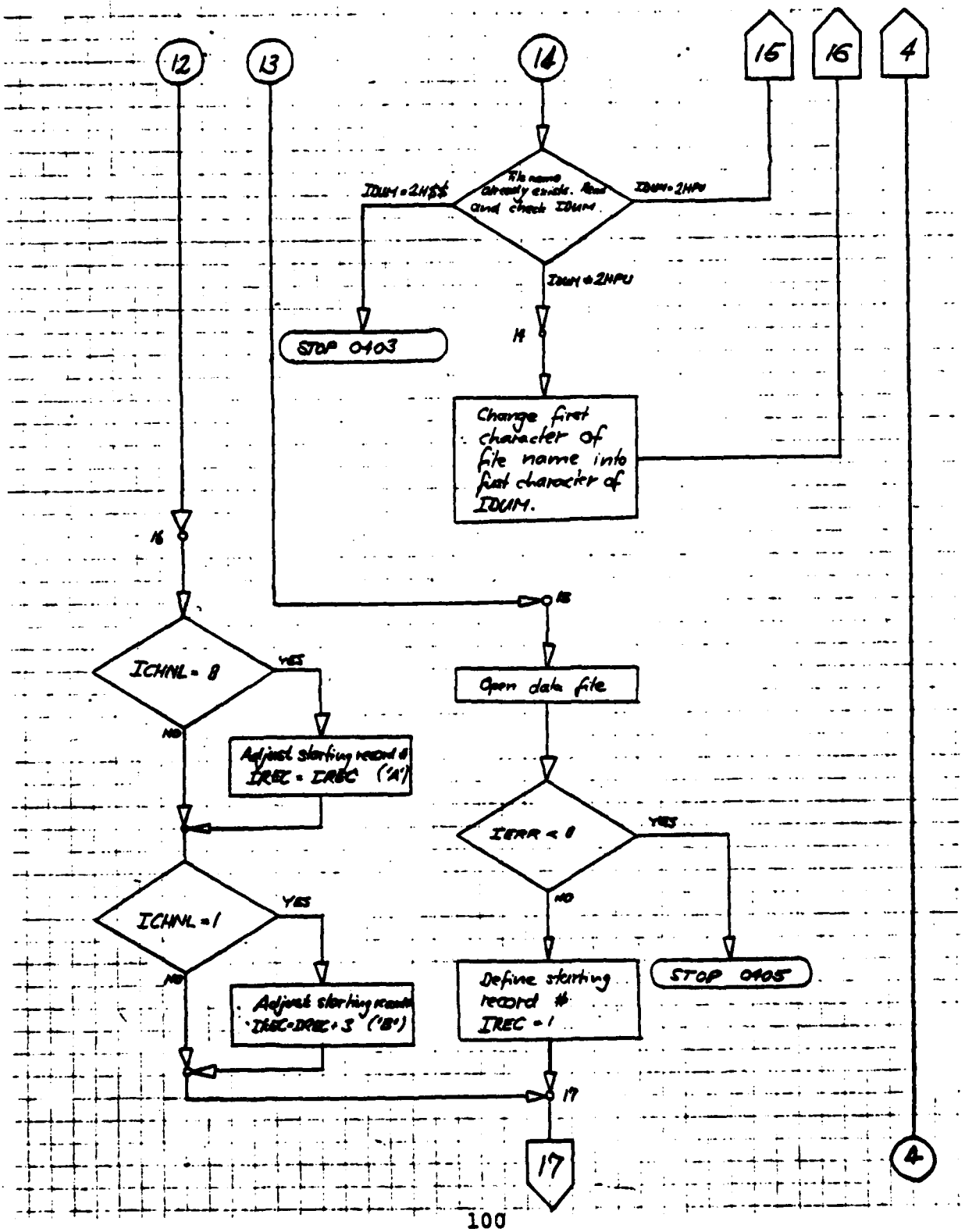

5
6
4
ICHNL = 0
YES
NO
ICHNL = 1
NO
YES
Get 'A' probe position and write values into IDXII
Get 'B' probe position and write values into IDXII
Wait KULITES. Indicate this in variable IDXIM
Inform operator about next data scan; wait for ready message.
read & check IDUM
IDUM=2HBB
STOP 8481
IDUM ≠ 2HBB
5
7
4

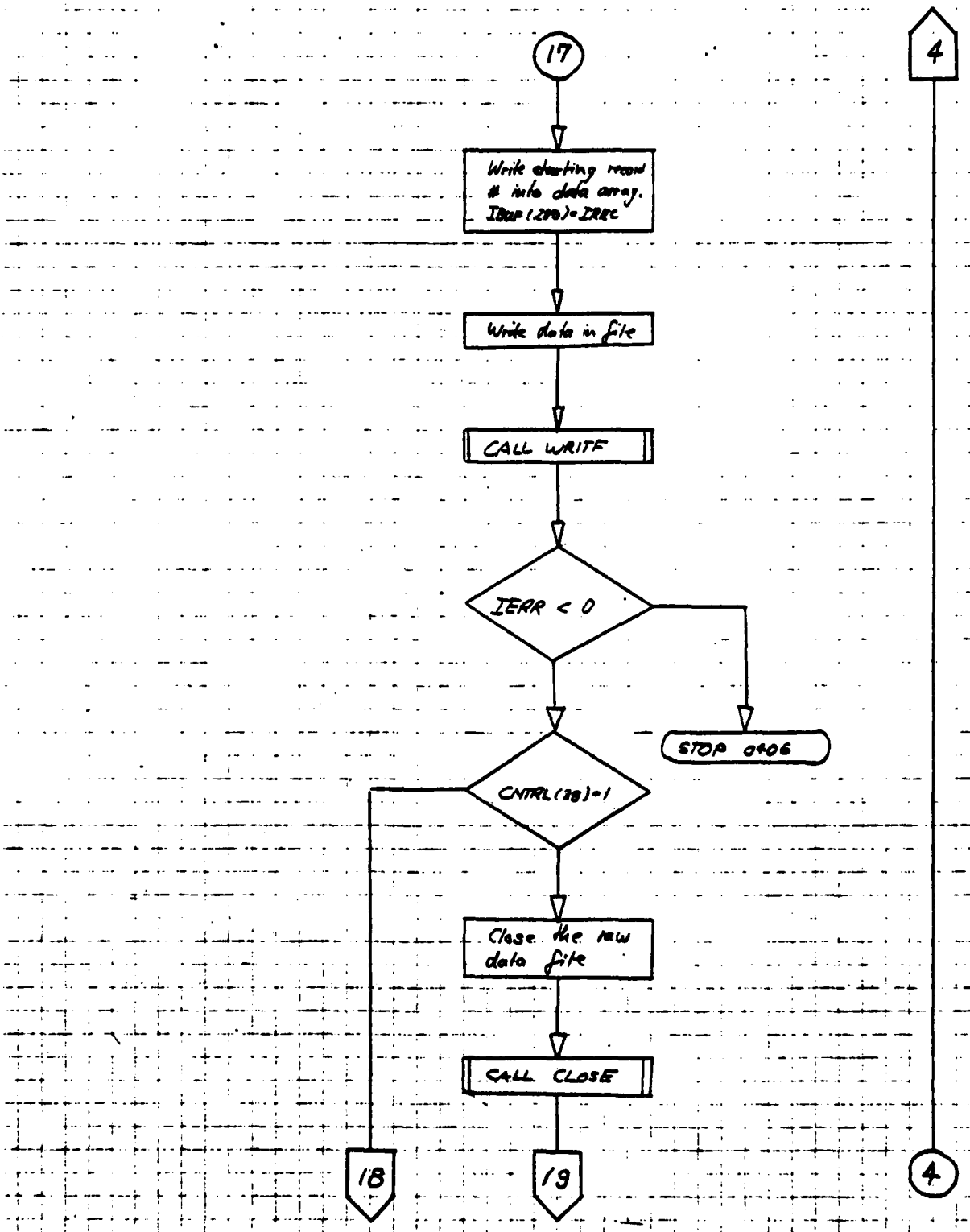

5
7
4
Define power mode IPAMO and block pair # IPAIR
IPAMO=CNTRL(220)
IPAIR=CNTRL(221)
IPAMO = 2
YES
Compensate phase angle between 'A'- and 'B'- probe by adding IADD, which depends on what channel is selected
ISTART=CNTRL(220)+IADD
ISTOP =CNTRL(222)+IADD
Define increment to step through two blade passages INCR and # of repetitions at each location IREP
INCR=CNTRL(223)
IREP=CNTRL(224);
5
8
4

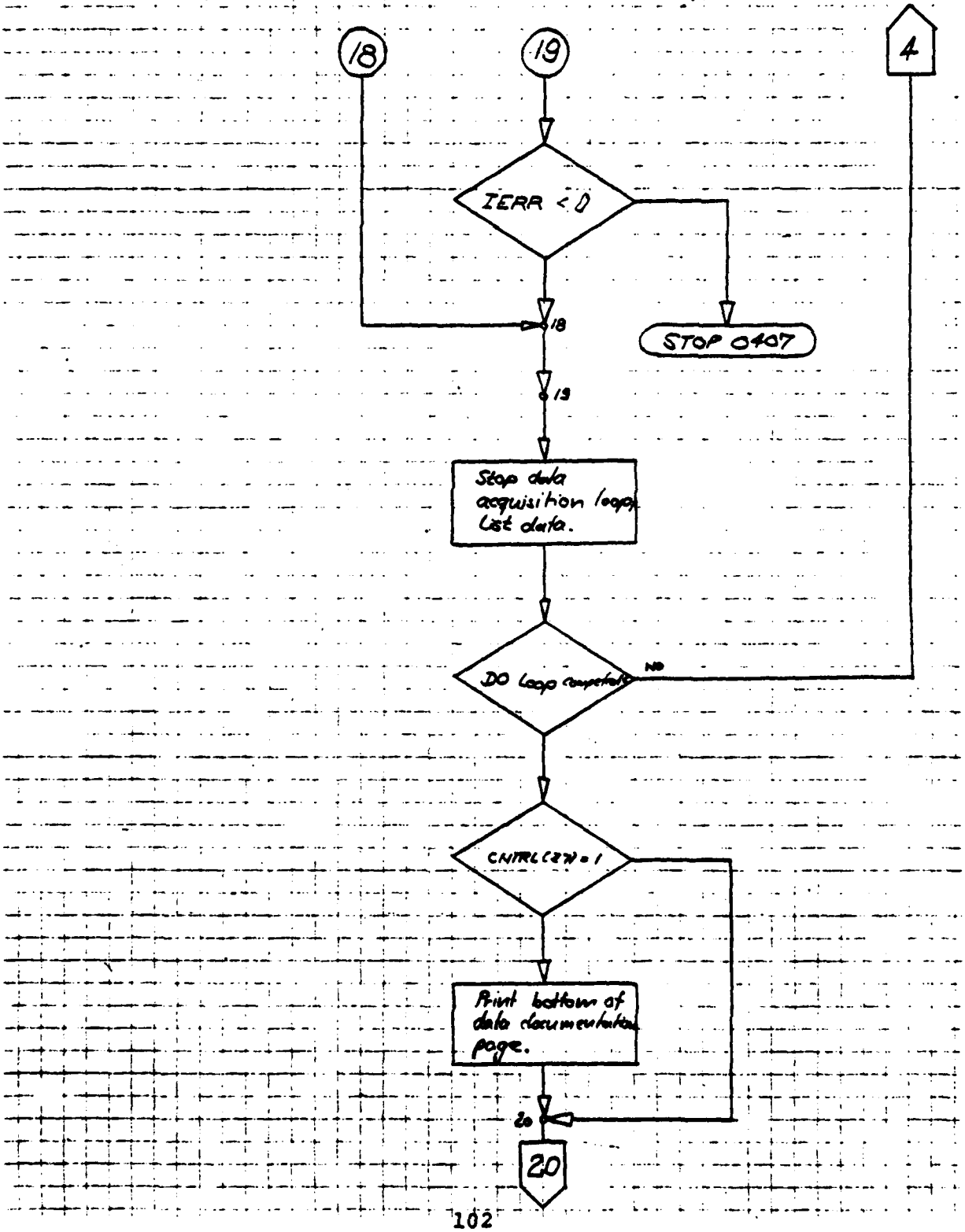

5
8
4
Acquire high speed data in paced run mode.
CALL RPACE
Write additional data into data array
CNTRL(40) = 1
NO
Display the just acquired data on auxiliary terminal CRT.
CALL PICTR
5
9
10
4

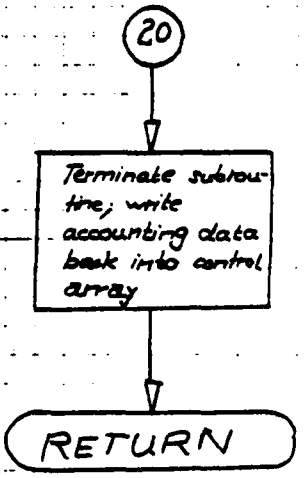

5
9
10
4
Reload the next step, read and check IDUM.
IDUM = 2HAE
IDUM = 2H-$8
STOP 0402
11
ISP < 0
YES
Read line count
ISP = 0
CNTRL(37) ≠ 1
NO
Print blank line to space output
12
ISP = ISP + 1
11
4


11
STOP 0404
4
13
NO
IERR < 0
Save data in file. Define security code ISECU and cartridge reference numbe ICR; write these data and file name into data array IBUF.
CALL PURGE
Purge the file with the already existing file name
CNTRL(39) = 1
NO
Create data file
CALL CREAT
YES
IERR > 0
NO
12
13
14
15
16
4


12
13
14
15
16
4
File name already exists. Read and check IDUM.
IDUM = 2H$$
IDUM = 2HPU
IDUM = 2HPU
STOP 0403
Change first character of file name into first character of IDUM.
ICHNL = 0
YES
NO
Adjust starting record # IREC = IREC ('A')
ICHNL = 1
YES
NO
Adjust starting record # IREC = IREC + 3 ('B')
Open data file
IERR < 0
YES
NO
STOP 0405
Define starting record # IREC = 1
17


17
4
Write starting record # into data array. IBUF(299)=IREC
Write data in file
CALL WRITF
IERR < 0
STOP 0406
CNTRL(38)=1
Close the raw data file
CALL CLOSE
18
19
4


18
19
4
IERR < 0
18
STOP 0407
19
Stop data acquisition loop. Last data.
DO Loop completed?
NO
CNTRL(27) = 1
Print bottom of data documentation page.
20
20


20
Terminate subroutine; write accounting data back into control array
RETURN

4.6. PROGRAM LISTING TXCO1

```
PAGE 0001  FTN.   2:47 PM  MON., 25  AUG., 1980

0001  FTN4,L
0002          BLOCK DATA
0003         *, / FMP    / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004          COMMON / FMP    / IDCB,IFILE,ISIZE,ISECU,ICR
0005          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006          END

 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
 ** NO WARNINGS **  NO ERRORS **
BLOCK COMMON FMP     SIZE =  00151
```

```
 PAGE 0002  FTN.   2:47 PM  MON., 25  AUG., 1980

0007          BLOCK DATA
0008         *, / CIBUF / IBUF(1664)
0009          COMMON / CIBUF / IBUF
0010          INTEGER IBUF(1664)
0011          END

  FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
  ** NO WARNINGS **  NO ERRORS **
 BLOCK COMMON CIBUF   SIZE =  01664
```

```
0012          BLOCK DATA
0013         *, / CONTR / CNTRL(256)

0014          COMMON / CONTR / CNTRL
0015          INTEGER CNTRL(256)
0016          END
```

FTN4 COMPILER: HP92060-16092_REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS **

BLOCK COMMON CONTR SIZE = 00256

```
0017          PROGRAM TXCO1 (3,99)
0018    C     .....................................................................
0019    C     .                                                                   .
0020    C     .  The operating system RTE IV B requests the data acquisition      .
0021    C     .  program TXCO for the one stage transsonic compressor to be       .
0022    C     .  split into several programs schduled by the father program       .
0023    C     .  program TXCO0.  This son program TXCO1 consists of the sub-      .
0024    C     .  routines ABSRV, CALIB, FREER and PACER.  These handle the        .
0025    C     .  acquisition of high speed data.  The data transfer between       .
0026    C     .  father and son program takes place via the control array         .
0027    C     .  file CONTR (disc file CNTRLF) and the data array IBUF (disc      .
0028    C     .  file IBUFF).                                                     .
0029    C     .  The utility subroutines ACQN, CNTL, CURVE, ICON, IPORT,          .
0030    C     .  PICTR, REWRF, RPACE, SCANR, TIME and WAIT are added.             .
0031    C     .  Author:    Hans M. Zebner                                        .
0032    C     .  Date  :    March 12, 1980                                        .
0033    C     .  A detailed program description is available in the TXCO log.     .
0034    C     .  Comment statements ans statement numbers in the source code      .
0035    C     .  match to yhe program description.  This program is part of       .
0036    C     .  the TXCO transonic compressor investigation program system.      .
0037    C     .                                                                   .
0038    C     .....................................................................
0039         *, First son program of father program TXCO.

0040          COMMON / CONTR / CNTRL
0041
0042          INTEGER CNTRL(256)
0043
0044          DATA NOLF  /006537B/
0045      101 FORMAT (9X"*20X""A2)
0046      102 FORMAT ("  TXCO0 : PROGRAM ABORTED!  NO SUBROUTINE HAS BE
0047         *EN INITIALIZED.")
0048      801 FORMAT ("CA")
0049     1001 FORMAT ("F1R7M3A1H0T3")
0050     1201 FORMAT ("PF4G6T")
0051     1501 FORMAT ("CA")
0052
0053          CALL REWRF (-1,2)
0054          LI = CNTRL(19)
0055          IF ( CNTRL(50)  .LT.  1  .OR.  CNTRL(50)  .GT.  4 ) GO TO 05
0056
0057    C     .....................................................................
0058    C     .                                                                   .
0059    C     .  Set interface bus and devices to remote control.                 .
0060    C     .                                                                   .
0061    C     .....................................................................
0062          CALL ABRT (7,2)
0063          CALL RMOTE (8)
0064          CALL RMOTE (10)
0065          CALL RMOTE (12)
0066          CALL RMOTE (15)
0067          WRITE ( 8, 801)
0068          WRITE (10,1001)
0069          WRITE (12,1201)
0070          WRITE (15,1501)
0071
0072    C     .....................................................................
0073    C     .                                                                   .
0074    C     .  Call subroutine indicated by CNTRL(50).                          .
0075    C     .                                                                   .
0076    C     .....................................................................
0077          ISTOP = CNTRL(50)
0078          IF ( CNTRL(50)  .EQ.  1 ) CALL ABSRV
0079          IF ( CNTRL(50)  .EQ.  2 ) CALL CALIB
0080          IF ( CNTRL(50)  .EQ.  3 ) CALL FREER (X1,X2,X3)
0081          IF ( CNTRL(50)  .EQ.  4 ) CALL PACER (1)
0082
0083    C     .....................................................................
0084    C     .                                                                   .
0085    C     .  Release interface bus and devices from remote control.           .
0086    C     .                                                                   .
0087    C     .....................................................................
0088          CALL CLEAR (7,1)
0089          CALL LOCL (7)
0090
0091          CALL REWRF (1,2)
```

```
PAGE 0005  TXC01  2:47 PM  MON., 25  AUG., 1980

0092          WRITE (LI, 101) NOLF
0093          GO TO (01,02,03,04) ISTOP
0094       01 STOP 0177
0095       02 STOP 0277
0096       03 STOP 0377
0097       04 STOP 0477
0098       05 WRITE (LI, 102)
0099          END


 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00257      COMMON = 00000
```

```
0100          SUBROUTINE ABSRV
0101    C     ..................................................................
0102    CCCCC .                                                                .
0103    C     .  Subroutine to acquire high speed data from the 1-stage axial  .
0104    C     .  transonic compressor using miniaturized probes equipped with  .
0105    C     .  KULITE semiconductor pressure transducers.                    .
0106    C     .  Author:    Hans Zebner                                        .
0107    C     .  Date:      August 12, 1980                                    .
0108    C     .  A detailed program description is available in the TXCO log.  .
0109    C     .  Comment statements and statement numbers in the source code   .
0110    C     .  match to the program description.  This subroutine is part of .
0111    C     .  the TXCO transonic compressor investigation program system.   .
0112    C     .                                                                .
0113    C     ..................................................................
0114          *, Takes data from the 'A' - 'B' - probe system.
0115
0116          COMMON / CIBUF / IBUF
0117          COMMON / CONTR / CNTRL
0118          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
0119
0120          INTEGER IBUF(1664)
0121          INTEGER CNTRL(256)
0122          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0123
0124          REAL POS(7),RBUF(64)
0125          INTEGER NOLF,NOCR(2),ICLR(3)
0126
0127          EQUIVALENCE (IBUF(1),RBUF(1))
0128
0129          DATA NOLF  /006537B/
0130          DATA NOCR  /000033B,040433B/
0131          DATA ICLR  /015524B,015515B,006537B/
0132          DATA IDCBS /144/
0133
0134    C     FORMATS ABSRV START
0135      101 FORMAT ("  Did you calibrate the type 'A' and type 'B' probes
0136         *  on line?          ",3("*"),2A2)
0137      102 FORMAT (A2)
0138      103 FORMAT (/" "79X""A2/"  Since you forgot to calibrate
0139         *these nice probes, I will do it right now."6X""A2/"  P
0140         *ress the RETURN key to continue the execution of the p
0141         *rogram!"16X""A2/" "79X""A2/)
0142      104 FORMAT ("  ABSRV : CALL CALIB")
0143      105 FORMAT (/"  Enter the following results from the on line cali
0144         *bration!" 5X""/
0145         *"     PBARO    PREF   AVRGEA   AVRGEB
0146         * SLOPEA   SECONA   SLOPEB   SECONB"/
0147         *"                 ",7("          "),A2)
0148      106 FORMAT (1X,F7.2,1X,7(F7.6,1X))
0149      107 FORMAT ("  How many yaw positions for the type 'A' and type '
0150         *B' probe?          ",2("*"),2A2)
0151      108 FORMAT ("WARNING: file "3A2" already exists!  Type PU to "
0152         *"allow purge or enter any char-",
0153         */,"                 acter but T to change file name."38X)
0154      109 FORMAT ("  ABSRV : PURGE "3A2":"A2":"A2)
0155      110 FORMAT (A1"1")
0156      111 FORMAT ("  ABSRV : File name "3A2" succesfully changed to "3A2)
0157      112 FORMAT ("  ABSRV : CREATE "3A2":"A2":"A2":"I1","I2":"I3)
0158      113 FORMAT (""15X"Read the probe positions;  Yaw Angle and Immersi
0159         *on."14X""A2)
0160      114 FORMAT (" Enter case angle"34X,2A2)
0161      115 FORMAT (/21X"Immersion"11X"Yaw Angle"/
0162         *24X"inches"19X"degrees"/
0163         *"  Combination probe "F10.3,10X,F10.3""/
0164         *"       Type 'A' probe "F10.3,10X,F10.3""/
0165         *"       Type 'B' probe "F10.3,10X,F10.3""//
0166         *"            Case angle "20X""F10.3""//
0167         *"Type  UP  to update these readings"/
0168         *"      TA  to take a data set at this constellation"/
0169         *"      "2A2)
0170      116 FORMAT ("  ABSRV : CALL PACER("I2")")
0171      117 FORMAT ( /"  "79X""A2/"  Check raw data from this"I2".
0172         * yaw position for obvious errors!"18X""A2/" "79X""
0173         *A2/
0174         *"  Type  RE  to repeat this point"48X""A2/
```

```
0175          *"          NE  to proceed to the next point"40X""A2/
0176          *"          EN  to terminate the 'A'-'B'-probe survey at
0177          *this radius/operating point"A2/
0178          *" "7X"  "70X""A2/
0179          *8X"  ",2(""),2A2)
0180       118 FORMAT ("  Error: You want to proceed to the"I2". positio
0181          *n, but the data array only"8X""A2/
0182          *"  can store"I2" positions, that you defined previously!"37X"
0183          *"A2)
0184       149 FORMAT (""((3A2)))
0185
0186       601 FORMAT (1H ,15(1H ),33HTransonic Compressor Test Run # ,I7)
0187       602 FORMAT (1H ,28(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
0188       603 FORMAT (1H ,"",28(1H ),6HTime: ,A2,1H.,A2,3H  h,/////)
0189       604 FORMAT (1H ,"",/////,8X
0190          *,"'A'  -  'B'  Probes Pass "I3" .",
0191          *//////)
0192       605 FORMAT (" "72X""//"*"41X""I2". Yaw position")
0193       606 FORMAT (/42X""I2". Yaw position"/" "72X"")
0194       607 FORMAT ("
```

",28(1H),6HTime: ,A2,1H.,A2,3H h)

```
0195  C     FORMATS ABSRV STOP
0196
0197
0198
0199  C     .....................................................................
0200  CCCC  :                                                                   :
0201  CCCC  :  Accounting.                                                      :
0202  CCCC  :                                                                   :
0203  C     .....................................................................
0204        IPAGE     = CNTRL(212)
0205        IDOC      = CNTRL(213)
0206        IDOCF     = CNTRL(213)
0207        IPAGE     = IPAGE+1
0208        IDOC      = IDOC+1
0209        IDOCF     = IDOCF+1
0210        ISECU     = CNTRL(31)
0211        ICR       = CNTRL(30)
0212        ISIZE(2)  = 128
0213        IFILE(2)  = ICON(CNTRL(4),0)
0214        IL        = 128
0215        ITYPE     = 1
0216        IFRST     = 2HT1
0217
0218
0219
0220  C     .....................................................................
0221  CCCC  :                                                                   :
0222  CCCC  :  I/O Assignments;  preset data array.                             :
0223  CCCC  :                                                                   :
0224  C     .....................................................................
0225        LI        = CNTRL(19)
0226        LO        = CNTRL(20)
0227        LS1       = CNTRL(71)
0228        LS2       = CNTRL(72)
0229        DO 01  I=1,768,1
0230     01 IBUF(I)= 0250$2B
0231
0232
0233
0234  C     .....................................................................
0235  CCCC  :                                                                   :
0236  CCCC  :  Ask operator, whether the 'A'-'B'-probe system has been          :
0237  CCCC  :  calibrated on line.                                             :
0238  CCCC  :                                                                   :
0239  C     .....................................................................
0240        WRITE (LI, 101) NOCR
0241        READ  (LI, 102) ICAL
0242        WRITE (LI, 149) ICLR
0243        IF ( ICAL  .EQ.  2Hss ) STOP 0101
0244        IF ( ICAL  .NE.  2HNO ) GO TO 02
0245
0246
0247
0248  C     .....................................................................
0249  C     :                                                                   :
```

```
0250   C      .  Call subroutine CALIB to calibrate 'A'-'B'-probe system.      .
0251   CC     .                                                               .
0252   CC     .................................................................
0253          WRITE (LI, 103) (NOLF,I=1,4,1)
0254          READ  (LI, 102) IDUM
0255          WRITE (LI, 149) (ICLR,I=1,7,1)
0256          IF ( IDUM  .EQ.  2H$$ ) STOP 0102
0257          WRITE (LI, 104)
0258          CALL CALIB
0259
0260
0261
0262   C      .................................................................
0263   CCC    .                                                               .
0264   CCCC   .  Enter calibration results;  print heading.                   .
0265   CCC    .                                                               .
0266   C      .................................................................
0267       02 WRITE (LI, 105) NOLF
0268          READ  (LI, 106) SLOPEA,SECONA,SLOPEB,SECONB,AVRGEA,AVRGEB,PBARO
0269          WRITE (LI, 149) (ICLR,I=1,4,1)
0270          CALL TIME (IMON,IYEAR,IDAY,IHOUR,IMIN)
0271          WRITE (LO, 601) CNTRL(4)
0272          WRITE (LO, 602) IMON,IYEAR,IDAY
0273          WRITE (LO, 603) IHOUR,IMIN
0274          WRITE (LO, 604) IPAGE
0275
0276
0277
0278   C      .................................................................
0279   CCC    .                                                               .
0280   CCCCC  .  Initialize data acquisition;  create raw data file of the    .
0281   CCCCC  .  correct size.  If the file name assigned to this data set     .
0282   CCCCC  .  already exists, the operator decides, whether to purge the    .
0283   CCCCC  .  already existing file (PU) or change this file name.          .
0284   CCC    .                                                               .
0285   C      .................................................................
0286          WRITE (LI, 107) NCCR
0287          READ  (LI,   *) NPOS
0288          WRITE (LI, 149) ICLR
0289          ISIZE(1) = 1+NPOS*6
0290          IF ( IDOCF  .LT.  100 ) GO TO 03
0291          IFRST    = 2HS1
0292          IDOCF    = IDOCF-100
0293       03 IFILE(1) = IFRST
0294          IFILE(3) = ICON(IDOCF,0)
0295       04 CALL CREAT (IDCB,IERR,IFILE,ISIZE,ITYPE,ISECU,ICR,IDCBS)
0296          IF ( IERR  .GT.  0 ) GO TO 06
0297          WRITE (LI, 108) IFILE
0298          READ  (LI, 102) IDUM
0299          WRITE (LI, 149) (ICLR,I=1,3)
0300          IF ( IDUM  .EQ.  2H$$ ) STOP 0103
0301          IF ( IDUM  .NE.  2HPU ) GO TO 05
0302          JSECU = ICON(ISECU,0)
0303          JCR   = ICON(ICR,0)
0304          WRITE (LI, 109) IFILE,JSECU,JCR
0305          CALL PURGE (IDCB,IERR,IFILE,ISECU,ICR)
0306          IF ( IERR  .LT.  0 ) STOP 0104
0307          GO TO 04
0308       05 CALL CODE
0309          WRITE (NEW,110) IDUM
0310          WRITE (LI, 111) IFILE,NEW,IFILE(2),IFILE(3)
0311          IFILE(1) = NEW
0312          GO TO 04
0313       06 CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0314          IF ( IERR  .LT.  0 ) STOP 0105
0315          DO 07 I=1,NPOS,1
0316          IREC   = 1+(I-1)*6
0317          CALL WRITF (IDCB,IERR,IBUF,768,IREC)
0318          IF ( IERR  .LT.  0 ) STOP 0106
0319       07 CONTINUE
0320          JSECU  = ICON(ISECU,0)
0321          JCR    = ICON(ICR,0)
0322          WRITE (LI, 112) IFILE,JSECU,JCR,ITYPE,ISIZE
0323
0324
```

```
0325
0326        C     .....................................................................
0327        C     .                                                                   .
0328        C     .  Position the raw data file is done by subroutine ABSRV.  The     .
0329        C     .  actual data are written in the data file by subroutine PACER.    .
0330        C     .  CNTRL(37) is set to 1 in order to suppress printing a heading    .
0331        C     .  in subroutine PACER.  CNTRL(39) is set to 1 to tell subrou-      .
0332        C     .  tine PACER not to create/open and close a new data file.         .
0333        C     .                                                                   .
0334        C     .....................................................................
0335              IPOS   = 0
0336           08 IPOS   = IPOS+1
0337              IF ( IPOS  .GT.  NPOS ) GO TO 19
0338           09 CNTRL(37) = 1
0339              CNTRL(38) = IPOS
0340              CNTRL(39) = 1
0341              IF ( IPOS  .EQ.  1 ) WRITE (LO, 605) IPOS
0342              IF ( IPOS  .GT.  1 ) WRITE (LO, 606) IPOS
0343              IREC   = 2+(IPOS-1)*6
0344              CALL POSNT (IDCB,IERR,1,IREC)
0345              IF ( IERR  .LT.  0 ) STOP 0107
0346
0347
0348
0349        C     .....................................................................
0350        C     .                                                                   .
0351        C     .  Check position of probes before acquiring data.                  .
0352        C     .                                                                   .
0353        C     .....................................................................
0354           10 WRITE (LI, 113) NOLF
0355              IC     = 1
0356              I2     = 1
0357              DO 11 J=30,35,1
0358              POS(I2)= SCANR(LS1,J,IC)
0359           11 I2     = I2+1
0360              DO 12 J=1,5,2
0361           12 POS(J) = POS(J)*1000.
0362              DO 13 J=2,6,2
0363           13 POS(J) = POS(J)*10000.
0364              WRITE (LI, 149) ICLR
0365              WRITE (LI, 114) NOCR
0366              READ  (LI,  *) POS(7)
0367              WRITE (LI, 149) ICLR
0368           14 WRITE (LI, 115) (POS(J),J=1,7,1),NOCR
0369              READ  (LI, 102) IDUM
0370              WRITE (LI, 149) (ICLR,I=1,12,1)
0371              IF ( IDUM  .EQ.  2HUP ) GO TO 10
0372              IF ( IDUM  .EQ.  2HTA ) GO TO 15
0373              IF ( IDUM  .EQ.  2H$$ ) STOP 0110
0374              GO TO 14
0375
0376
0377
0378        C     .....................................................................
0379        C     .                                                                   .
0380        C     .  Acquire data in subroutine PACER.                                .
0381        C     .                                                                   .
0382        C     .....................................................................
0383           15 WRITE (LI, 116) IREC
0384              CALL PACER (IREC)
0385
0386
0387
0388        C     .....................................................................
0389        C     .                                                                   .
0390        C     .  Select the next step:                                            .
0391        C     .    RE        repeat the data acquisition at this yaw position     .
0392        C     .    NE        proceed to the next yaw position                     .
0393        C     .    EN        terminate the survey at this operating point         .
0394        C     .                                                                   .
0395        C     .....................................................................
0396           16 WRITE (LI, 117) NOLF,IPOS,(NOLF,I=1,6,1),NOCR
0397              READ  (LI, 102) IDUM
0398              WRITE (LI, 149) (ICLR,I=1,9,1)
0399              IF ( IDUM  .EQ.  2HNE ) GO TO 08
```

```
0400          IF ( IDUM  .EQ.  2HRE ) GO TO 09
0401          IF ( IDUM  .EQ.  2HEN ) GO TO 17
0402          IF ( IDUM  .EQ.  2H$$ ) STOP 0111
0403          GO TO 16
0404
0405
0406
0407    C     ..................................................................
0408    CCCCC .                                                                :
0409    CCCCC .  Stop data acquisition.  Write additional data (i.e. barome-   :
0410    CCCCC .  tric pressure, calibration results, number of points and yaw  :
0411    CCCCC .  positions into first record (Directory) of the data file.     :
0412    CCCCC .                                                                :
0413    C     ..................................................................
0414       17 CONTINUE
0415          DO 18  I=1,128,1
0416       18 IBUF(I) = 0250$2B
0417          IBUF(1) = 256
0418          IBUF(2) = NPOS
0419          RBUF(2) = PBARO
0420          RBUF(3) = PREF
0421          RBUF(4) = AVRGEA
0422          RBUF(5) = AVRGEB
0423          RBUF(6) = SLOPEA
0424          RBUF(7) = SECONA
0425          RBUF(8) = SLOPEB
0426          RBUF(9) = SECONB
0427          CALL TIME (IBUF(96),IBUF(104),IBUF(112),IBUF(120),IBUF(128))
0428          IREC    = 1
0429          CALL WRITF (IDCB,IERR,IBUF,IL,IREC)
0430          IF ( IERR  .LT.  0 ) STOP 0112
0431          CALL CLOSE (IDCB,IERR)
0432          IF ( IERR  .LT.  0 ) STOP 0113
0433          WRITE (LO, 607) IBUF(120),IBUF(128)
0434
0435
0436
0437    C     ..................................................................
0438    CCCCC .                                                                :
0439    CCCCC .  Terminate subroutine;  write accounting variables back into   :
0440    CCCCC .  control array.                                                :
0441    CCCCC .                                                                :
0442    C     ..................................................................
0443          CNTRL(212) = IPAGE
0444          CNTRL(213) = IDOC
0445          CNTRL( 50) = -1
0446          RETURN
0447
0448
0449
0450    C     ..................................................................
0451    CCCCC .                                                                :
0452    CCCCC .  Error returns.                                                :
0453    CCCCC .                                                                :
0454    C     ..................................................................
0455       19 WRITE (LI, 118) IPOS,NOLF,NPOS,NOLF
0456          GO TO 17
0457          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 02498 COMMON = 00060

```
PAGE 0011  FTN.   2:47 PM  MON., 25  AUG., 1980

0458          SUBROUTINE CALIB
0459    C     .....................................................
0460    C     .                                                   .
0461    C     .  Subroutine to control the on line calibration of the KULITE
0462    C     .  type 'A'-'B'-probe system.
0463    C     .  Author:     Hans Zebner
0464    C     .  Date:       August 13, 1980
0465    C     .  A detailed program description is available in the TXCO log.
0466    C     .  Comment statements and statement numbers in the source code
0467    C     .  match to the program description.  This subroutine is part of
0468    C     .  the TXCO transonic compressor investigation program system.
0469    C     .                                                   .
0470    C     .....................................................
0471          *, On-line calibration of KULITE probes.
0472
0473          COMMON / CONTR / CNTRL
0474
0475          INTEGER CNTRL(256)
0476
0477          REAL AVOLT(10),BVOLT(10),RPRES(10),DMM(10)
0478          INTEGER NOCR(2),ICLR(3),ITIME(5),IO(5)
0479
0480          DATA NOLF  /006537B/
0481          DATA NOCR  /000033B,040433B/
0482          DATA ICLR  /015524B,015515B,006537B/
0483
0484    C     FORMATS CALIB START
0485      101 FORMAT (/" "79X""A2/"  Apply defined reference pressur
0486         *e to KULITE pressure transducers!  Input DMM   "A2/
0487         *"  multimeter read out to initialize calibration,  RE
0488         *to repeat this part of the "A2/
0489         *"  calibration  or  EN to terminate the on line calibr
0490         *ation!"21X""A2/" "79X""/
0491         *"                "A2)
0492      102 FORMAT (5A2)
0493      103 FORMAT (F10.6)
0494      104 FORMAT ("  CALIB : CALL FREER")
0495      105 FORMAT ("  Switch PACER to free run mode;  then press
0496         *CR to continue!   "3A2)
0497      106 FORMAT ("  Switch PACER to pacer run mode;  then press
0498         * CR to continue! "3A2)
0499      107 FORMAT ("  CALIB : CALL PACER("I2")")
0500      108 FORMAT ("  Error: You did not perform a calibration at all
0501         *!"30X""A2)
0502      109 FORMAT ("  Error: Please, ask yourself honestly, whether j
0503         *ust one point is sufficient"4X""A2/
0504         *"  to give an accurate calibration curve fit?  I frankly doubt
0505         * it!"15X""A2)
0506      149 FORMAT ((3A2))
0507
0508      601 FORMAT (1H ,15(1H ),33HTranssonic Compressor Test Run # ,I7)
0509      602 FORMAT (1H ,28(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
0510      603 FORMAT (1H ,"",28(1H ),6HTime: ,A2,1H.,A2,3H  h,/////)
0511      604 FORMAT (1H ,"",/////,8X
0512         *"  On line Calibration Page "I2".  "
0513         *,/////)
0514      605 FORMAT (/"  "I1". reference pressure applied"/)
0515      606 FORMAT (/"  Calibration done! Take pacer run data".
0516         *""/)
0517      607 FORMAT (//1X"Calibration results:"/)
0518      608 FORMAT (1X"AVOLT("I1")="F8.6"  BVOLT("I1")="F8.6"  RPRES("I1")="
0519         *F8.6"  DMM("I1")="F8.6)
0520      609 FORMAT (/1X"Type 'A' Probe : "//
0521         *1X"SLOPE ="F10.6"  SECON ="F10.6/)
0522      610 FORMAT (/1X"Type 'B' Probe : "//
0523         *1X"SLOPE ="F10.6"  SECON ="F10.6/)
0524      611 FORMAT ("

",28(1H ),6HTime: ,A2,1H.,A2,3H  h)
```

```
      C     FORMATS CALIB STOP
      C
      CXXCC  Accounting.
      C
```

```
0533  C     ..................................................................
0534        IPAGE       = CNTRL(214)
0535        IPAGE       = IPAGE+1
0536
0537
0538
0539  C     ..................................................................
0540  CCCCC .                                                                .
0541  CCCCC .  I/O Assignments;  print heading.                              .
0542  CCCCC .                                                                .
0543  C     ..................................................................
0544        LI = CNTRL(19)
0545        LO = CNTRL(20)
0546        CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0547        WRITE (LO, 601) CNTRL(4)
0548        WRITE (LO, 602) IMON,IDAY,IYEAR
0549        WRITE (LO, 603) IHOUR,IMIN
0550        WRITE (LO, 604) IPAGE
0551        IPTS = 0
0552
0553
0554
0555  C     ..................................................................
0556  CCCCC .                                                                .
0557  CCCCC .  Limit on line calibration calibration to two A/D channels,    .
0558  CCCCC .  that contain the type 'A' and 'B' probe output voltage.       .
0559  CCCCC .  The current value of CNTRL(230) is temporaryly stored in      .
0560  CCCCC .  the variable ITEMP.                                           .
0561  CCCCC .                                                                .
0562  C     ..................................................................
0563        IPREV = CNTRL(230)
0564        CNTRL(230) = 2
0565
0566
0567
0568  C     ..................................................................
0569  CCCCC .                                                                .
0570  CCCCC .  Select the next step:                                         .
0571  CCCCC .    RE                Repeat this point                         .
0572  CCCCC .    EN                Terminate the on line calibration         .
0573  CCCCC .    any number        Proceed to the next point                 .
0574  CCCCC .                                                                .
0575  C     ..................................................................
0576     01 WRITE (LI, 101) (NOLF,I=1,5,1)
0577        READ  (LI, 102) IO
0578        WRITE (LI, 149) (ICLR,I=1,8,1)
0579        IF ( IO(1)  .EQ.  2HRE ) GO TO 01
0580        IF ( IO(1)  .EQ.  2HEN ) GO TO 10
0581        IF ( IO(1)  .EQ.  2H$$ ) STOP 0201
0582        IDUM = 0
0583        DO 02 I=1,5,1
0584     02 IF ( IO(I)  .NE.  2H   ) IDUM = 1
0585        IF ( IDUM   .EQ.  0    ) GO TO 01
0586        CALL CODE
0587        READ (IO, 103) DM
0588
0589
0590
0591  C     ..................................................................
0592  CCCCC .                                                                .
0593  CCCCC .  Take free run data at a defined reference pressure.           .
0594  CCCCC .                                                                .
0595  C     ..................................................................
0596     03 IPTS = IPTS+1
0597        DMM(IPTS) = DM
0598     04 WRITE (LI, 104)
0599        WRITE (LI, 105) NOCR
0600        READ  (LI, 102) IDUM
0601        WRITE (LI, 149) ICLR
0602        IF ( IDUM  .EQ.  2H$$ ) STOP 0202
0603        CNTRL(37) = 1
0604        CNTRL(38) = IPTS
0605        WRITE (LO, 605) IPTS
0606        CALL FREER (AVOLT(IPTS),BVOLT(IPTS),RPRES(IPTS))
0607
```

```
0608
0609
0610      C     ..................................................................
0611      C     .                                                                .
0612      C     .  Select the next step:                                         .
0613      C     .    RE                 Repeat this point                        .
0614      C     .    EN                 Terminate the on line calibration        .
0615      C     .    any number         Proceed to the next point                .
0616      C     .                                                                .
0617      C     ..................................................................
0618           05   WRITE (LI, 101) (NOLF,I=1,5,1)
0619                READ  (LI, 102) IO
0620                WRITE (LI, 149) (ICLR,I=1,8,1)
0621                IF ( IO(1)  .EQ.  2HRE ) GO TO 04
0622                IF ( IO(1)  .EQ.  2HEN ) GO TO 07
0623                IF ( IO(1)  .EQ.  2H$$ ) STOP 0203
0624                IDUM = 0
0625                DO 06 I=1,5,1
0626           06   IF ( IO(I)  .NE.  2H   ) IDUM = 1
0627                IF ( IDUM   .EQ.  0    ) GO TO 05
0628                CALL CODE
0629                READ (IO,103) DM
0630                GO TO 03
0631
0632
0633
0634      C     ..................................................................
0635      C     .                                                                .
0636      C     .  Take paced run data at one defined reference pressure.        .
0637      C     .                                                                .
0638      C     ..................................................................
0639           07 IF ( IPTS .EQ. 1 ) GO TO 11
0640              CNTRL(37) = 1
0641              CNTRL(38) = 1
0642              WRITE (LI, 106) NOCR
0643              READ  (LI, 102) IDUM
0644              WRITE (LI, 149) ICLR
0645              IF ( IDUM  .EQ.  2H$$ ) STOP 0204
0646              IREC = 1
0647              WRITE (LI, 107) IREC
0648              WRITE (LO, 606)
0649              CALL PACER (IREC)
0650
0651
0652
0653      C     ..................................................................
0654      C     .                                                                .
0655      C     .  Calculate linear curve fit through data points.               .
0656      C     .                                                                .
0657      C     ..................................................................
0658              WRITE (LO, 607)
0659              DO 08 I=1,IPTS,1
0660           08 WRITE (LO, 608) I,AVOLT(I),I,BVOLT(I),I,RPRES(I),I,DMM(I)
0661              CALL CURVE (IPTS,AVOLT,RPRES,SLOPE,SECON)
0662              WRITE (LO, 609) SLOPE,SECON
0663              CALL CURVE (IPTS,BVOLT,RPRES,SLOPE,SECON)
0664              WRITE (LO, 610) SLOPE,SECON
0665
0666
0667      C     ..................................................................
0668      C     .                                                                .
0669      C     .  Terminate subroutine;  write accounting variables back into   .
0670      C     .  control array;  set CNTRL(230) back to its previous value.    .
0671      C     .                                                                .
0672      C     ..................................................................
0673           09 CNTRL( 37) = -1
0674              CNTRL( 38) = 1
0675              CNTRL( 50) = -2
0676              CNTRL(214) = IPAGE
0677              CNTRL(230) = IPREV
0678              CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0679              WRITE (LO, 611) IHOUR,IMIN
0680              RETURN
0681
0682
```

```
0683
0684   CC     ..........................................................
0685   CCCC   :                                                        :
0686   CCCC   :  Error returns.                                       :
0687   CC     :                                                        :
0688   CC     ..........................................................
0689        10 WRITE (LI, 108) NOLF
0690           GO TO 09
0691        11 WRITE (LI, 109) (NOLF,I=1,2,1)
0692           GO TO 09
0693
0694
0695
0696           END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01525 COMMON = 00000

```
0697          SUBROUTINE FREER (AVOLT,BVOLT,PREFR)
0698    C     .....................................................................
0699    CCC   .                                                                   .
0700    CCC   .  Subroutine to acquire data using the HP 5610A A/D converter,    .
0701    CCC   .  if the A/D converter is operated in free run mode.               .
0702    CCC   .  Author:     Hans Zebner                                          .
0703    CCC   .  Date:       August 14, 1980                                      .
0704    CCC   .  A detailed program description is available in the TXCO log.    .
0705    CCC   .  Comment statements and statement numbers in the source code     .
0706    CCC   .  match to the program description.  This subroutine is part of   .
0707    CCC   .  the TXCO transonic compressor investigation program system.      .
0708    CC    .                                                                   .
0709    C     .....................................................................
0710         *, Takes data from KULITE probes; A/D free run mode.
0711
0712          COMMON / CIBUF / IBUF
0713          COMMON / CONTR / CNTRL
0714          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
0715
0716          INTEGER IBUF(1664)
0717          INTEGER CNTRL(256)
0718          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0719
0720          INTEGER NOLF,NOCR(2),ICLR(3),IOXIM(9)
0721
0722          DATA NOLF  /006537B/
0723          DATA NOCR  /0000338,040433B/
0724          DATA ICLR  /015524B,015515B,006537B/
0725          DATA FSVLT /1.00/
0726          DATA IDCBS /1441/
0727    C     FORMATS FREER START
0728      101 FORMAT ("0"27X"acquiring additional data"27X"0"A2)
0729      102 FORMAT ("B "F7.6,F9.6)
0730      103 FORMAT ("A "F7.6,F9.6)
0731      104 FORMAT (1BH       Wall KULITE)
0732      105 FORMAT ("0"28X"acquiring data from A/D"28X"0"A2)
0733      106 FORMAT ("0"24X"calculating the average voltage"24X"0"A2)
0734      107 FORMAT ("0"26X"storing data in file "3A2,26X"0"A2)
0735      108 FORMAT ("WARNING: file "3A2" already exists!  Type PU to "
0736         *"allow purge or enter any char-",
0737         */,"          acter but T to change file name."38X)
0738      109 FORMAT ("  FREER : PURGE "3A2":"A2":"A2)
0739      110 FORMAT (A1"2")
0740      111 FORMAT ("  FREER : File name "3A2" succesfully changed to "3A2)
0741      148 FORMAT (/3A2)
0742      149 FORMAT ((3A2))
0743
0744      601 FORMAT (1H ,15(1H ),33HTranssonic Compressor Test Run # ,I7)
0745      602 FORMAT (1H ,28(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
0746      603 FORMAT (1H ,"",28(1H ),6HTime: ,A2,1H.,A2,3H  h,/////)
0747      604 FORMAT (1H ,"",/////,8X
0748         *,"   Free Run Page "I3" .  "
0749         *////)
0750      605 FORMAT (/1X"0"72X"0"/1X"0"72X"0"/
0751         *          1X"          A/D input
0752         *                        "/
0753         *          1X" # sample  channel
0754         *              # points"/1X"0"72X"0"/
0755         *          1X"Combination probe:"54X"0"/
0756         *          1X"Immersion   yaw a.      p1       p23       p4
0757         *   Tt       dT                "/1X"0"72X"0"/
0758         *          1X"KULITE probe:     reference    rotor  average
0759         *              starting"/
0760         *          1X"Immersion   yaw a. pressure      RPM  voltage
0761         * data in file record #"/1X"0"72X"0"/)
0762      606 FORMAT (1H )
0763      607 FORMAT (1X"0"72X"0"/1X"0"2I9,45X,I9"0"/1X"0"7F9.6,"
0764         *"9X"0"/1X"0"9A2,F9.6,I9,F9.6,6X,3A2":"A2":"A2,I9"0"/
0765         *1X"0"72X"0")
0766      608 FORMAT ("
```

",28(1H),6HTime: ,A2,1H.,A2,3H h)

```
0767     901 FORMAT (" ERROR DETECTED IN PROGRAM FREER"/
0768    *            " CALL EXEC (1,20,IBUF(1),"I4","I2",4)")
0769     902 FORMAT (" A REGISTER IS "K6"      B REGISTER IS "K6/)
0770  C      FORMATS FREER STOP
0771
```

```
0772
0773
0774      C     .....................................................................
0775      CCCC  .                                                                   .
0776      CCCC  .  Accounting.                                                      .
0777      CCCC  .                                                                   .
0778      C     .....................................................................
0779            IPAGE     = CNTRL(216)
0780            IDOC      = CNTRL(217)
0781            IDOCF     = CNTRL(217)
0782            IPAGE     = IPAGE+1
0783            ISECU     = CNTRL(31)
0784            ICR       = CNTRL(30)
0785            ISIZE(1)  = 13
0786            ISIZE(2)  = 128
0787            IFILE(2)  = ICON(CNTRL(4),0)
0788            ITYPE     = 1
0789            IFRST     = 2HT2
0790            ISP       = 0
0791            IL        = 1664
0792
0793
0794
0795      C     .....................................................................
0796      CCCC  .                                                                   .
0797      CCCC  .  I/O Assignments;  preset data array.                             .
0798      CCCC  .                                                                   .
0799      C     .....................................................................
0800            DO 01 I=1,1664,1
0801         01 IBUF(I)   = 177777B
0802            LI        = CNTRL(19)
0803            LO        = CNTRL(20)
0804            LS1       = CNTRL(71)
0805            LS2       = CNTRL(72)
0806            ISV1      = CNTRL(61)
0807            ISV4      = CNTRL(64)
0808            NRPT1     = CNTRL(230)
0809            NRPT2     = CNTRL(251)
0810            NRPT3     = NRPT2+1
0811            IMASK     = 177700B
0812            IW        = CNTRL(250)
0813
0814
0815
0816      C     .....................................................................
0817      CCCC  .                                                                   .
0818      CCCC  .  Print heading, unless CNTRL(37) is set to 1.                     .
0819      CCCC  .                                                                   .
0820      C     .....................................................................
0821            IF ( CNTRL(37)  .EQ.  1 ) GO TO 02
0822            CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0823            WRITE (LO, 601) CNTRL(4)
0824            WRITE (LO, 602) IMON,IDAY,IYEAR
0825            WRITE (LO, 603) IHOUR,IMIN
0826            WRITE (LO, 604) IPAGE
0827         02 IF ( CNTRL(38)  .EQ.  1 ) WRITE (LO, 605)
0828
0829
0830
0831      C     .....................................................................
0832      CCCC  .                                                                   .
0833      CCCC  .  Start data acquisition loop.                                     .
0834      CCCC  .                                                                   .
0835      C     .....................................................................
0836            DO 19 J1=1,NRPT1,1
0837            WRITE (LI, 101) NOLF
0838            ICHNL = CNTRL(230+J1)
0839            IDOC  = IDOC+1
0840            IDOCF = IDOCF+1
0841            IF ( IDOCF  .LT.  100 ) GO TO 03
0842            IFRST    = 2HS2
0843            IDOCF    = IDOCF-100
0844         03 IFILE(1) = IFRST
0845            IFILE(3) = ICON(IDOCF,0)
0846
```

```
0847
0848
0849      C     .....
0850      C     :
0851      CCCC  :  Acquire additional data required for reduction procedure.
0852      C     :
0853      C     .....
0854            FREQ  = SCANR(LS1,17,2)
0855            FREQ  = FREQ*10.0
0856            CIM   = SCANR(LS1,30,1)
0857            CYAW  = SCANR(LS1,31,1)
0858            PREF  = SCANR(LS1,37,1)
0859            PREFR = PREF
0860            P1    = ACQN(ISV4, 3,IW)
0861            P23   = ACQN(ISV4, 4,IW)
0862            P4    = ACQN(ISV4, 5,IW)
0863            E     = SCANR(LS2,18,1)
0864            DE    = SCANR(LS2,19,1)
0865
0866
0867
0868      C     .....
0869      C     :
0870      CCCC  :  Get correct probe positions.
0871      C     :
0872      C     .....
0873            IF ( ICHNL .EQ. 0 ) GO TO 04
0874            IF ( ICHNL .EQ. 1 ) GO TO 05
0875            GO TO 06
0876
0877
0878
0879      C     .....
0880      C     :
0881      CCCC  :  Type 'A' KULITE probe (on A/D input channel 0).
0882      C     :
0883      C     .....
0884        04  XIM   = SCANR(LS1,32,1)
0885            YAW   = SCANR(LS1,33,1)
0886            CALL CODE
0887            WRITE (IOXIM,102) XIM,YAW
0888            GO TO 07
0889
0890
0891
0892      C     .....
0893      C     :
0894      CCCC  :  Type 'B' KULITE probe (on A/D input channel 1).
0895      C     :
0896      C     .....
0897        05  XIM   = SCANR(LS1,34,1)
0898            YAW   = SCANR(LS1,35,1)
0899            CALL CODE
0900            WRITE (IOXIM,103) XIM,YAW
0901            GO TO 07
0902
0903
0904
0905      C     .....
0906      C     :
0907      CCCC  :  Wall KULITE (on A/D input channels 2 and higher).
0908      C     :
0909      C     .....
0910        06  CALL CODE
0911            WRITE (IOXIM,104)
0912
0913
0914
0915      C     .....
0916      C     :
0917      CCCC  :  Preset unused elements of data array IBUF.
0918      C     :
0919      C     .....
0920        07  IF ( NRPT2 .EQ. 1664 ) GO TO 09
0921            DO 08 J2=NRPT3,1664,1
```

```
0922      08    IBUF(J2) = 000000B
0923
0924
0925
0926  C          .....................................................
0927  C          .                                                   .
0928  C          . Read voltages from A/D converter into data array IBUF;
0929  C          . Starting element is IBUF(1);  A/D is operated in the
0930  C          . free run mode (4).                                .
0931  C          .                                                   .
0932  C          .....................................................
0933      09    WRITE (LI, 105) NOLF
0934            CALL EXEC (1+100000B,20,IBUF(1),NRPT2,ICHNL,4)
0935            GO TO 11
0936      10    GO TO 12
0937      11    CALL ABREG (IA,IB)
0938            GO TO 21
0939
0940
0941
0942  C          .....................................................
0943  C          .                                                   .
0944  C          . Calculate average voltage.                        .
0945  C          .                                                   .
0946  C          .....................................................
0947      12    WRITE (LI, 106) NOLF
0948            DO 13 J2=1,NRPT2,1
0949      13    IBUF(J2)=IAND(IBUF(J2),IMASK)
0950            AVRGE = 0.0
0951            DO 14 J2=1,NRPT2,1
0952      14    AVRGE = AVRGE+FLOAT(IBUF(J2))
0953            AVRGE = FSVLT*((AVRGE/32768.0)/NRPT2)
0954
0955
0956
0957  C          .....................................................
0958  C          .                                                   .
0959  C          . Write average KULITE voltage (after amplification and
0960  C          . conversion to digital notation) into AVOLT and BVOLT, de-
0961  C          . pending on what signal has been digitized;  The type 'A'
0962  C          . probe is on A/D input channel 0 and the type 'B' on 1.
0963  C          .                                                   .
0964  C          .....................................................
0965            IF ( ICHNL  .EQ.  0 ) AVOLT = AVRGE
0966            IF ( ICHNL  .EQ.  1 ) BVOLT = AVRGE
0967
0968
0969
0970          IF ( ISP  .LT.  5 ) GO TO 15
0971          ISP   = 0
0972          IF ( CNTRL( 37)  .NE.  1 ) WRITE (LO, 606)
0973      15  ISP   = 1+ISP
0974
0975
0976
0977  C          .....................................................
0978  C          .                                                   .
0979  C          . Save data in file.                                .
0980  C          .                                                   .
0981  C          .....................................................
0982      16  WRITE (LI, 107) IFILE,NOLF
0983          CALL CREAT (IDCB,IERR,IFILE,ISIZE,ITYPE,ISECU,ICR,IDCBS)
0984          IF ( IERR  .GT.  0 ) GO TO 18
0985          WRITE (LI, 108) IFILE
0986          READ  (LI, 149) IDUM
0987          WRITE (LI, 149) (ICLR,I=1,3,1)
0988          IF ( IDUM  .EQ.  2H$$ ) STOP 0301
0989          IF ( IDUM  .NE.  2HPU ) GO TO 17
0990          ISECU = ICON(ISECU,0)
0991          JCR   = ICON(ICR,0)
0992          WRITE (LI, 109) IFILE,ISECU,ICR
0993          CALL PURGE (IDCB,IERR,IFILE,ISECU,ICR)
0994          IF ( IERR  .LT.  0 ) STOP 0302
0995          GO TO 16
0996      17  CALL CODE
```

```
0997            WRITE (NEW,110) IDUM
0998            WRITE (LI, 111) IFILE,NEW,IFILE(1),IFILE(2)
0999            IFILE(1) = NEW
1000            GO TO 16
1001         18 CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
1002            IF ( IERR .LT. 0 ) STOP 0303
1003            CALL WRITF (IDCB,IERR,IBUF,IL)
1004            IF ( IERR .LT. 0 ) STOP 0304
1005            CALL CLOSE (IDCB,IERR,0)
1006            IF ( IERR .LT. 0 ) STOP 0305
1007            WRITE (LI, 148) ICLR
1008            JSECU = ICON(ISECU,0)
1009            JCR   = ICON(ICR,0)
1010
1011
1012
1013 CCCCC  .....................................................................
1014 CCCCC  .                                                                   .
1015 CCCCC  .  Step data acquisition loop.                                      .
1016 CCCCC  .                                                                   .
1017 CCCCC  .....................................................................
1018         19 WRITE (LO, 607) IDOC,ICHNL,NRPT2,CIN,CYAW,P1,P23,P4,E,DE,IOXIN,P
1019            *  REF,FREQ,AVRGE,IFILE,JSECU,JCR,IREC
1020
1021
1022
1023 CCCCC  .....................................................................
1024 CCCCC  .                                                                   .
1025 CCCCC  .  Terminate subroutine;  write accounting variables back          .
1026 CCCCC  .  into control array.                                             .
1027 CCCCC  .                                                                   .
1028 CCCCC  .....................................................................
1029            IF ( CNTRL(37) .EQ. 1 ) GO TO 20
1030            CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
1031            WRITE (LO, 608) IHOUR,IMIN
1032         20 CNTRL( 37) = -3
1033            CNTRL( 38) = 1
1034            CNTRL( 50) = -3
1035            CNTRL(216) = IPAGE
1036            CNTRL(217) = IDOC
1037            RETURN
1038
1039
1040
1041 CCCCC  .....................................................................
1042 CCCCC  .                                                                   .
1043 CCCCC  .  Error returns from EXEC calls;  output error message to the    .
1044 CCCCC  .  line printer and look what's in the A and B register.            .
1045 CCCCC  .                                                                   .
1046 CCCCC  .....................................................................
1047         21 WRITE ( 6, 901) NRPT2,ICHNL
1048            WRITE ( 6, 902) IA,IB
1049            GO TO 20
1050
1051
1052
1053            END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01829 COMMON = 00000

```
1054          SUBROUTINE PACER (IREC)
1055    C     .....................................................
1056    C     .                                                   .
1057    C     .  Subroutine to acquire data using the HP 5610A A/D converter,  .
1058    C     .  if the A/D converter is triggered through the pacer (paced    .
1059    C     .  run mode).                                       .
1060    C     .  Author:     Hans Zebner                          .
1061    C     .  Date:       August 25, 1980                      .
1062    C     .  A detailed program description is available in the TXCO log.  .
1063    C     .  Comment statements and statement numbers in the source code   .
1064    C     .  match to the program description.  This subroutine is part of .
1065    C     .  the TXCO transonic compressor investigation program system.    .
1066    C     .                                                   .
1067    C     .....................................................
1068          *, Takes data from KULITE probes; A/D paced run mode.

1069          COMMON / CIBUF / IBUF
1070          COMMON / CONTR / CNTRL
1071          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
1072
1073          INTEGER IBUF(1664)
1074          INTEGER CNTRL(256)
1075          INTEGER IDCB(144),IFILE(3),ISIZE(2)
1076
1077          INTEGER NOLF,NOCR(2),ICLR(3),IBUF1(384),IOXIM(9),IAVRGE(2)
1078
1079          EQUIVALENCE (IBUF(1),IBUF1(1)),(IAVRGE(1),AVRGE)
1080
1081          DATA NOLF  /006537B/
1082          DATA NOCR  /000033B,040433B/
1083          DATA ICLR  /015524B,015515B,006537B/
1084          DATA IDCBS /144/
1085    C     FORMATS  PACER  START
1086      101 FORMAT ("1X"acquiring additional data required for reductio"
1087         *"n procedure"10X"A2)
1088      102 FORMAT ("A"F8.6,F9.6)
1089      103 FORMAT ("B"F8.6,F9.6)
1090      104 FORMAT (18H       Wall KULITE)
1091      105 FORMAT ("/""79X""/" The next signal to be digitized
1092         * in paced run mode is on A/D input channel"I3". "/
1093         *" Plug in a lead from the amplifier output to the oscilloscope
1094         *, if you desire    "/
1095         *" to monitor the data acquisition.  Don't forget the c
1096         *omparator output signal!  "/""79X""/
1097         *" Press CR to continue the program execution!"35X"
1098         *"/""79X""/)
1099      106 FORMAT ("1X"displaying wave form on terminal LU#"I2,20X"
1100         *"A2)
1101      107 FORMAT ("/""79X""/" CHECK digitized output on a
1102         *uxiliary console against amplifier output fed into "/
1103         *" A/D input channel "I2" .  Press CR if data a
1104         *re OK.  If an error is suspected    "/
1105         *" type RE to repeat this data acquisition!"38X""/
1106         *""79X""/)
1107      108 FORMAT (""26X"storing data in file "3A2,26X""A2)
1108      109 FORMAT ("WARNING: File "3A2" already exists!  Type ",
1109         *"PU to allow purge or enter any char-",
1110         */"          acter but T to change file name."38X)
1111      110 FORMAT ("  PACER : PURGE "3A2": "A2": "A2)
1112      111 FORMAT (A1"3")
1113      112 FORMAT ("  PACER : File name "3A2" succesfully changed to "3A2)
1114      113 FORMAT (I5,A2)
1115      148 FORMAT (//3A2))
1116      149 FORMAT (3A2)
1117      601 FORMAT (1H ,15(1H ),33HTranssonic Compressor Test Run # ,I7)
1118      602 FORMAT (1H ,28(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
1119      603 FORMAT (1H ,"",28(1H ),6HTime: ,A2,1H:,A2,3H  h,/////)
1120      604 FORMAT (1H ,"",////,8X
1121         *,"  Paced Run Page "I3".  "
1122         */////)
1123      605 FORMAT (/1X""72X""/1X""72X""/
1124         *          1X"                A/D input    pacer    blade    start   in
1125         *cre-      step  # repe-"/
1126         *          1X" # sample  channel     mode     pair     count
1127         *ment     count  titions"/1X""72X""/
1128         *          1X"Combination probe: "54X""/
```

```
      *           1X"immersion   yaw a.        p1       p23         p4
      *  Tt         dT           "/1X""72X""/
      *           1X"Probe                reference     rotor  average
      *                starting"/
      *           1X"immersion   yaw a. pressure         RPM  voltage
      * data in file record #"/1X""72X""/)
  609 FORMAT (1H )
  610 FORMAT (1X""72X""/1X""8I9""/1X""7F9.6,9X""
     */" "9A2,F9.6,I9,F9.6,6X,3A2":"A2":"A2,I9""/" "72X"
     *")
  611 FORMAT ("
```

```
",20(1H ),6HTime: ,A2,1H.,A2,3H  h)
```

```
C     FORMATS PACER STOP



C       ..........................................................................
C       :                                                                        :
C       :  Accounting.                                                           :
C       :                                                                        :
C       ..........................................................................
        IF ( CNTRL(39)  .EQ.  1 ) GO TO 01
        IPAGE      = CNTRL(218)
        IDOC       = CNTRL(219)
        IDOCF      = CNTRL(219)
        IPAGE      = IPAGE+1
        ISECU      = CNTRL(31)
        ICR        = CNTRL(30)
        ISIZE(1)   = 3
        ISIZE(2)   = 128
        IFILE(2)   = ICON(CNTRL(4),0)
        ITYPE      = 1
        IFRST      = 2HT3
     01 ISP        = 0
        IL         = 384



C       ..........................................................................
C       :                                                                        :
C       :  I/O Assignments;  preset data array.                                  :
C       :                                                                        :
C       ..........................................................................
        DO 02 I=1,384,1
     02 IBUF1(I) = 177777B
        DO 03 I=265,296,1
     03 IBUF1(I) = 0
        LI         = CNTRL(19)
        LO         = CNTRL(20)
        LA         = CNTRL(21)
        LS1        = CNTRL(71)
        LS2        = CNTRL(72)
        ISV1       = CNTRL(61)
        ISV4       = CNTRL(64)
        NRPT1      = CNTRL(230)
        IW         = CNTRL(250)



C       ..........................................................................
C       :                                                                        :
C       :  Print heading, unless CNTRL(37) is set to 1.                          :
C       :                                                                        :
C       ..........................................................................
        IF ( CNTRL(37)  .EQ.  1 ) GO TO 04
        CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
        WRITE (LO, 601) CNTRL(4)
        WRITE (LO, 602) IMON,IDAY,IYEAR
        WRITE (LO, 603) IHOUR,IMIN
        WRITE (LO, 604) IPAGE
     04 IF ( CNTRL(38)  .EQ.  1 ) WRITE (LO, 605)



C         ........................................................................
C         :                                                                      :
```

```
1204   C        :  Start data acquisition loop.                              :
1205   CC       :                                                            :
1206   C        :............................................................:
1207             DO 19 Ji=1,NRPTi,1
1208             WRITE (LI,101) NOLF
1209             ICHNL = CNTRL(230+Ji)
1210             IF ( CNTRL(39)  .EQ.  1 ) GO TO 06
1211             IDOC  = IDOC+1
1212             IDOCF = IDOCF+1
1213             IF ( IDOCF  .LT.  100 ) GO TO 05
1214             IFRST = 2HS3
1215             IDOCF = IDOCF-100
1216       05    IFILE(1) = IFRST
1217             IFILE(3) = ICON(IDOCF,0)
1218
1219
1220
1221   C        ..............................................................
1222   CCC      :                                                            :
1223   CCCC     :  Acquire additional data required for reduction procedure. :
1224   CCC      :                                                            :
1225   C        :............................................................:
1226       06    FREQ  = SCANR(LSi,17,2)
1227             FREQ  = FREQ*10.0
1228             CIM   = SCANR(LSi,30,1)
1229             CYAW  = SCANR(LSi,31,1)
1230             PREF  = SCANR(LSi,37,1)
1231
1232             P1    = ACQN(ISV4, 3,IW)
1233             P23   = ACQN(ISV4, 4,IW)
1234             P4    = ACQN(ISV4, 5,IW)
1235             E     = SCANR(LS2,18,1)
1236             DE    = SCANR(LS2,19,1)
1237
1238
1239
1240   C        ..............................................................
1241   CCC      :                                                            :
1242   CCCC     :  Get correct probe positions.                              :
1243   CCC      :                                                            :
1244   C        :............................................................:
1245             IF ( ICHNL  .EQ.  0 ) GO TO 07
1246             IF ( ICHNL  .EQ.  1 ) GO TO 08
1247             GO TO 09
1248
1249
1250
1251   C        ..............................................................
1252   CCC      :                                                            :
1253   CCCC     :  Type 'A' KULITE probe (on A/D input channel 0).           :
1254   CCC      :                                                            :
1255   C        :............................................................:
1256       07    XIM   = SCANR(LSi,32,1)
1257             YAW   = SCANR(LSi,33,1)
1258             CALL CODE
1259             WRITE (IOXIM,102) XIM,YAW
1260             IADD = 0
1261             IBUF1(280) = 1
1262             GO TO 10
1263
1264
1265
1266   C        ..............................................................
1267   CCC      :                                                            :
1268   CCCC     :  Type 'B' KULITE probe (on A/D input channel 1).           :
1269   CCC      :                                                            :
1270   C        :............................................................:
1271       08    XIM   = SCANR(LSi,34,1)
1272             YAW   = SCANR(LSi,35,1)
1273             CALL CODE
1274             WRITE (IOXIM,103) XIM,YAW
1275             IADD = 576
1276             IBUF1(280) = 2
1277             GO TO 10
1278
```

```
1279
1280
1281 C     .....................................................................
1282 C     .                                                                   .
1283 C     .  Wall KULITE (on A/D input channels 2 and higher).                .
1284 C     .                                                                   .
1285 C     .....................................................................
1286       09   CALL CODE
1287            WRITE (IOXIM,104)
1288            IBUF1(280) = 9
1289            XIM = -99.
1290            YAW = -99.
1291
1292
1293
1294 C     .....................................................................
1295 C     .                                                                   .
1296 C     .  Inform operator about next data scan;  wait for ready            .
1297 C     .  message;  continue.                                              .
1298 C     .                                                                   .
1299 C     .....................................................................
1300       10   WRITE (LI, 105) ICHNL
1301            READ  (LI, 149) IDUM
1302            WRITE (LI, 149) (ICLR,I=1,10,1)
1303            IF ( IDUM  .EQ.  2H$$ ) STOP 0401
1304
1305
1306
1307 C     .....................................................................
1308 C     .                                                                   .
1309 C     .  Acquire high speed data in paced run mode.                       .
1310 C     .                                                                   .
1311 C     .....................................................................
1312            IPAMO = CNTRL(220)
1313            IPAIR = CNTRL(221)
1314            IF ( IPAMO  .EQ.  2 ) ISTART = CNTRL(222)+IADD
1315            IF ( IPAMO  .EQ.  2 ) ISTOP  = CNTRL(224)+IADD
1316            INCR  = CNTRL(223)
1317            IRPT  = CNTRL(225)
1318            CALL RPACE (ICHNL,IPAMO,IPAIR,ISTART,INCR,ISTOP,IRPT,AVRGE,0,0)
1319
1320
1321
1322 C     .....................................................................
1323 C     .                                                                   .
1324 C     .  Write additional data into data array.                           .
1325 C     .                                                                   .
1326 C     .....................................................................
1327            IBUF1(265) = IDOC
1328            IBUF1(266) = ICHNL
1329            IBUF1(267) = IPAMO
1330            IBUF1(268) = IPAIR
1331            IBUF1(269) = ISTART
1332            IBUF1(270) = INCR
1333            IBUF1(271) = ISTOP
1334            IBUF1(272) = IRPT
1335
1336            IBUF1(273) = CIM  * 1000000.
1337            IBUF1(274) = CYAW * 1000000.
1338            IBUF1(275) = P1   * 1000000.
1339            IBUF1(276) = P23  * 1000000.
1340            IBUF1(277) = P4   * 1000000.
1341            IBUF1(278) = E    * 1000000.
1342            IBUF1(279) = DE   * 1000000.
1343
1344            IBUF1(281) = XIM  * 1000000.
1345            IBUF1(282) = YAW  * 1000000.
1346            IBUF1(283) = PREF * 1000000.
1347            IBUF1(284) = RPM
1348
1349            IBUF1(290) = CNTRL(4)
1350            IBUF1(291) = CNTRL(5)
1351            IBUF1(292) = CNTRL(6)
1352
1353            IBUF1(293) = IAVRGE(1)
```

```
1354            IBUF1(294) = IAVRGE(2)
1355
1356            CALL TIME (IBUF1(352),IBUF1(360),IBUF1(368),IBUF1(376),IBUF1(384
1357          *  ))
1358
1359
1360
1361    C       .....................................................................
1362    CCCCC   .                                                                   .
1363    CCCCC   .  Display the just acquired wave on terminal CRT, if               .
1364    CCCCC   .  CNTRL(40) is set to 1.  The character used for the "plot"        .
1365    CCCCC   .  is defined by CNTRL(249).                                        .
1366    CCCCC   .                                                                   .
1367    C       .....................................................................
1368            WRITE (LI, 148) ICLR
1369            WRITE (LI, 149) ICLR
1370            IF ( CNTRL(40)  .NE.  1 ) GO TO 11
1371            WRITE (LI, 106) LA,NOLF
1372            J111 = 1
1373            J222 = 1
1374            CALL PICTR (LA,IDOC,J111,J222,CNTRL(249),DUM)
1375
1376
1377
1378    C       .....................................................................
1379    CCCCC   .                                                                   .
1380    CCCCC   .  Select the next step:                                            .
1381    CCCCC   .     RE              Repeat this point                             .
1382    CCCCC   .     anything else   Proceed to the next point                     .
1383    CCCCC   .                                                                   .
1384    C       .....................................................................
1385            WRITE (LI, 107) ICHNL
1386            READ  (LI, 149) IDUM
1387            WRITE (LI, 149) (ICLR,I=1,8,1)
1388            IF ( IDUM  .EQ.  2H$$ ) STOP 0402
1389            IF ( IDUM  .EQ.  2HRE ) GO TO 06
1390
1391        11  IF ( ISP  .LT. 5 ) GO TO 12
1392            ISP   = 0
1393            IF ( CNTRL(37)  .NE.  1 ) WRITE (LO, 609)
1394        12  ISP   = ISP+1
1395
1396
1397
1398    C       .....................................................................
1399    CCCCC   .                                                                   .
1400    CCCCC   .  Save data in file. There are two options. The raw data file      .
1401    CCCCC   .  is either created/opened and closed by subroutine PACER          .
1402    CCCCC   .  (CNTRL(39) is set to anything but to 1) or this subroutine      .
1403    CCCCC   .  is called from subroutine ABSRV, which already has created/     .
1404    CCCCC   .  opened and positioned the raw data file and will close it       .
1405    CCCCC   .  (CNTRL(39) is set to 1).  In both cases the raw data are        .
1406    CCCCC   .  written in file by this subroutine PACER.                        .
1407    CCCCC   .                                                                   .
1408    C       .....................................................................
1409        13  WRITE (LI, 108) IFILE,NOLF
1410            JSECU = ICON(ISECU,0)
1411            JCR   = ICON(ICR,0)
1412            IBUF1(257) = IFILE(1)
1413            IBUF1(258) = IFILE(2)
1414            IBUF1(259) = IFILE(3)
1415            IBUF1(261) = JSECU
1416            IBUF1(263) = JCR
1417            IF ( CNTRL(39)  .EQ.  1 ) GO TO 16
1418            CALL CREAT (IDCB,IERR,IFILE,ISIZE,ITYPE,ISECU,ICR,IDCBS)
1419            IF ( IERR  .GT.  0 )  GO TO 15
1420            WRITE (LI, 109) IFILE
1421            READ  (LI, 149) IDUM
1422            WRITE (LI, 149) (ICLR,I=1,3,1)
1423            IF ( IDUM  .EQ.  2H$$ ) STOP 0403
1424            IF ( IDUM  .NE.  2HPU ) GO TO 14
1425            WRITE (LI, 110) IFILE,JSECU,JCR
1426            CALL PURGE (IDCB,IERR,IFILE,ISECU,ICR)
1427            IF ( IERR  .LT.  0 ) STOP 0404
1428            GO TO 13
```

```
1429        14   CALL CODE
1430             WRITE (NEW,111) IDUM
1431             WRITE (LI, 112) IFILE,NEW,IFILE(2),IFILE(3)
1432             IFILE(1) = NEW
1433             GO TO 13
1434        15   CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
1435             IF ( IERR  .LT.  0 ) STOP 0405
1436             IREC     = 1
1437             GO TO 17
1438        16   IF ( ICHNL  .EQ.  0 ) IREC = IREC
1439             IF ( ICHNL  .EQ.  1 ) IREC = IREC+3
1440        17   IBUF1(289) = IREC
1441             CALL WRITF (IDCB,IERR,IBUF1,IL,IREC)
1442             IF ( IERR       .LT.  0 ) STOP 0406
1443             IF ( CNTRL(39)  .EQ.  1 ) GO TO 18
1444             CALL CLOSE (IDCB,IERR,0)
1445             IF ( IERR  .LT.  0 ) STOP 0407
1446             JSECU    = ICON(ISECU,0)
1447             JCR      = ICON(ICR,0)
1448        18   WRITE (LI, 148) ICLR
1449
1450
1451
1452   C         .....................................................................
1453   CCCC      :                                                                   :
1454   CCCCC     :  Step data acquisition loop.                                      :
1455   CCCC      :                                                                   :
1456   C         .....................................................................
1457        19   WRITE (LO, 610) (IBUF1(J2),J2=265,272,1),CIM,CYAW,P1,P23,P4,E,DE
1458          *  ,IOXIM,PREFR,FREQ,AVRGE,IFILE,JSECU,JCR,IREC
1459
1460
1461
1462   C         .....................................................................
1463   CCCCC     :                                                                   :
1464   CCCCCC    :  Terminate subroutine;  write accounting data back                :
1465   CCCCC     :  into control array.                                              :
1466   CCCC      :                                                                   :
1467   C         .....................................................................
1468             IF ( CNTRL(37)  .EQ.  1 ) GO TO 20
1469             CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
1470             WRITE (LO, 611) IHOUR,IMIN
1471        20   CNTRL( 37) = -4
1472             CNTRL( 38) = 1
1473             CNTRL( 39) = -4
1474             CNTRL( 50) = -4
1475             CNTRL(218) = IPAGE
1476             CNTRL(219) = IDOC
1477             RETURN
1478
1479
1480
1481             END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 02561 COMMON = 00000

5. PROGRAM TXCO2

5.1. DESCRIPTION

TXCO2 is a son program of the father program TXCOØ, by which it is scheduled if one of the following operations is desired:

5 - Radial survey using the combination probe

6 - Scan through all steady state data

When scheduled by TXCOØ, which suspends operation while the son program TXCO2 executes, the program TXCO2 reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control and programs the digital voltmeter (DVM), the scanners and the counter. CNTRL(50) is the actual decision variable to select and call the subroutine, which performs the desired operation. When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note that each stop code ending in 77 indicates correct execution of a subroutine.

CNTRL(50)	Subroutine	STOP Code
5	COMB	TXCO2 : STOP 0577
6	STDY	TXCO2 : STOP 0677

Any other STOP code indicates a mistake and with the help of a program list the operator can trace the problem. The first two digits of the STOP code identify the subroutine. An example: if the program stops at STOP code 0604, the first

two digits read 6 and tells the operator that it was subroutine STDY which encountered problems. The last two digits read 04 (no error would give 77). A program list reveals that the failure occurred after attempting to purge an existing data file using FMP (File Management Package) subroutine PURGE near line 752. Maybe the cartridge, where the raw data are directed, has not been mounted with the MC-command from FMGR. STOP codes are crucial to a complex program system in order to rapidly detect and salvage problems, even during a test run.

EXTERNALS: REWRF, ABERT, RMOTE, COMB, STDY, CLEAR, LOCL

COMMON BLOCKS: FMP, CIBUF, CONTR

FORTRAN conventions for the HP 21 MX computer request COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

BLOCK DATA Subroutine	Arrays & Variables	Length in Words
FMP	IDCB,IFILE,ISIZE,ISECIA,ICR	$227B = 151_{10}$
CIBUF	IBUF	3200B = 1664
CONTR	CNTRL	400B = 256

The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. COMMON block CIBUF is designed to take the largest raw data array in the TXCO data acquisition and reduction system - IBUF(1664) in subroutine FREER. The largest data array in TXCO2 is PDAT (24, 21) with 1008 = 2*21*24 words. The TXCO2 subroutines only partially use the COMMON area. The COMMON

block CONTR allocates the space for the control array CNTRL. Since each individual subroutine saves the data prior to terminating, the buffer area for the raw data can be shared by more than one subroutine or function.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL(50) is less than 5 or greater than 6, no subroutine can be selected and the program terminates outputting an error message (FORMAT 102) to the standard input device; i.e. the terminal.

PROCEDURE: For more detailed information study the flow chart and the information given in the subroutine descriptions.

DATA FILE: None

VARIABLES IN BLOCK DATA FMP:

IDCB (144)	integer	data control block
IFILE (3)	integer	array to contain file name
ISIZE (2)	integer	array to contain # of records in the first and record length in the second 16-bit word
ISECU	integer	security code of data file
ICR	integer	cartridge reference number, where data file is located

VARIABLES IN BLOCK DATA CIBUF:

IBUF (1664)	integer	buffer array for the raw data

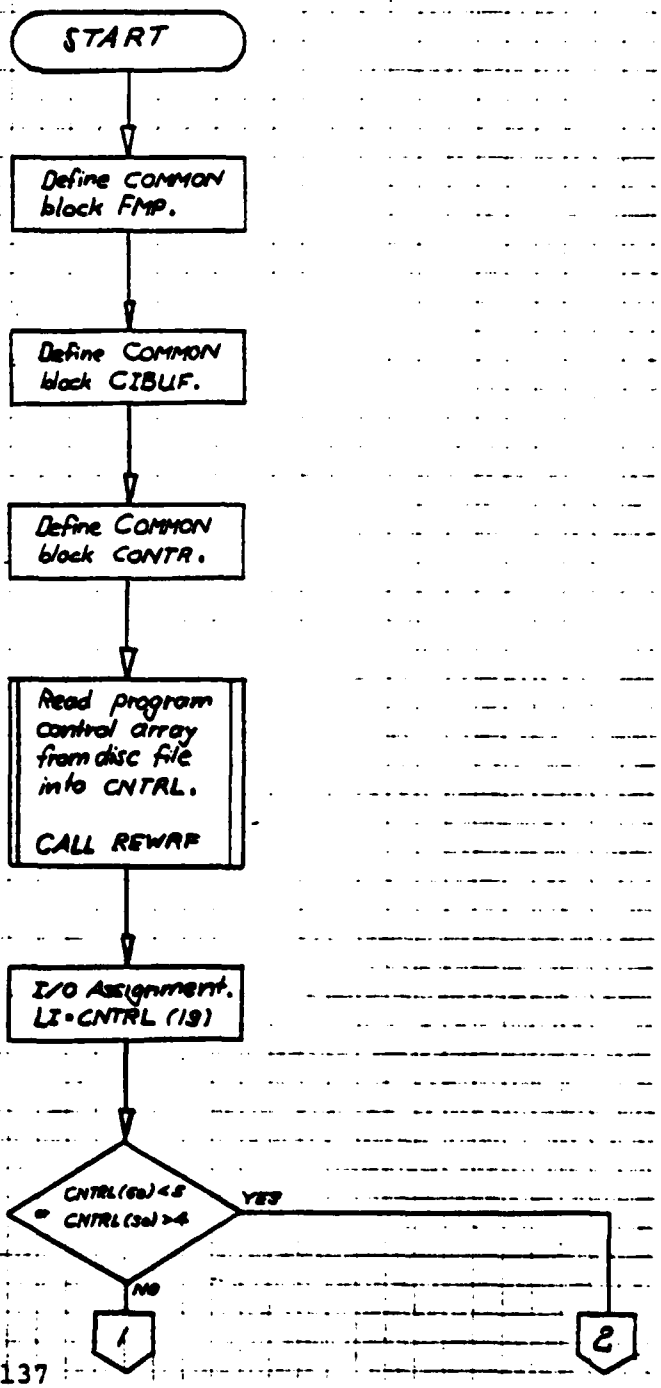
VARIABLES IN BLOCK DATA CONTR:

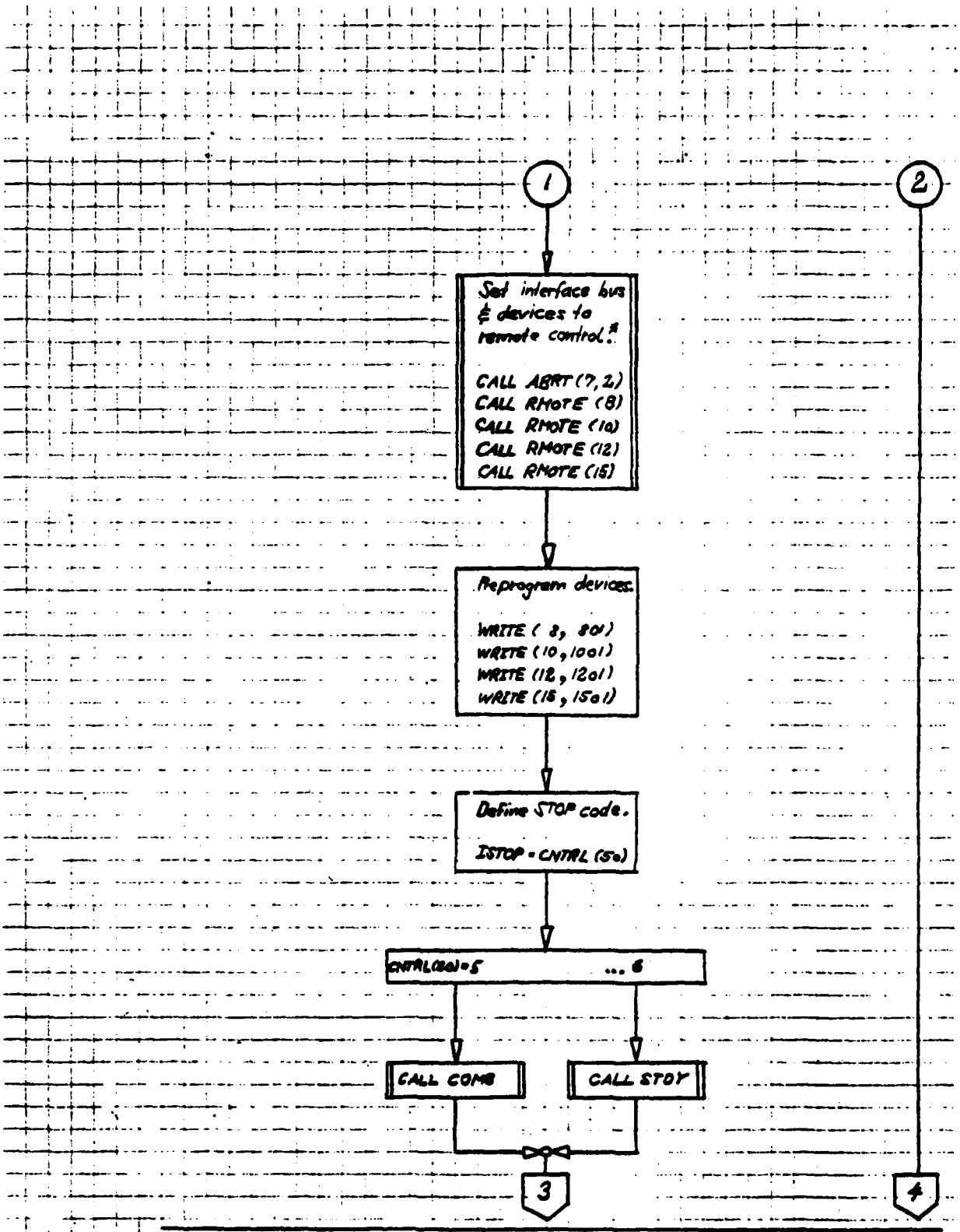
CNTRL (256)	integer	program control array

VARIABLES IN PROGRAM TXCO2:

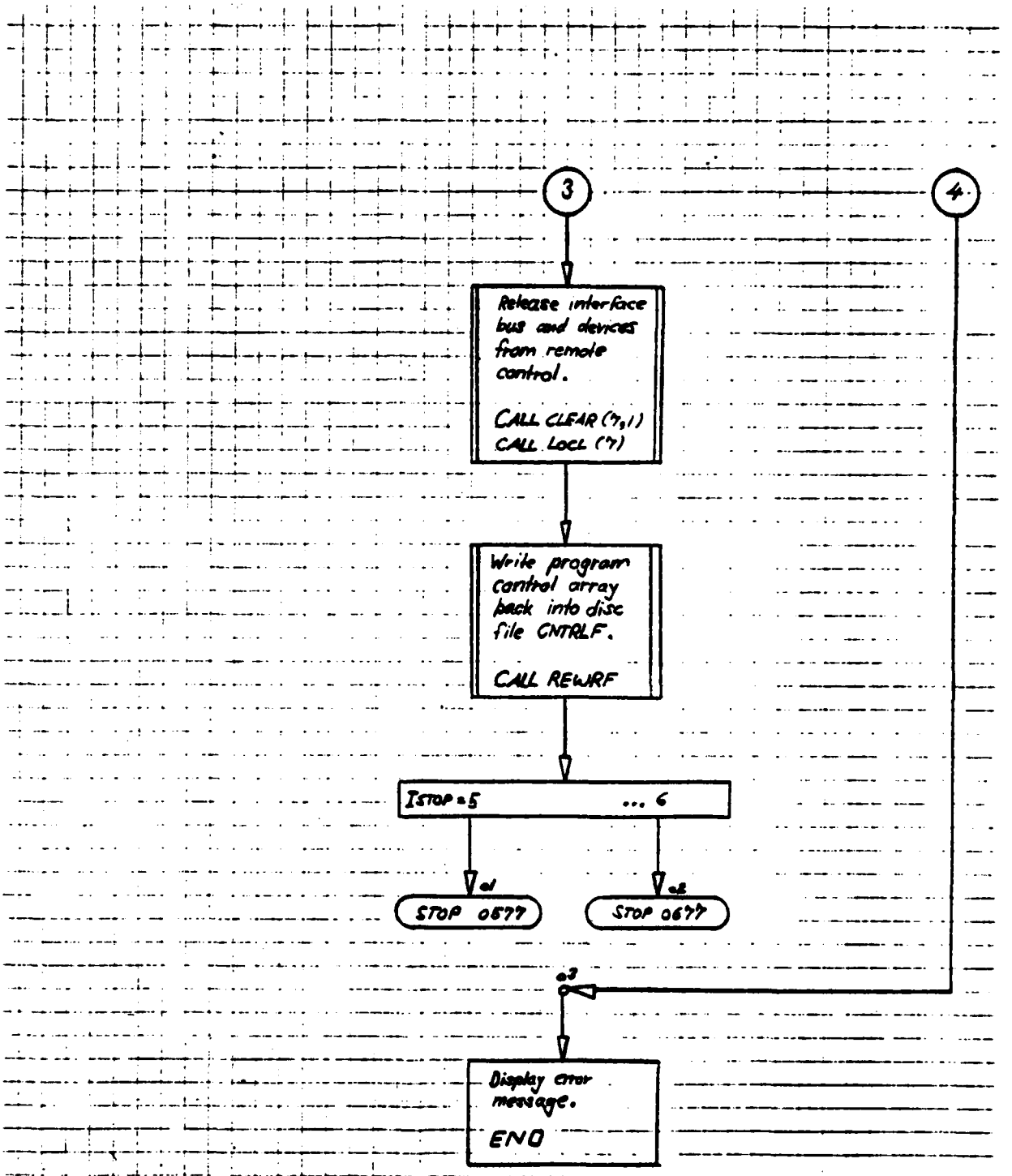
CNTRL (256)	integer	program control array
NOLF	integer	suppresses line feed
LI	integer	LU# of standard input device (terminal)
ISTOP	integer	control variable to select STOP code

FLOW CHART PROGRAM TXCO2:





* LU ASSIGNMENTS: 8 Scanner #1
10 Digital Voltmeter (DVM)
12 Digital Counter
15 Scanner #2
7 HP. Interface bus


3
4
Release interface bus and devices from remote control.
CALL CLEAR (7,1)
CALL LOCL (7)
Write program control array back into disc file CNTRLF.
CALL REWRF
ISTOP = 5
... 6
STOP 0677
STOP 0677
Display error message.
END

5.2. SUBROUTINE COMB:

PURPOSE: Acquisition of flow data from the transonic 1-stage axial compressor using a pneumatic 4-hole combination probe. The data necessary for the reduction procedure (PROGRAM REDCO: Reduce Combination probe data) are recorded also. Up to 24 different radial positions can be recorded. Taking more than one scan at one and the same radial position should be avoided, because the reduction program (originally written by R. Shreeve for the Laboratory's HP 9830 calculator and rewritten by F. Neuhoff for the more advanced HP 21 MX computer) is not set up for this condition.

ARGUMENTS: None

EXTERNALS: TIME, SCANR, ACQN, CREAT, PURGE, OPEN, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed explanation refer to the TXCO2 description.

MNEMONIC ABBREVIATIONS:

RE ... Repeat data acquisition at this radial position.

NE ... Proceed to the next radial position.

EN ... End survey at this operating condition.

UP ... Update position readings of probes prior to data taking.

TA ... Initialisation command to take data.

TR ... Transfer raw data to HP 9830.

ST ... Store raw data in 21 MX disc file.

PU ... Allow purge of an existing data file.

ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the accounting data, assigned the I/O references, COMB asks the operator whether the radial survey takes place ahead of the (IPOS = 1) rotor or behind the (IPOS = 2) rotor. IPOS later will be used to identify the correct data port (see Appendix A.1: Data Locations). The raw data array IBUF - which is set equivalent to real array PDAT (Prob Data; used instead of IBUF, which is an integer array) is first preset with zeroes. Before the operator goes ahead and allows the subroutine to gather data (Input: TA; see key to raw data array), he can monitor the probe positions by updating its reading (Input: UP), until the probe is manually set to the desired position. Upon completion of the data scan the acquired data are printed and the next step depends on the operator's decision. If a preliminary check reveals erroneous data, the scan at this radial position should be repeated (Input: RE). If the data are correct, the operator either proceeds to the next radial position (Input: NE) or terminates the radial combination probe survey (Input: EN) at this operating condition. The subroutine then asks where to dump the data. When this routine was developed the data reduction program for the combination probe was not available in the 21 MX system, hence the option to transfer the data to the 9830 calculator (Input: TR) was used. But the data can as well be stored in a 21 MX disc file (Input: ST). If the raw data file with the automatically

determined name already exists, the operator either allows overwriting the existing file (Input: PU) or renames the current data file (Input: any alphabetic character other than T). The subroutine terminates printing the data file name at the bottom of the data documentation page.

DATA FILE: The default file name is T5rrss (rr ... ASCII converted run #; ss ... ASCII converted sequential #).

VARIABLES:

IBUF (1664)	integer	buffer array
CNTRL (256)	integer	control array
IDCB (144)	integer	data control block, used for FMP calls
IFILE (3)	integer	array to contain current file name
ISIZE (2)	integer	specifies # of records and record length
ISECU	integer	security code of data file
ICR	integer	cartridge reference number, where the raw data file is located
JSECU	integer	ASCII converted security code
JCR	integer	ASCII converted cartridge reference number
PDAT(21,24)	real	raw data array, set equivalent to IBUF

POS (7)	real	array to contain probe positions
NOLF	integer	suppresses line feed
NOCR (2)	integer	suppresses line feed and carriage RETURN
ICLR (3)	integer	clears line above cursor
IDCBS	integer	length of data control block
IENTR	integer	multiple entry flag
IDOC	integer	count of current program run
IDOCF	integer	count of current data file sequential #
ITYPE	integer	type of raw data file
IL	integer	number of words to be transferred in FMP calls
LI	integer	LU# of standard input device (terminal)
LO	integer	LU# of standard output device (line printer)
IPOS	integer	Flag to indicate measurement location
IMON	integer	ASCII converted month of current year
IYEAR	integer	ASCII converted last two digits of current year
IDAY	integer	ASCII converted day of the month
IHOUR	integer	ASCII converted hour of the day (24 h clock)
IMIN	integer	ASCII converted minute of the hour

JI	integer	Subscript for data array PDAT
IS	integer	LU# of the selected scanner
IC	integer	Instrument code (DVM ... 1 and digital counter ... 3)
I2	integer	Subscript for position array POS
J3	integer	Contains channel of desired scanner
IDUM	integer	Decision variable
IW	integer	Determines delay in tens of milliseconds between closing S/V port and DVM reading
JO	integer	Number of selected S/V
SUM	real	Variable used to compute average
ISYNCH	integer	Synchronisation variable to coordinate data transfer 21 MX ⟶ 9830
NEW	integer	Scratch variable used to rename files

Key to data array PDAT

PDAT (1,J1) Barometric pressure p_{BARO}

PDAT (2,J1) KULITE reference pressure

PDAT (3,J1) Combination probe pressure p_1

PDAT (4,J1) Combination probe pressure p_{23}

PDAT (5,J1) Combination probe pressure p_4

PDAT (6,J1) Total pressure ahead of compressor p_t

PDAT (7,J1) Static port in casing #2 , S_2

PDAT (8,J1) Static port in casing #13 , S_{13}

PDAT (9,J1) Average reading of 4 static ports in Hub (#2 thru 5) $(H_1 + H_2 + H_3 + H_4)/4$

PDAT (10,J1) Pressure ahead of compressor flow meter orifice P_1 nozzle compr

PDAT (11,J1) Temperature ahead of compressor flow meter orifice T_1 nozzle compr

PDAT (12,J1) Pressure drop across compressor flow meter orifice $p_{nozzle\ compr}$

PDAT (13,J1) Temperature reading from reference probe T_{ref}

PDAT (14,J1) Differential temperature reading from the combination probe to the reference probe T_{probe}

PDAT (15,J1) Radial immersion of the combination probe

PDAT (16,J1) Yaw angle of the combination probe

PDAT (17,J1) Case angle

PDAT (18,J1) Compressor RPM

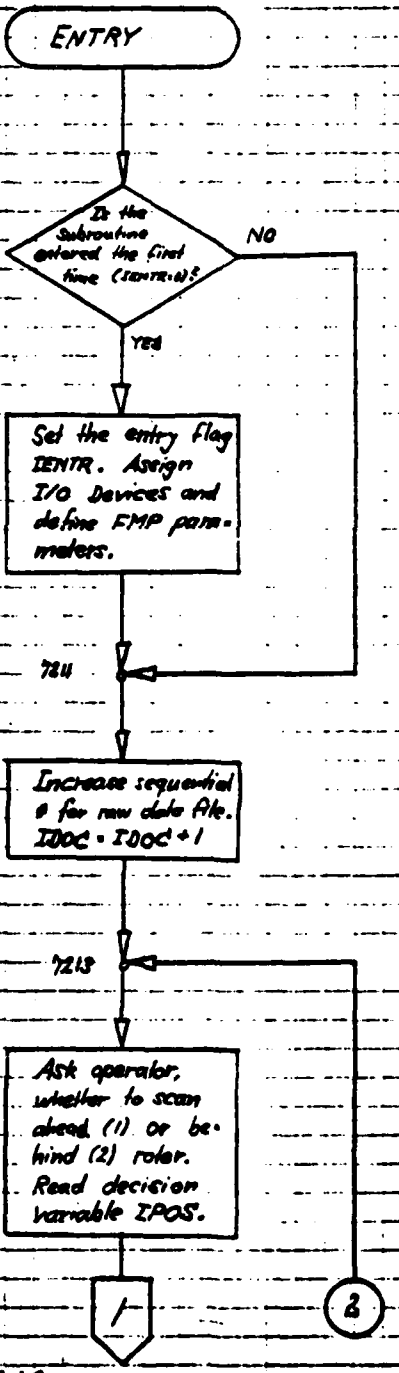
PDAT (19,J1) Test run #

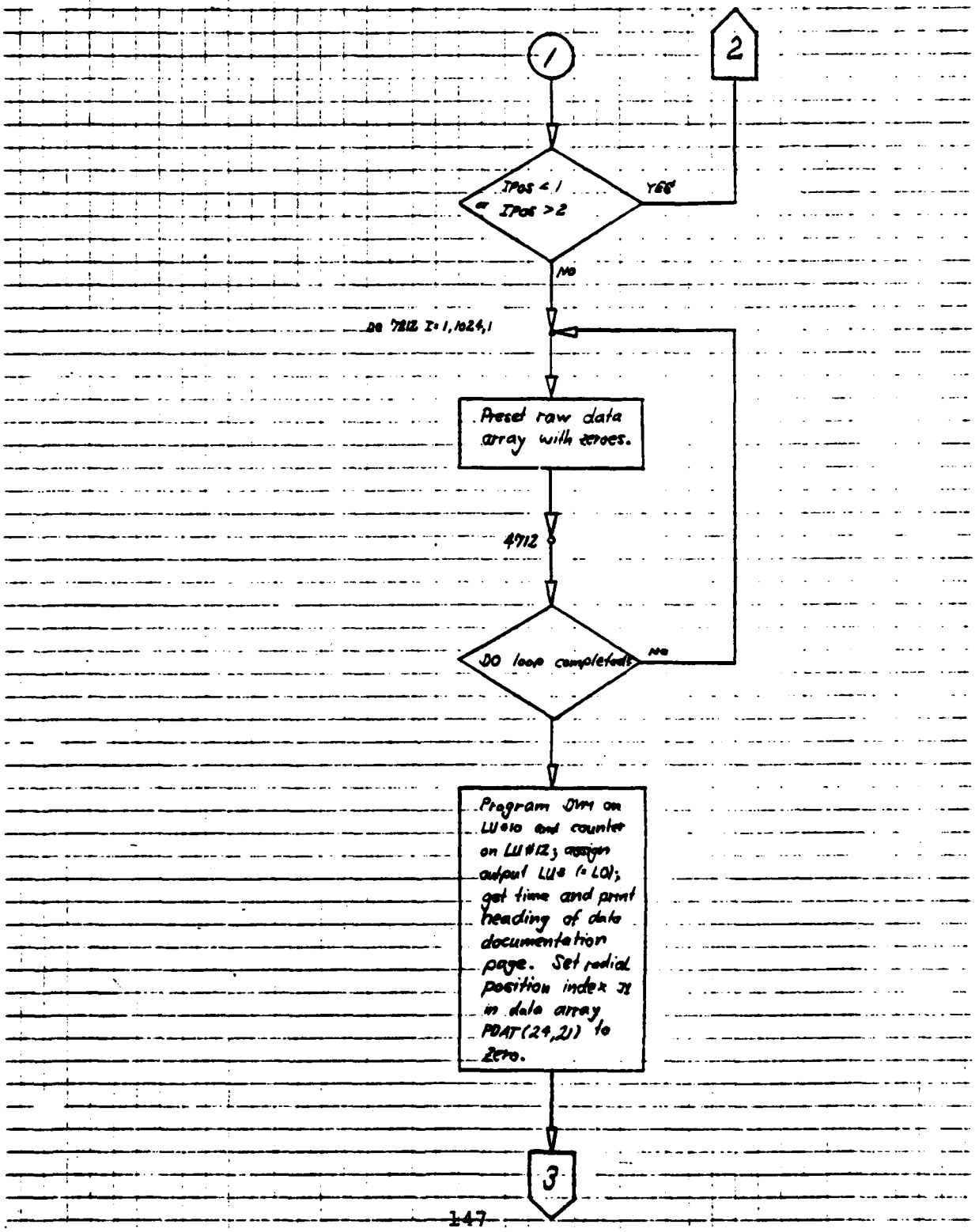
PDAT (20,J1) Test # of this run

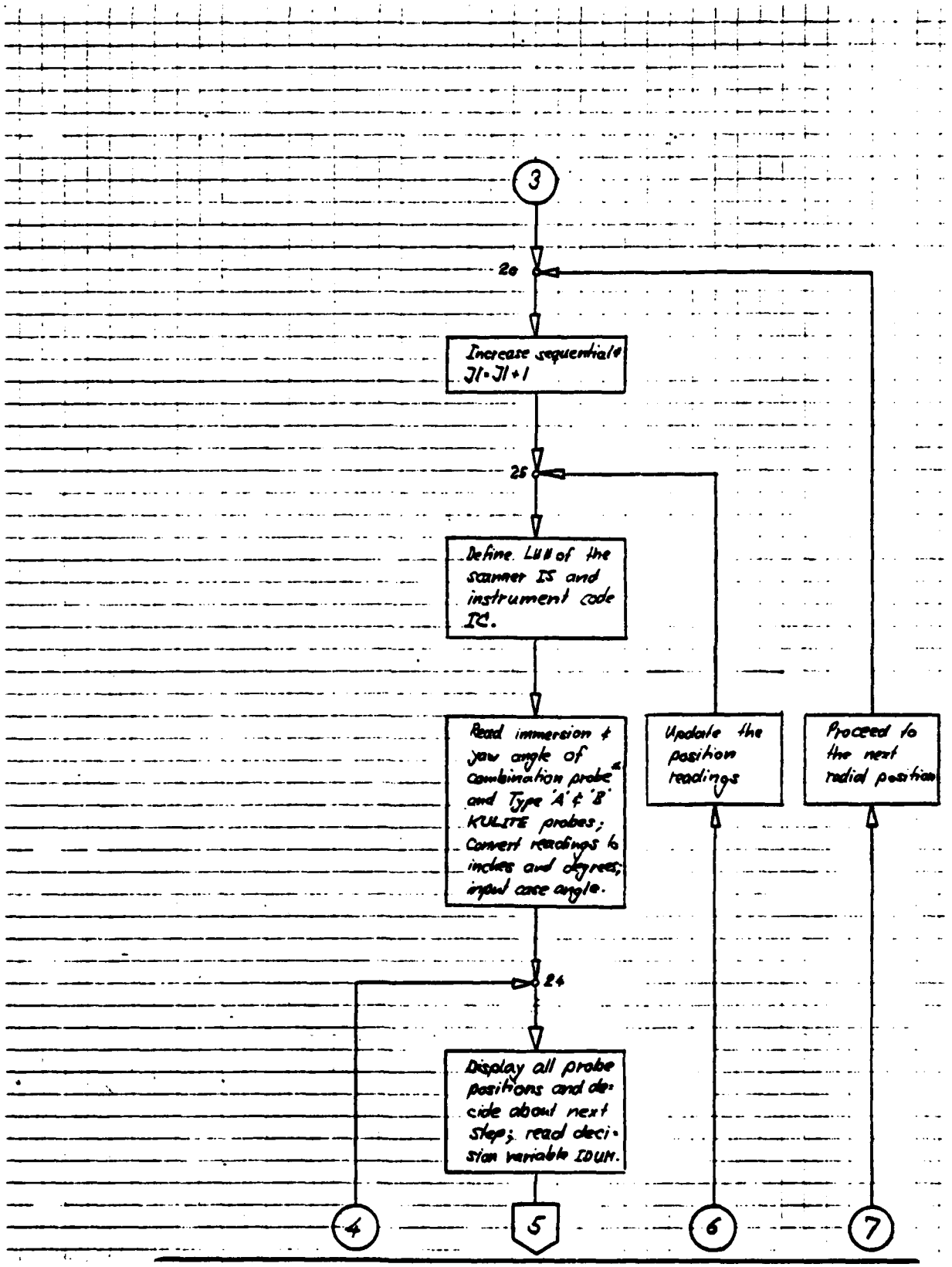
PDAT (21,J1) Point # of this test

J1 = 1 ... 24 indicates # of radial position.

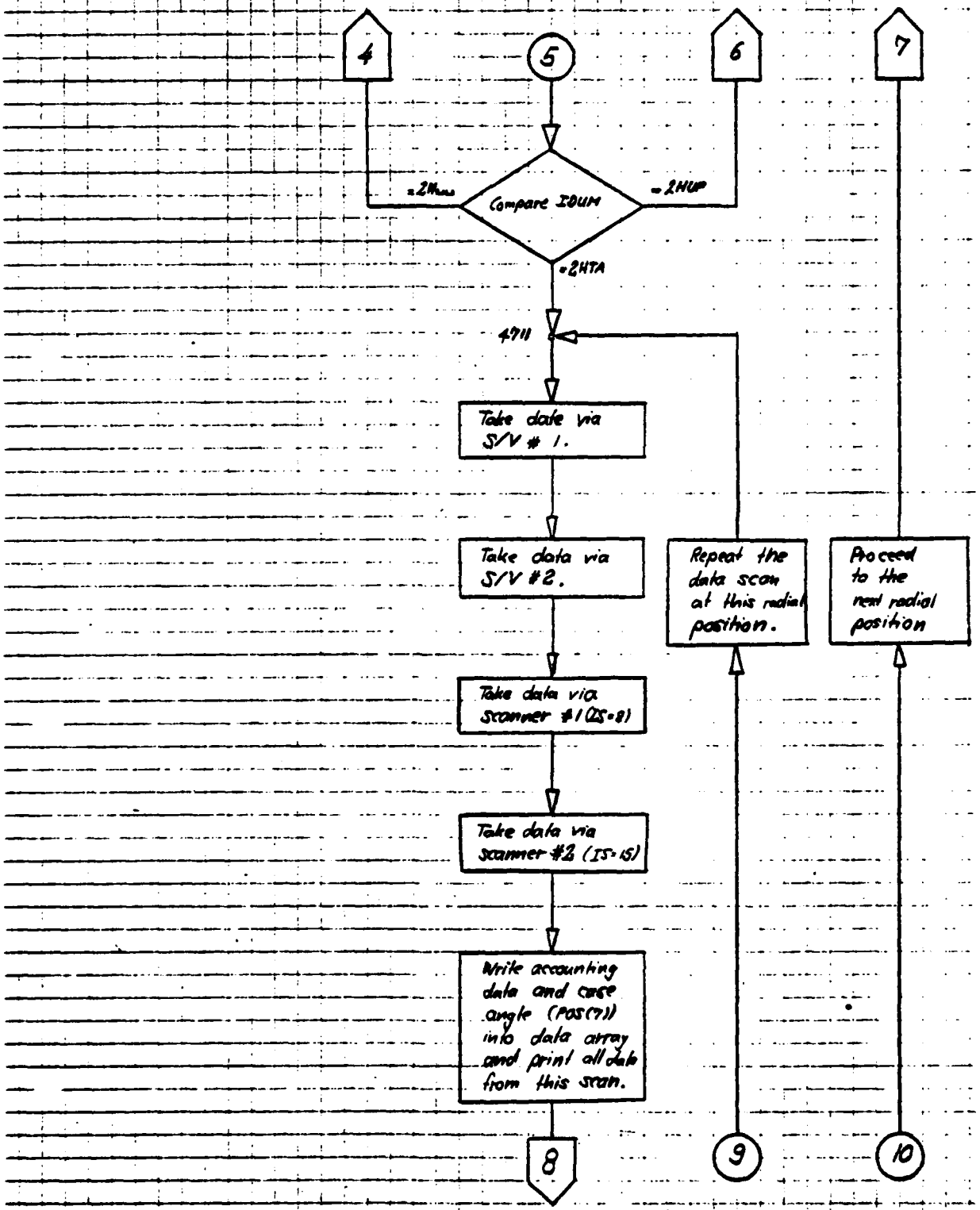
FLOW CHART SUBROUTINE COMB

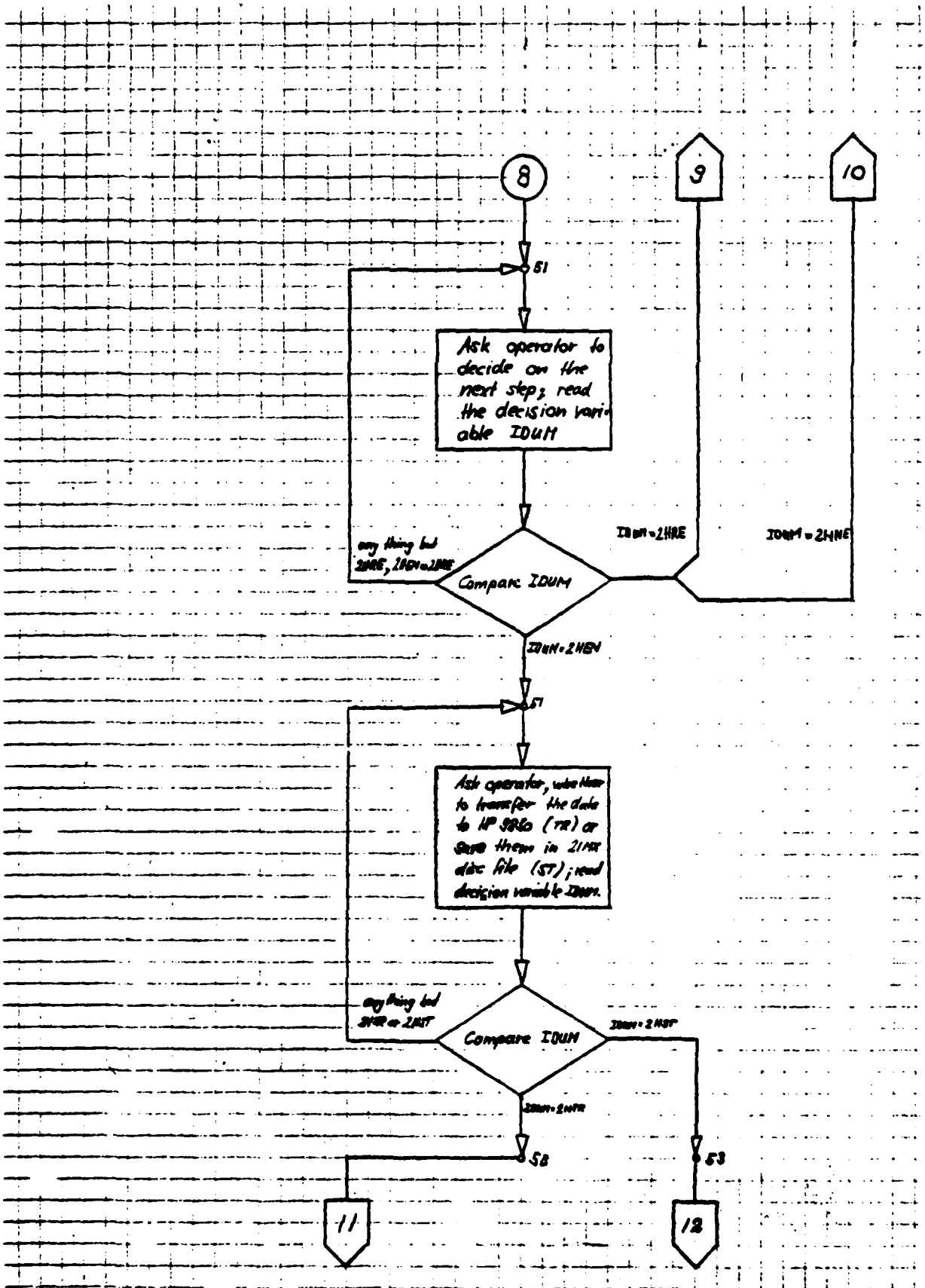


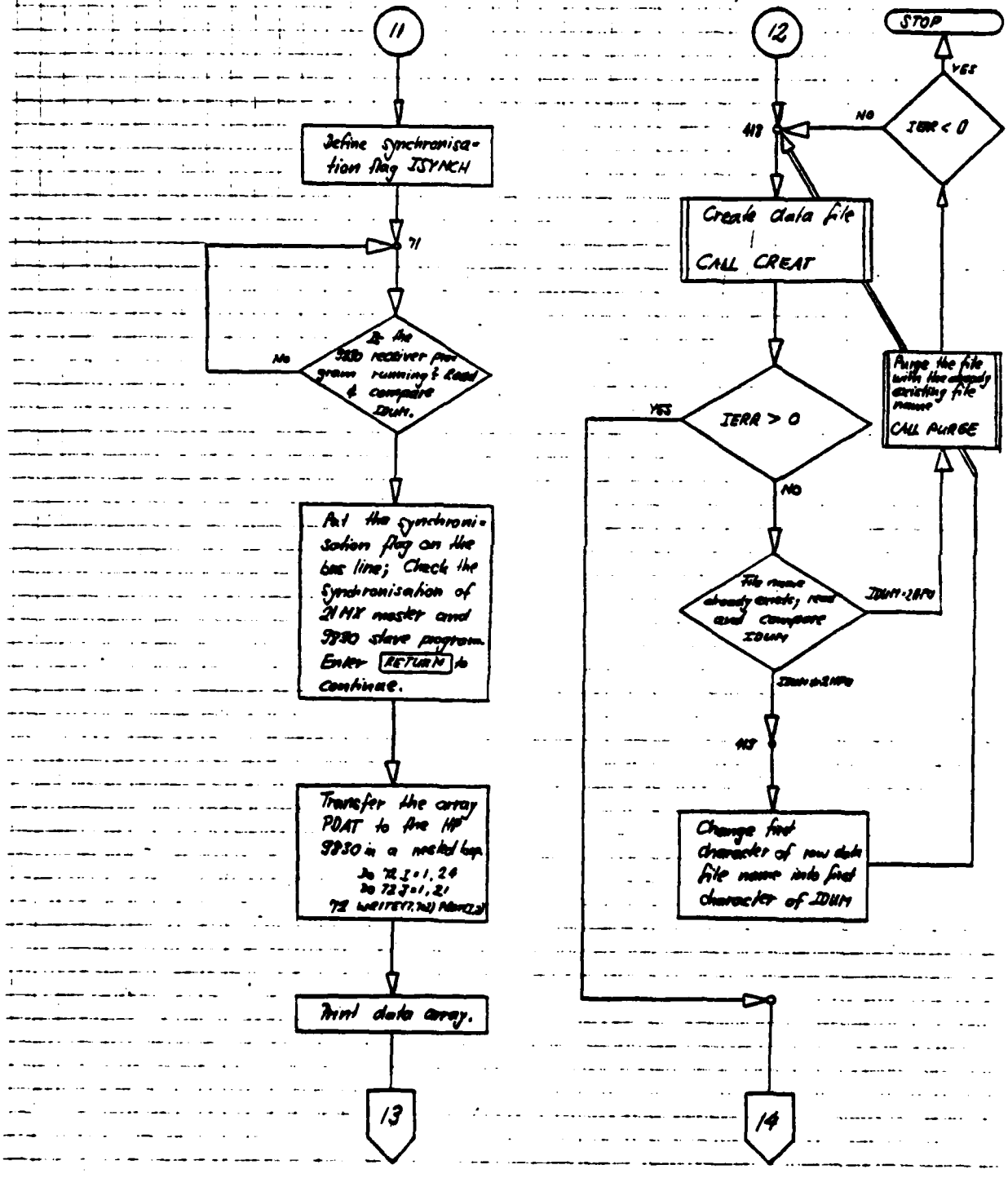

1
2
IPos < 1
or IPos > 2
YES
NO
DO 7812 I=1,1024,1
Preset raw data array with zeroes.
4712
DO loop completed?
NO
Program DVM on LU#10 and counter on LU#12; assign output LU# (=LO); get time and print heading of data documentation page. Set radial position index JI in data array PDAT(24,2) to zero.
3

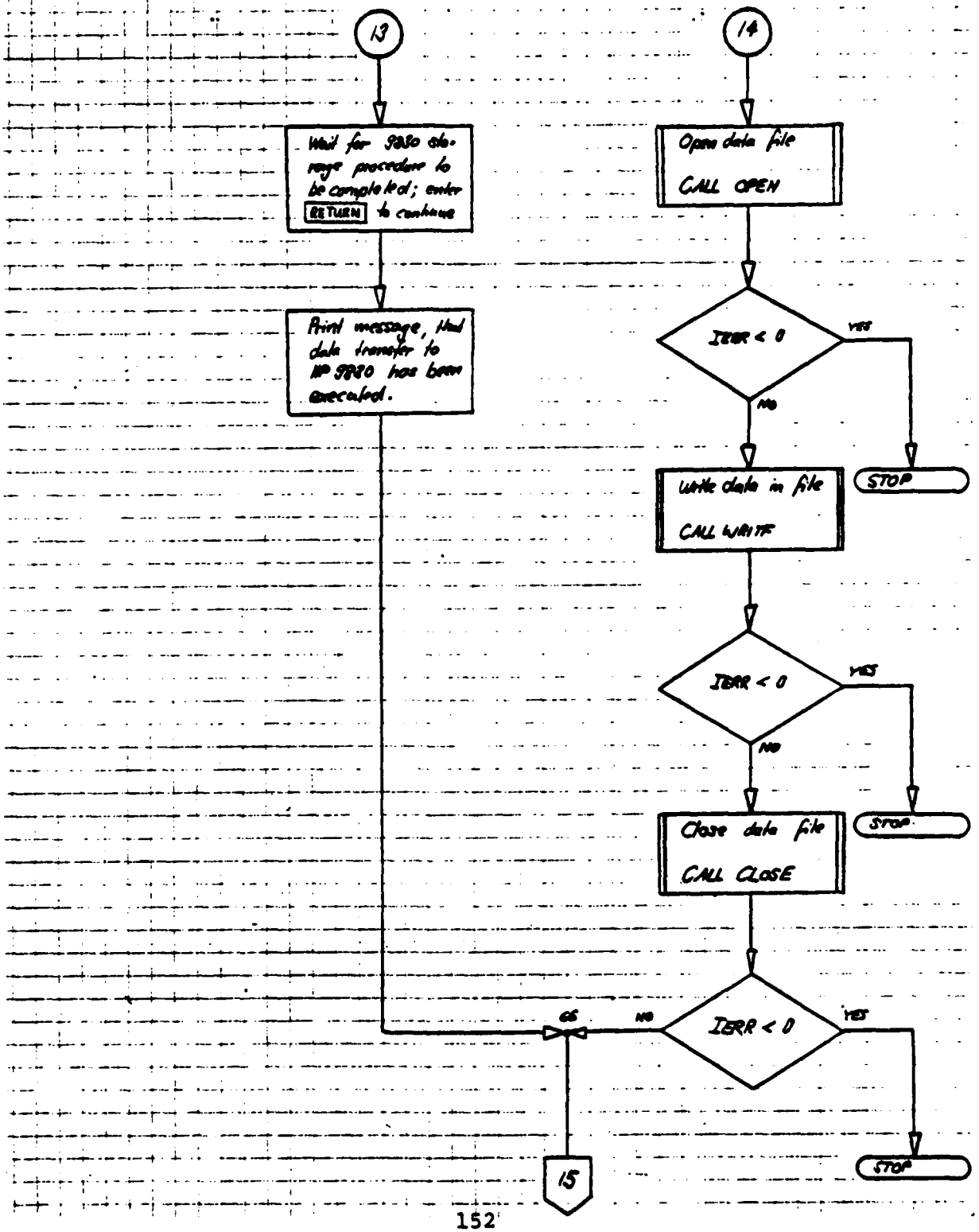


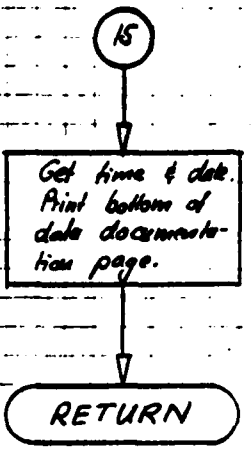
*) Refer to Appendix A1: work sheet data locations. The location of immersion and yaw angle for the combination probe depends on where to take ten (10) rd.


4
5
6
7
Compare IDUM
=2HTA
4711
Take data via S/V # 1.
Take data via S/V #2.
Take data via scanner #1 (IS=8)
Take data via scanner #2 (IS=15)
Write accounting data and case angle (POS(7)) into data array and print all data from this scan.
8
Repeat the data scan at this radial position.
9
Proceed to the next radial position
10


8
9
10
51
Ask operator to decide on the next step; read the decision variable IDUM
anything but 2HRE, IDUM=2HRE
Compare IDUM
IDUM=2HRE
IDUM=2HNE
IDUM=2HEN
57
Ask operator, whether to transfer the data to HP 9826 (TR) or store them in 21MX disc file (ST); read decision variable IDUM.
anything but 2HTR or 2HST
Compare IDUM
IDUM=2HST
IDUM=2HTR
58
53
11
12


11
Define Synchronisation flag ISYNCH
71
Is the 9330 receiver program running & Read & compare IDUM.
No
Put the synchronisation flag on the bus line; Check the synchronisation of 2MX master and 9330 slave program. Enter RETURN to continue.
Transfer the array FDAT to the 9330 in a nested loop
Do 72 I=1,24
Do 72 J=1,21
72 WRITE(7,74) FDAT(I,J)
Print data array.
13
12
STOP
Yes
No
IERR < 0
413
Create data file
CALL CREAT
IERR > 0
Yes
No
Purge the file with the already existing file name
CALL PURGE
File name already exists; read and compare IDUM
IDUM=2MX
IDUM≠2MX
413
Change first character of new data file name into first character of IDUM
14


13
Wait for 9830 storage procedure to be completed; enter RETURN to continue
Print message, that data transfer to HP 9830 has been executed.
14
Open data file
CALL OPEN
IERR < 0
YES
NO
STOP
Write data in file
CALL WRITE
IERR < 0
YES
NO
STOP
Close data file
CALL CLOSE
IERR < 0
YES
NO
GO
STOP
15


15
Get time & date. Print bottom of data documentation page.
RETURN

5.3. SUBROUTINE STDY:

PURPOSE: Acquisition of flow data from the transonic 1-stage axial compressor using the steady state instrumentation. All data, gathered by this subroutine, will be used for the reduction (PROGRAM REDST: Reduce Steady state data; see section 8.3 of this report). The data array is designed to both resemble the data source location matrix (see Appendix A.1) and to discriminate groups of similar data by blank lines (Appendix A.2). CH3(1) through CH3(5), which contain all the pressures needed to calculate the flow rates, and CH3(6), which is left blank, separates this group of data from the next one. The reason is to allow the investigator a quick check and verification of crucial data. The "units" of the readings depend on the amplifier settings, but usually each channel is calibrated to allow the operator to read voltages as a quantity in engineering units. As far as possible, amplifier drift is traced by the program (CH1(1), CH2(1) etc.). (The author is indebted to Laboratory's manager, Mr. Jim Hammer, who, with admirable patience, instructed the author in how best to handle data and data systems.)

ARGUMENTS: IRUN; if IRUN is set to 0 (zero), taking pressure readings from the Scanivalves (S/V) is skipped. This option was needed when the subroutine was first written so that frequent debugging runs did not put additional loads on the S/V's. The standard entry is: IRUN = 1. Only then will the reduction program REDST perform correctly.

EXTERNALS: TIME, ACQN, SCANR, CREAT, PURGE, OPEN, WRITF, CLOSE

COMMON BLOCKS: CIBUF, CONTR, FMP. For detailed explanation refer to the TXCO2 description

MNEMONIC ABBREVIATIONS:

RP ... Repeat data acquisition at this operating condition

RT ... End data acquisition and return to calling program

PU ... Allow purge of an existing data file

ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the accounting data and defined the I/O references, STDY presets all elements of the raw data array with -0.999999, which definitely never will occur as a data reading. Next, unless IRUN equals 0 (zero), the pressures on Scanivalves 1 and 4 are read. The voltages from scanner #1 (LU# = 8) and scanner #2 (LU# 15) are read next, not depending on the value of IRUN. Note, that the allocation for the voltages in the raw data array provides blocks of similar data, separated by blank lines. A set of control parameters (CNTRL (1) thru CNTRL (6); and CNTRL (15) and the case angle - which needs to be put in manually - completes the steady state data. Then the raw data is printed to allow the operator to look at and to verify the newly acquired data. The data scan can either be repeated (Input: RP) or the subroutine terminates (Input: ST) storing the data in a type 1 disc file. If the automatically determined name for the data file already exists,

the operator either allows overwriting the existing file (Input: PU) or renames the current data file (Input: any alphabetic character other than T). Finally data file name and time are printed at the bottom of the data documentation page.

DATA FILE: The default file name is T4rrss (rr ... ASCII converted run #; ss ... ASCII converted sequential #); see Appendix A.2: Steady State Data Array. CH4 (1) through CH4 (26) are not used, because the reduction program will write its results into these slots.

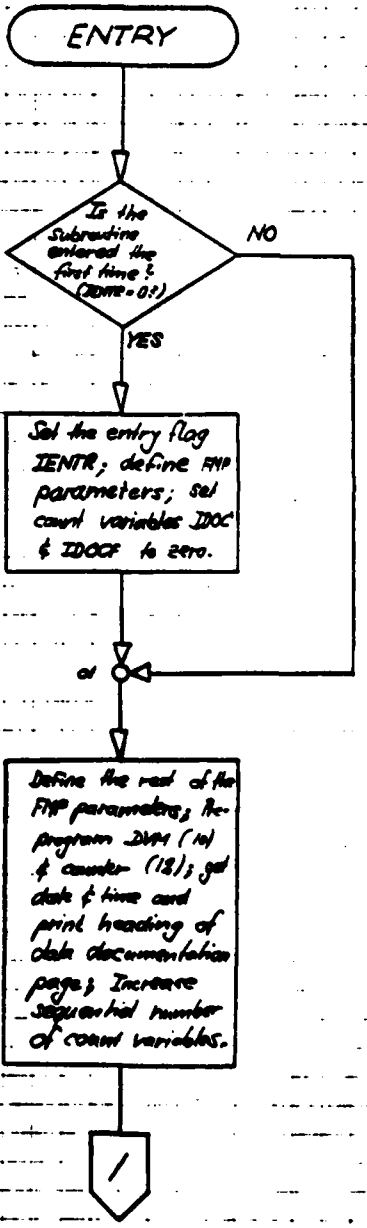
VARIABLES:

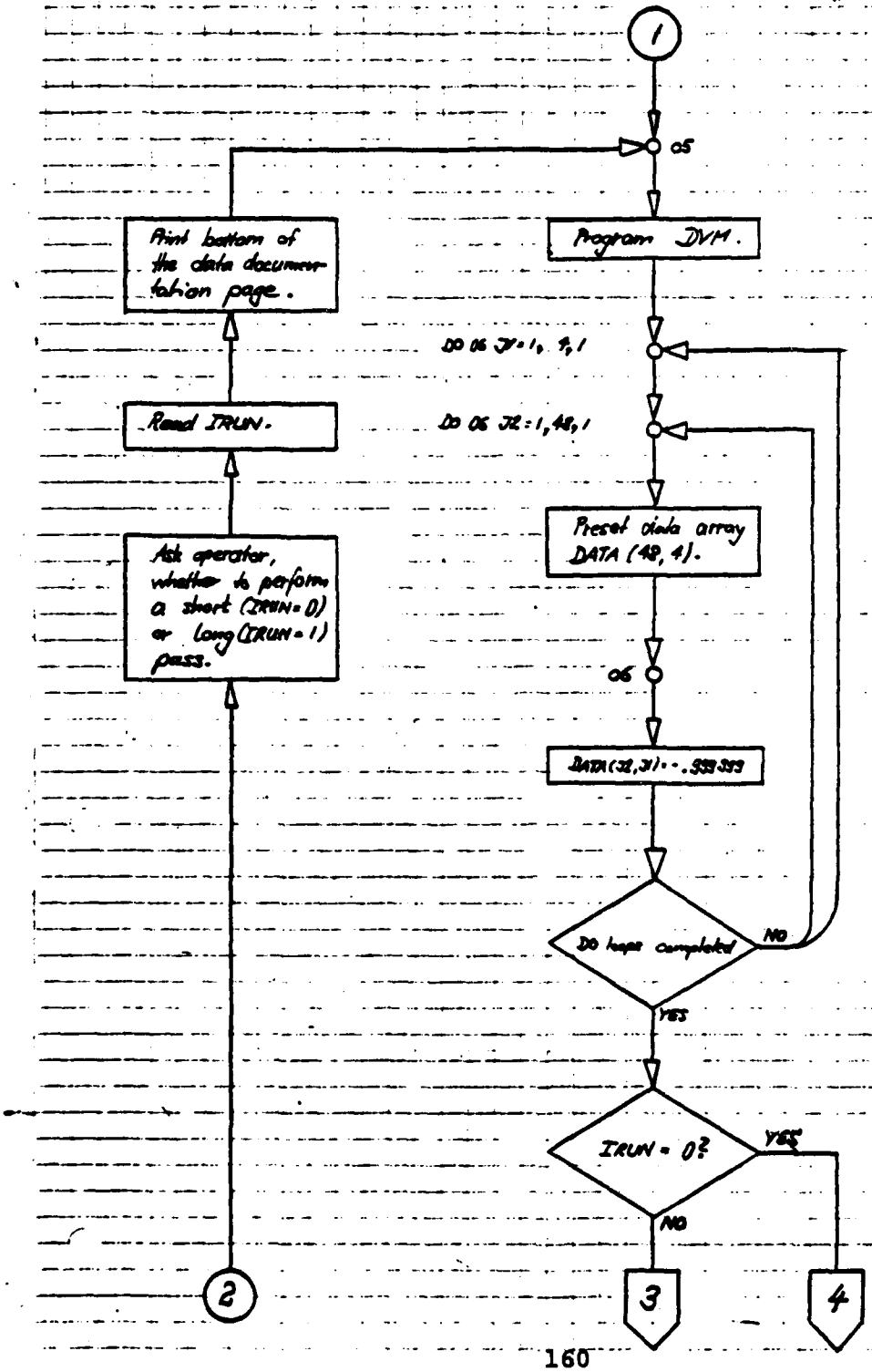
IBUF (1664)	integer	buffer array
CNTRL (256)	integer	control array
IDCB (144)	integer	data control block, used for FMP calls
IFILE (3)	integer	array to contain current file name
ISIZE (2)	integer	specifies # of records and record length
ISECU	integer	security code of data file
ICR	integer	cartridge reference number, where the raw data file is located
JSECU	integer	ASCII converted security code
JCR	integer	ASCII converted cartridge reference number
DATA (48,4)	real	raw data array, set equivalent to IBUF

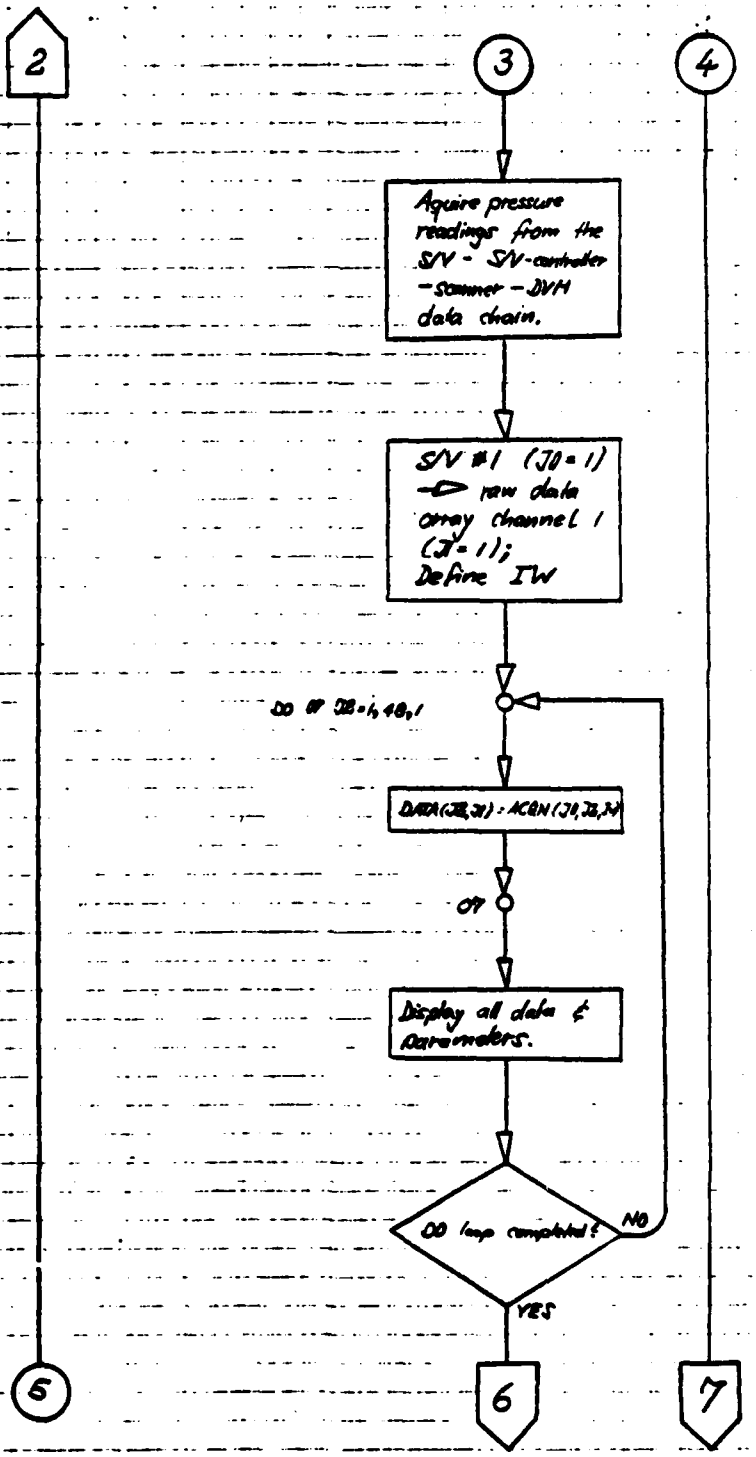
NOLF	integer	suppresses line feed
NOCR (2)	integer	suppresses line feed and carriage RETURN
ICLR (3)	integer	clears line above cursor
IENTR	integer	multiple entry flag
IDOC	integer	count of current program run
IDOCF	integer	count of current data file sequential #
ITYPE	integer	type of raw data file
IL	integer	number of words to be transferred in FMP calls
IFRST	integer	temporary buffer variable
LI	integer	LU # of standard input device (terminal)
LO	integer	LU # of standard output device (line printer)
J1	integer	subscript for data array DATA, specifies channel
J2	integer	subscript for data array DATA
IRUN	integer	control variable
JO	integer	number of selected S/V
IW	integer	determines delay in tens of milliseconds between closing S/V port and DVM reading
TARE1	real	drift of amplifier S/V#1 during test run

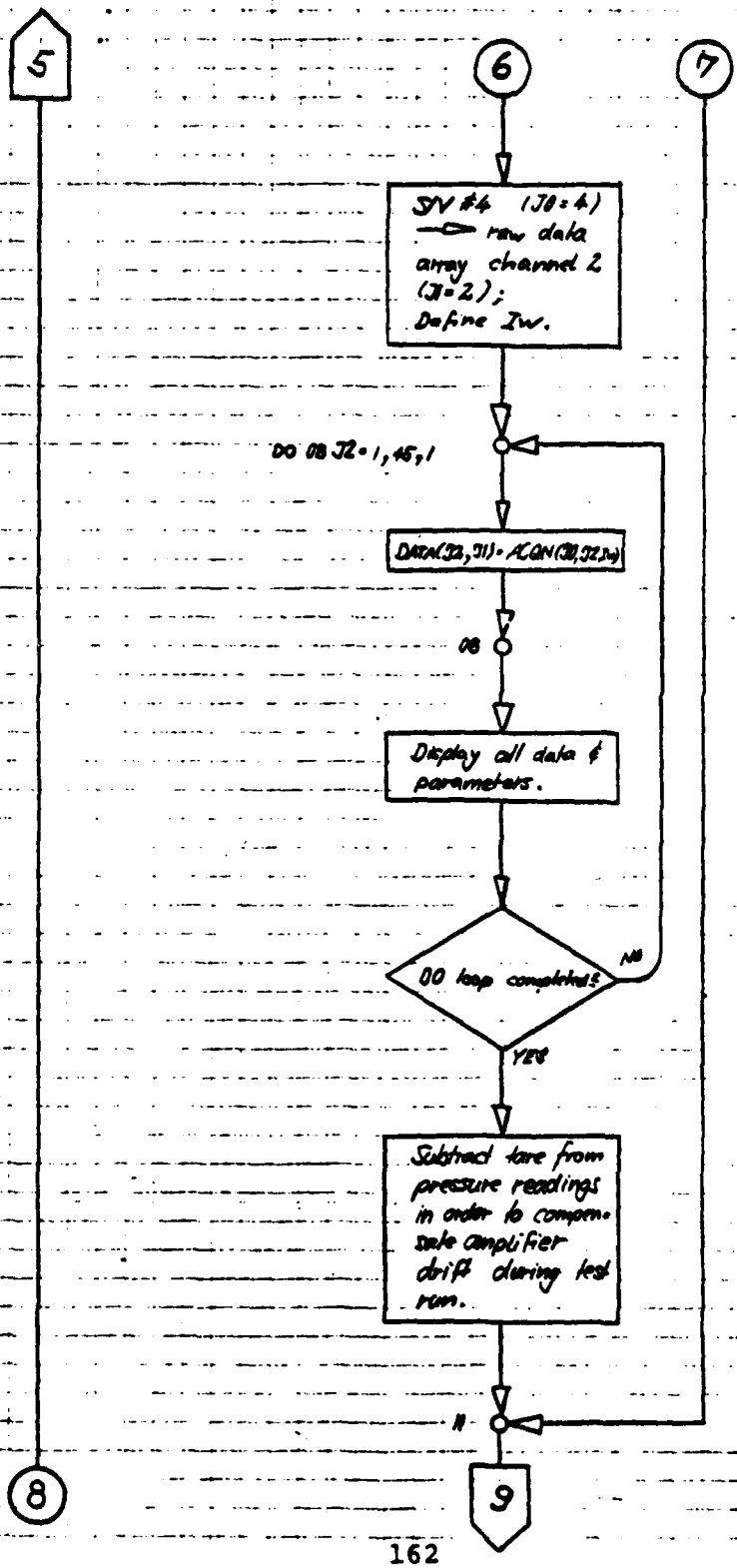
TARE2	real	drift of amplifier S/V #4 during test run
IS	integer	LU # of the selected scanner
IC	integer	instrument code (DVM ... 1 and digital counter ... 2)
J3	integer	contains channel of scanner
NO(2)	integer	ASCII text to be printed, if value of DATA (J2,J1) = -.999899
IDUM	integer	decision variable
IMON	integer	ASCII converted month of current year
IDAY	integer	ASCII converted day of the month
IYEAR	integer	ASCII converted last two digits of current year
IHOUR	integer	ASCII converted hour of the day (24 h clock)
IMIN	integer	ASCII converted minute of the hour
NEW	integer	scratch variable used to rename files

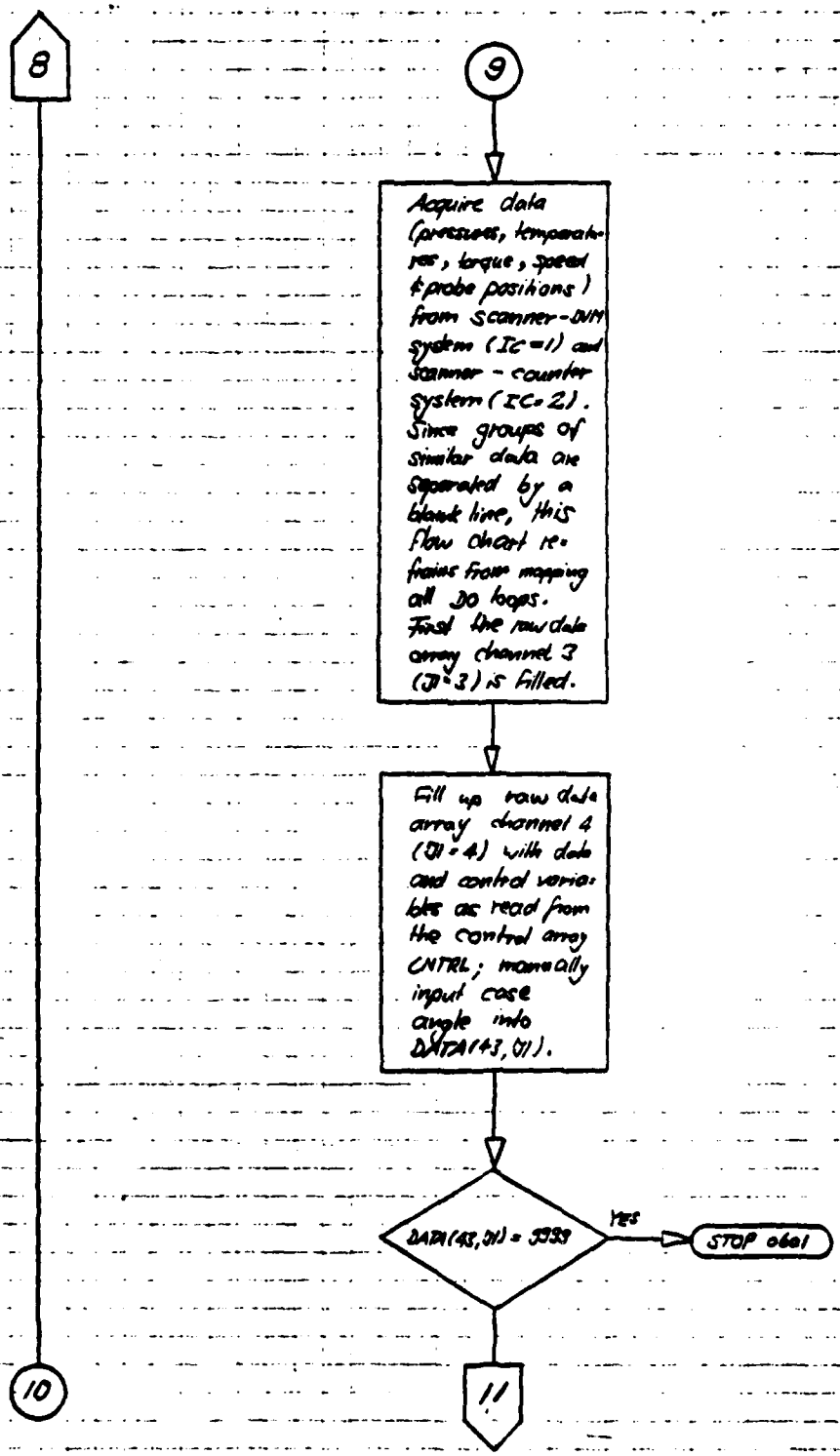
FLOW CHART SUBROUTINE STOY

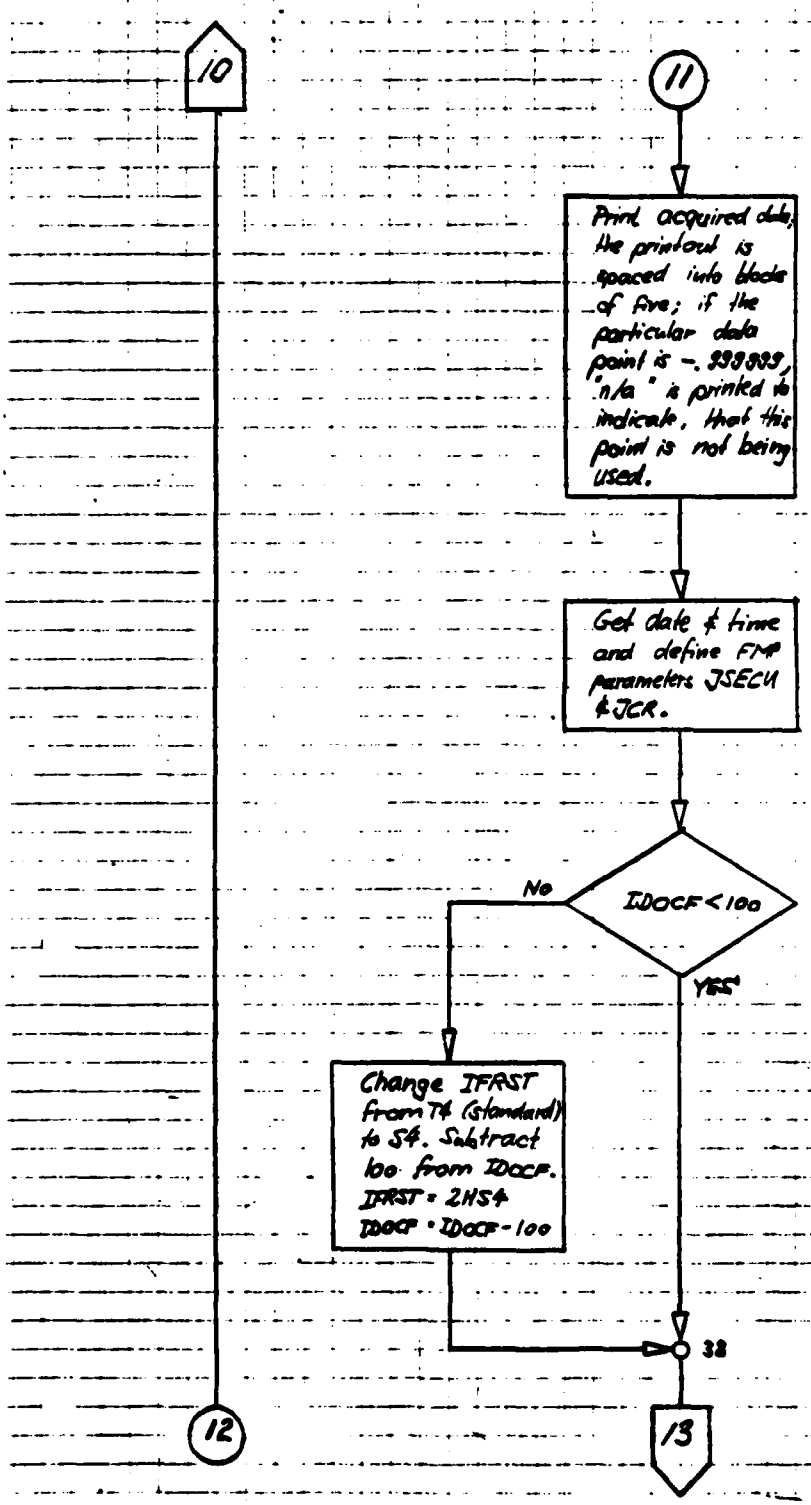


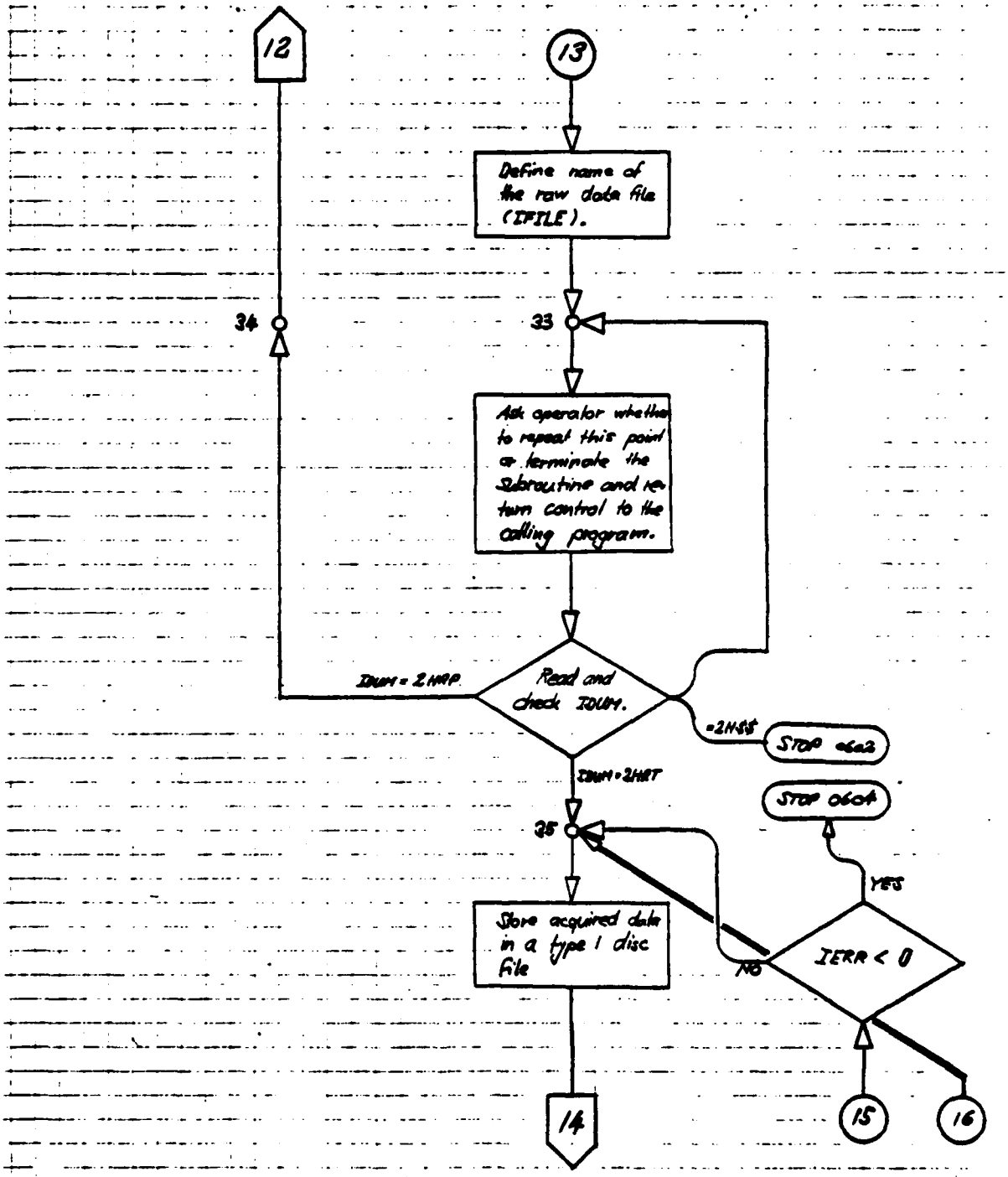

1
05
Program DVM.
DO 06 J=1, 4, 1
DO 06 JR=1,48,1
Preset data array DATA(48,4).
06
DATA(JR,J)=.333333
DO loops completed
NO
YES
IRUN = 0?
YES
NO
3
4
Print bottom of the data documentation page.
Read IRUN.
Ask operator, whether to perform a short (IRUN=0) or long (IRUN=1) pass.
2

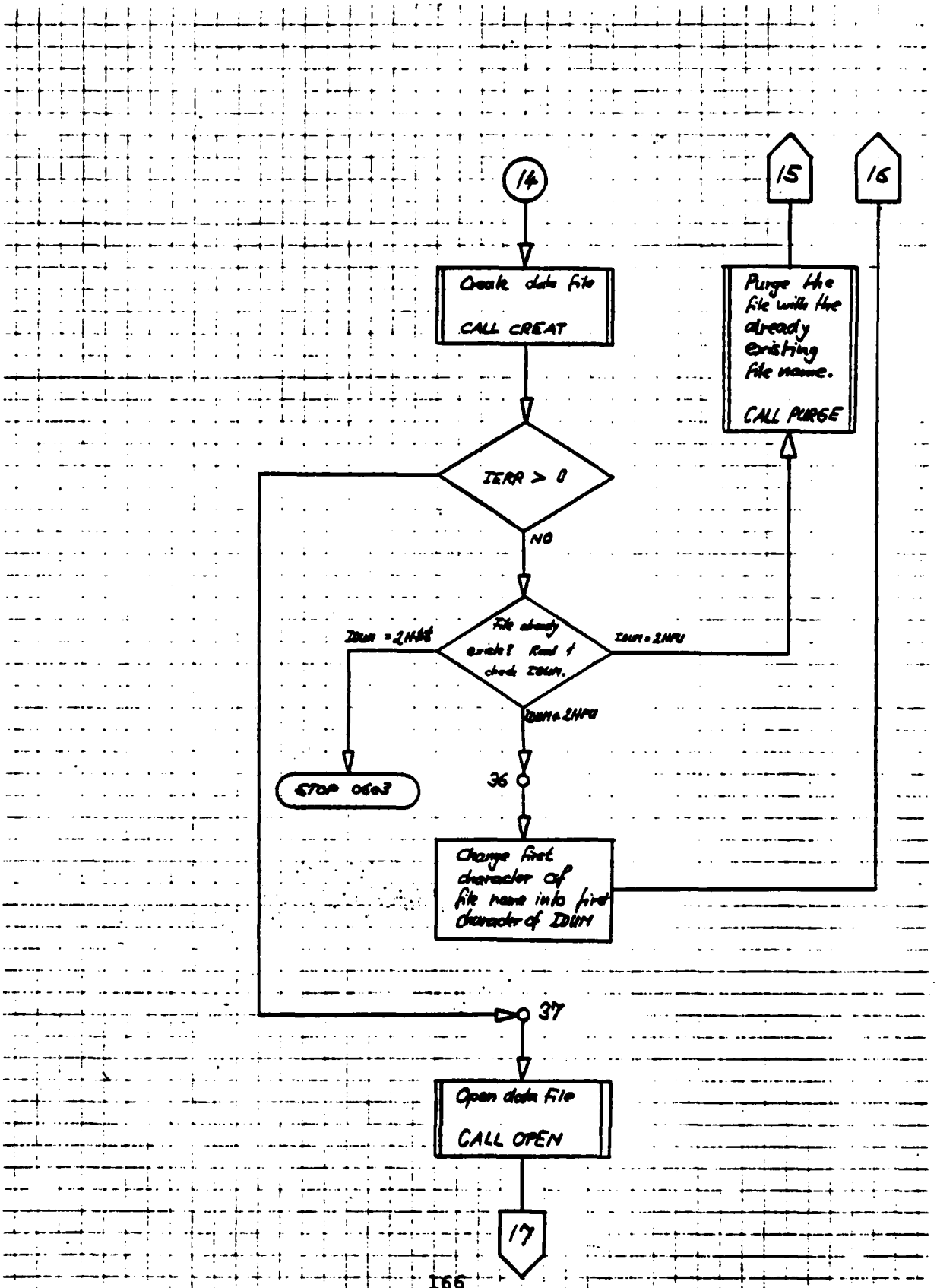

2
3
4
Aquire pressure readings from the S/V - S/V-controller - scanner - DVM data chain.
S/V #1 (J0 = 1) → raw data array channel 1 (J = 1); Define IW
DO 07 JB = 1, 48, 1
DATA(JB, J) = ACQN(J1, J2, J)
07
Display all data & parameters.
DO loop completed?
NO
YES
6
6
7

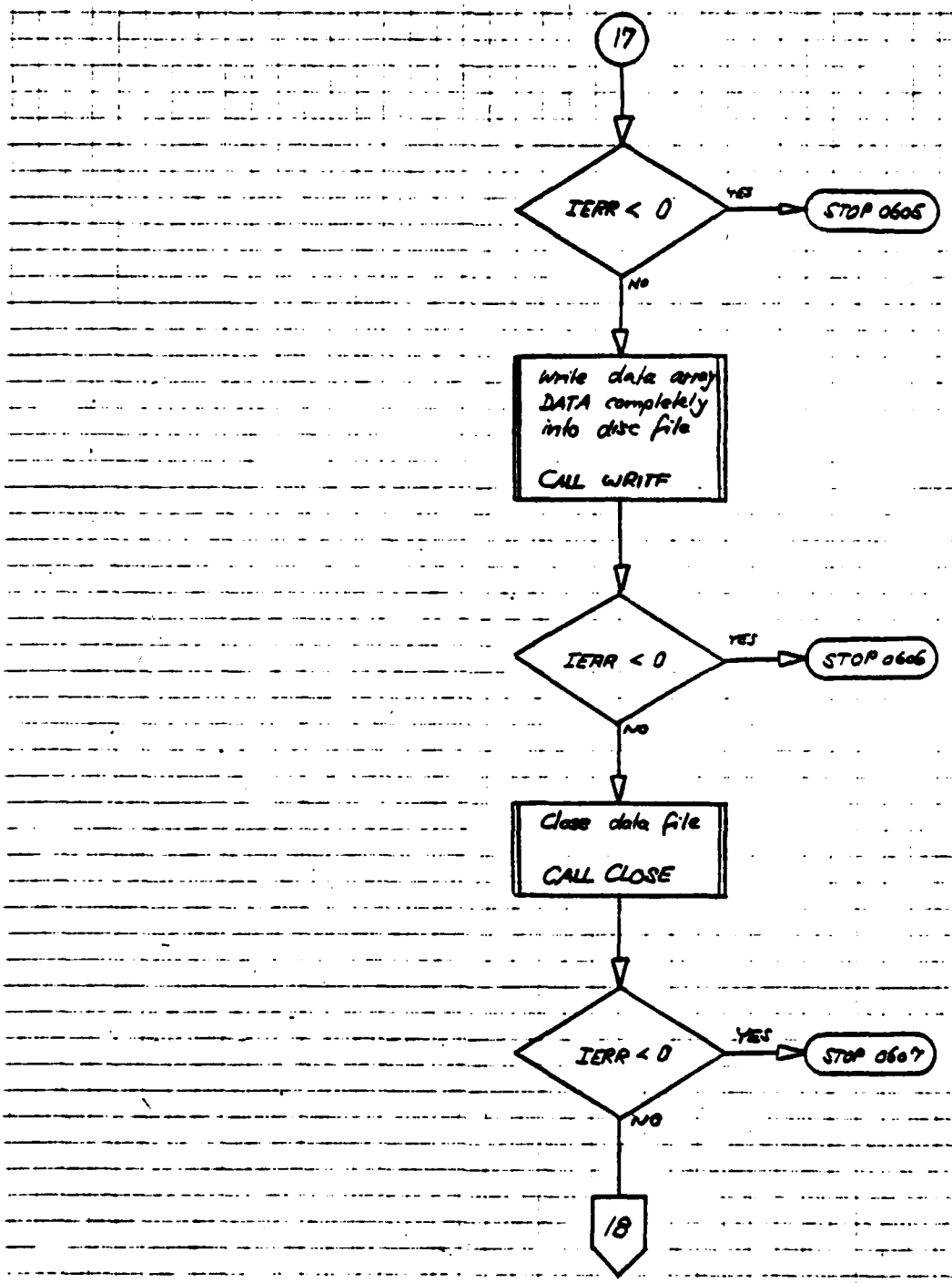

5
6
7
SV #4 (J0=4) → raw data array channel 2 (J1=2); Define Iw.
DO 08 J2=1,45,1
DATA(J2,J1)=ACQN(J2,J2Iw)
08
Display all data & parameters.
DO loop completed?
NO
YES
Subtract tare from pressure readings in order to compensate amplifier drift during test run.
11
8
9

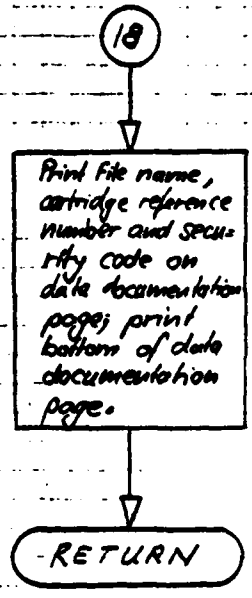

8
9
Acquire data (pressures, temperatures, torque, speed & probe positions) from scanner-DVM system (IC=1) and scanner-counter system (IC=2). Since groups of similar data are separated by a blank line, this flow chart refrains from mapping all DO loops. First the raw data array channel 3 (J=3) is filled.
Fill up raw data array channel 4 (J=4) with data and control variables as read from the control array CNTRL; manually input case angle into DATA(43,J1).
DATA(43,J1) = 3939
YES
STOP 0601
10
11


10
11
Print acquired data, the printout is spaced into blocks of five; if the particular data point is -.999999, "n/a" is printed to indicate, that this point is not being used.
Get date & time and define FMP parameters JSECN & JCR.
IDOCF<100
No
YES
Change IFRST from 74 (standard) to 54. Subtract 100 from IDOCF.
IFRST = 2HS4
IDOCF = IDOCF-100
38
12
13


12
13
Define name of the raw data file (IFILE).
34
33
Ask operator whether to repeat this point or terminate the subroutine and return control to the calling program.
IDUM = 2HMP
Read and check IDUM.
=2H-55
STOP 0602
IDUM = 2HET
STOP 0604
35
YES
Store acquired data in a type 1 disc file
NO
IERR < 0
14
15
16


14
15
16
Create data file
CALL CREAT
Purge the file with the already existing file name.
CALL PURGE
IERR > 0
NO
File already exists? Read 1 check IDUM.
IDUM = 2HAA
IDUM = 2HFU
IDUM = 2HFU
STOP 0603
36
Change first character of file name into first character of IDUM
37
Open data file
CALL OPEN
17


17
IERR < 0
YES
STOP 0605
NO
write data array DATA completely into disc file
CALL WRITF
IERR < 0
YES
STOP 0606
NO
Close data file
CALL CLOSE
IERR < 0
YES
STOP 0607
NO
18


18
Print file name, cartridge reference number and security code on data documentation page; print bottom of data documentation page.
RETURN

5.4. PROGRAM LISTING TXCO2

```
PAGE 0001  FTN.  4:12 PM  TUE., 23  SEP., 1980

0001   FTN4,L
0002          BLOCK DATA
0003         *, / FMP    / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004          COMMON / FMP    / IDCB,IFILE,ISIZE,ISECU,ICR
0005          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006          END

 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
 **  NO WARNINGS **  NO ERRORS **
BLOCK COMMON FMP     SIZE =  00151
```

```
 PAGE 0002  FTN.   4:12 PM  TUE., 23  SEP., 1980

0007          BLOCK DATA
0008         *, / CIBUF / IFUF(1664)
0009          COMMON / CIBUF / IBUF
0010          INTEGER IBUF(1664)
0011          END

  FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
  ** NO WARNINGS **  NO ERRORS **
 BLOCK COMMON CIBUF  SIZE = 01664
```

```
PAGE 0003  FTN.   4:12 PM  TUE., 23  SEP., 1980

0012          BLOCK DATA
0013         *, / CONTR / CNTRL(256)

0014          COMMON / CONTR / CNTRL
0015          INTEGER CNTRL(256)
0016          END

 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
 ** NO WARNINGS **  NO ERRORS **
BLOCK COMMON CONTR  SIZE =  00256
```

```
0017          PROGRAM TXCO2 (3,99)
0018    C     .....................................................................
0019    C     .                                                                   .
0020    C     .  The operating system RTE IV B requests the data acquisition      .
0021    C     .  program TXCO for the one stage transsonic compressor to be       .
0022    C     .  split into several programs scheduled by the father program      .
0023    C     .  TXCO.  This son program TXCO2 consists of the subroutines        .
0024    C     .  COMB and STDY.  These codes handle the acquisition of the        .
0025    C     .  steady state data and a survey cunducted with the combination    .
0026    C     .  probe.  The data transfer between father and son program         .
0027    C     .  takes place via the control array CNTRL (disc file CNTRLF)       .
0028    C     .  and the data array IBUF (disc file IBUFF).                       .
0029    C     .                                                                   .
0030    C     .  The utility subroutines ACQN, CNTL, CURVE, ICON, IPORT,          .
0031    C     .  PICTR, REWRF, RPACE, SCANR, TIME and WAIT are added.             .
0032    C     .                                                                   .
0033    C     .  Author:    Hans M. Zebner                                        .
0034    C     .  Date  :    March 12, 1980                                        .
0035    C     .                                                                   .
0036    C     .  A detailed program description is available in the TXCO log.     .
0037    C     .                                                                   .
0038    C     .....................................................................
0039          *, Second son program of father program TXCO.

0040
0041          COMMON / CONTR / CNTRL
0042
0043          INTEGER CNTRL(256)
0044
0045          DATA NOLF /006537/
0046
0047      101 FORMAT (9X""20X""A2)
0048      102 FORMAT ("  TXCO2 : PROGRAM ABORTED!  NO SUBROUTINE HAS BE
0049     *EN INITIALIZED.")
0050      801 FORMAT ("CA")
0051     1001 FORMAT ("F1R7M3A1H0T3")
0052     1201 FORMAT ("PF4G6T")
0053     1501 FORMAT ("CA")
0054
0055          CALL REWRF (-1,2)
0056          LI = CNTRL(19)
0057          IF ( CNTRL(50)  .LT.  5   .OR.   CNTRL(50)  .GT.  6 ) GO TO 03
0058
0059    C     .....................................................................
0060    C     .                                                                   .
0061    C     .  Set interface bus and devices to remote control.                 .
0062    C     .                                                                   .
0063    C     .....................................................................
0064          CALL ABRT (7,2)
0065          CALL RMOTE ( 8)
0066          CALL RMOTE (10)
0067          CALL RMOTE (12)
0068          CALL RMOTE (15)
0069          WRITE ( 8, 801)
0070          WRITE (10,1001)
0071          WRITE (12,1201)
0072          WRITE (15,1501)
0073
0074    C     .....................................................................
0075    C     .                                                                   .
0076    C     .  Call subroutine indicated by CNTRL(50).                          .
0077    C     .                                                                   .
0078    C     .....................................................................
0079          ISTOP = CNTRL(50)
0080          IF ( CNTRL(50)  .EQ.  5 ) CALL COMB
0081          IF ( CNTRL(50)  .EQ.  6 ) CALL STDY(CNTRL(51))
0082
0083    C     .....................................................................
0084    C     .                                                                   .
0085    C     .  Release interface bus and devices from remote control.           .
0086    C     .                                                                   .
0087    C     .....................................................................
0088          CALL CLEAR (7,1)
0089          CALL LOCL (7)
0090
0091          CALL REWRF (1,2)
```

```
PAGE 0005  TXC02  4:12 PM  TUE., 23  SEP., 1980

0092          WRITE (LI, 101) NOLF
0093          GO TO (01,02) ISTOP
0094       01 STOP 0577
0095       02 STOP 0677
0096       03 WRITE (LI, 102)
0097          END

 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 00221      COMMON = 00000
```

```
0098          SUBROUTINE COMB
0099    C     .....................................................
0100    C     .                                                   .
0101    C     .  Acquires data from the transsonic 1-stage axial compressor.
0102    C     .  Gathers data required for a probe survey with the
0103    C     .  combination probe.  The raw data then are transferred to
0104    C     .  the HP 9830 calculator for reduction.            .
0105    C     .                                                   .
0106    C     .....................................................
0107          *, Takes data from the combination probe.
0108          COMMON / CIBUF / IBUF
0109          COMMON / CONTR / CNTRL
0110          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
0111          REAL PDAT(21,24),POS(7)
0112          INTEGER CNTRL(256),IDCB(144),IFILE(3),ISIZE(2)
0113          INTEGER IBUF(1664)
0114          INTEGER NOLF,NOCR(2),ICLR(3)
0115          EQUIVALENCE (IBUF(1),PDAT(1,1))
0116          DATA NOLF  /006537B/
0117          DATA NOCR  /000033B,040433B/
0118          DATA ICLR  /015524B,015515B,006537B/
0119          DATA IDCBS /144/
0120    C     FORMATS COMB START
0121      100 FORMAT (A1"S")
0122      101 FORMAT ("WARNING: file "3A2" already exists!  Type PU to "
0123     *"allow purge or enter any char-",
0124     */,"          acter but T to change file name."3BX)
0125      102 FORMAT ("File name change succesful! "3A2" changed to "3A2)
0126      103 FORMAT (""29X"gathering probe data!"29X""A2)
0127      105 FORMAT (""26X"storing data in file "3A2,26X""A2)
0128      107 FORMAT (A2)
0129      210 FORMAT (" Make sure, that the 9830 receiver program runs!  Ty
0130     *pe YES to continue:"2A2)
0131      211 FORMAT (" "27X"Transferring data to 9830"27X""2A2)
0132      212 FORMAT (" "79X/2A2"  Data transfer completed.  Print transferr
0133     *ed data?  Enter LU# or NO!    "2A2)
0134      213 FORMAT (" Repeat data transfer?  Enter YES or NO!
0135     *    "2A2)
0136      215 FORMAT (" Waiting for 9830 storage procedure.  Type C
0137     *R to continue!    "2A2)
0138      216 FORMAT (" Check synchronisation of master and slave program!
0139     * Type CR to continue!      "2A2)
0140      116 FORMAT (" "16X"Probe survey at this constellation completed"17
0141     *X""A2//
0142     *"Type  TR  to transfer the data to HP 9830 calculator"/
0143     *"      ST  to save the Data in HP 21MX disc file "3A2":"A2
0144     *":"A2/
0145     *"          "2A2)
0146      117 FORMAT (" "25X"Check raw data of this scan!"26X""A2//
0147     *"Type  RE  to repeat this point"/
0148     *"      NE  to proceed to the next point"/
0149     *"      EN  to terminate the survey at this constellation"/
0150     *"          "2A2)
0151      118 FORMAT (21X"Immersion"11X"Yaw Angle"/
0152     *24X"inches"19X"  "/
0153     *" Combination probe "F10.3,10X,F10.3""/
0154     *"      Type 'A' probe "F10.3,10X,F10.3""/
0155     *"      Type 'B' probe "F10.3,10X,F10.3""//
0156     *"           Case angle "20X""F10.3""//
0157     *"Type  UP  to update these readings"/
0158     *"      TA  to take a data set at this constellation"/
0159     *"          "2A2)
0160      119 FORMAT (" Enter case angle"34X,2A2)
0161      120 FORMAT (" Is this combination probe survey done before (",
0162     *"1) or after (2) the rotor? "2A2)
0163      121 FORMAT (" PDAT("I2","I2") = ACQN("I2","I2","I3")    has been e
0164     *xecuted;  result is "F10.6""A2)
0165      122 FORMAT (" PDAT("I2","I2") = SCANR("I2","I2","I1")    has been exe
0166     *cuted;  result is "F10.6""A2)
0167      147 FORMAT (I1)
0168      148 FORMAT (I2)
0169      149 FORMAT ((3A2))
0170      189 FORMAT ((21F6.4))
0171      401 FORMAT (1H ,"" /////,36X
0172     *,"Probe Survey Documentation Pages"
```

```
0173          *//////)
0174       602 FORMAT (1H ,45(1H ),33HTranssonic Compressor Test Run # ,I7)
0175       603 FORMAT (1H ,58(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
0176       604 FORMAT (1H ,"",58(1H ),6HTime: ,A2,1H.,A2,3H  h,//////)
0177       605 FORMAT (1H0,"Constellation #"I3""//1H ,9F10.6/
0178          *1H ,60X,3F10.6/1H ,70X,2F10.6/1H ,20X,2F10.6,F10.1,4I10/)
0179       606 FORMAT (1H0,"Data transferred to HP 9830 file "10X".")
0180       607 FORMAT (1H0,"Data saved in file "3A2":"A2":"A2".")
0181       610 FORMAT ("
```

",58(1H),6HTime: ,A2,1H.,A2,3H h)

```
0182     701 FORMAT (I13)
0183     702 FORMAT (F13.6)
0184     900 FORMAT (" A REGISTER IS "K6"      B REGISTER IS "K6/)
0185    1001 FORMAT ("F1R7M3A1H0T3")
0186    1201 FORMAT ("PF4G6T")
0187  C      FORMATS COMB STOP
0188         IF ( IENTR  .NE.  0 ) GO TO 7211
0189         IENTR = 1
0190         IDOC  = 0
0191         IDOCF = 0
0192         ISECU = CNTRL(31)
0193         ICR   = CNTRL(30)
0194         ITYPE = 1
0195         IL    = 1024
0196         ISIZE(1) = 8
0197         ISIZE(2) = 128
0198         LI = CNTRL(19)
0199    7211 IDOC  = IDOC+1
0200    7213 WRITE (LI, 120) NOCR
0201         READ  (LI,   *) IPOS
0202         WRITE (LI, 149) ICLR
0203         IF ( IPOS  .LT.  1   .OR.  IPOS  .GT.  2 ) GO TO 7213
0204         IDOCF = IDOCF+1
0205         DO 7212 I=1,1024,1
0206    7212 IBUF(I)=0
0207         WRITE (10,1001)
0208         WRITE (12,1201)
0209         LO = CNTRL(20)
0210         CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0211         WRITE (LO,602) CNTRL(4)
0212         WRITE (LO,603) IMON,IDAY,IYEAR
0213         WRITE (LO,604) IHOUR,IMIN
0214         WRITE (LO,601)
0215         J1 = 0
0216      20 J1 = J1+1
0217      25 IS = 8**CNTRL(7)
0218         IC = 1
0219         I2 = 1
0220         IF ( IPOS  .EQ.  2 ) J3 = 30
0221         IF ( IPOS  .EQ.  1 ) J3 = 38
0222         POS(I2) = SCANR(IS,J3,IC)
0223         I2 = I2+1
0224         IF ( IPOS  .EQ.  2 ) J3 = 31
0225         IF ( IPOS  .EQ.  1 ) J3 = 39
0226         POS(I2) = SCANR(IS,J3,IC)
0227         I2 = I2+1
0228         DO 21 J3=32,35,1
0229         POS(I2) = SCANR(IS,J3,IC)
0230      21 I2 = I2+1
0231         DO 22 I2=1,5,2
0232      22 POS(I2) = POS(I2)*1000.
0233         DO 23 I2=2,6,2
0234      23 POS(I2) = POS(I2)*10000.
0235         WRITE (LI, 119) NOCR
0236         READ  (LI,   *) POS(7)
0237         WRITE (LI, 149) ICLR
0238      24 WRITE (LI, 118) (POS(I2),I2=1,7),NOCR
0239         READ  (LI, 149) IDUM
0240         WRITE (LI, 149) (ICLR,I=1,11)
0241         IF ( IDUM  .EQ.  2HTA ) GO TO 4711
0242         IF ( IDUM  .EQ.  2HUP ) GO TO 25
0243         GO TO 24
0244    4711 IW = CNTRL(250)
0245         WRITE (LI, 103) NOLF
0246  C      ..................................................................
0247  C      .                                                                .
```

```
0248  C      . Gather data recorded via S/V#1 (J0=1).                    .
0249  CC     .                                                          .
0250  C      ............................................................
0251         J0 = 1
0252         J3 = 7
0253         J2 = 6
0254         PDAT(J2,J1) = ACQN(J0,J3,IW)
0255         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0256         J3 = 9
0257         J2 = 7
0258         PDAT(J2,J1) = ACQN(J0,J3,IW)
0259         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0260         J3 = 20
0261         J2 = 8
0262         PDAT(J2,J1) = ACQN(J0,J3,IW)
0263         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0264         SUM = 0.
0265         DO 1 J3=30,33
0266         J2 = 9
0267         PDAT (J2,J1) = ACQN(J0,J3,IW)
0268         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0269       1 SUM = SUM+PDAT(J2,J1)
0270         PDAT(J2,J1) = SUM/4.
0271  C      ............................................................
0272  CCC    .                                                          .
0273  CCC    .  Gather data recorded via S/V#4 (J0=4).                  .
0274  CCC    .                                                          .
0275  C      ............................................................
0276         J0 = 4
0277         IF ( IPOS  .EQ.  2 ) J3 = 3
0278         IF ( IPOS  .EQ.  1 ) J3 = 7
0279         J2 = 3
0280         PDAT(J2,J1) = ACQN(J0,J3,IW)
0281         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0282         IF ( IPOS  .EQ.  2 ) J3 = 4
0283         IF ( IPOS  .EQ.  1 ) J3 = 8
0284         J2 = 4
0285         PDAT(J2,J1) = ACQN(J0,J3,IW)
0286         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0287         IF ( IPOS  .EQ.  2 ) J3 = 5
0288         IF ( IPOS  .EQ.  1 ) J3 = 9
0289         J2 = 5
0290         PDAT(J2,J1) = ACQN(J0,J3, 1)
0291         WRITE (LI, 121) J2,J1,J0,J3,IW,PDAT(J2,J1),NOLF
0292  C      ............................................................
0293  CCC    .                                                          .
0294  CCC    .  Gather data recorded via scanner#1 (IS=8).              .
0295  CCC    .                                                          .
0296  C      ............................................................
0297         IS = 8
0298         IC = 2
0299         J3 = 17
0300         J2 = 18
0301         PDAT(J2,J1) = SCANR(IS,J3,IC)
0302         WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0303         IC = 1
0304         J3 = 25
0305         J2 = 1
0306         PDAT(J2,J1) = SCANR(IS,J3,IC)
0307         WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0308         J3 = 26
0309         J2 = 10
0310         PDAT(J2,J1) = SCANR(IS,J3,IC)
0311         WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0312         J3 = 27
0313         J2 = 12
0314         PDAT(J2,J1) = SCANR(IS,J3,IC)
0315         WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0316         IF ( IPOS  .EQ.  2 ) J3 = 30
0317         IF ( IPOS  .EQ.  1 ) J3 = 38
0318         J2 = 15
0319         PDAT(J2,J1) = SCANR(IS,J3,IC)
0320         WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0321         IF ( IPOS  .EQ.  2 ) J3 = 31
0322         IF ( IPOS  .EQ.  1 ) J3 = 39
```

```
0323          J2 = 16
0324          PDAT(J2,J1) = SCANR(IS,J3,IC)
0325          WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0326          J3 = 37
0327          J2 = 2
0328          PDAT(J2,J1) = SCANR(IS,J3,IC)
0329          WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0330    C     .....................................................................
0331    CC    .                                                                   .
0332    CC    . Gather data recorded via scanner#2 (IS=15).                       .
0333    CC    .                                                                   .
0334    C     .....................................................................
0335          IS = 15
0336          J2 = 11
0337          SUM = 0.
0338          DO 2 J3=4,5
0339          PDAT(J2,J1) = SCANR(IS,J3,IC)
0340          WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0341        2 SUM = SUM+PDAT(J2,J1)
0342          PDAT(J2,J1) = SUM/2.
0343          J3 = 18
0344          J2 = 13
0345          PDAT(J2,J1) = SCANR(IS,J3,IC)
0346          WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0347          J3 = 19
0348          J2 = 14
0349          PDAT(J2,J1) = SCANR(IS,J3,IC)
0350          WRITE (LI, 122) J2,J1,IS,J3,IC,PDAT(J2,J1),NOLF
0351    C     .....................................................................
0352    CC    .                                                                   .
0353    CC    .  Gather the rest of the required data.                            .
0354    CC    .                                                                   .
0355    C     .....................................................................
0356          PDAT(19,J1) = CNTRL(4)
0357          PDAT(20,J1) = CNTRL(5)
0358          PDAT(21,J1) = CNTRL(6)
0359          PDAT(17,J1) = POS(7)
0360          WRITE (LO, 605) J1,(PDAT(I,J1),I=1,21)
0361       50 WRITE (LI, 117) NOLF,NOCR
0362          READ  (LI, 149) IDUM
0363          WRITE (LI, 149) (ICLR,I=1,6)
0364          IF ( IDUM  .EQ.  2HRE ) GO TO 4711
0365          IF ( IDUM  .EQ.  2HNE ) GO TO 20
0366          IF ( IDUM  .EQ.  2HEN ) GO TO 51
0367          GO TO 50
0368       51 IFILE(1) = 2HTS
0369          IFILE(2) = ICON(CNTRL(4),0)
0370          IFILE(3) = ICON(IDOCF,0)
0371          JSECU = ICON(ISECU,0)
0372          JCR   = ICON(ICR,0)
0373          WRITE (LI, 116) NOLF,IFILE,JSECU,JCR,NOCR
0374          READ  (LI, 149) IDUM
0375          WRITE (LI, 149) (ICLR,I=1,5)
0376          IF ( IDUM  .EQ.  2HTR ) GO TO 52
0377          IF ( IDUM  .EQ.  2HST ) GO TO 53
0378          GO TO 51
0379       52 ISYNCH = 9830
0380    C     .....................................................................
0381    CC    .                                                                   .
0382    CC    .  Data transfer to HP 9830 for reduction.  No storage on 21MX!     .
0383    CC    .                                                                   .
0384    C     .....................................................................
0385       71 WRITE (LI, 210) NOCR
0386          READ  (LI, 149) IDUM
0387          WRITE (LI, 149) ICLR
0388          IF ( IDUM  .NE.  2HYE ) GO TO 71
0389          WRITE ( 7, 701) ISYNCH
0390          WRITE (LI, 216) NOCR
0391          READ  (LI,   *) IDUM
0392          WRITE (LI, 149) ICLR
0393          WRITE (LI, 211) NOLF
0394          DO 72 I=1,21
0395          DO 72 J=1,24
0396       72 WRITE ( 7, 702) PDAT(I,J)
0397          WRITE (LO, 189) ((PDAT(I,J),I=1,21),J=1,24)
```

```
0398          WRITE (LI, 149) ICLR
0399          WRITE (LI, 215) NOCR
0400          READ  (LI,   *) IDUM
0401          WRITE (LI, 149) ICLR
0402          WRITE (LO, 606)
0403          GO TO 66
0404    C     ..................................................................
0405    C     .                                                                .
0406    C     .  Save data on 21MX disc.  No transfer to HP 9830.              .
0407    C     .                                                                .
0408    C     ..................................................................
0409       53 CONTINUE
0410      418 WRITE (LI,105) (IFILE(J2),J2=1,3),NOLF
0411          CALL CREAT (IDCB,IERR,IFILE,ISIZE,ITYPE,ISECU,ICR,IDCBS)
0412          IF ( IERR  .GT.  0 ) GO TO 420
0413          WRITE (LI,101) (IFILE(J2),J2=1,3)
0414          READ  (LI,107) IDUM
0415          WRITE (LI,149) ICLR
0416          IF ( IDUM  .NE.  2HPU ) GO TO 419
0417          CALL PURGE (IDCB,IERR,IFILE,ISECU,ICR)
0418          IF ( IERR  .LT.  0 ) STOP 15
0419          GO TO 418
0420      419 CALL CODE
0421          WRITE (NEW,100) IDUM
0422          WRITE (LI,102) (IFILE(J2),J2=1,3),NEW,(IFILE(J2),J2=2,3)
0423          IFILE(1) = NEW
0424          GO TO 418
0425      420 CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0426          IF ( IERR  .LT.  0 ) STOP 16
0427          CALL WRITF (IDCB,IERR,PDAT,IL)
0428          IF ( IERR  .LT.  0 ) STOP 17
0429          CALL CLOSE (IDCB,IERR,0)
0430          IF ( IERR  .LT.  0 ) STOP 20
0431          WRITE (LI,149) ICLR
0432          WRITE (LO, 607) IFILE,JSECU,JCR
0433       66 CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0434          WRITE (LO,610) IHOUR,IMIN
0435          RETURN
0436          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 03462 COMMON = 00000

```
0437          SUBROUTINE STDY (IRUN)
0438    C     .....................................................................
0439    C     .                                                                   .
0440    C     .  Acquires data from the transsonic 1-stage axial compressor.      .
0441    C     .  This subroutine takes care of the steady state data, either      .
0442    C     .  in a long or a short run (IRUN=1 resp. 0).                       .
0443    C     .                                                                   .
0444    C     .....................................................................
0445          *, Takes data from steady state system.

0446          COMMON / CIBUF / IBUF
0447          COMMON / CONTR / CNTRL
0448          COMMON / FMP   / IDCB,IFILE,ISIZE,ISECU,ICR
0449          INTEGER IBUF(1664),ITIME(5),IDATA(8,4),NO(2),NOLF,NOCR(2),ICLR(3)
0450          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0451          INTEGER CNTRL(256)
0452          REAL DATA(48,4)
0453          EQUIVALENCE (IBUF(1),DATA(1,1))
0454          DATA NOLF  /006537B/
0455          DATA NOCR  /000033B,040433B/
0456          DATA ICLR  /015524B,015515B,006537B/
0457    C     FORMATS  STDY  START
0458      100 FORMAT (A1"4")
0459      101 FORMAT ("WARNING: File "3A2" already exists!  Type ",
0460         *"PU to allow purge or enter any char-",
0461         */,"          after but T to change file name."38X)
0462      102 FORMAT ("  STDY  : File name "3A2" succesfully changed to "3A2)
0463      103 FORMAT ("  STDY  : PU,"3A2":"A2":"A2)
0464      104 FORMAT (I2)
0465      105 FORMAT (""26X"storing data in file "3A2,26X""A2)
0466      106 FORMAT (" Enter case angle  :  "R1"J"2A2)
0467      110 FORMAT (/"Look at the just acquired data! Decide and enter"//
0468         *"RP  ...  to repeat the data acquisition"/
0469         *"RT  ...  to return to the calling program"//
0470         *"  "A2)
0471      112 FORMAT (/"Do you want"//
0472         *"0  ...  a short run"/
0473         *"1  ...  a long run"//" "A2)
0474      115 FORMAT (4F16.6)
0475      120 FORMAT (" DATA("I2","I1") = ACQN("I1","I2","I3")    has been execu
0476         *ted;  result is "F10.6""A2)
0477      121 FORMAT (" DATA("I2","I1") = SCANR("I2","I2","I1")    has been exec
0478         *uted;  result is "F10.6""A2)
0479      147 FORMAT (I1)
0480      149 FORMAT ((3A2))
0481      601 FORMAT (1H1,""/////,36X
0482         *,"Steady State Data Documentation P
0483         *//////)
0484      602 FORMAT (1H ,45(1H ),33HTranssonic Compressor Test Run # ,I7)
0485      603 FORMAT (1H ,58(1H ),6HDate: ,A2,1H/,A2,1H/,A2)
0486      604 FORMAT (1H ,""58(1H ),6HTime: ,A2,1H.,A2,3H  h,/////)
0487      605 FORMAT (1H ,46(1H ),""4I15"")
0488      606 FORMAT (1H ,36(1H ),I2,8(1H ),""4(A1,7A2)"",I7)
0489      607 FORMAT (1H ,46(1H ),4(A1,7A2))
0490      608 FORMAT (1H ,36(1H ),I2,8(1H ),4(A1,7A2),I7)
0491      609 FORMAT (1H ,46(1H ),4I15,//)
0492      610 FORMAT (1H ,36X,"Long run performed.  Above printed data are sav
0493         *ed in file "3A2":"A2":"I2,/
0494         *1H ,36X,"Note : "2A2" ... no data acquired.")
0495      611 FORMAT (1H ,36X,"Short run performed.  Above printed data are sa
0496         *ved in file "3A2":"A2":"I2,/
0497         *1H ,36X,"Note : "2A2" ... no data acquired.")
0498      612 FORMAT ("
```

",58(1H),6HTime: ,A2,1H.,A2,3H h,/////)

```
0499     900 FORMAT (" A REGISTER IS "K6"      B REGISTER IS "K6/)
0500    1001 FORMAT ("F1R7M3A1H0T3")
0501    1201 FORMAT ("PF4G6T")
0502  C      FORMATS STUDY STOP
0503         IF ( IENTR  .NE.  0 ) GO TO  1
0504         IENTR     = 1
0505         IDOC      = 0
0506         IDOCF     = 0
0507         ISECU     = CNTRL(31)
0508         ICR       = CNTRL(30)
0509         ITYPE     = 1
0510         IL        = 384
0511         IDCBS     = 144
```

```
0512       01 ISIZE(1) = 3
0513          ISIZE(2) = 128
0514          WRITE (10,1001)
0515          WRITE (12,1201)
0516          IFRST    = 2HT4
0517          LI       = CNTRL(19)
0518          LO       = CNTRL(20)
0519          CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0520          WRITE (LO,602) CNTRL(4)
0521          WRITE (LO,603) IMON,IDAY,IYEAR
0522          WRITE (LO,604) IHOUR,IMIN
0523          WRITE (LO,601)
0524          IDOC  = 1+IDOC
0525          IDOCF = 1+IDOCF
0526       05 WRITE (10,1001)
0527       C  .....................................................................
0528       CC .                                                                   .
0529       CC . Preset data array DATA(48,4) with the dummy variable -.999999     .
0530       CC . to make trouble shooting easier.                                  .
0531       CC .                                                                   .
0532       C  .....................................................................
0533          DO 06 J1=1,4,1
0534          DO 06 J2=1,48,1
0535       06 DATA(J2,J1) = -0.999999
0536          IF ( IRUN  .EQ.  0 ) GO TO 11
0537       C  .....................................................................
0538       CC .                                                                   .
0539       CC .  Acquire pressures from scannivalve-scanner-DVM system.  Only     .
0540       CC .  performs in a long run.                                          .
0541       CC .                                                                   .
0542       C  .....................................................................
0543          J0    = 1
0544          J1    = 1
0545          IW    =  CNTRL(250)
0546          DO 07 J2=1,48,1
0547          DATA(J2,J1) = ACQN(J0,J2,IW)
0548       07 WRITE (LI, 120) J2,J1,J0,J2,IW,DATA(J2,J1),NOLF
0549          J0    = 4
0550          J1    = 2
0551          IW    = CNTRL(250)
0552          DO 08 J2=1,45,1
0553          DATA(J2,J1) = ACQN(J0,J2,IW)
0554       08 WRITE (LI, 120) J2,J1,J0,J2,IW,DATA(J2,J1),NOLF
0555       C  .....................................................................
0556       CC .                                                                   .
0557       CC .  Subtract tare from pressure readings.                            .
0558       CC .                                                                   .
0559       C  .....................................................................
0560          TARE1        = DATA( 1, 1)
0561          TARE2        = DATA( 1, 2)
0562          J1 = 1
0563          DO 09 J2=1,48,1
0564       09 DATA(J2,J1) = DATA(J2,J1)-TARE1
0565          J1 = 2
0566          DO 10 J2=1,45,1
0567       10 DATA(J2,J1) = DATA(J2,J1)-TARE2
0568       C  .....................................................................
0569       CC .                                                                   .
0570       CC .  Acquire data (pressures, temperatures, torque, speed, probe      .
0571       CC .  positions) from scanner-DVM or scanner-counter system.           .
0572       CC .                                                                   .
0573       C  .....................................................................
0574       11 J1 = 3
0575          IS = 0
0576          IC = 1
0577          J3 = 25
0578          DO 12 J2=1,5,1
0579          DATA(J2,J1) = SCANR(IS,J3,IC)
0580          WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0581       12 J3 = J3+1
0582          J3 = 30
0583          DO 13 J2=7,12,1
0584          DATA(J2,J1) = SCANR(IS,J3,IC)
0585          WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0586       13 J3 = J3+1
```

```
0587            J3 = 0
0588            IS = 15
0589            DO 14 J2=14,17,1
0590            DATA(J2,J1) = SCANR(IS,J3,IC)
0591            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0592         14 J3 = J3+1
0593            J3 = 4
0594            DO 15 J2=19,25,1
0595            DATA(J2,J1) = SCANR(IS,J3,IC)
0596            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0597         15 J3 = J3+1
0598            J3 = 12
0599            DO 16 J2=27,31,1
0600            DATA(J2,J1) = SCANR(IS,J3,IC)
0601            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0602         16 J3 = J3+1
0603            IS = 8
0604            J3 = 37
0605            J2 = 33
0606            DATA(J2,J1) = SCANR(IS,J3,IC)
0607            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0608            IS = 15
0609            J3 = 40
0610            DO 17 J2=34,45,1
0611            DATA(J2,J1) = SCANR(IS,J3,IC)
0612            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0613         17 J3 = J3+1
0614            J3 = 52
0615            DO 18 J2=47,48,1
0616            DATA(J2,J1) = SCANR(IS,J3,IC)
0617            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0618         18 J3 = J3+1
0619            IS = 8
0620            IC = 2
0621            J1 = 4
0622            J3 = 19
0623            J2 = 45
0624            DATA(J2,J1) = SCANR(IS,J3,IC)
0625            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0626            J3 = 17
0627            J2 = 46
0628            DATA(J2,J1) = SCANR(IS,J3,IC)
0629            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0630            IC = 1
0631            J3 = 36
0632            J2 = 47
0633            DATA(J2,J1) = SCANR(IS,J3,IC)
0634            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0635            IS = 15
0636            J3 = 19
0637            J2 = 29
0638            DATA(J2,J1) = SCANR(IS,J3,IC)
0639            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0640            J3 = 28
0641            J2 = 30
0642            DATA(J2,J1) = SCANR(IS,J3,IC)
0643            WRITE (LI, 121) J2,J1,IS,J3,IC,DATA(J2,J1),NOLF
0644            DATA(33,J1) = CNTRL(  4)
0645            DATA(34,J1) = CNTRL(  5)
0646            DATA(35,J1) = CNTRL(  6)
0647            DATA(36,J1) = CNTRL(  2)
0648            DATA(37,J1) = CNTRL(  1)
0649            DATA(38,J1) = CNTRL(  3)
0650            DATA(39,J1) = CNTRL( 15)
0651            WRITE (LI, 106) NOCR(1),NOCR
0652            READ  (LI,   *) DATA(43,J1)
0653            WRITE (LI, 149) ICLR
0654            IF ( DATA(43,J1)  .EQ. 9999 ) STOP 0601
0655         C  ..................................................................
0656         CCC :                                                                :
0657         CCC :  Print acquired data.                                          :
0658         CCC :                                                                :
0659         C  ..................................................................
0660            NO(1) = 2H n
0661            NO(2) = 2H/a
```

```
0662          WRITE (LO, 605) (J2,J2=1,4)
0663          J1=1
0664          CALL CODE
0665          WRITE (IDATA,115) (DATA(J1,J2),J2=1,4)
0666          DO 20 J2=1,4,1
0667          IF ( DATA(J1,J2)   .NE.  -.999999 ) GO TO 20
0668          IDATA( 7,J2) = NO(1)
0669          IDATA( 8,J2) = NO(2)
0670          DO 19 J3=1,6,1
0671       19 IDATA(J3,J2) = 2H
0672       20 CONTINUE
0673          WRITE (LO, 608) J1,((IDATA(J3,J2),J3=1,8),J2=1,4),J1
0674          DO 26 J1=2,47,1
0675          IF ( (J1/5)*5   .NE.  J1 ) GO TO 23
0676          CALL CODE
0677          WRITE (IDATA,115) (DATA(J1,J2),J2=1,4)
0678          DO 22 J2=1,4,1
0679          IF ( DATA(J1,J2)   .NE.  -.999999 ) GO TO 22
0680          IDATA( 7,J2) = NO(1)
0681          IDATA( 8,J2) = NO(2)
0682          DO 21 J3=1,6,1
0683       21 IDATA(J3,J2) = 2H
0684       22 CONTINUE
0685          WRITE (LO, 606) J1,((IDATA(J3,J2),J3=1,8),J2=1,4),J1
0686          GO TO 26
0687       23 CALL CODE
0688          WRITE (IDATA,115) (DATA(J1,J2),J2=1,4)
0689          DO 25 J2=1,4,1
0690          IF ( DATA(J1,J2)   .NE.  -.999999 ) GO TO 25
0691          IDATA( 7,J2) = NO(1)
0692          IDATA( 8,J2) = NO(2)
0693          DO 24 J3=1,6,1
0694       24 IDATA(J3,J2) = 2H
0695       25 CONTINUE
0696          WRITE (LO, 607) ((IDATA(J3,J2),J3=1,8),J2=1,4)
0697       26 CONTINUE
0698          J1=48
0699          CALL CODE
0700          WRITE (IDATA,115) (DATA(J1,J2),J2=1,4)
0701          DO 28 J2=1,4,1
0702          IF ( DATA(J1,J2)   .NE.  -.999999 ) GO TO 28
0703          IDATA( 7,J2) = NO(1)
0704          IDATA( 8,J2) = NO(2)
0705          DO 27 J3=1,6,1
0706       27 IDATA(J3,J2) = 2H
0707       28 CONTINUE
0708          WRITE (LO, 606) J1,((IDATA(J3,J2),J3=1,8),J2=1,4),J1
0709          WRITE (LO, 609) (J2,J2=1,4)
0710          CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN)
0711          ISECU = ICON(ISECU,0)
0712          ICR   = ICON(ICR,0)
0713          IF  ( IDOCF  .LT.  100 ) GO TO 32
0714          IFRST = 2H84
0715          IDOCF = IDOCF-99
0716       32 IFILE(1) = IFRST
0717          IFILE(2) = ICON(CNTRL(4),0)
0718          IFILE(3) = ICON(IDOCF,0)
0719    C     ..................................................................
0720    C     .                                                                .
0721    C     .  Ask operator whether to repeat the data acquisition at this   .
0722    C     .  setting (RP) or to return to the calling program (RT).        .
0723    C     .                                                                .
0724    C     ..................................................................
0725       33 WRITE (LI, 110) NOLF
0726          READ  (LI, 149) IDUM
0727          WRITE (LI, 149) (ICLR,I=1,7)
0728          IF ( IDUM  .EQ.  2HRP ) GO TO 34
0729          IF ( IDUM  .EQ.  2HRT ) GO TO 35
0730          IF ( IDUM  .EQ.  2Hss ) STOP 0602
0731          GO TO 33
0732       34 WRITE (LI, 112) NOLF
0733          READ  (LI, 147) IRUN
0734          WRITE (LI, 149) (ICLR,I=1,7)
0735          WRITE (LO, 612) IHOUR,IMIN
0736          WRITE (LO, 604) IHOUR,IMIN
```

```
0737          GO TO 05
0738    C     ..................................................................
0739    C     .                                                                .
0740    C     .  Store acquired data on a disc type 1 file.                    .
0741    C     .                                                                .
0742    C     ..................................................................
0743       35 WRITE (LI, 105) IFILE,NOLF
0744          CALL CREAT (IDCB,IERR,IFILE,ISIZE,ITYPE,ISECU,ICR,IDCBS)
0745          IF ( IERR  .GT.  0 ) GO TO 37
0746          WRITE (LI, 101) IFILE
0747          READ  (LI, 149) IDUM
0748          WRITE (LI, 149) (ICLR,I=1,3)
0749          IF ( IDUM  .EQ.  2H$$ ) STOP 0603
0750          IF ( IDUM  .NE.  2HPU ) GO TO 36
0751          WRITE (LI, 103) IFILE,JSECU,JCR
0752          CALL PURGE (IDCB,IERR,IFILE,ISECU,ICR)
0753          IF ( IERR  .LT.  0 ) STOP 0604
0754          GO TO 35
0755       36 CALL CODE
0756          WRITE (NEW,100) IDUM
0757          WRITE (LI, 102) IFILE,NEW,IFILE(2),IFILE(3)
0758          IFILE(1) = NEW
0759          GO TO 35
0760       37 CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0761          IF ( IERR  .LT.  0 ) STOP 0605
0762          CALL WRITF (IDCB,IERR,DATA,IL)
0763          IF ( IERR  .LT.  0 ) STOP 0606
0764          CALL CLOSE (IDCB,IERR,0)
0765          IF ( IERR  .LT.  0 ) STOP 0607
0766          IF ( IRUN  .EQ.  1 ) WRITE (LO,610) IFILE,ISECU,ICR,NO
0767          IF ( IRUN  .EQ.  0 ) WRITE (LO,611) IFILE,JSECU,ICR,NO
0768          WRITE (LO,612) IHOUR,IMIN
0769          RETURN
0770          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 03193 COMMON = 00000

6. PROGRAM TXCO3

6.1. DESCRIPTION

TXCO3 is a son program of the father program TXCOØ, by which it is scheduled if one of the following operations is desired:

7 - Check the instrumentation

8 - Manipulate the program control array CNTRL.

When scheduled by TXCOØ, which suspends operation while the son program TXCO3 executes, the program TXCO3 reads the program control array from the disc, sets the HP interface bus and the measurement and control devices to remote control, preprograms the digital voltmeter (DVM), the scanners and the counter. CNTRL(50) is the actual decision variable to select and call the subroutine, which performs the desired operation. When this subroutine has terminated, the interface bus and the devices are released from remote control and the control array is written into a disc file, so that the next TXCO module can read it. The correct termination of each subroutine can be verified by checking the stop codes. Note that all stop codes ending on 77 indicate correct execution of a subroutine.

CNTRL(50)	Subroutine	STOP Code
7	CHECK	TXCO3 : STOP 0777
8	CHNGE	TXCO3 : STOP 1077

EXTERNALS: REWRF, ABRT, RMOTE, CHECK, CHNGE, CLEAR, LOCL

COMMON BLOCKS: FMP, CIBUF, CONTR.

The FORTRAN-IV compiler for the HP 21 MX computer requests COMMON blocks to be predefined in a BLOCK DATA subroutine prior to using a COMMON block in a program, subroutine or function.

BLOCK DATA Subroutine	Arrays & Variables	Length in Words
FMP	IDCB,IFILE,ISIZE,ISECU,ICR	$227B = 151_{10}$
CIBUF	IBUF	$3200B = 1664_{10}$
CONTR	CNTRL	$400B = 256_{10}$

The arrays and variables allocated by the COMMON block FMP are frequently used for the data transfer from and to the disc. COMMON block CIBUF is designed to take the largest raw data array in the TXCO data acquisition and reduction system - IBUF(1664) in subroutine FREER. The program modules CHECK and CHNGE do not use the complete area allocated by CIBUF. COMMON block CONTR allocates the space for the control array CNTRL.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: If CNTRL(50) is less than 7 or greater than 8, no subroutine has been selected and the program terminates outputting an error message (FORMAT #102) to the standard input device, i.e. the terminal.

PROCEDURE: For more detailed information, study the flow chart and the information given in the subroutine descriptions.

DATA FILE: None

VARIABLES IN BLOCK DATA FMP:

ICDB (144)	integer	data control block
IFILE (3)	integer	array to contain file name
ISIZE (2)	integer	array to contain # of records in the first and record length in 16-bit-words in the second word
ISECU	integer	security code of data file
ICR	integer	cartridge reference number, where data file is located

VARIABLES IN BLOCK DATA CIBUF:

IBUF (1664)	integer	buffer array for the raw data

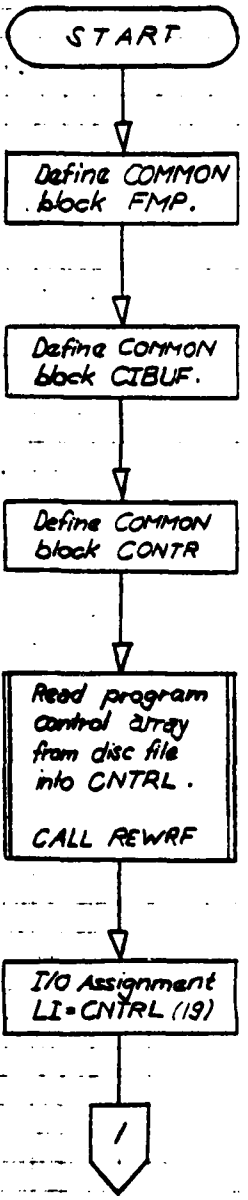
VARIABLES IN BLOCK DATA CONTR:

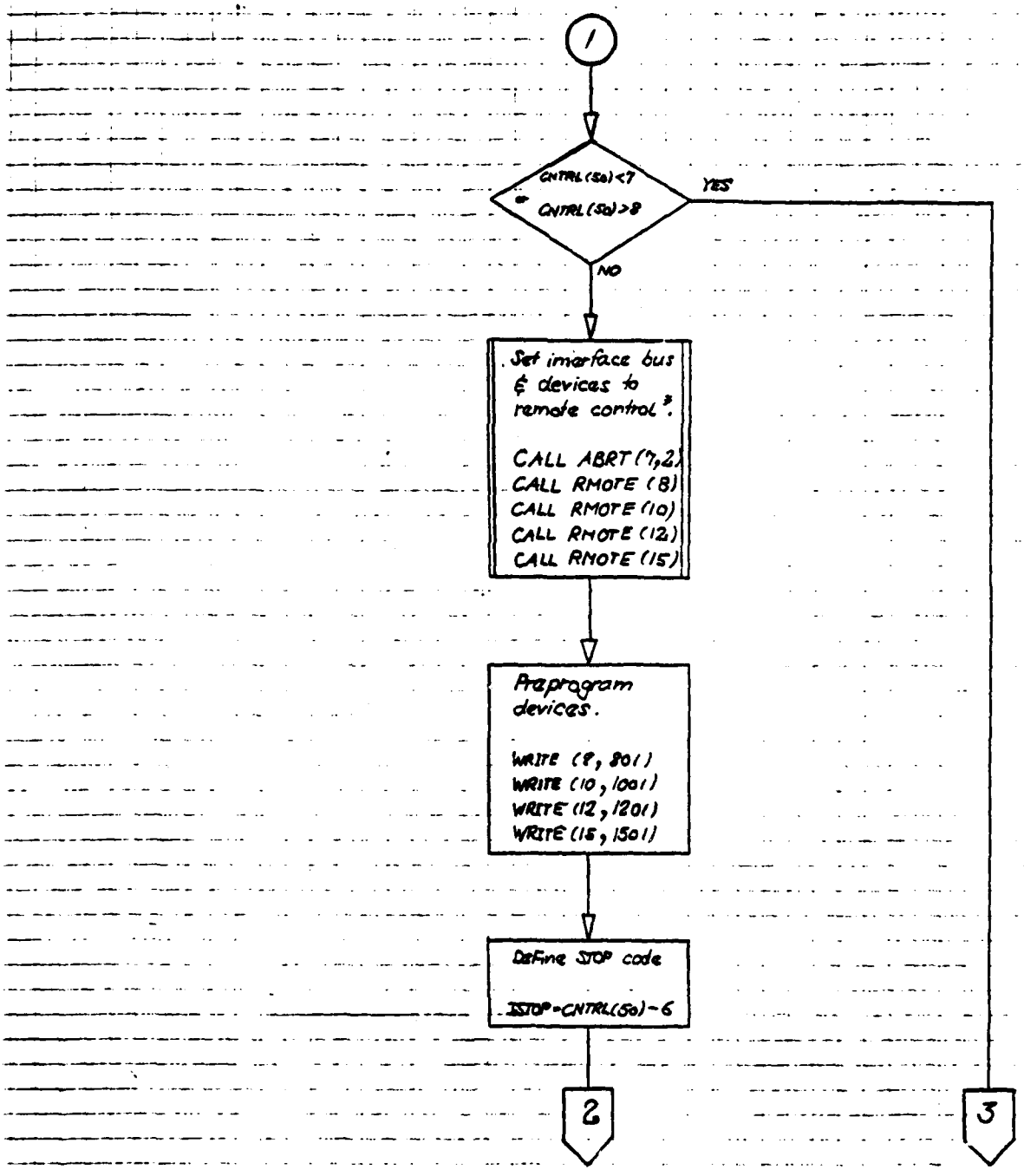
CNTRL (256)	integer	program control array

VARIABLES IN PROGRAM TXCO3:

CNTRL (256)	integer	program control array
NOLF	integer	suppresses line feed
LI	integer	LU # of standard input device (terminal)
ISTOP	integer	control variable to select STOP code

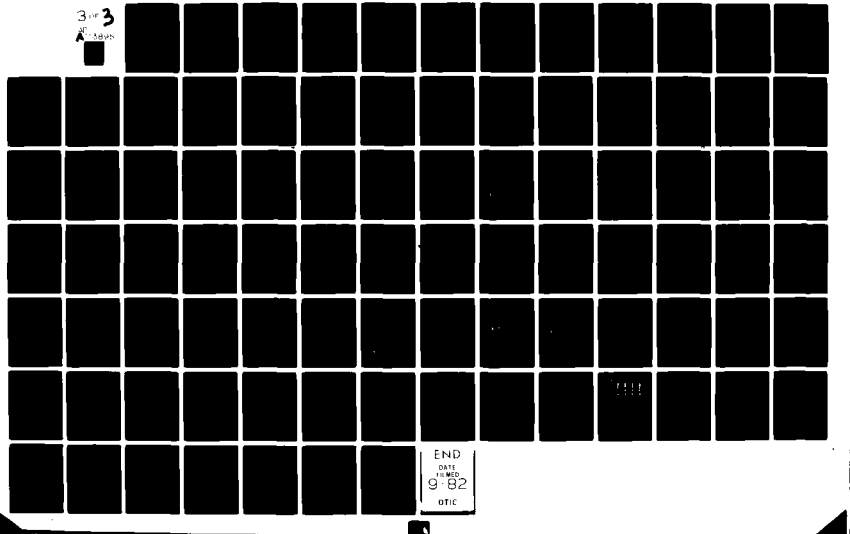
FLOW CHART PROGRAM TXC03

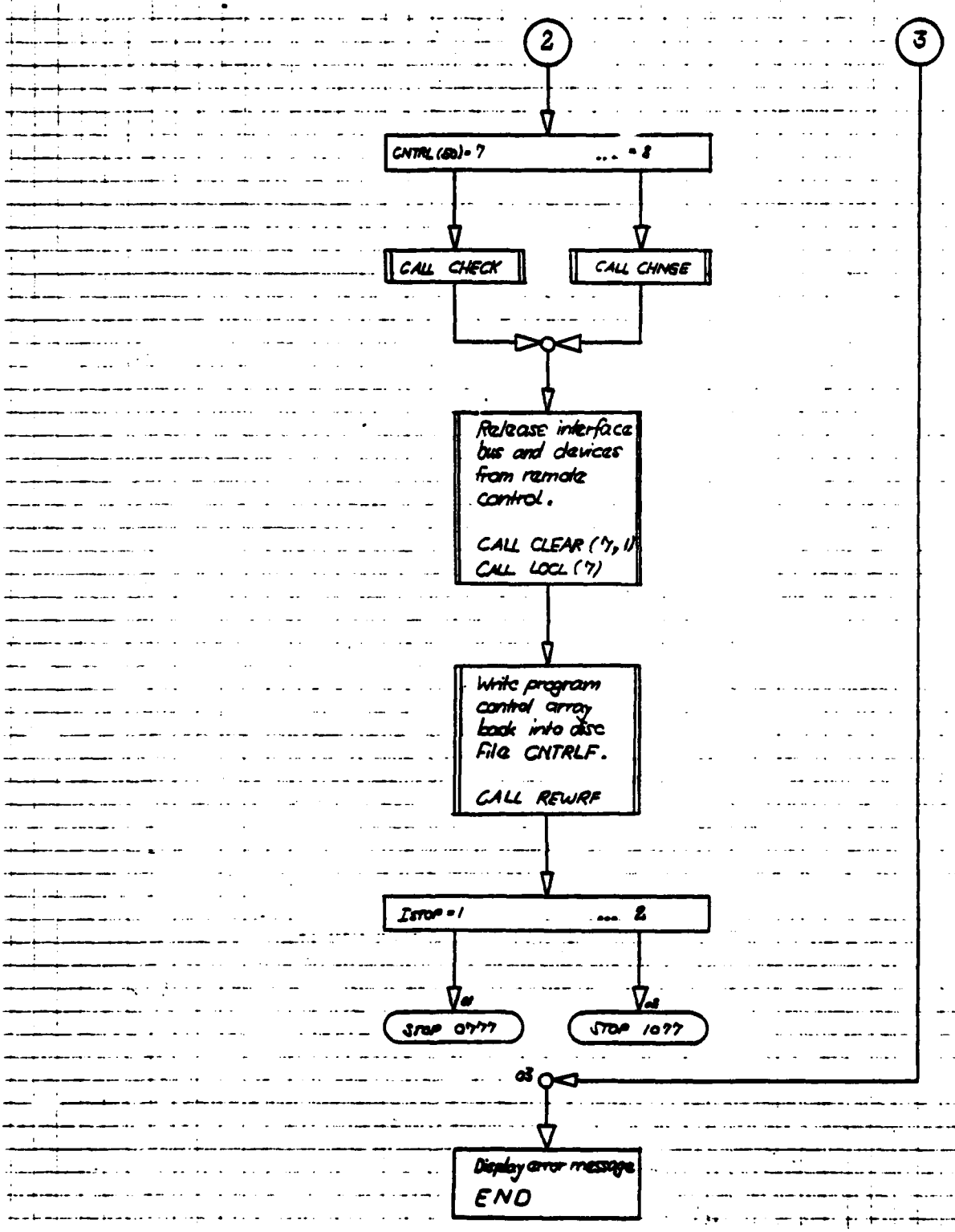




*) Lu Assignments:
7 HP Interface Bus
8 Scanner #1
10 Digital Voltmeter
12 Counter
15 Scanner #2

AD-A113 895 BDM CORP MONTEREY CA F/G 5/8
TRANSONIC COMPRESSOR: PROGRAM SYSTEM TXCO FOR DATA ACQUISITION --ETC(U)
OCT 80 H ZEBNER N00014-78-C-0204
UNCLASSIFIED NPS-67-80-02CR NL




2
3
CNTRL(60)=7 ... =8
CALL CHECK
CALL CHNGE
Release interface bus and devices from remote control.
CALL CLEAR (7,1)
CALL LOCL (7)
Write program control array back into disc file CNTRLF.
CALL REWRF
ISTOP=1 ... 2
STOP 07777
STOP 10777
Display error message
END

6.2. SUBROUTINE CHECK:

PURPOSE: This subroutine enables the investigator to (independently from the data acquisition modules ABSRV, CALIB, FREER, PACER, COMB and STDY) check all data locations to troubleshoot or verify the transonic compressor test rig measurement system.

ARGUMENTS: LO ; this variable specifies the output unit where the protocol of the check is directed to. In any case, the data are displayed on the standard input device (terminal LI) and if LO is equal to LI, double output is suppressed. The selection of LO = 6 (line printer) is an appropriate choice for a hardcopy of the check protocol.

EXTERNALS: ACQN, SCANR, RPACE

COMMON BLOCK: CONTR; for detailed explanation refer to the TXCO3 description.

MNEMONIC ABBREVIATIONS: None

ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the I/O reference, CHECK asks the operator which particular system should be checked.

Instrumentation code	system being checked
1	S/V-S/V controller - scanner - DVM - system

Instrumentation Code	System Being Checked
2	amplifier - scanner - DVM - system
3	Pacer

The operator then selects the desired code and the program branches.

i) S/V - S/V controller - scanner - DVM - system

The operator has to input the number (1 thru 5) of the S/V, the low port and the high port. Erroneous input will cause the program to re-request the data. If S/V #2 is selected and either low or high port are odd, they will be increased to the next even number. In increments of 1 (2 resp., if S/V #2 is addressed) the subroutine steps from low to high port, taking a reading of each. The result is displayed and printed immediately. Upon completion the operator is asked whether another check shall be done. The answer is YES or NO, and if YES is entered, SUBROUTINE CHECK is run again from the beginning.

ii) amplifier - scanner - DVM - system

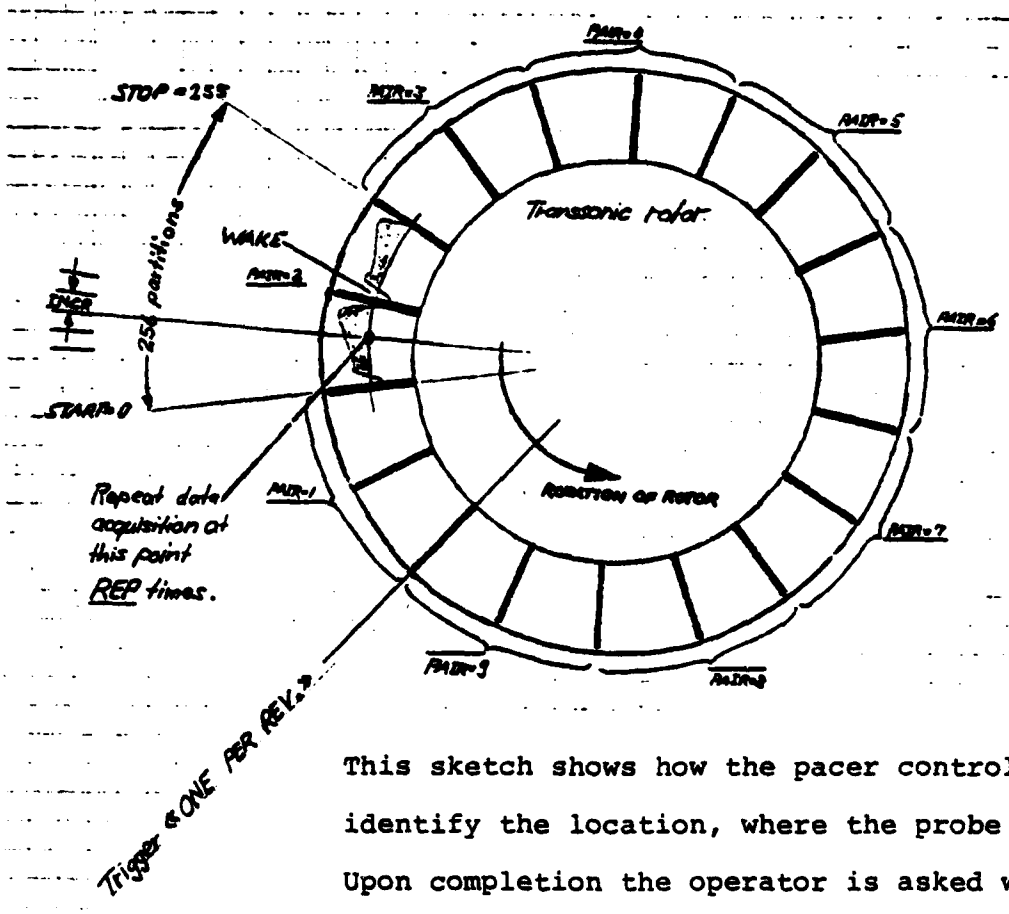
The operator has to input the number (1 or 2) of the scanner, the low channel and the high channel. Erroneous input of the scanner # will cause the program to re-request the data, whereas no check is made whether the boundaries for the scan, low channel ILOW and high channel IHIGH, are correct. In increments of 1 the subroutine steps from low channel to high channel, taking a reading at each port. The result is displayed and printed immediately. Upon completion, the operator is

asked whether another check shall be done. The answer is YES or NO, and if YES is entered, SUBROUTINE CHECK is run again from the beginning.

iii) Pacer

The operator has to input the pacer control parameters:

ADCHNL	A/D analog input channel to be selected by the A/D converter multiplexer.
PAMO	Pacer mode = 1 allows pacer to trigger A/D conversion at the specified position in any blade interval. The variable PAIR is ignored. = 2 causes pacer to select blade pair # PAIR.
PAIR	# of blade pair selected (1 - 9)
START	Start count to step through blade passage
INCR	Increment to step through blade passage
STOP	Stop count to step through blade passage
REP	Number of repetitions at each individual point



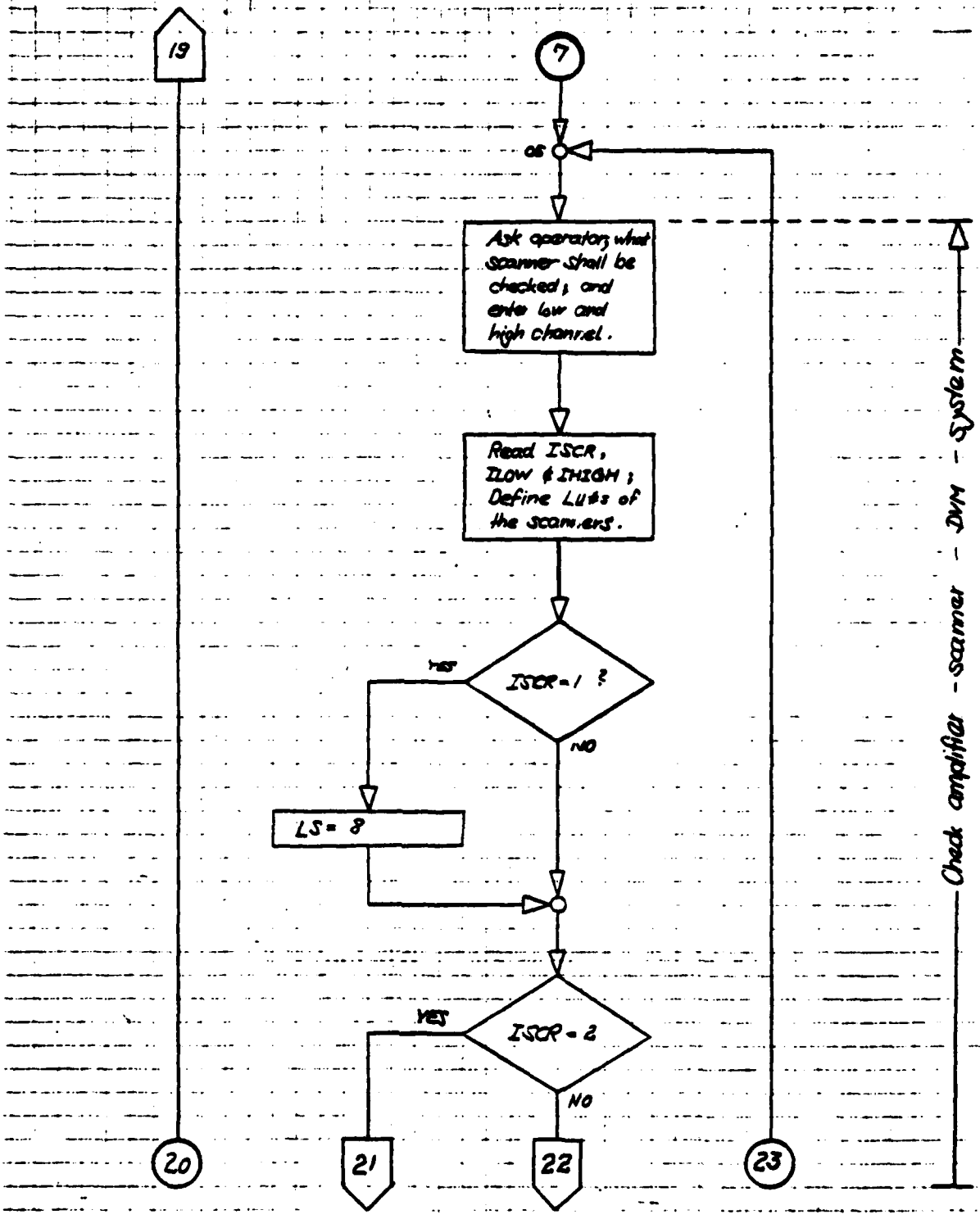
This sketch shows how the pacer control parameters identify the location, where the probe takes data. Upon completion the operator is asked whether another check shall be done. The answer is YES or NO, and if YES is entered, start to read this section SUBROUTINE CHECK again.

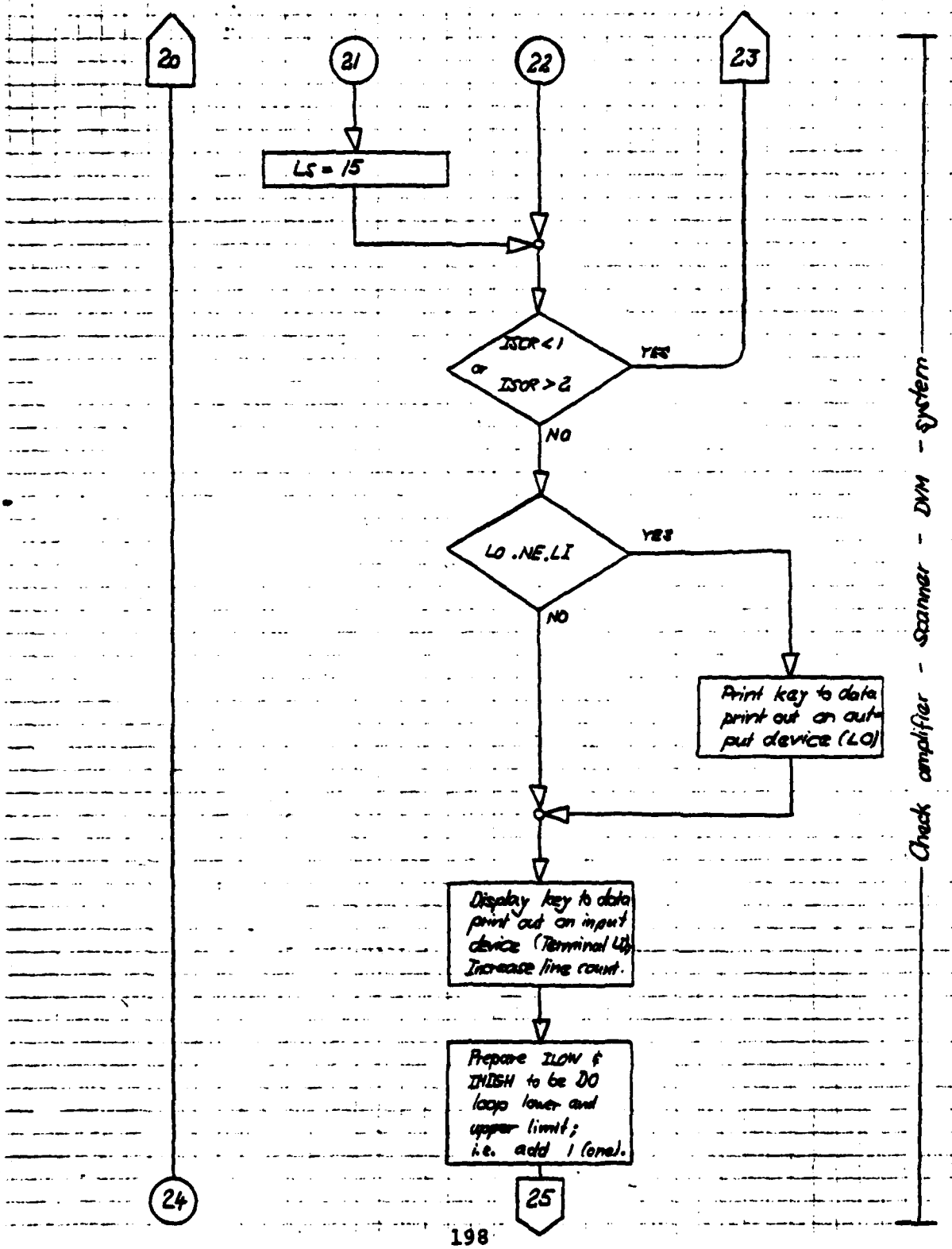
DATA FILE: None

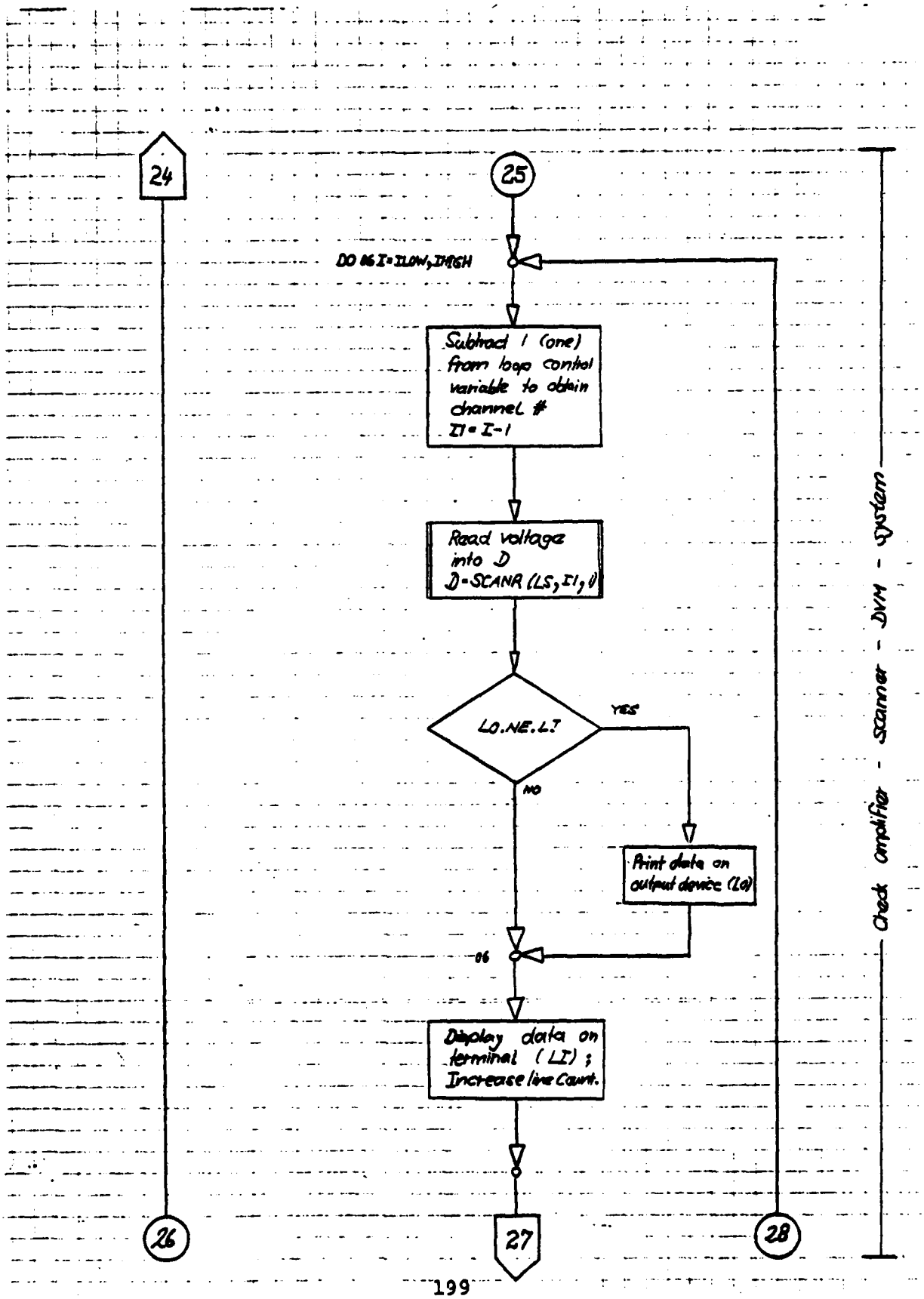
VARIABLES:

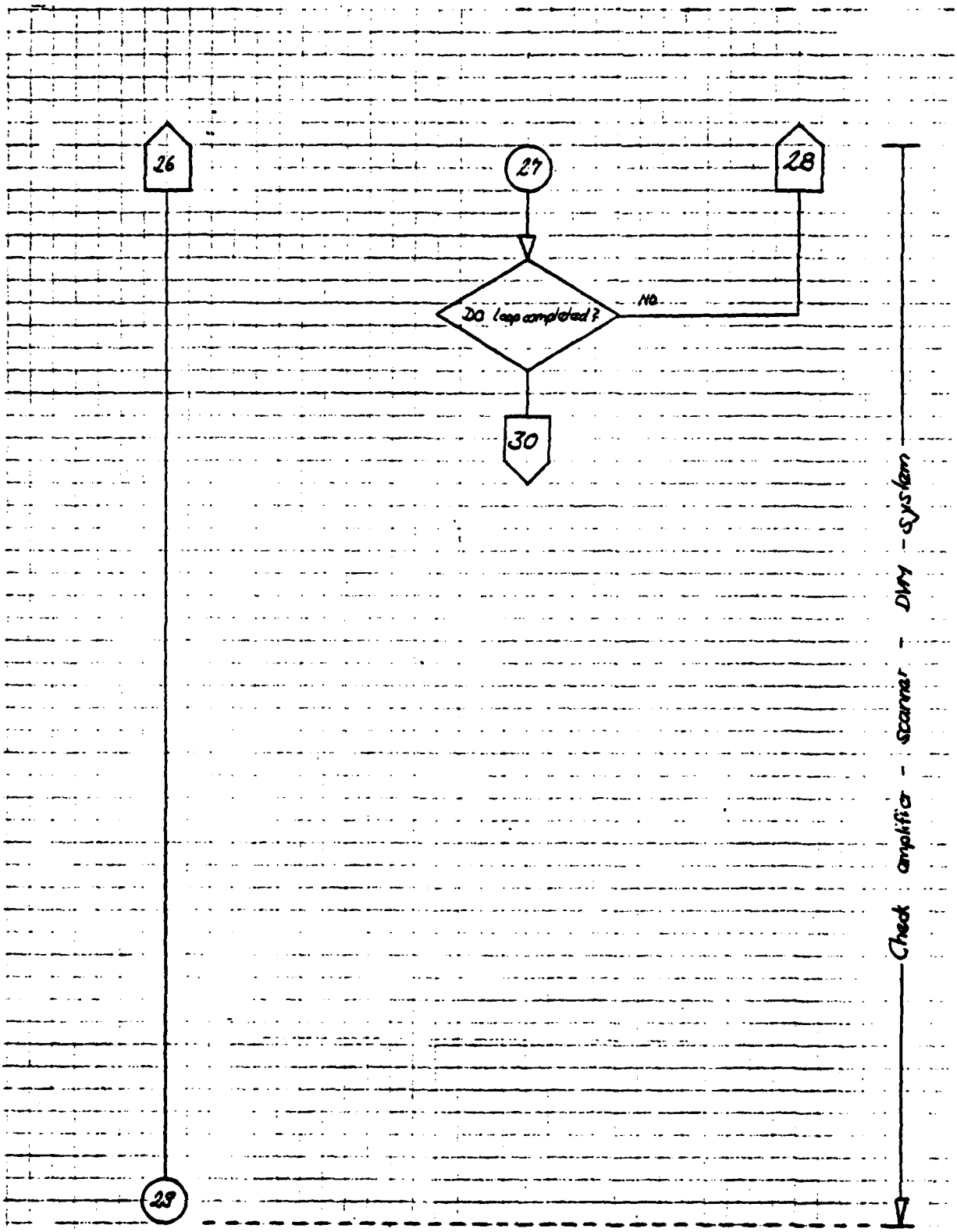
LO	integer	LU# of standard output device (line printer)
CNTRL(256)	integer	program control array
NOLF	integer	suppresses line feed
NOCR(2)	integer	suppresses line feed and carriage return
ICLR(3)	integer	clears line above cursor

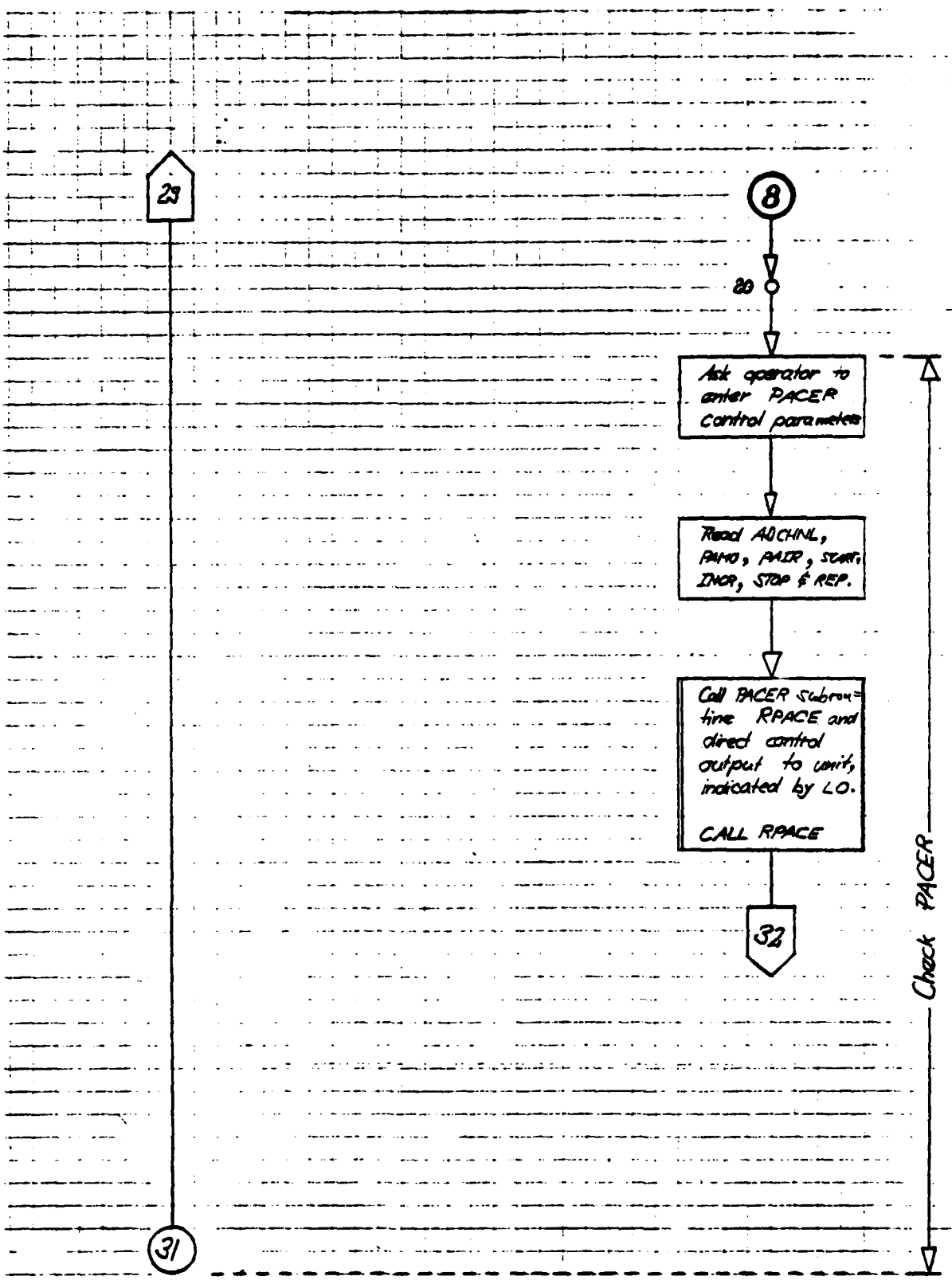
ADCHNL	integer	Pacer control parameter	see section iii) for detailed explanation
PAMO	integer	Pacer control parameter	
PAIR	integer	Pacer control parameter	
START	integer	Pacer control parameter	
INCR	integer	Pacer control parameter	
STOP	integer	Pacer control parameter	
REP	integer	Pacer control parameter	
LI	integer	LU# of standard interactive input device (system terminal)	
LINES	integer	line count	
IDIUM	integer	decision variable	
IPORT	integer	# of desired S/V (1 — 5)	
ILOW	integer	low port of desired S/V	
IHIGH	integer	high port of desired S/V	
ISTEP	integer	increment to step from low to high port	
IW	integer	delay between closing S/V port and taking the DVM reading in tens of ms.	
V	real	pressure reading (raw data)	
ISCR	integer	# of desired scanner (1 or 2)	
ILOW	integer	low channel of desired scanner	
IHIGH	integer	high channel of desired scanner	
LS	integer	LU# of the desired scanner	
D	real	voltage reading (raw data)	
AVRGE	real	average voltage as returned from subroutine RPACE	

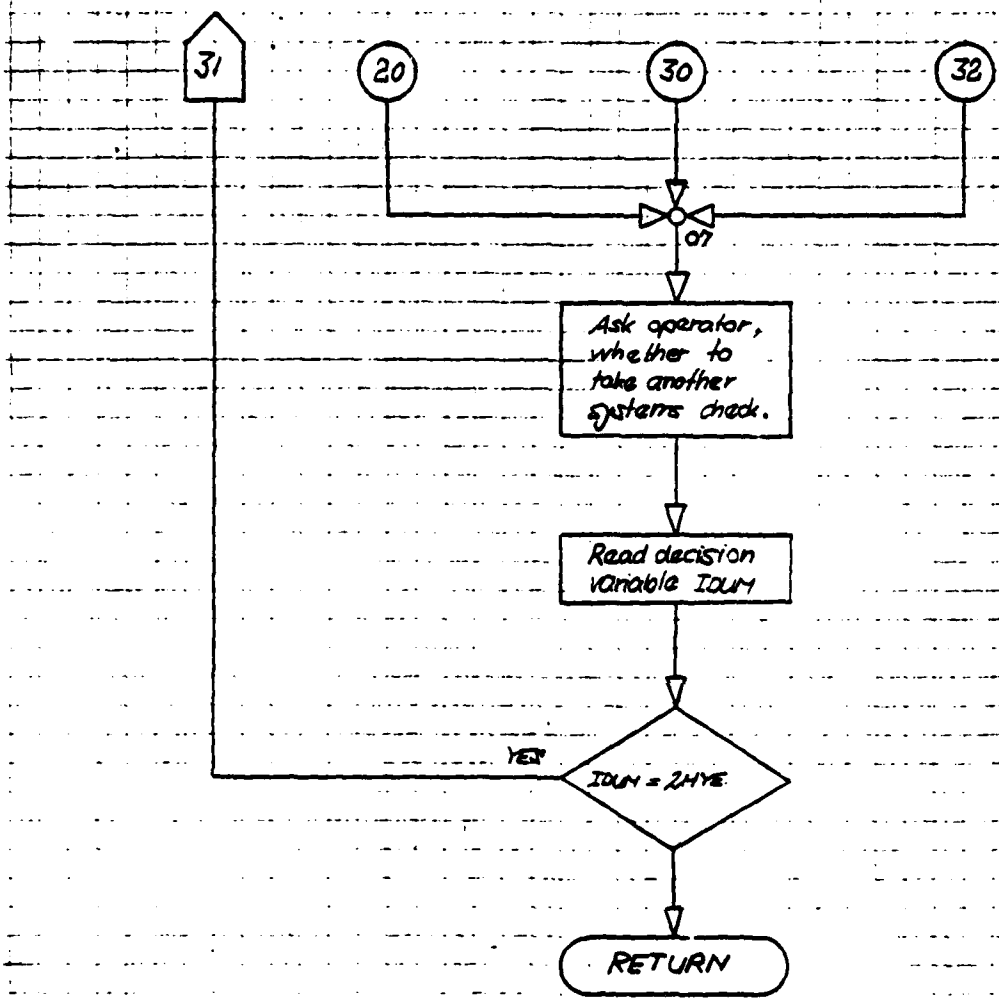

19
7
OS
Ask operator, what scanner shall be checked, and enter low and high channel.
Read ISCR, ILOW & IHIGH; Define LU#s of the scanners.
ISCR = 1 ?
YES
NO
LS = 8
ISCR = 2
YES
NO
20
21
22
23
Check amplifier - scanner - DVM - System

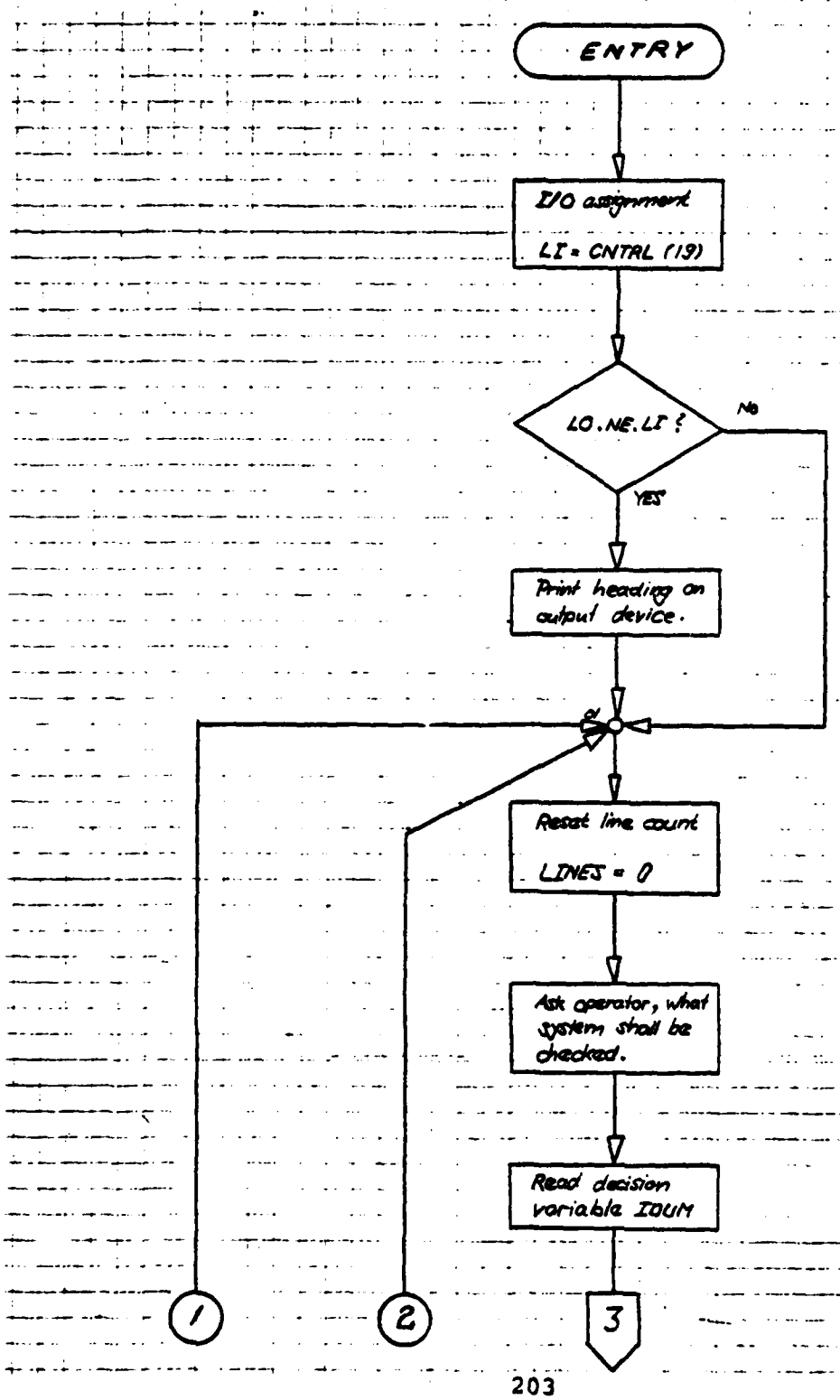

20
21
22
23
LS = 15
ISOR < 1
or
ISOR > 2
YES
NO
LO .NE. LI
YES
NO
Print key to data print out on output device (LO)
Display key to data print out on input device (Terminal U)
Increase line count.
Prepare ILOW & INISH to be DO loop lower and upper limit; i.e. add 1 (one).
24
25
Check amplifier - Scanner - DVM - System

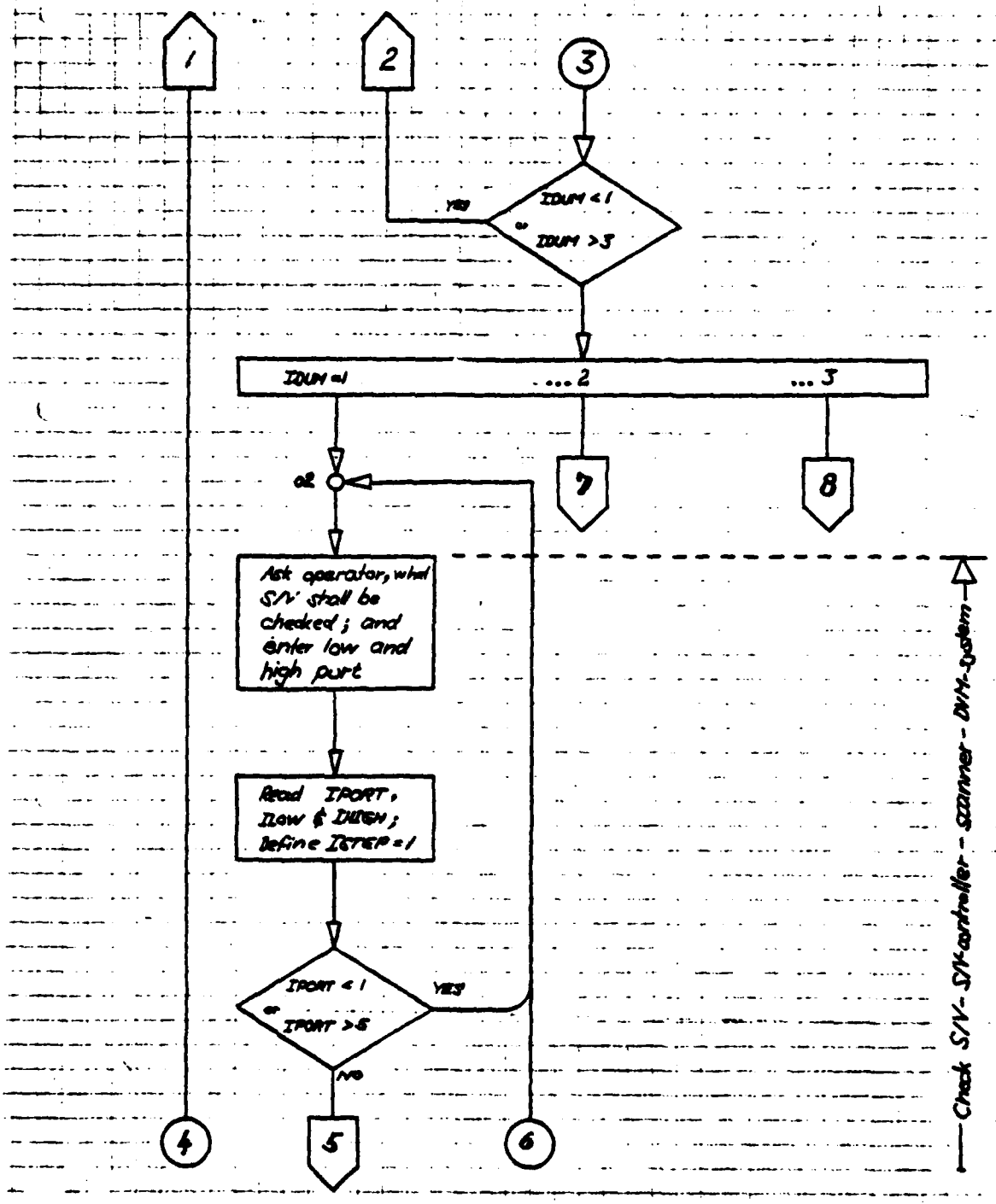

24
25
DO 86 I=ILOW,IHIGH
Subtract 1 (one) from loop control variable to obtain channel #
II=I-1
Read voltage into D
D=SCANR(LS,II,1)
LO.NE.L?
YES
NO
Print data on output device (LO)
86
Display data on terminal (LI); Increase line count.
26
27
28
Check amplifier - scanner - DVM - system

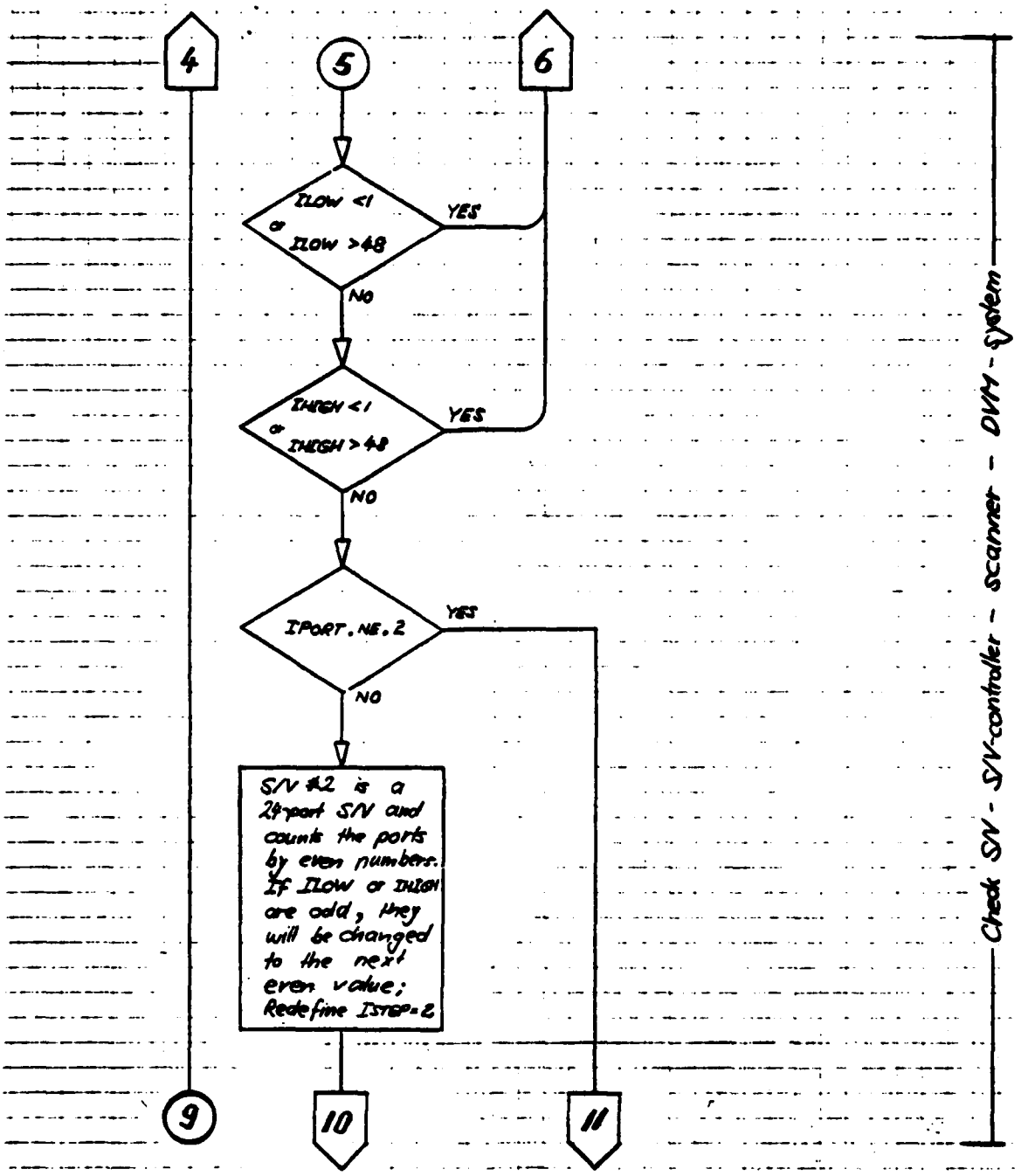

26
27
28
DO loop completed?
NO
30
29
Check amplifier - scanner - DVM - system

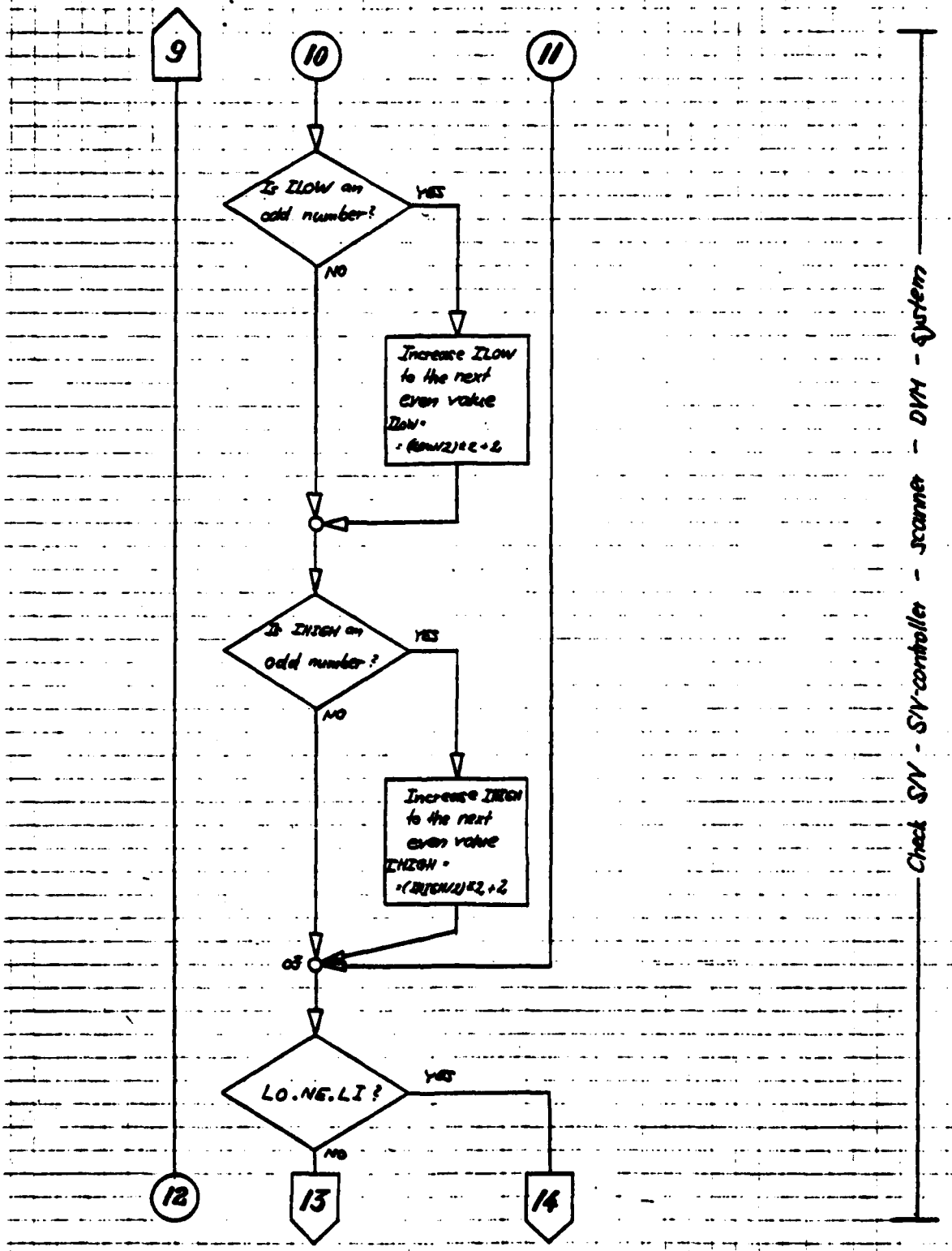

29
8
20
Ask operator to enter PACER control parameters
Read AICHNL, PAMO, PAIR, STAR. INCR, STOP & REP.
Call PACER Subroutine RPACE and direct control output to unit, indicated by LO.
CALL RPACE
32
31
Check PACER

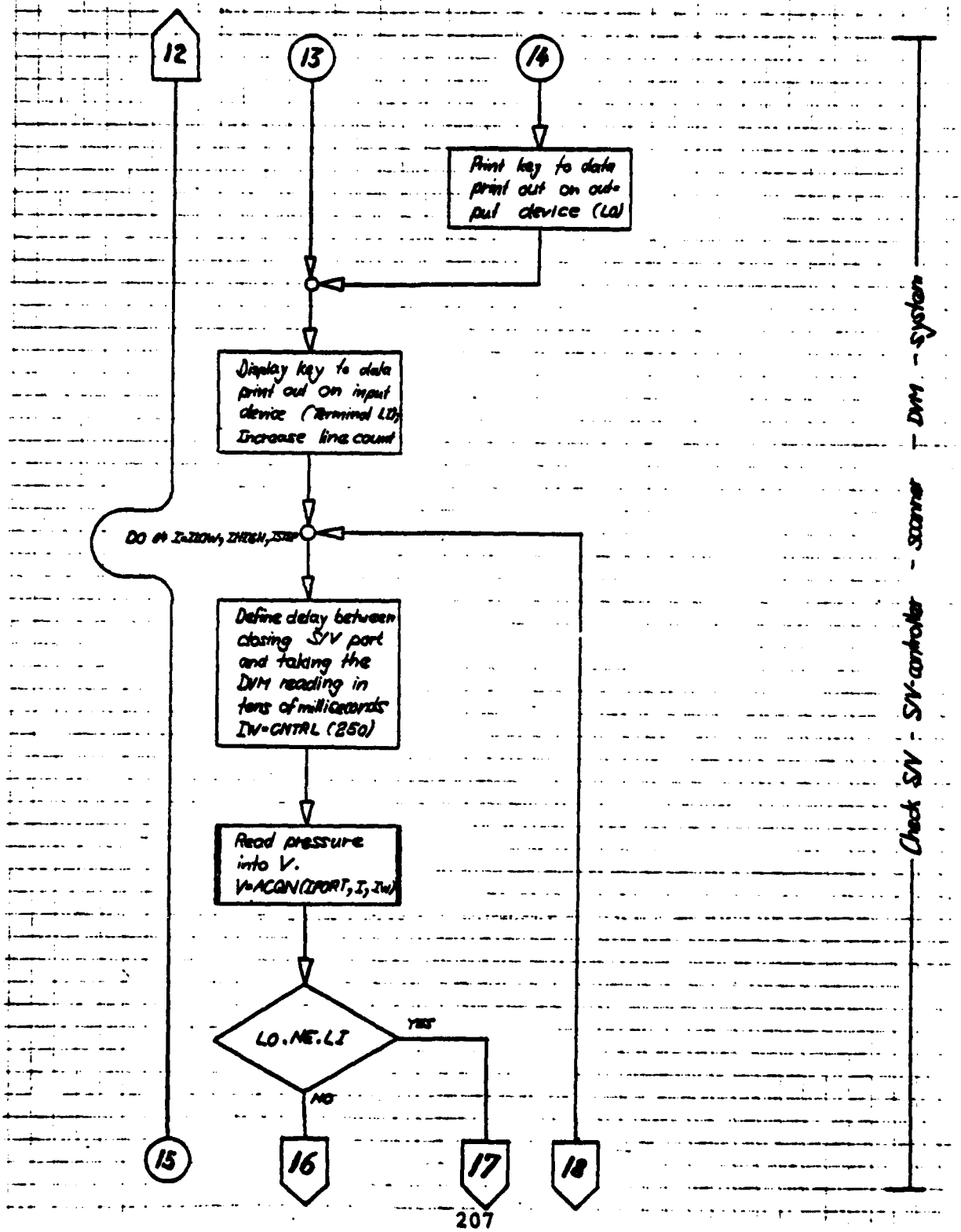

31
20
30
32
07
Ask operator, whether to take another systems check.
Read decision variable IOUM
YES
IOUM = 2HYE
RETURN

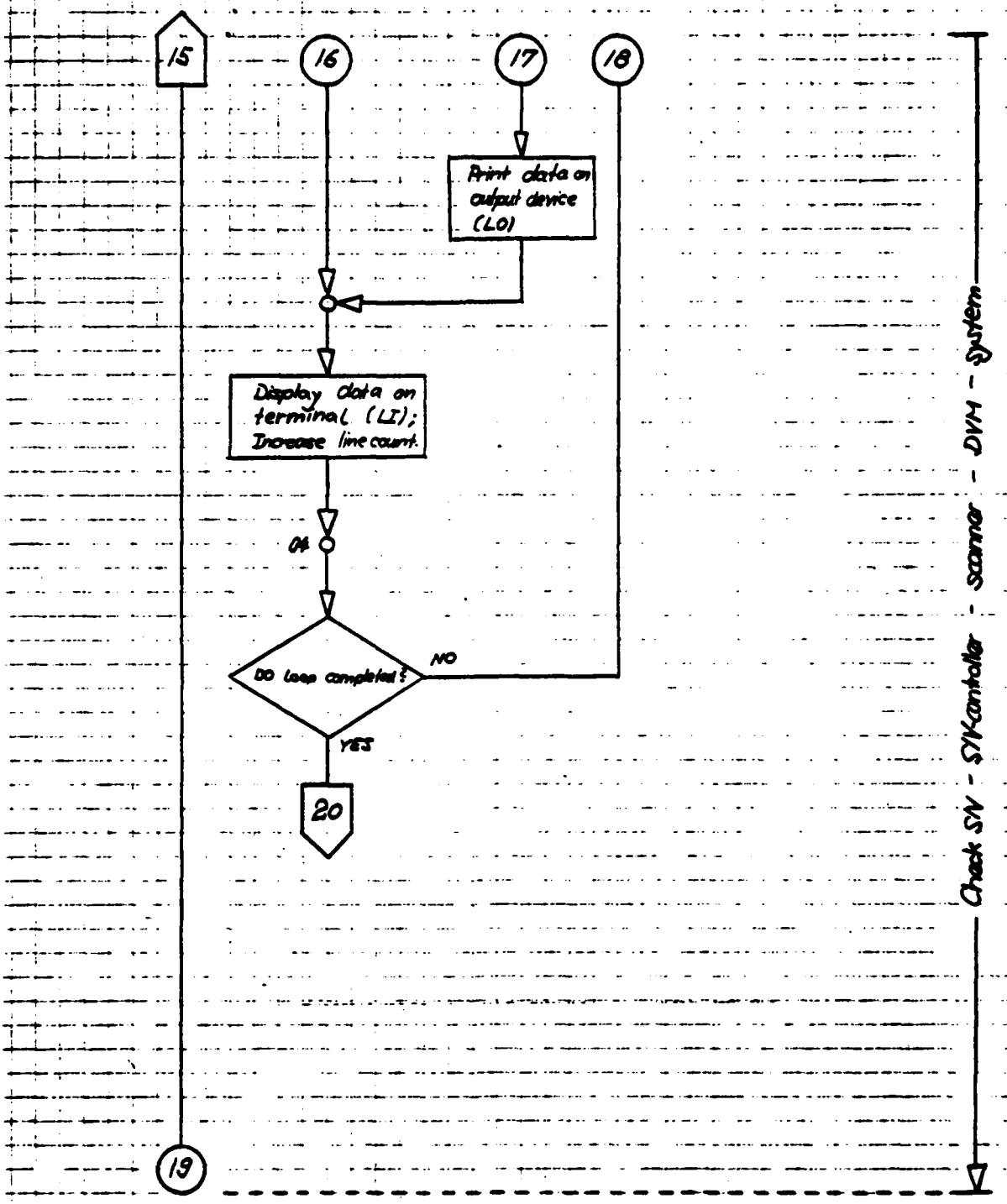

ENTRY
I/O assignment
LI = CNTRL (19)
LO.NE.LI ?
No
YES
Print heading on output device.
Reset line count
LINES = 0
Ask operator, what system shall be checked.
Read decision variable IOUM
1
2
3


1
2
3
IDUM < 1
or
IDUM > 3
YES
IDUM = 1
... 2
... 3
a2
7
8
Ask operator, what S/V shall be checked; and enter low and high port
Read IPORT, ILow & IUPEN; Define ISTEP = 1
IPORT < 1
or
IPORT > 6
YES
NO
4
5
6
Check S/V - S/V-controller - scanner - D/A-system


4
5
6
ILOW <1
or
ILOW >48
YES
NO
IHIGH <1
or
IHIGH > 48
YES
NO
IPORT.NE.2
YES
NO
S/V #2 is a 24-port S/N and counts the ports by even numbers. If ILOW or IHIGH are odd, they will be changed to the next even value; Redefine ISTEP=2
9
10
11
Check S/V - S/V-controller - scanner - DVM - system


9
10
11
Is ILOW an odd number?
YES
NO
Increase ILOW to the next even value
ILOW =
=(ILOW/2)*2+2
Is IHIGH an odd number?
YES
NO
Increase IHIGH to the next even value
IHIGH =
=(IHIGH/2)*2+2
OS
LO.NE.LI?
YES
NO
12
13
14
Check S/V - S/V-controller - scanner - DVH - System


12
13
14
Print key to data print out on output device (LO)
Display key to data print out on input device (Terminal LU); Increase line count
DO #4 I=1,ICOUNT, IINCR, ISTEP
Define delay between closing SV port and taking the DVM reading in tens of milliseconds IW=CNTRL (250)
Read pressure into V. V=ACQN(IPORT, I, IW)
LO.NE.LI
YES
NO
15
16
17
18
Check SV - SV-exhaler - scanner - DVM - system


15
16
17
18
Print data on output device (LO)
Display data on terminal (LI); Increase line count
DO loop completed?
NO
YES
20
19
Check SV - SVcontroller - scanner - DVH - system

6.3. SUBROUTINE CHNGE:

PURPOSE: Change any element of the program control array CNTRL on line and display any element of CNTRL.

ARGUMENTS: None

EXTERNALS: CODE, REWRF

COMMON BLOCK: CONTR; for detailed explanation refer to the TXCO3 description.

MNEMONIC ABBREVIATIONS:

C ... Change CNTRL (i) to new value

D ... Display current value of CNTRL (i)

R ... Return to the calling program

Note: C is followed by the value of ≪i≫ and the new value for ≪CNTRL (i)≫ and D is followed by the value of ≪i≫.

ERROR MESSAGES: None

PROCEDURE: For more detailed information, study the flow chart. After having assigned the I/O reference and reset the line count, the operator is asked what to do. The input in the first inverse video box specifies where to branch to.

i) Change CNTRL (i) to new value Input: C

The control character C is followed by the value of ≪i≫ and the new value for ≪CNTRL (i)≫. Each of the latter data items has to be aligned to the right margin of the two

following inverse video boxes. If the input is to be compiled as ASCII code, the identifier 2H has to precede the two input characters.

EXAMPLE A) Suppose, the time delay IW between closing a S/V port and taking the DVM reading shall be changed to 1 second, which is 100*10 milliseconds. Enter

[C] [250] [100]

and press the RETURN key. The subroutine responds displaying the message CNTRL (250) changed from 80 to 100. Where 80 is the previous value of CNTRL (250).

EXAMPLE B) The character, used to display the just acquired wave form in subroutine PACER shall be changed to the asterisk (= 2H*␣). Enter

[C] [249] [2H*␣]

blank, because subroutine PICTR outputs this item in A1-Format

and press the [RETURN] key. The subroutine responds displaying

CNTRL (249) changed from 2H+␣ to 2H*␣ .

where the add sign (= 2H+␣) was previously used for the plot.

ii) Display actual value of CNTRL (i) [Input: D]

The control character C is followed by the value of ≪i≫ and, only if the element CNTRL (i) shall be displayed in ASCII-mode, the string ≪2H≫ . The data for ≪i≫ has to be aligned to right margin of the second inverse video box and ≪2H≫ has to be centered in the third box.

EXAMPLE A) Display the value for the cartridge reference number, where the raw data files are located. Appendix A.3 (Program Control Array) reveals that you have to look into CNTRL(30). Enter

[D] [30] [ ]

and press the RETURN key. The subroutine responds by informing you that

The actual value of CNTRL (30) is 26.

EXAMPLE B) Display the first two characters of the name of the raw data file, which are written into CNTRL (32). Since the file name is ASCII coded, the ASCII-identifier ≪2H≫ must not be forgotten. Enter

[D] [32] [2H]

and press the [RETURN] key. The subroutine responds by informing you, that

The actual value of CNTRL (32) is 2HT5.

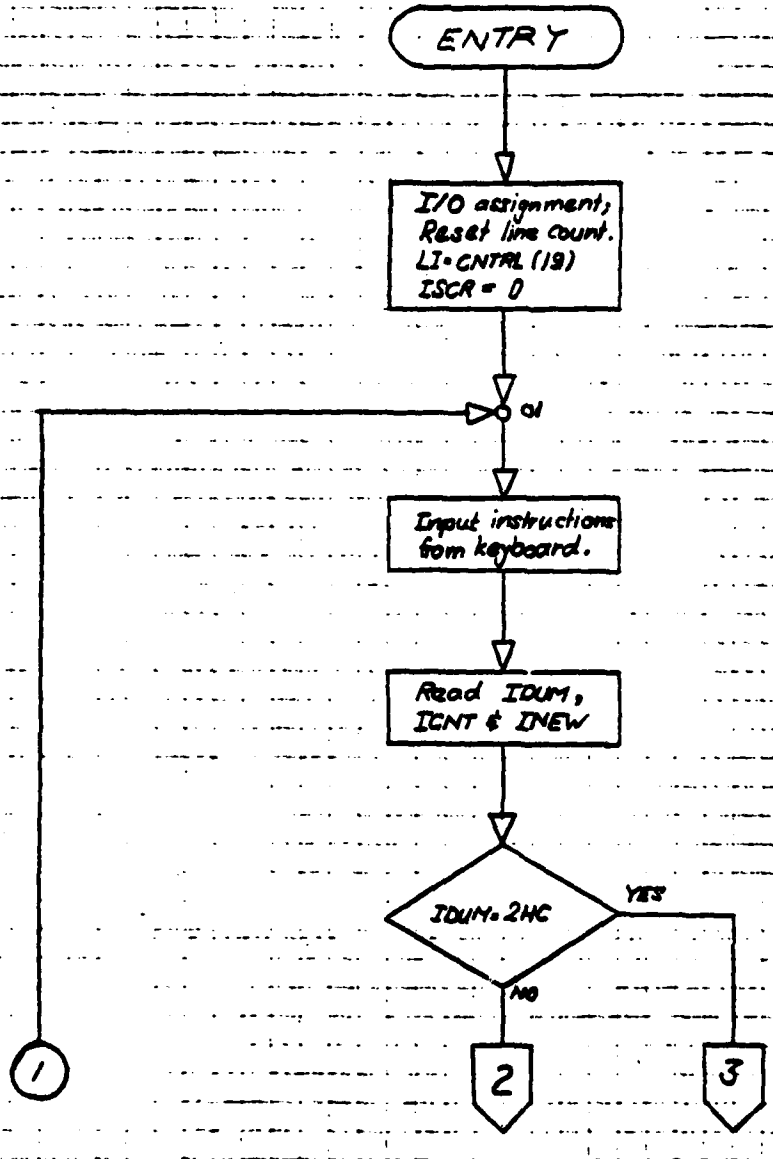
This information reveals that the last data acquisition was a combination probe survey, since there all data file names start with ≪T5≫ .

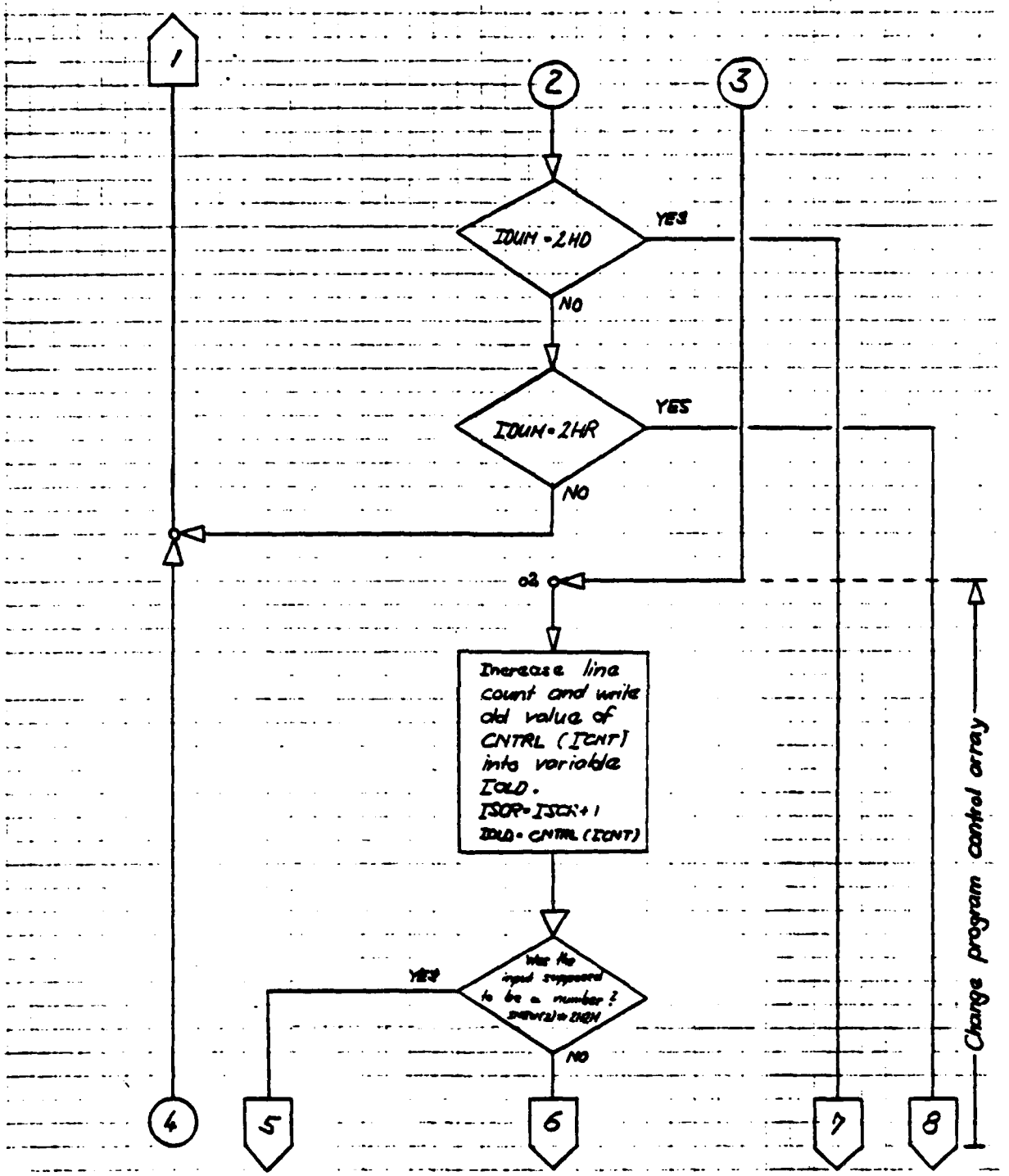
iii) Return [Input: R]

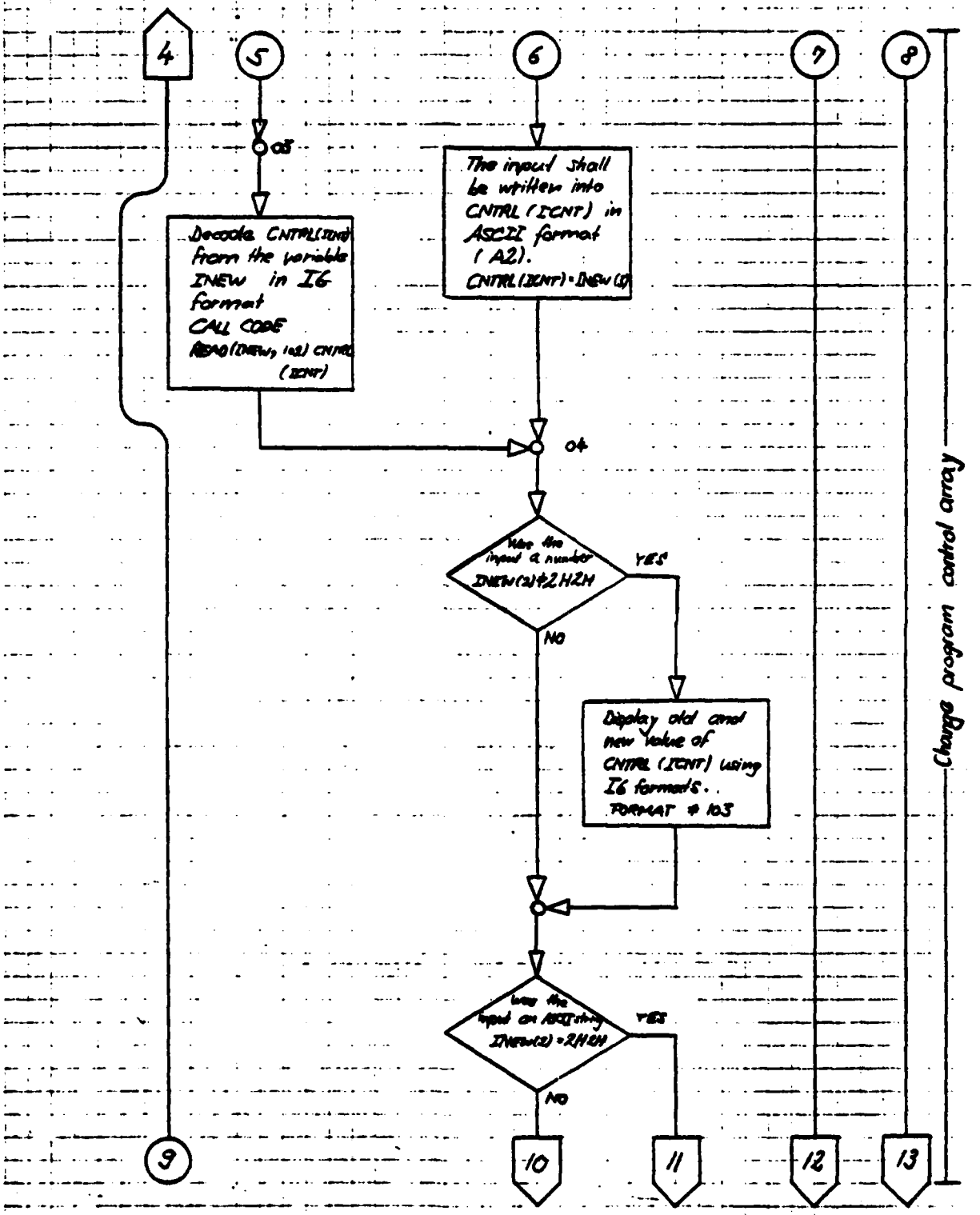
Enter R; Now the subroutine asks, whether to clear the informative responses, displayed by this subroutine previously. Inputting anything else but NO initializes the program to clear the screen.

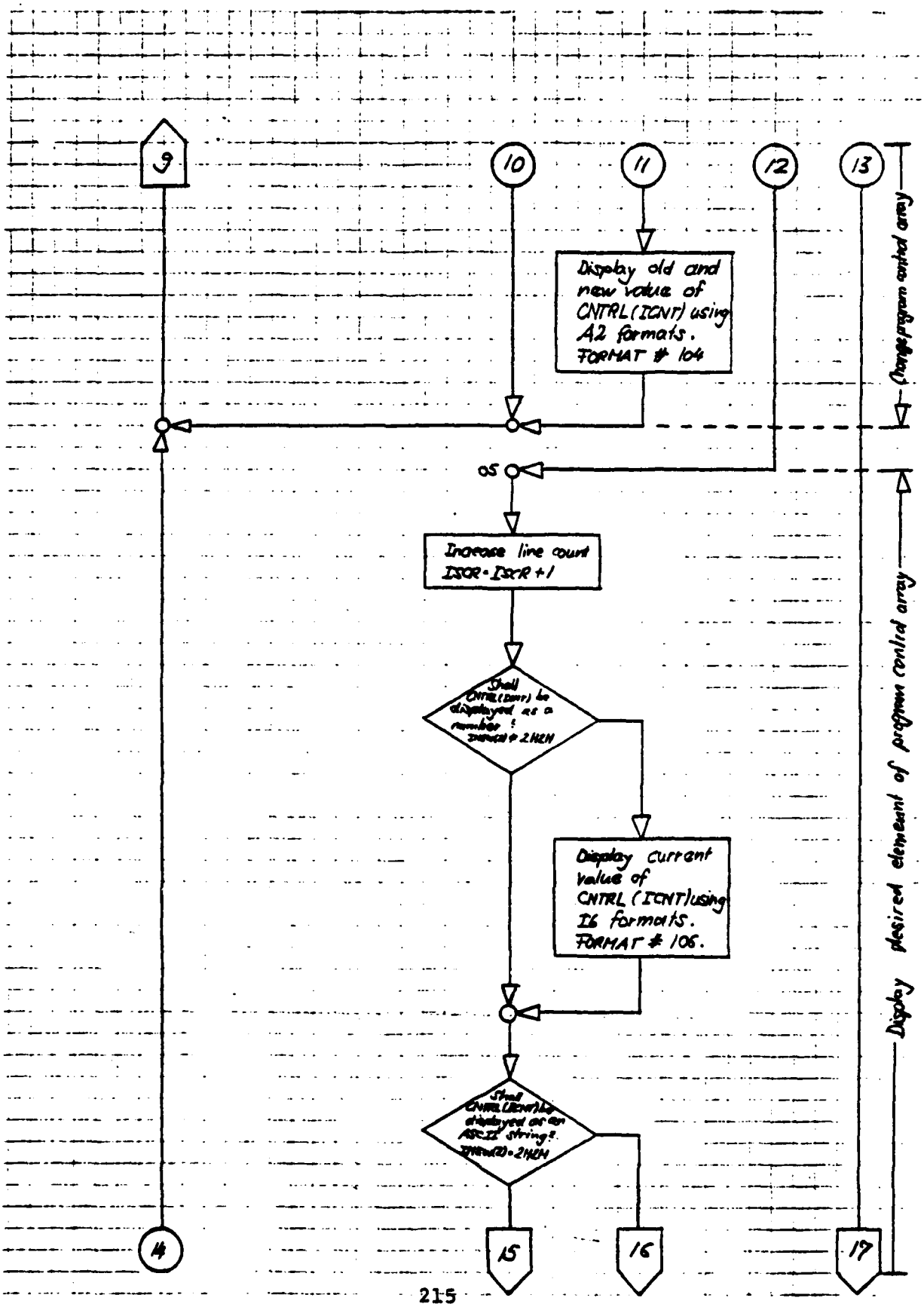
Then the subroutine terminates writing the modified control array back into disc file.

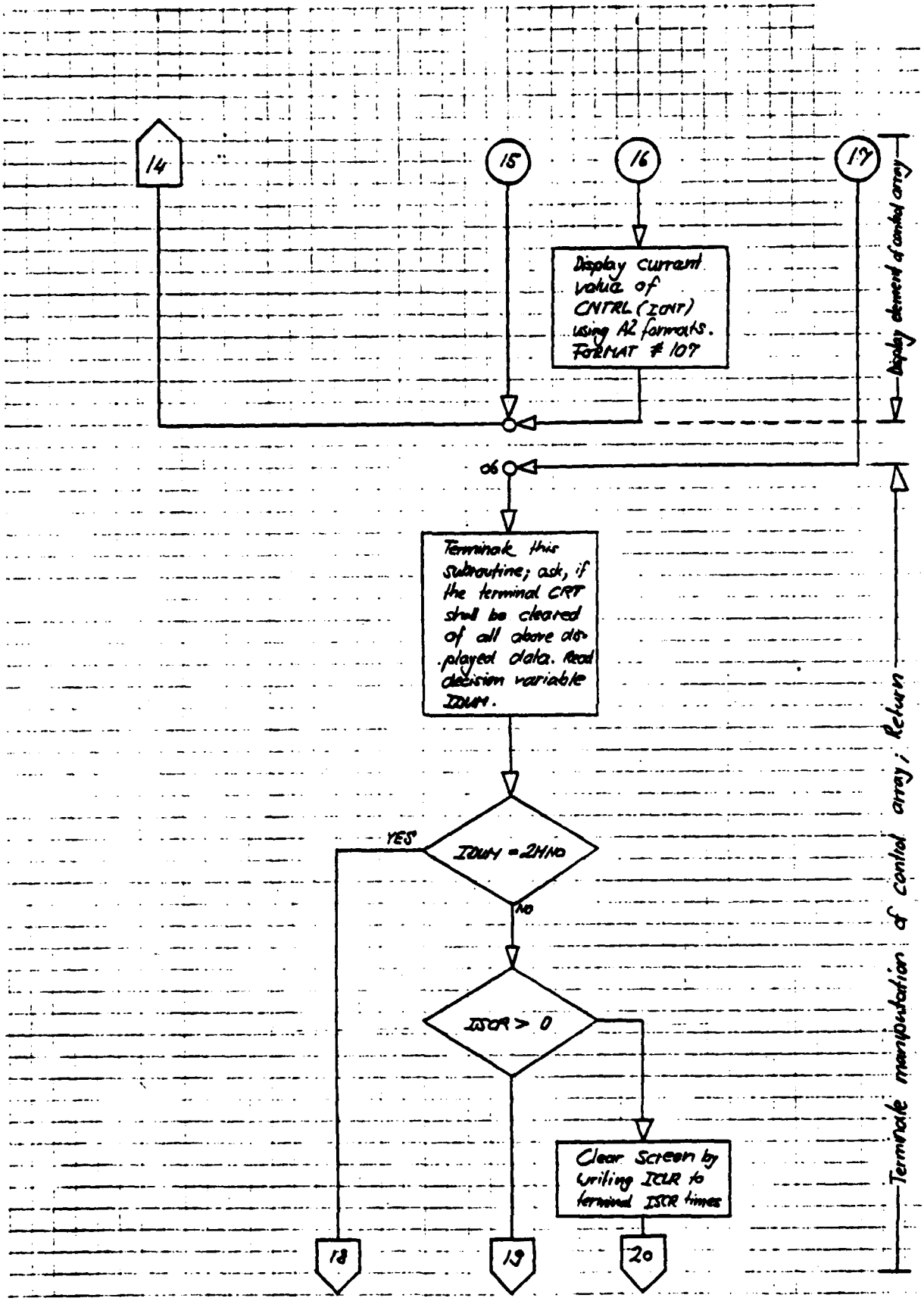
FLOW CHART SUBROUTINE CHNGE

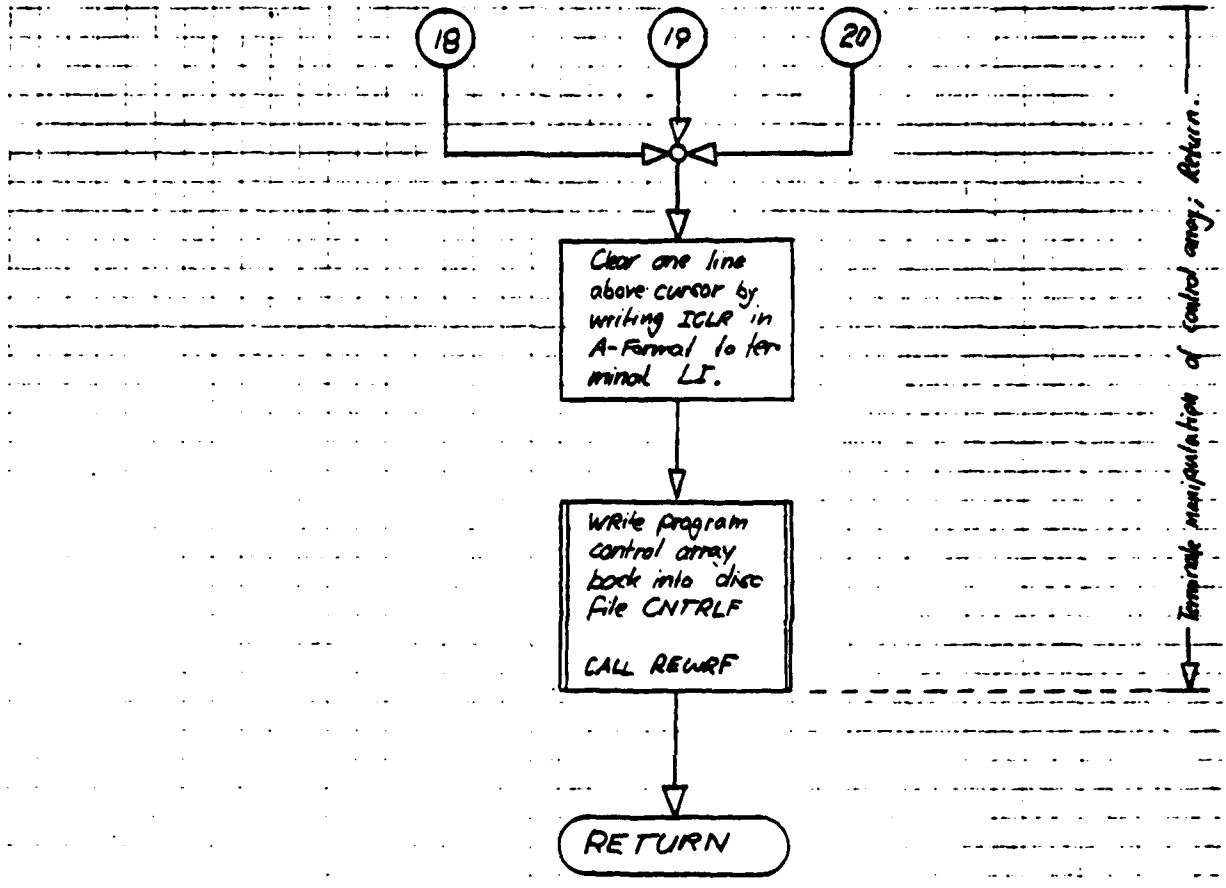



1
2
3
IOUM = 2HD
YES
NO
IOUM = 2HR
YES
NO
o2
Increase line count and write old value of CNTRL (ICNT) into variable IOLD.
ISOP = ISOP + 1
IOLD = CNTRL (ICNT)
Was the input supposed to be a number? (return) = 2HN
YES
NO
4
5
6
7
8
Change program control area


4
5
6
7
8
05
Decode CNTRL(ICNT) from the variable INEW in I6 format
CALL CODE
READ(INEW, 102) CNTRL (ICNT)
The input shall be written into CNTRL (ICNT) in ASCII format (A2).
CNTRL(ICNT) = INEW(1)
04
Was the input a number INEW(2)=2H2H
YES
NO
Display old and new value of CNTRL (ICNT) using I6 formats.
FORMAT # 103
Was the input an ASCII string INEW(2) = 2H1H
YES
NO
9
10
11
12
13
Change program control array


9
10
11
12
13
Display old and new value of CNTRL(ICNT) using A2 formats. FORMAT # 104
Change program control area
05
Increase line count ISCR = ISCR + 1
Shall CNTRL(ICNT) be displayed as a number ? IWHICH = 2HNM
Display current value of CNTRL(ICNT) using I6 formats. FORMAT # 106.
Shall CNTRL(ICNT) be displayed as an ASCII string? IWHICH = 2HCH
Display desired element of program control array
4
15
16
17


14
15
16
17
Display current value of CNTRL(IONT) using A2 formats. FORMAT # 107
Display element of control array
06
Terminate this subroutine; ask, if the terminal CRT shall be cleared of all above displayed data. Read decision variable IOWH.
IOWH = 2HNO
YES
NO
ISCR > 0
Clear Screen by writing ICLR to terminal ISCR times
18
19
20
Terminate manipulation of control array; Return


18
19
20
Clear one line above cursor by writing ICLR in A-Format to terminal LI.
WRite program control array back into disc file CNTRLF
CALL REWRF
RETURN
Terminate manipulation of control array; Return.

6.4. PROGRAM LISTING TXCO3

```
PAGE 0001  FTN.   4:12 PM  TUE., 23  SEP., 1980

0001  FTN4,L
0002        BLOCK DATA
0003       *  / FMP    / IDCB(144),IFILE(3),ISIZE(2),ISECU,ICR
0004        COMMON / FMP    / IDCB,IFILE,ISIZE,ISECU,ICR
0005        INTEGER IDCB(144),IFILE(3),ISIZE(2)
0006        END

 FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)
 ** NO WARNINGS **  NO ERRORS **
BLOCK COMMON FMP     SIZE =  00151
```

```
0007          BLOCK DATA
0008         *, / CIBUF / IFUF(1664)
0009          COMMON / CIBUF / IBUF
0010          INTEGER IBUF(1664)
0011          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS **

BLOCK COMMON CIBUF SIZE = 01664

```
0012          BLOCK DATA
0013         *, / CONTR / CNTRL(256)
0014          COMMON / CONTR / CNTRL
0015          INTEGER CNTRL(256)
0016          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS **

BLOCK COMMON CONTR SIZE = 00256

```
0017          PROGRAM TXCO3 (3,99)
0018    C     .....................................................................
0019    C     .                                                                   .
0020    C     .  The operating system RTE IV B requests the data acquisition      .
0021    C     .  program TXCO for the one stage transsonic compressor to be       .
0022    C     .  split into several programs scheduled by the father program      .
0023    C     .  TXCO.  This son program TXCO3 consists of the subroutines        .
0024    C     .  CHECK and CHNGE.  These codes handle the equipment check and     .
0025    C     .  the on line modification of the control array.  The data         .
0026    C     .  transfer between father and son programs take place via the      .
0027    C     .  control array CNTRL (disc file CNTRLF) and the data array        .
0028    C     .  IBUF (disc file IBUFF).                                          .
0029    C     .                                                                   .
0030    C     .  The utility subroutines ACQN, CNTL, CURVE, ICON, IPORT,          .
0031    C     .  PICTR, REWRF, RPACE, SCANR, TIME and WAIT are added.             .
0032    C     .                                                                   .
0033    C     .  Author:    Hans M. Zebner                                        .
0034    C     .  Date  :    March 12, 1980                                        .
0035    C     .                                                                   .
0036    C     .  A detailed program description is available in the TXCO log.     .
0037    C     .                                                                   .
0038    C     .....................................................................
0039          *, third son program of father program TXCO.

0040          COMMON / CONTR / CNTRL
0042          INTEGER CNTRL(256)
0044
0045
0046          DATA NOLF /0065378/
0047
0048     101  FORMAT (9X""20X""A2)
0049     102  FORMAT ("  IXCO3 : PROGRAM ABORTED!  NO SUBROUTINE HAS BE
0050         *EN INITIALIZED.")
0051     801  FORMAT ("CA")
0052    1001  FORMAT ("F1R7M3A1H0T3")
0053    1201  FORMAT ("PF4G6T")
0054    1501  FORMAT ("CA")
0055
0056          CALL REWRF (-1,2)
0057          LI = CNTRL(19)
0058          IF ( CNTRL(50)  .LT.  7   .OR.   CNTRL(50)  .GT.  8 ) GO TO 03
0059
0060    C     .....................................................................
0061    C     .                                                                   .
0062    C     .  Set interface bus and devices to remote control.                 .
0063    C     .                                                                   .
0064    C     .....................................................................
0065          CALL ABRT (7,2)
0066          CALL RMOTE (7,0)
0067          CALL RMOTE (10)
0068          CALL RMOTE (12)
0069          CALL RMOTE (15)
0070          WRITE ( 0, 801)
0071          WRITE (10,1001)
0072          WRITE (12,1201)
0073          WRITE (15,1501)
0074
0075    C     .....................................................................
0076    C     .                                                                   .
0077    C     .  Call subroutine indicated by CNTRL(50).                          .
0078    C     .                                                                   .
0079    C     .....................................................................
0080          ISTOP = CNTRL(50) - 6
0081          IF ( CNTRL(50)  .EQ.  7 ) CALL CHECK(CNTRL(51))
0082          IF ( CNTRL(50)  .EQ.  8 ) CALL CHNGE
0083
0084    C     .....................................................................
0085    C     .                                                                   .
0086    C     .  Release interface bus and devices from remote control.           .
0087    C     .                                                                   .
0088    C     .....................................................................
0089          CALL CLEAR (7,1)
0090          CALL LOCL (7)
0091
```

```
0092          CALL REWRF (1,2)
0093          WRITE (LI, 101) NOLF
0094          GO TO (01,02) ISTOP
0095       01 STOP 0777
0096       02 STOP 1077
0097       03 WRITE (LI, 102)
0098          END
```

FTN4 COMPILER: HP92060-16892 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00220 COMMON = 00000

```
0099          SUBROUTINE CHECK (LO)
0100     C    .....................................................................
0101     C    .                                                                   .
0102     C    .  Instrumentation check.                                           .
0103     C    .  Author:    Hans M. Zebner                                        .
0104     C    .  Date  :    January 11, 1980                                      .
0105     C    .  Detailed program description is available in TXCO log;  The      .
0106     C    .  Comment statements match to the flow chart explanations.         .
0107     C    .                                                                   .
0108     C    .....................................................................
0109          *, Checks the instrumentation.
0110
0111          COMMON / CONTR / CNTRL
0112
0113          INTEGER CNTRL(256),NOLF,NOCR(2),ICLR(3)
0114          INTEGER ADCHNL,PAMO,PAIR,START,INCR,STOP,REP
0115
0116     C    FORMATS CHECK START
0117       100 FORMAT ("1SUBROUTINE CHECK; CONTROL OUT
0118          *  ")
0119       101 FORMAT (/"Enter instrumentation code"//
0120          *"1  ...   S/V"/
0121          *"2  ...   Scanner"/
0122          *"3  ...   Pacer"//" "A2)
0123       102 FORMAT (/"Enter S/V #, LO, HI"21X,2A2)
0124       103 FORMAT (/" Scannivalve #"I3/" Port"5X"Voltage")
0125       104 FORMAT (I4,F13.6"")
0126       105 FORMAT (/"Enter scanner #, LO, HI"17X,2A2)
0127       106 FORMAT (/5X"Scanner #"I3/" Chan"8X"Data")
0128       109 FORMAT (" Enter"//"A/D channel Pacer mode ",
0129          *"Blade pair Start count Increment ",
0130          *"Stop count # repetitions"A2//
0131          *7(" "),A2)
0132       110 FORMAT (7(I3,1X))
0133       111 FORMAT ("A/D channel Pacer mode Blade pair
0134          *_Start count Increment Stop count",
0135          *" # repetitions"A2/I11,I11,I11,I11,I12,I10,I11,I14,A2)
0136       107 FORMAT ("Enter YES to repeat"21X,2A2)
0137       108 FORMAT ("

Check completed")
```

```
0138     149 FORMAT ((3A2))
0139    1001 FORMAT ("F1R7M3A1H1T3")
0140    C    FORMATS CHECK STOP .
0141         DATA NOLF  /006537B/
0142         DATA NOCR  /000033B,040433B/
0143         DATA ICLR  /015524B,015515B,006537B/
0144
0145         LI = CNTRL(19)
0146         IF ( LO  .NE.  LI ) WRITE (LO,100)
0147      01 LINES = 0
0148    C    ..................................................................
0149    CCCC :                                                                :
0150    CCCC :  Ask operator what the instrument code is.                     :
0151    CCCC :                                                                :
0152    C    ..................................................................
0153         WRITE (LI, 101) NOLF
0154         READ  (LI,   *) IDUM
0155         WRITE (LI, 149) (ICLR,I=1,8)
0156         IF ( IDUM  .LT.  1  .OR.  IDUM  .GT.  3 ) GO TO 01
0157         GO TO (02,05,20) IDUM
0158    C    ..................................................................
0159    CCCC :                                                                :
0160    CCCC :  Check S/V;  Input S/V #, low port, high port.                 :
0161    CCCC :                                                                :
0162    C    ..................................................................
0163      02 WRITE (LI, 102) NOCR
0164         READ  (LI,   *) IPORT,ILOW,IHIGH
0165         WRITE (LI, 149) (ICLR,I=1,2)
0166         ISTEP = 1
0167         IF ( IPORT  .LT.  01     .OR.  IPORT  .GT.  05 ) GO TO 02
0168         IF ( ILOW   .LT.  01     .OR.  ILOW   .GT.  48 ) GO TO 02
0169         IF ( IHIGH  .LT.  ILOW   .OR.  IHIGH  .GT.  48 ) GO TO 02
0170         IF ( IPORT  .NE.  2 ) GO TO 03
0171         ISTEP = 2
0172         IF ( (ILOW /2)*2  .NE.  ILOW  ) ILOW =(ILOW /2)*2+2
0173         IF ( (IHIGH/2)*2  .NE.  IHIGH ) IHIGH=(IHIGH/2)*2+2
```

```
0174        03 IF ( LO  .NE.  LI ) WRITE (LO,103) IPORT
0175           WRITE (LI, 103) IPORT
0176           LINES = LINES+3
0177     C     .....................................................................
0178     C     :                                                                   :
0179     CCCC  :  Read & output voltages.                                          :
0180     C     :                                                                   :
0181     C     .....................................................................
0182           WRITE (10,1001)
0183           DO 04 I=ILOW,IHIGH,ISTEP
0184           IW = CNTRL(250)
0185           V = ACQN(IPORT,I,IW)
0186           IF ( LO  .NE.  LI  ) WRITE (LO, 104) I,V
0187           WRITE (LI, 104) I,V
0188        04 LINES = LINES+1
0189           GO TO 07
0190     C     .....................................................................
0191     C     :                                                                   :
0192     CCCC  :  Check scanner;  Input scanner #, low channel, high channel.      :
0193     C     :                                                                   :
0194     C     .....................................................................
0195        05 WRITE (LI, 105) NOCR
0196           READ  (LI,   *) ISCR,ILOW,IHIGH
0197           WRITE (LI, 149) (ICLR,I=1,2)
0198           IF ( ISCR  .EQ.  1 ) LS =  8
0199           IF ( ISCR  .EQ.  2 ) LS = 15
0200           IF ( ISCR  .LT.  1  .OR.  ISCR  .GT.  2 ) GO TO 05
0201           IF ( LO  .NE.  LI ) WRITE (LO, 106) ISCR
0202           WRITE (LI, 106) ISCR
0203           LINES = LINES+3
0204           ILOW = ILOW+1
0205           IHIGH = IHIGH+1
0206     C     .....................................................................
0207     C     :                                                                   :
0208     CCCC  :  Read & output voltages.                                          :
0209     C     :                                                                   :
0210     C     .....................................................................
0211           WRITE (10,1001)
0212           DO 06 I=ILOW,IHIGH
0213           I1 = I-1
0214           D = SCANR(LS,I1,1)
0215           IF ( LO  .NE.  LI ) WRITE (LO, 104) I1,D
0216           WRITE (LI, 104) I1,D
0217        06 LINES = LINES+1
0218           GO TO 07
0219
0220     C     .....................................................................
0221     C     :                                                                   :
0222     CCCC  :  Check pacer;  Input pacer control parameters.                    :
0223     C     :                                                                   :
0224     C     .....................................................................
0225        20 WRITE (LI, 109) (NOLF,I=1,2)
0226           READ  (LI, 110) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP
0227           WRITE (LI, 149) (ICLR,I=1,5)
0228           WRITE (LI, 111) NOLF,ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,NOLF
0229           LINES = LINES+2
0230           CALL RPACE (ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AVRGE,1,LO)
0231           LINES = LINES+3
0232
0233     C     .....................................................................
0234     C     :                                                                   :
0235     CCCC  :  Ask whether to perform more test samples.                        :
0236     C     :                                                                   :
0237     C     .....................................................................
0238        07 WRITE (LI, 107) NOCR
0239           READ  (LI, 149) IDUM
0240           LINES = LINES+1
0241           WRITE (LI, 149) (ICLR,I=1,LINES)
0242           IF ( IDUM  .EQ.  2HYE ) GO TO 01
0243           IF ( LO  .EQ.  6 ) WRITE (LO, 108)
0244           RETURN
0245           END
```

```
PAGE 0008  CHECK  4:12 PM  TUE., 23  SEP., 1980
 FTN4 COMPILER: HP92060-16092 REV. 1926 (790438)
 **  NO WARNINGS **  NO ERRORS **   PROGRAM = 01030      COMMON = 00000
```

```
0246          SUBROUTINE CHNGE
0247    C     .....................................................................
0248    CCCCC .                                                                   .
0249    C     .  Modify control array CNTRL(256) interactively.                   .
0250    C     .  Author:     Hans M. Zebner                                       .
0251    C     .  Date   :    January 11, 1980                                     .
0252    C     .  Detailed program description is available in TXCO log;  The      .
0253    C     .  Comment statements match to the flow chart explanations.         .
0254    CCCCC .                                                                   .
0255    C     .....................................................................
0256          *, Enables user to change control array on-line.

0257          COMMON / CONTR / CNTRL
0258          INTEGER CNTRL(256)
0259          INTEGER INEW(3),NOLF,NOCR(2),ICLR(3)
0260
0261    C     FORMATS CHNGE START
0262      100 FORMAT (/" Enter"//
0263         *" C, I and new value of CNTRL(I)  ...   to
0264         * change the desired element I"/
0265         *" D, I"28X"...    to display the desired element I"
0266         */" R   "28X"...    to return to the calling program"//
0267         *"                "A2)
0268      101 FORMAT (A1,1X,I3,1X,3A2)
0269      102 FORMAT (I6)
0270      103 FORMAT ("CNTRL("I3") changed from "I6" to "I6" .")
0271      104 FORMAT ("CNTRL("I3") changed from   2H"A2" to   2H"A2" .")
0272      105 FORMAT (" Clear screen? YES or NO  "2A2)
0273      106 FORMAT ("The actual value of CNTRL("I3") is"I7" .")
0274      107 FORMAT ("The actual value of CNTRL("I3") is   2H"A2" .")
0275      149 FORMAT ((3A2))
0276    C     FORMATS CHNGE STOP
0277
0278          DATA NOLF   /006537B/
0279          DATA NOCR   /000033B,040433B/
0280          DATA ICLR   /015524B,015515B,006537B/
0281          LI    = CNTRL(19)
0282          ISCR  = 0
0283
0284
0285
0286    C     .....................................................................
0287    CCCCC .                                                                   .
0288    C     .  Input instructions from keyboard.                                .
0289    CCCCC .                                                                   .
0290    C     .....................................................................
0291       01 WRITE (LI, 100) NOLF
0292          READ  (LI, 101) IDUM,ICNT,INEW
0293          WRITE (LI, 149) (ICLR,I=1,8)
0294          IF ( IDUM  .EQ.  2HC  ) GO TO 02
0295          IF ( IDUM  .EQ.  2HD  ) GO TO 05
0296          IF ( IDUM  .EQ.  2HR  ) GO TO 06
0297          GO TO 01
0298
0299
0300
0301    C     .....................................................................
0302    CCCCC .                                                                   .
0303    C     .  Change desired element of control array CNTRL.                   .
0304    CCCCC .                                                                   .
0305    C     .....................................................................
0306       02 ISCR = ISCR+1
0307          IOLD  = CNTRL(ICNT)
0308          IF ( INEW(2)  .NE.  2H2H ) GO TO 03
0309          CNTRL(ICNT) = INEW(3)
0310          GO TO 04
0311       03 CALL CODE
0312          READ (INEW, 102) CNTRL(ICNT)
0313       04 IF ( INEW(2)  .NE.  2H2H ) WRITE (LI, 103) ICNT,IOLD,CNTRL(ICNT)
0314          IF ( INEW(2)  .EQ.  2H2H ) WRITE (LI, 104) ICNT, IOLD, CNTRL(ICNT)
0315
0316          GO TO 01
0317
0318
0319
0320    C     .....................................................................
```

```
PAGE 0010  CHNGE  4:12 PM  TUE., 23  SEP., 1980

0321  C
0322  C     :  Display desired element of control array CNTRL.                :
0323  C     :                                                                  :
0324  C     :..................................................................:
0325        05 ISCR = ISCR+1
0326           IF ( INEW(2)  .NE.  2H2H ) WRITE (LI, 106) ICNT,CNTRL(ICNT)
0327           IF ( INEW(2)  .EQ.  2H2H ) WRITE (LI, 107) ICNT, CNTRL(ICNT)
0328
0329           GO TO 01
0330
0331
0332
0333  C     :..................................................................:
0334  C     :                                                                  :
0335  C     :  Terminate modification of control array CNTRL;  write it        :
0336  C     :  back to the disc;  return to calling program.                   :
0337  C     :                                                                  :
0338  C     :..................................................................:
0339        06 WRITE (LI, 105) NOCR
0340           READ  (LI, 149) IDUM
0341           IF ( IDUM  .EQ.  2HNO    ) GO TO 07
0342           IF ( ISCR  .GT.  0 ) WRITE (LI, 149) (ICLR,I=1,ISCR)
0343        07 WRITE (LI, 149) ICLR
0344           CALL REWRF (1,2)
0345           RETURN
0346           END

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

**  NO WARNINGS **  NO ERRORS **   PROGRAM = 00581      COMMON = 00000
```

7. UTILITY SUBROUTINE PACKAGE TXCOU

7.1. Description

Subroutines and functions, which are commonly used by either TXCO2, TXCO2 or TXCO3, are contained in the utility package TXCOU (source code is saved in file &TXCOU; relocatable binary code is saved in file %TXCOU). Thus the length of the TXCO1, TXCO2 or TXCO3 source files can be kept to minimum, which allows editing and recompiling TXCO1, -2, -3 separately, which saves time. When loading TXCO1, -2, -3, the load of the utility subroutines has to be included using the multiple search loader command (MS,%TXCOU). Since the utility subroutines and functions are short and straightforward, the comment statements and program explanation included in the code serve to describe them. The present section briefly outlines the utility subroutine package.

Name	Purpose	Author
ACQN	Positions Scanivalve (S/V) and reads DVM	Geopfarth
CNTL	Closes scanner channels, which control the S/V controller, HG78K	Geopfarth
CURVE	Computes coefficients for a linear curve fit	McGuire
ICON	Converts a one or two digit integer into a two character ASCII string	Geopfarth

Name	Purpose	Author
IPORT	Interrogates S/V controller and returns the present port #	Geopfarth
PICTR	Uses the (24 x 80 dot) CRT of a terminal for a graphics display of data acquired with the PACER	Zebner
REWRF	Data transfer disc file to program array and vice versa	Zebner
RPACE	Triggers A/D through the PACER and calculates the average voltage	Zebner
SCANR	Closes scanner channel and reads the DVM or digital counter	Geopfarth
TIME	Obtains date and time in ASCII-format	Zebner
WAIT	Causes a defined time delay	Geopfarth (Original) Zebner (Modification)

REAL FUNCTION ACQN

Arguments: INTEGER: IVALVE, IADES, IW

IVALVE - - - Desired S/V #

IADES - - - Desired port # of S/V

IW - - - - - Time delay in tens of ms between closing transducer relay and taking the DVM reading

Example: The pressure on S/V #4, Port #18 is to be read with the time delay to be 0.5 sec (= 500 ms = 50 x 10 ms). The correct call is

```
IVALVE = 4
IPORT = 18
IW = 50
PRES = ACQN (IVALVE, IPORT, IW)
```

or

```
PRES = ACQN (4, 18, 50)
```

In both cases the DVM reading is written into the real variable PRES.

It is desirable to step forwards systematically and sequentially through the required parts of a S/V in order to reduce unnecessary wear. Whenever a S/V is scanned, the operator should watch the data system closely. In some cases (e.g. if the HP 9830 is brought on line) the HP-Interface bus and the devices may be downed. If this happens when the program ACQN has closed the scanner channels (on scanner #1) which either resets or advances the S/V the S/V relay will burn out. To prevent damage, the operator must turn off the power to scanner #1 immediately, then bring the data system up again using the UP-command (see HP manuals).

SUBROUTINE CNTL (ICHAN, IDEL, ISTEP, K)

Arguments: INTEGER: ICHAN, IDEL, ISTEP, K

ICHAN - - - Channel # of scanner #1 (LU# = 8)

IDEL - - - Number of repetitions to close the scanner channel

ISTEP - - - Increment to step from 1 through IDEL

K - - - - - Function code

K = 1 Close for 10 ms, wait for 150 ms; Repeat "IDEL" times; return

K = 2 Close for 10 ms, wait for 4 sec; return

K = 3 Close; return

An example is unnecessary since the only subroutine to use SUBROUTINE CNTL is the REAL FUNCTION ACQN, which is itself a utility. The user won't have to deal with CNTL.

SUBROUTINE CURVE (N, X, Y, SLOPE, SECON)

Arguments: INTEGER: N

REAL: X(N), Y(N), SLOPE, SECON

N - - - - Number of data points

X(N) - - - Abscissa of data points

Y(N) - - - Ordinals of data points

SLOPE - - Slope of linear curve fit

SECON - - Intercept of linear curve fit

Example: Suppose the following (N = 6) pairs of data points shall be approximated by a linear curve fit:

X(1) = 1.0 Y(1) = 105.6

X(2) = 2.5 Y(2) = 105.4

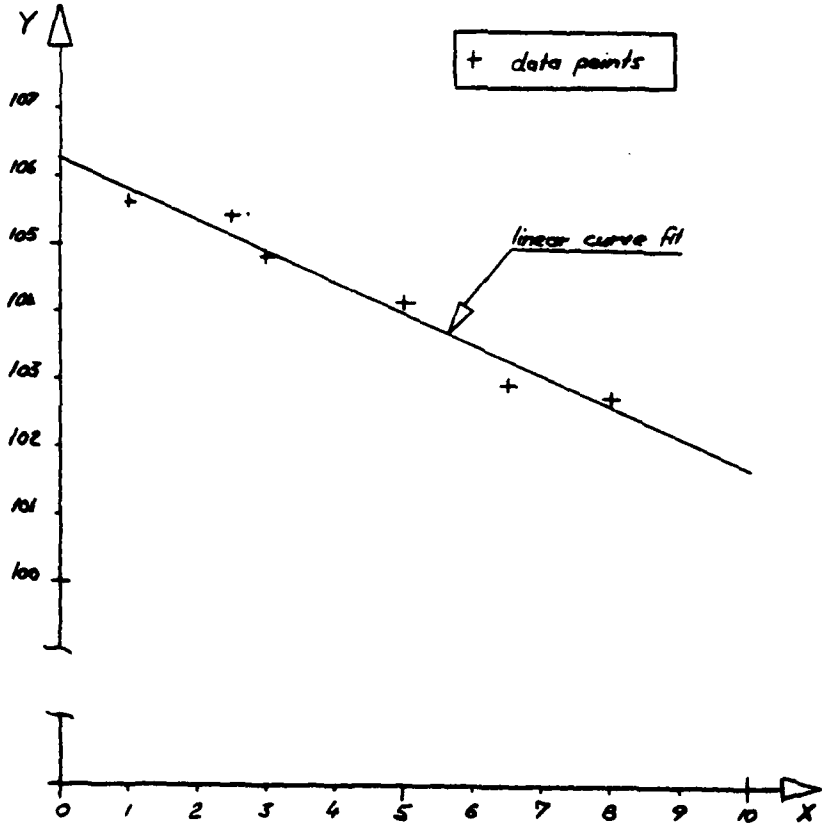
X(3) = 3.0 Y(3) = 104.8

X(4) = 5.0 Y(4) = 104.1

X(5) = 6.5 Y(5) = 102.9

X(6) = 8.0 Y(6) = 102.7

The situation is shown in the following sketch:



To obtain the slope and the intercept of the linear curve fit (which is derived using the least squares criterion), program

CALL CURVE (6, X, Y, SLOPE, SECON)

and the results will be returned from SUBROUTINE CURVE

SLOPE = -.461

SECON = 106.247

INTEGER FUNCTION ICON

Arguments: INTEGER: I, N

I, N - - - Two integer numbers to be added

ICON - - - The result of the addition (which should not exceed two digits), but ASCII converted.

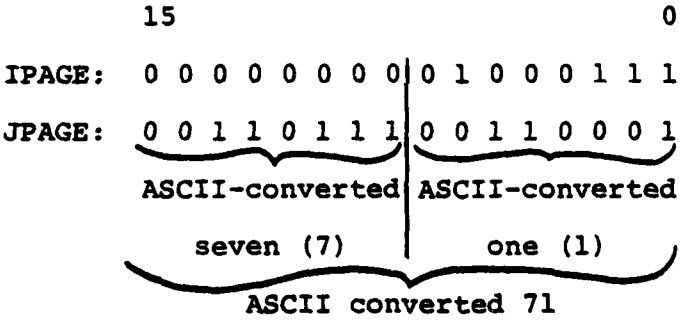
Example: Suppose the data documentation page IPAGE shall be converted to an ASCII-string named JPAGE. The correct call is:

JPAGE = ICON (IPAGE, Ø)

or

JPAGE = ICON (Ø, IPAGE)

The bit structure, if IPAGE is 71_{10}, is as follows:



INTEGER FUNCTION IPORT

Arguments: INTEGER: IVALVE

IVALVE - - - Desired S/V (1 through 5)

Example: The call

IWHERE = IPORT(5)

returns the present port # of S/V #5 and writes it into the variable IWHERE.

SUBROUTINE PICTR

Arguments: INTEGER: LO, NUMBER, NEWPG, ICL, ISIGN

REAL: AMPL

LO - - - - Defines terminal LU# (either 1 or 18)

NUMBR - - Identifier to appear in the "drawing"

NEWPG - - No significance! Will be altered by PICTR

ICL - - - No significance! Will be altered by PICTR

ISIGN - - Character to be used for the drawing

AMPL - - - Amplitude range of the raw data returned to the calling program

Subroutine PICTR is designed for the TXCO1-subroutine PACER, from which it gets the data through the COMMON block CIBUF. PICTR uses the enhanced display capabilities of the video terminal to produce a "drawing" which, of course, is of limited resolution. It allows the operator to verify the acquired raw paced run data qualitatively by checking the "drawing" against the oscilloscope display. See the listing of subroutine PACER, if an example is needed.

SUBROUTINE REWRF

Arguments: INTEGER: IREWRF, IWHATA

IREWRF - - Determines whether the array indicated by IWHATA shall be read from a disc file into an array or whether it shall be written into a disc file

IREWRF = -1 Read data from disc file into array

IREWRF = +1 Write data from array into disc file

IWHATA - - Specifies the type of data to be transferred

IWHATA = 1 Array IBUF ↔ disc file IBUFF

IWHATA = 2 Array CNTRL ↔ disc file CNTRLF

This subroutine relieves the individual TXCO1, -2, -3 subroutines from the routine task of data transfer between disc and program (CP).

Example: The four applications are:

i) Read the program control array from disc file CNTRLF::26 into array CNTRL: CALL REWRF (-1, 2)

ii) Write the program control array from array CNTRL into disc file CNTRLF::26: CALL REWRF (1, 2)

iii) Read the raw data buffer from disc file IBUFF::26 into array IBUF: CALL REWRF (-1, 1)

iv) Write the raw data buffer from array IBUF into disc file IBUFF::26: CALL REWRF (1, 1)

SUBROUTINE RPACE

Arguments: INTEGER: ADCHNL, PAMO, PAIR, START, INCR, STOP, REP, IPRINT, LO

REAL: AVRGE

ADCHNL - - A/D input channel to be selected (0...15)

PAMO - - - Pacer mode (1 or 2)

PAIR - - - Blade pair (1...9)

START - - - Start location (in counts) for data scan across the blade passage

INCR - - - Step size (in counts) to scan across the blade passage

STOP - - - Stop location (in counts) for data scan across the blade passage.

REP - - - - Number of repetitions at each location in the blade passage

IPRINT - - Flag to decide whether to output intermediate information

IPRINT = 1 Print all intermediate data and suppress pointer at the terminal

IPRINT = Ø Suppress printed output and initialize pointer at the terminal

LO - - - - LU# of device for printed output

AVRGE - - - Average voltage of paced run data

This subroutine is the control program to acquire data from the A/D converter in the synchronized PACER mode. (See also the description of subroutine PACER (Section 4.5), where the synchronized sampling is explained.) Although the (decoded) voltages from the A/D converter are REAL numbers between -1.0 volt and +1.0 volt, the data storage uses an integer array. Before writing the voltages into the data array, they are multiplied by 10,000. Note that if one of the PACER control parameters is out of the defined range, it is set to a default value without outputting any warning. The subroutine contains a large number of I/O statements

which were necessary during the development of the TXCO program system. In order to increase speed, the unnecessary statements should be removed. A further improvement would be a conversion from FORTRAN to ASSEMBLER programming language. An example of a call to subroutine RPACE is contained in the description and listing of subroutine PACER (Section 4.5).

REAL FUNCTION SCANR

Arguments: INTEGER: LU, ICHAN, K

LU - - - - Logical Unit # of the desired scanner (either 8 - scanner #1, or 15 - scanner #2)

ICHAN - - - Scanner channel (integer)

K - - - - - Instrument code

K = 1 Read the DVM

K = 2 Read the digital counter

Example A: To obtain the torque reading from the transonic compressor test rig, which is fed into scanner #1 (LU = 8), channel 36, program

```
LU = 8
ICHAN = 36
K = 1
TORQUE = SCANR (LU, ICHAN, K)
or
TORQUE = SCANR (8, 36, 1)
```

In both cases the DVM reading is written into the real variable TORQUE.

Example B: To obtain the RPM of the Allis-Chalmers (central air supply) compressor, program

```
LU = 8
ICHAN = 15
K = 2
RPMAC = SCANR (LU, ICHAN, K)
or
RPMAC = SCANR (8, 15, 2)
```

In both cases the reading of the digital counter is written into the real variable RPMAC.

Example	Instrument Code	Instrument Read
A	1	Digital Voltmeter
B	2	Digital Counter

SUBROUTINE TIME

Arguments: INTEGER: IMON, IDAY, IYEAR, IHOUR, IMIN, ISEC

IMON - - - ASCII converted month of the year

IYEAR - - - ASCII converted last two digits of current year

IDAY - - - ASCII converted day of the month

IHOUR - - - ASCII converted hour (24-hour clock) of the day

IMIN - - - ASCII converted minute of the hour

ISEC - - - ASCII converted second of the minute

IMON, IDAY and IYEAR are obtained from the program control array; IHOUR, IMIN and ISEC are obtained from the system

clock through an EXEC call; all variables are returned to the calling program.

Example: Suppose the system clock is set correctly and the control array CNTRL is defined, i.e., CNTRL was read from the disc file and adjusted to the actual conditions; then the following code

```
      .
      .
      .
603 FORMAT ("Date & Time: "A2"/"A2"/"A2,2X,
   *A2":"A2)
    CALL TIME (IMON,IDAY,IYEAR,IHOUR,IMIN,ISEC)
    WRITE (6,6Ø9) IMON,IDAY,IYEAR,IHOUR,IMIN
      .
      .
      .
```

produces the following output:

```
      .
      .
      .
Date & Time:  Ø9/27/8Ø  21:57
      .
      .
      .
```

SUBROUTINE WAIT

Arguments: INTEGER: TWAIT

TWAIT - - - Time delay in tens of milliseconds

Example: To cause a defined time delay of 5,7 sec (= 5700 ms = 570 x 10 ms), program

```
ITWAIT = 570
CALL WAIT (ITWAIT)
```

or

```
CALL WAIT (570)
```

7.2. PROGRAM LISTING TXCOU

```
 PAGE 0001  FTN.   9:57 PM  SAT., 27  SEP., 1980

0001   FTN4,L
0002         REAL FUNCTION ACQN (IVALVE,IADES,IW)
0003   C     .....................................................................
0004   CCCCC .                                                                   .
0005   CCCCC .  Position scannivalve IVALVE to port IADES and define ACQN        .
0006   CCCCC .  the DVM output voltage.  A time delay of (IWAIT*10) ms occurs    .
0007   CCCCC .  between port selection and voltage measurement.  The DVM is      .
0008   CCCCC .  triggered by issuing HP-IB subroutine TRIGR.                     .
0009   CCCCC .  Author:    Robert N. Geopfarth                                   .
0010   CCCCC .  Date:      January 31, 1979                                      .
0011   CCCCC .  A detailed program description is available in TXCO log.  The   .
0012   CCCCC .  variables are:                                                   .
0013   CCCCC .  IVALVE ...   Desired S/V.                                        .
0014   CCCCC .  IADES  ...   Desired S/V port #.                                 .
0015   CCCCC .  IAPR   ...   Present S/V port #.                                 .
0016   CCCCC .  ICHAN  ...   ASCII converted scanner channel.                   .
0017   CCCCC .  ACQN   ...   Transducer voltage as read from DVM.                .
0018   CCCCC .  IW     ...   Time delay factor.                                  .
0019   CCCCC .                                                                   .
0020   C     .....................................................................
0021         *, Positions scannivalve and reads DVM.   Utilities.
0022     801 FORMAT ("C")
0023     901 FORMAT (/" ****   ERROR DETECTED IN REAL FUNCTION ACQN;"/
0024        *          " ****        CHECK FOR BAD PARAMETER IN CALL!"/
0025        *          " ****   IVALVE ="I3"    IADES ="I3"     IW ="I4/)
0026
0027         ISTEP = 1
0028         IF ( IVALVE  .LT.  1   .OR.   IVALVE  .GT.   5 ) GO TO  06
0029         IF ( IADES   .LT.  1   .OR.   IADES   .GT.  48 ) GO TO  06
0030         IF ( IVALVE  .EQ.  2   .OR.   IVALVE  .EQ.   3 ) ISTEP = 2
0031
0032   C     .....................................................................
0033   CCCCC .                                                                   .
0034   CCCCC .  Compare present port # to desired port#.                         .
0035   CCCCC .                                                                   .
0036   C     .....................................................................
0037      01 IAPR = IPORT(IVALVE)
0038         IDEL = IADES-IAPR
0039         IF ( IDEL ) 02,03,04
0040
0041   C     .....................................................................
0042   CCCCC .                                                                   .
0043   CCCCC .  Desired port below present port;  reset S/V                      .
0044   CCCCC .                                                                   .
0045   C     .....................................................................
0046      02 ICHAN = ICON(IVALVE,4)
0047         K = 2
0048         GO TO 05
0049
0050   C     .....................................................................
0051   CCCCC .                                                                   .
0052   CCCCC .  Present port is present port;  close X-ducer relay & read        .
0053   CCCCC .                                                                   .
0054   C     .....................................................................
0055      03 ICHAN = ICON(IVALVE,9)
0056         K = 3
0057         GO TO 05
0058
0059   C     .....................................................................
0060   CCCCC .                                                                   .
0061   CCCCC .  Desired port is above present port;  advance S/V.                .
0062   CCCCC .                                                                   .
0063   C     .....................................................................
0064      04 ICHAN = ICON(IVALVE,-1)
0065         K = 1
0066
0067   C     .....................................................................
0068   CCCCC .                                                                   .
0069   CCCCC .  Control S/V.                                                     .
0070   CCCCC .                                                                   .
0071   C     .....................................................................
0072      05 CALL CNTL (ICHAN,IDEL,ISTEP,K)
0073         IF ( K  .NE.  3 ) GO TO 01
0074
0075   C     .....................................................................
0076   C     .
```

```
0077  C      . Pause and read transducer output voltage.                    .
0078  C      . .                                                           .
0079  C      .....................................................................
0080           CALL WAIT (IW)
0081           CALL TRIGR (10)
0082           READ  (10,   *) DUM
0083           CALL TRIGR (10)
0084           READ  (10,   *) ACQN
0085           WRITE ( 8, 801)
0086           RETURN
0087
0088  C      .....................................................................
0089  C      .                                                                   .
0090  C      .  Error encountered;  output error message;  return.               .
0091  C      .                                                                   .
0092  C      .....................................................................
0093       06 WRITE ( 6, 901) IVALVE,IADES,IW
0094           RETURN
0095           END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00237 COMMON = 00000

```
0096          SUBROUTINE CNTL(ICHAN,IDEL,ISTEP,K)
0097         *, Closes scanner channel.
0098    C
0099    CCCCC          THIS PROGRAM CLOSES SCANNER CHANNEL "ICHAN"
0100    CCCCC          "IDEL" TIMES IN STEPS OF "ISTEP" BASED UPON
0101    CCCCC          PROGRAM OPTIONS SPECIFIED BY "K". (ICHAN MUST
0102    CCCCC          BE AN ASCII-CONVERTED INTEGER.)
0103    CCCCC
0104    CCCCC           K          FUNCTION
0105    CCCCC          ---  ------------------------------
0106    CCCCC           1   CLOSE FOR 10-MS  WAIT FOR
0107    CCCCC               150-MS.  REPEAT "IDEL" TIMES.
0108    CCCCC
0109    CCCCC           2   CLOSE FOR 10-MS  WAIT FOR
0110    CCCCC               4-SEC  RETURN.
0111    CCCCC
0112    CCCCC           3   CLOSE  RETURN.
0113    CCCCC          ---  ------------------------------
0114    CCCCC
0115    CCCCC     AUTHOR: R.N. GEOPFARTH,LT USN
0116    CCCCC     DATE:   JAN 79
0117    C
0118          GO TO(100,200,300),K
0119    C
0120    100   DO 10 I=1,IDEL,ISTEP
0121          WRITE(8,60)ICHAN
0122          CALL WAIT(1)
0123          WRITE(8,62)
0124          CALL WAIT(15)
0125    10    CONTINUE
0126          RETURN
0127    C
0128    200   WRITE(8,60)ICHAN
0129          CALL WAIT(1)
0130          WRITE(8,62)
0131          CALL WAIT(400)
0132          RETURN
0133    C
0134    300   WRITE(8,60)ICHAN
0135          RETURN
0136    C
0137    60    FORMAT(A2)
0138    62    FORMAT("C")
0139          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00088 COMMON = 00000

```
0140          SUBROUTINE CURVE (N,X,Y,SLOPE,SECON)
0141    C     .....................................................
0142    CCCCC .                                                   .
0143    CCCCC .  Compute linear curve fit using least square root method.
0144    CCCCC .  Author:     Alan G. McGuire                      .
0145    CCCCC .  Date:       February 21, 1980                    .
0146    CCCCC .  A detailed program description is available in TXCO log.  The .
0147    CCCCC .  comment statements and statement numbers match to the ones .
0148    CCCCC .  used in the flow chart.                          .
0149    CCCCC .                                                   .
0150    C     .....................................................
0151          *, Computes linear curve fit.
0152          REAL X(N),Y(N)
0153          SUMP0 = 0.
0154          SUME0 = 0.
0155          SUMVP = 0.
0156          SUME2 = 0.
0157          DO 1 I=1,N,1
0158          SUME0 = SUME0 + X(I)
0159          SUMP0 = SUMP0 + Y(I)
0160          SUMVP = SUMVP + (X(I)*Y(I))
0161       01 SUME2 = SUME2 + X(I)*X(I)
0162          RN = FLOAT(N)
0163          SNUM = (RN*SUMVP) - (SUME0*SUMP0)
0164          SDEN = (RN*SUME2) - (SUME0*SUME0)
0165          SLOPE = SNUM/SDEN
0166          SECON = (SUMP0-(SLOPE*SUME0))/RN
0167          RETURN
0168          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00176 COMMON = 00000

```
PAGE 0005  FTN.   9:57 PM  SAT., 27  SEP., 1980

0169          INTEGER FUNCTION ICON (I,N)
0170    C     .....................................................................
0171    CCC   .                                                                   .
0172    CCC   .  Converts integer numbers into ASCI string.                      .
0173    CCC   .  Author:     Robert N. Geopfarth                                  .
0174    CCC   .  Date:       January 31, 1979                                     .
0175    CCC   .  Because of the simplicity of the program the program             .
0176    CCC   .  description is included in this box.                             .
0177    CCC   .  I, N    ...   integer numbers to be added.                       .
0178    CCC   .  IC      ...   integer number to be converted into ASCII.         .
0179    CCC   .  ICON    ...   2 - character ASCII string to be returned          .
0180    CCC   .                                                                   .
0181    CCC   .....................................................................
0182          *, Converts integer to ASCII-string.
0183      100 FORMAT (I2)
0184
0185          IC = I+N
0186          IF ( IC  .LT.  10 ) GO TO 01
0187
0188          CALL CODE
0189          WRITE (ICON,100) IC
0190          RETURN
0191
0192       01 ICON = IC+30060B
0193          RETURN
0194          END

  FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

  ** NO WARNINGS ** NO ERRORS **   PROGRAM = 00036      COMMON = 00000
```

```
0195          INTEGER FUNCTION IPORT (IVALVE)
0196         *, Interrogates scanivalve.
0197    C
0198    C               THIS PROGRAM INTERROGATES SCANIVALVE
0199    C               "IVALVE" AND CONVERTS PORT ADDRESS
0200    C               INTO A DECIMAL VALUE.
0201    C
0202    C               VARIABLES:
0203    C
0204    C                 IVALVE = DESIRED S/V
0205    C                 IP     = S/V INPUT BUFFER
0206    C                 MSD    = MOST SIGNIF. DIGIT
0207    C                 LSD    = LEAST SIGNIF. DIGIT
0208    C                 IPORT  = DECIMAL S/V ADDRESS
0209    C
0210    C       AUTHOR: R.N. GEOPFARTH,LT USN
0211    C       DATE:   DEC 78
0212    C
0213          LU = 14 + 2100B
0214          CALL EXEC(2,LU,IVALVE*256,-1)
0215          CALL EXEC(1,LU,IP,-1)
0216          IP=IP/256
0217          MSD = IAND(IP/16,7B)
0218          LSD = IAND(IP,17B)
0219          IPORT = 10*MSD + LSD
0220          RETURN
0221          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00064 COMMON = 00000

```
0222          SUBROUTINE PICTR (LO,NUMBR,NEWPG,ICL,ISIGN,AMPL)
0223    C     ..................................................................
0224    CCC   :                                                                :
0225    CCC   :  Use terminal screen for grafic display.                       :
0226    CCC   :                                                                :
0227    CCC   ..................................................................
0228          *  Use CRT to display the acquired data.                         *
0229          COMMON / CIBUF / IBUF
0230          REAL CLR(64)
0231          INTEGER IBUF(1664)
0232          INTEGER NOLF,NOCR(2),ICLR(3),PLOT(8)
0233          INTEGER BLACK(2),GREY(2),WHITE(2),BG(2),LN(2),TEXT(20),HJF(3)
0234          DATA NOLF  /006537B/
0235          DATA NOCR  /000033B,040433B/
0236          DATA ICLR  /015524B,015515B,006537B/
0237          DATA BLACK /015446B,062100B/
0238          DATA WHITE /015446B,062102B/
0239          DATA GREY  /015446B,062112B/
0240          DATA HJF   /015510B,015512B,015506B/
0241          DATA ICLEAR /2H  /
0242    C     FORMATS  PICTR  START
0243     1801 FORMAT (2A2,79X,3A2)
0244     1803 FORMAT (A2"a"I3"r"I3"CFigure"I3"; "20A2)
0245     1805 FORMAT (2A2,8X,2A2,6SX,2A2,6X,3A2)
0246     1806 FORMAT (A2"a"I3"r"I3"C"F6.3)
0247     1807 FORMAT (2A2,8X,2A2,1X,4(2A2,15X,2A2,1X),2A2,6X)
0248     1809 FORMAT (" "F8.3,4F16.3)
0249     1817 FORMAT (8A2)
0250     1819 FORMAT (A2"a"I3"r"I3"C"A1,2A2)
0251    C     FORMATS  PICTR  STOP
0252          IF ( IFRST  .EQ.  1 ) GO TO 21
0253          IFRST = 1
0254          NEWPG = 1
0255       21 IF ( NEWPG  .EQ.  1 ) ICL = 0
0256          IF ( LO  .EQ.  1 ) GO TO 01
0257          IF ( LO  .EQ.  18 ) GO TO 02
0258       01 LN(1) = GREY(1)
0259          LN(2) = GREY(2)
0260          GO TO 03
0261       02 LN(1) = BLACK(1)
0262          LN(2) = BLACK(2)
0263       03 BG(1) = WHITE(1)
0264          BG(2) = WHITE(2)
0265          IF ( ICL  .NE.  1 ) GO TO 25
0266    C     ..................................................................
0267    CCC   :                                                                :
0268    CCC   :  Clear dots in frame w/o erasing the frame.                    :
0269    CCC   :                                                                :
0270    CCC   ..................................................................
0271       11 IROW = 11
0272          ICOL = 7
0273          CALL CODE
0274          WRITE (PLOT,1819) BLACK(1),IROW,ICOL,ICLEAR,NOCR
0275          WRITE (LO,1817) PLOT
0276          DO 13 I=1,64
0277          DO 12 J1=1,21
0278          J = J1-11
0279          X = J
0280       12 IF ( CLR(I)  .GE.  X-0.5  .AND.  CLR(I)  .LT.  X+0.5 ) NUPDN = J
0281          IROW = 11-NUPDN
0282          ICOL = 7+I
0283          CALL CODE
0284          WRITE (PLOT,1819) BLACK(1),IROW,ICOL,ICLEAR,NOCR
0285       13 WRITE (LO,1817) PLOT
0286    C     ..................................................................
0287    CCC   :                                                                :
0288    CCC   :  Get the curve in the format required by PICTR.                :
0289    CCC   :                                                                :
0290    C     ..................................................................
0291       25 J=0
0292          DO 04 I=1,256,4
0293          J=J+1
0294       04 CLR(J) = IBUF(I)/10000.
0295          XMIN = CLR(1)
0296          XMAX = CLR(1)
0297          IF ( ICL  .EQ.  1 ) GO TO 43
```

```
0298          DO 41 I=1,64
0299          IF ( CLR(I)  .GT.  XMAX ) XMAX = CLR(I)
0300       41 IF ( CLR(I)  .LT.  XMIN ) XMIN = CLR(I)
0301          AMPL = XMAX
0302          IF ( ABS(XMIN)  .GT.  XMAX ) AMPL = -XMIN
0303          IF ( AMPL  .LE.  0.001  .AND.  AMPL  .GT.  0.000 ) AMPL= 0.001
0304          IF ( AMPL  .LE.  0.002  .AND.  AMPL  .GT.  0.001 ) AMPL= 0.002
0305          IF ( AMPL  .LE.  0.010  .AND.  AMPL  .GT.  0.005 ) AMPL= 0.010
0306          IF ( AMPL  .LE.  0.020  .AND.  AMPL  .GT.  0.010 ) AMPL= 0.020
0307          IF ( AMPL  .LE.  0.050  .AND.  AMPL  .GT.  0.020 ) AMPL= 0.050
0308          IF ( AMPL  .LE.  0.100  .AND.  AMPL  .GT.  0.050 ) AMPL= 0.100
0309          IF ( AMPL  .LE.  0.200  .AND.  AMPL  .GT.  0.100 ) AMPL= 0.200
0310          IF ( AMPL  .LE.  0.500  .AND.  AMPL  .GT.  0.200 ) AMPL= 0.500
0311          IF ( AMPL  .LE.  1.000  .AND.  AMPL  .GT.  0.500 ) AMPL= 1.000
0312          IF ( AMPL  .LE.  2.000  .AND.  AMPL  .GT.  1.000 ) AMPL= 2.000
0313          IF ( AMPL  .LE.  5.000  .AND.  AMPL  .GT.  2.000 ) AMPL= 5.000
0314          IF ( AMPL  .LE. 10.000  .AND.  AMPL  .GT.  5.000 ) AMPL=10.000
0315       43 CONTINUE
0316          DO 42 I=1,64
0317       42 CLR(I) = CLR(I)*(10.0/AMPL)
0318          IF ( NEWPG  .NE.  1 ) GO TO 31
0319    C     .....................................................................
0320    CCCCC :                                                                   :
0321    CCCCC :  New frame.                                                       :
0322    CCCCC :                                                                   :
0323    C     .....................................................................
0324       05 WRITE (LO,1817) HJF(1),HJF(2),NOLF
0325          WRITE (LO,1801) BG,BLACK
0326          WRITE (LO,1805) BG,LN,BG,BLACK
0327          DO 06 I=1,9
0328       06 WRITE (LO,1807) BG,LN,BG,LN,BG,LN,BG,LN,BG,LN,BG
0329          WRITE (LO,1805) BG,LN,BG,BLACK
0330          DO 07 I=1,9
0331       07 WRITE (LO,1807) BG,LN,BG,LN,BG,LN,BG,LN,BG,LN,BG
0332          WRITE (LO,1805) BG,LN,BG,BLACK
0333          WRITE (LO,1801) BG,BLACK
0334          WRITE (LO,1801) BG,BLACK,NOLF
0335    C     .....................................................................
0336    CCCCC :                                                                   :
0337    CCCCC :  Label the existing frame.                                        :
0338    CCCCC :                                                                   :
0339    C     .....................................................................
0340       31 ZERO  = 0.
0341          AMPLM = -AMPL
0342          IROW  = 0
0343          ICOL  = 8
0344          WRITE (LO,1803) BLACK(1),IROW,ICOL,NUMBR,TEXT
0345          IROW  = 1
0346          ICOL  = 2
0347          WRITE (LO,1806) BLACK(1),IROW,ICOL,AMPL
0348          IROW  = 11
0349          WRITE (LO,1806) BLACK(1),IROW,ICOL,ZERO
0350          IROW  = 21
0351          WRITE (LO,1806) BLACK(1),IROW,ICOL,AMPLM
0352    C     .....................................................................
0353    CCCCC :                                                                   :
0354    CCCCC :  Plot curve into frame.                                           :
0355    CCCCC :                                                                   :
0356    C     .....................................................................
0357       08 CONTINUE
0358          DO 10 I=1,64
0359          DO 09 J1=1,21
0360          J = J1-11
0361          X = J
0362       09 IF ( CLR(I)  .GE.  X-0.5  .AND.  CLR(I)  .LT.  X+0.5 ) NUPDN = J
0363          IROW = 11-NUPDN
0364          ICOL = 7+I
0365          CALL CODE
0366          WRITE (PLOT,1819) BLACK(1),IROW,ICOL,ISIGN,NOCR
0367       10 WRITE (LO,1817) PLOT
0368          IROW = 11
0369          ICOL = 7
0370          CALL CODE
0371          WRITE (PLOT,1819) BLACK(1),IROW,ICOL,ICLEAR,NOCR
0372          WRITE (LO,1817) PLOT
0373          RETURN
```

```
0374          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01345 COMMON = 00000

```
0375          SUBROUTINE REWRF (IREWR,IWHATA)
0376    C     .....................................................................
0377    CCC   .                                                                   .
0378    CCC   .  This subroutine reads (IREWR = -1) or writes (IREWR = +1) of    .
0379    CCC   .  of a array specified by IWHATA.                                  .
0380    CCC   .  Author:    Hans M. Zebner                                        .
0381    CCC   .  Date  :    Febraury 08, 1980                                     .
0382    CCC   .  Detailed program description is available in TXCO log;  The     .
0383    CCC   .  Comment statements match to the flow chart explanations.         .
0384    CCC   .                                                                   .
0385    C     .....................................................................
0386          *  Data transfer disc    array.
0387          COMMON / CIBUF / IBUF
0388          COMMON / CONTR / CNTRL
0389    C     COMMON / CA    / A
0390          COMMON / FMP   / IDCB,IFILE,ISIZE
0391    C     REAL A(256)
0392          INTEGER IBUF(1664)
0393          INTEGER IDCB(144),IFILE(3),ISIZE(2)
0394          INTEGER NOLF,NOCR(2),ICLR(3)
0395          DATA NOLF  /006537B/
0396          DATA NOCR  /000033B,040433B/
0397          DATA ICLR  /015524B,015515B,006537B/
0398    C     FORMATS  REWRF  START
0399      101 FORMAT ("  REWRF : ARRAY IBUF(1664)   DISC FILE IBUFF
0400         *:00:26")
0401      102 FORMAT ("  REWRF : DISC FILE IBUFF:00:26   ARRAY IBUF(1664)")
0402      103 FORMAT ("  REWRF : ARRAY CNTRL(256)   DISC FILE CNTRLF:00:26"
0403         *)
0404      104 FORMAT ("  REWRF : DISC FILE CNTRLF:00:26   ARRAY CNTRL(256)"
0405         *)
0406      105 FORMAT ("  REWRF : ARRAY A(256)   DISC FILE AF:00:26")
0407      106 FORMAT ("  REWRF : DISC FILE AF:00:26   ARRAY A(256)")
0408      107 FORMAT ("  REWRF : ERROR RETURN (IWHATA ="I3")")
0409      108 FORMAT ("IBUFF ")
0410      109 FORMAT ("CNTRLF")
0411      110 FORMAT ("AF    ")
0412      121 FORMAT (" CALL OPEN  (IDCB,IERR,"3A2","I2","I2","I2","I4"
0413         *)  failed; STOP"21X"")
0414      122 FORMAT (" CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),I
0415         *DUM,IDUM,ISIZE(2)) failed; STOP"4)
0416      123 FORMAT (" CALL RWNDF (IDCB,IERR)  failed; STOP"42X"")
0417      124 FORMAT (" CALL READF (IDCB,IERR,IBUF,"I3","I2","I2")  fai
0418         *led; STOP"27X"")
0419      125 FORMAT (" CALL WRITF (IDCB,IERR,IBUF,"I3","I2","I2")  fai
0420         *led; STOP"26X"")
0421      126 FORMAT (" CALL READF (IDCB,IERR,CNTRL,"I3","I2","I2")  fa
0422         *iled; STOP"27X"")
0423      127 FORMAT (" CALL WRITF (IDCB,IERR,CNTRL,"I3","I2","I2")  fa
0424         *iled; STOP"26X"")
0425      128 FORMAT (" CALL READF (IDCB,IERR,A,"I3","I2","I2")  failed
0426         *; STOP"27X"")
0427      129 FORMAT (" CALL WRITF (IDCB,IERR,A,"I3","I2","I2")  failed
0428         *; STOP"26X"")
0429      130 FORMAT (" CALL CLOSE (IDCB,IERR,0)  failed; STOP"40X""
0430         *)
0431          LI = LOGLU(ISESSN)
0432          ISECU = 0
0433          ICR = 26
0434          IF ( IWHATA  .LT.  1   .OR.   IWHATA  .GT.  2 ) GO TO 40
0435          GO TO (10,20) IWHATA
0436
0437
0438
0439    C     .....................................................................
0440    CCC   .                                                                   .
0441    CCC   .  Integer array IBUF being written back and forth.                 .
0442    CCC   .                                                                   .
0443    C     .....................................................................
0444       10 CALL CODE
0445          WRITE (IFILE,108)
0446          CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0447          IF ( IERR  .GE.  0 ) GO TO 11
0448          WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0449          STOP 1
0450       11 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
```

```
0451          IF ( IERR  .GE.  0 ) GO TO 12
0452          WRITE (LI, 122)
0453          STOP 2
0454       12 CALL RWNDF (IDCB,IERR)
0455          IF ( IERR  .GE.  0 ) GO TO 13
0456          WRITE (LI, 123)
0457          STOP 3
0458       13 ISIZE(1) = ISIZE(1)/2
0459          IL          = ISIZE(1)*ISIZE(2)
0460          IF ( IREWR  .EQ.  -1 ) GO TO 14
0461          IF ( IREWR  .EQ.  +1 ) GO TO 15
0462       14 CALL READF (IDCB,IERR,IBUF,IL)
0463          IF ( IERR  .GE.  0 ) WRITE (LI, 102)
0464          IF ( IERR  .GE.  0 ) GO TO 16
0465          WRITE (LI, 124) IL,LEN,NUM
0466          STOP 4
0467       15 CALL WRITF (IDCB,IERR,IBUF,IL)
0468          IF ( IERR  .GE.  0 ) WRITE (LI, 101)
0469          IF ( IERR  .GE.  0 ) GO TO 16
0470          WRITE (LI, 125) IL,LEN,NUM
0471          STOP 5
0472       16 CALL CLOSE (IDCB,IERR,0)
0473          IF ( IERR  .GE.  0 ) GO TO 17
0474          WRITE (LI, 130)
0475          STOP 6
0476       17 RETURN
0477
0478
0479
0480    C     .....................................................................
0481    CCC   :                                                                   :
0482    CCC   :  Integer array CNTRL being written back and forth.                :
0483    CCC   :                                                                   :
0484    C     .....................................................................
0485       20 CALL CODE
0486          WRITE (IFILE,109)
0487          CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0488          IF ( IERR  .GE.  0 ) GO TO 21
0489          WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0490          STOP 11
0491       21 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0492          IF ( IERR  .GE.  0 ) GO TO 22
0493          WRITE (LI, 122)
0494          STOP 12
0495       22 CALL RWNDF (IDCB,IERR)
0496          IF ( IERR  .GE.  0 ) GO TO 23
0497          WRITE (LI, 123)
0498          STOP 13
0499       23 ISIZE(1) = ISIZE(1)/2
0500          IL          = ISIZE(1)*ISIZE(2)
0501          IF ( IREWR  .EQ.  -1 ) GO TO 24
0502          IF ( IREWR  .EQ.  +1 ) GO TO 25
0503       24 CALL READF (IDCB,IERR,CNTRL,IL)
0504          IF ( IERR  .GE.  0 ) WRITE (LI, 104)
0505          IF ( IERR  .GE.  0 ) GO TO 26
0506          WRITE (LI, 126) IL,LEN,NUM
0507          STOP 14
0508       25 CALL WRITF (IDCB,IERR,CNTRL,IL)
0509          IF ( IERR  .GE.  0 ) WRITE (LI, 103)
0510          IF ( IERR  .GE.  0 ) GO TO 26
0511          WRITE (LI, 127) IL,LEN,NUM
0512          STOP 15
0513       26 CALL CLOSE (IDCB,IERR,0)
0514          IF ( IERR  .GE.  0 ) GO TO 27
0515          WRITE (LI, 130)
0516          STOP 16
0517       27 RETURN
0518
0519
0520
0521    C     .....................................................................
0522    CCC   :                                                                   :
0523    CCC   :  Real array A being written back and forth.                       :
0524    CCC   :                                                                   :
0525    C     .....................................................................
0526    C  30 CALL CODE
```

```
0527   C       WRITE (IFILE,110)
0528   C       CALL OPEN  (IDCB,IERR,IFILE,IOPTN,ISECU,ICR,IDCBS)
0529   C       IF ( IERR  .GE.  0 ) GO TO 31
0530   C       WRITE (LI, 121) IFILE,IOPTN,ISECU,ICR,IDCBS
0531   C       STOP 21
0532   C  31 CALL LOCF  (IDCB,IERR,IDUM,IDUM,IDUM,ISIZE(1),IDUM,IDUM,ISIZE(2))
0533   C       IF ( IERR  .GE.  0 ) GO TO 32
0534   C       WRITE (LI, 122)
0535   C       STOP 22
0536   C  32 CALL RWNDF (IDCB,IERR)
0537   C       IF ( IERR  .GE.  0 ) GO TO 33
0538   C       WRITE (LI, 123)
0539   C       STOP 23
0540   C  33 ISIZE(1) = ISIZE(1)/2
0541   C       IL        = ISIZE(1)*ISIZE(2)
0542   C       IF ( IREWR  .EQ.  -1 ) GO TO 34
0543   C       IF ( IREWR  .EQ.  +1 ) GO TO 35
0544   C  34 CALL READF (IDCB,IERR,A,IL)
0545   C       IF ( IERR  .GE.  0 ) WRITE (LI, 106)
0546   C       IF ( IERR  .GE.  0 ) GO TO 36
0547   C       WRITE (LI, 128) IL,LEN,NUM
0548   C       STOP 24
0549   C  35 CALL WRITF (IDCB,IERR,A,IL)
0550   C       IF ( IERR  .GE.  0 ) WRITE (LI, 105)
0551   C       IF ( IERR  .GE.  0 ) GO TO 36
0552   C       WRITE (LI, 129) IL,LEN,NUM
0553   C       STOP 25
0554   C  36 CALL CLOSE (IDCB,IERR,0)
0555   C       IF ( IERR  .GE.  0 ) GO TO 37
0556   C       WRITE (LI, 130)
0557   C       STOP 26
0558   C  37 RETURN
0559
0560
0561
0562   C       .....................................................................
0563   C       :                                                                   :
0564   C       :  Error;  IWHATA is not defined.                                   :
0565   C       :                                                                   :
0566   C       .....................................................................
0567      40   WRITE (LI, 107) IWHATA
0568           IWHATA = -IWHATA
0569           RETURN
0570           END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 01148 COMMON = 00000

```
0571          SUBROUTINE RPACE (ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AVRGE,IPRIN
0572         *T,LO)
0573     C    .....................................................................
0574     C    .                                                                   .
0575     C    .  Interface program to trigger HP A/D converted through pacer.     .
0576     C    .  Author:    Hans M. Zebner                                        .
0577     C    .  Date:      March 20, 1980                                        .
0578     C    .  Detailed program description is available in TXCO log.           .
0579     C    .                                                                   .
0580     C    .....................................................................
0581         *, Triggers A/D through Pacer.
0582
0583          COMMON / CIBUF / BUFR
0584          COMMON / CONTR / CNTRL
0585
0586          INTEGER BUFR(1664),CNTRL(256)
0587          INTEGER ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,NOLF,ICLR(3),WHERE,WH
0588         *EREP
0589
0590          DATA NOLF /006537B/
0591          DATA ICLR /015524B,015515B,006537B/
0592          DATA MASK /177700B/
0593          DATA FSVLTG /.1E01/
0594
0595     C    FORMATS RPACE START
0596       101 FORMAT (/ 9X" ENTERED SUBROUTINE RPACE ("I2","I1","I2","I6","I2","
0597         *I6","I2",AVRGE,"I1","I2")")
0598       102 FORMAT (10X"AVRGE"3X"ICOUNT"3X"IBLAD"1X"REP"5X"IBUF"4X"IBUF"5X"RBU
0599         *F"1X"BUFR(ICOUNT)")
0600       103 FORMAT ("Pointer at       "63X"A2)
0601       104 FORMAT ((" "A2))
0602       105 FORMAT (20X,I4,1X,K7,2X,K2,1X,K8,I8,1X,F8.6,1X,I12,A2)
0603       106 FORMAT (1X,F14.1,5X,I4,1X,K7,I4,1X,K8,I8,1X,F8.6,1X,I12)
0604       107 FORMAT ( 7X" COMPLETED SUBROUTINE RPACE ("I2","I1","I2","I6","I2",
0605         *"I6","I2","F5.3","I1","I2")")
0606       108 FORMAT ("Done"10X"
0607
0608       149 FORMAT ((3A2))
0609     C    FORMATS RPACE STOP .
0610
0611
0612
0613     C    .....................................................................
0614     C    .                                                                   .
0615     C    .  Check the input variables.  If one is out of range, it is        .
0616     C    .  set to the default value.  No warning is displayed.              .
0617     C    .                                                                   .
0618     C    .....................................................................
0619          LI = CNTRL(19)
0620          IF ( IPRINT  .EQ.   0 ) GO TO 01
0621          LINES = 0
0622          WRITE (LI, 101) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,IPRINT,LO
0623          WRITE (LI, 102)
0624          IF ( LO      .EQ.  LI ) GO TO 01
0625          WRITE (LO, 101) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,IPRINT,LO
0626          WRITE (LO, 102)
0627       01 IF ( ADCHNL  .GT.   15   .OR.   ADCHNL  .LT.    0 ) ADCHNL =   0
0628          IF ( PAMO    .GT.    2   .OR.   PAMO    .LT.    0 ) PAMO   =   1
0629          IF ( PAIR    .GT.    9   .OR.   PAIR    .LT.    2 ) PAIR   =   1
0630          IF ( START   .LT.    0                             ) START  =   0
0631          IF ( INCR    .LT.    1                             ) INCR   =   1
0632          IF ( REP     .LT.    1                             ) REP    =   1
0633     C    .....................................................................
0634     C    .                                                                   .
0635     C    .  Check input variables for logical errors.                        .
0636     C    .                                                                   .
0637     C    .....................................................................
0638          IF ( STOP    .LE.  START     ) STOP = START+1
0639          IF ( INCR    .GT.  STOP-START ) INCR = STOP-START
0640
0641
0642
0643     C    .....................................................................
0644     C    .                                                                   .
0645     C    .  Get adjusted START and STOP, depending on selected PAMO.         .
0646     C    .                                                                   .
```

```
0647  C     ..................................................................
0648        GO TO (03,02) PAMO
0649     02 START = (START + 256*PAIR) + 100000B
0650        STOP  = (STOP  + 256*PAIR) + 100000B
0651     03 IF ( IPRINT  .NE.  0 ) GO TO 05
0652        ICOUNT = 0
0653        DO 04 I=START,STOP,INCR
0654     04 ICOUNT = ICOUNT+1
0655        IDIFF  = ICOUNT
0656        ICORR  = 1
0657        IF ( IDIFF  .GT.  64 ) GO TO 05
0658        ICORR  = (64/IDIFF)
0659     05 ICOUNT = 0
0660
0661
0662
0663  C     ..................................................................
0664  CCC   .                                                                .
0665  CCC   .  Start acquisition loop.                                       .
0666  CCC   .                                                                .
0667  C     ..................................................................
0668        AVRGE = 0
0669        WHEREP = 1
0670        IF ( IPRINT  .EQ.  0 ) WRITE (LI, 103) NOLF
0671        DO 10 I=START,STOP,INCR
0672        ICOUNT = ICOUNT + 1
0673        WHERE = (ICOUNT*64)/IDIFF
0674        IF ( WHERE  .GT.  WHEREP  .AND.  IPRINT  .EQ. 0 ) GO TO 06
0675        GO TO 07
0676     06 WHEREP = WHERE
0677        WRITE (LI, 104) (NOLF,K=1,ICORR)
0678     07 CONTINUE
0679
0680
0681
0682  C     ..................................................................
0683  CCC   .                                                                .
0684  CCC   .  Repeat A/D conversion at selected point REP times.            .
0685  CCC   .                                                                .
0686  C     ..................................................................
0687        RBUF = 0
0688        BUFR(ICOUNT) = 0
0689        DO 08 J=1,REP,1
0690        CALL EXEC (3,19)
0691        CALL EXEC (1,19,IRPM,1,I)
0692        CALL EXEC (1,20,IBUF,1,ADCHNL,0)
0693        IBUF = IAND(IBUF,MASK)
0694        RBUF = FLOAT(IBUF)/32768. + RBUF
0695        IF ( IPRINT  .EQ.  0                          ) GO TO 08
0696        WRITE (LI, 105) ICOUNT,I,J,IBUF,IBUF,RBUF,BUFR(ICOUNT),NOLF
0697        IF ( LO      .EQ.  LI    .OR.  LO   .EQ.  6 ) GO TO 08
0698        WRITE (LO, 105) ICOUNT,I,J,IBUF,IBUF,RBUF,BUFR(ICOUNT),NOLF
0699     08 CONTINUE
0700        BUFR(ICOUNT) = ((RBUF*FSVLTG)/REP)*10000
0701
0702
0703
0704        AVRGE = AVRGE + BUFR(ICOUNT)
0705        IF ( IPRINT  .EQ.  0 ) GO TO 10
0706        WRITE (LI, 106) AVRGE,ICOUNT,I,REP,IBUF,IBUF,RBUF,BUFR(ICOUNT)
0707        LINES = LINES+1
0708        IF ( LINES  .LT.  20 ) GO TO 09
0709        WRITE (LI, 149) (ICLR,K=1,LINES)
0710        LINES = 0
0711     09 IF ( LO     .EQ.  LI ) GO TO 10
0712        WRITE (LO, 106) AVRGE,ICOUNT,I,REP,IBUF,IBUF,RBUF,BUFR(ICOUNT)
0713     10 CONTINUE
0714  C     ..................................................................
0715  CCC   .                                                                .
0716  CCC   .  Stop data acquisition loop.                                   .
0717  CCC   .                                                                .
0718  C     ..................................................................
0719
0720
0721
0722        AVRGE = (AVRGE/ICOUNT)/10000
```

```
0723          IF ( IPRINT  .EQ.   0 ) GO TO 11
0724          LINES = LINES+1
0725          WRITE (LI, 149) (ICLR,I=1,LINES)
0726          WRITE (LI, 107) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AVRGE,IPRINT,
0727         *LO
0728          IF ( LO      .EQ.  LI ) GO TO 11
0729          WRITE (LO, 107) ADCHNL,PAMO,PAIR,START,INCR,STOP,REP,AVRGE,IPRINT,
0730         *LO
0731
0732
0733
0734       11 CONTINUE
0735          IF ( IPRINT  .NE.   0 ) GO TO 12
0736          WRITE (LI, 108)
0737          WRITE (LI, 149) ICLR
0738       12 RETURN
0739          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00973 COMMON = 00000

```
0740          REAL FUNCTION SCANR (LU,ICHAN,K)
0741     C    ..................................................................
0742     C    :                                                                :
0743     C    :  Close relay ICHAN on scanner LU and read the instrument       :
0744     C    :  indicated by K.                                               :
0745     C    :  Author:     Robert N. Geopfarth                               :
0746     C    :  Date:       February 31, 1979                                 :
0747     C    :  Detailed program description is available in TXCO log;  the   :
0748     C    :  variables are:                                                :
0749     C    :  LU       ...    LU# of desired scanner (8 or 15).             :
0750     C    :  ICHAN    ...    Scanner channel (integer).                    :
0751     C    :  IC       ...    Scanner channel (ASCII).                      :
0752     C    :  K        ...    Instrument code ( DVM = 1 / Counter = 2 ).    :
0753     C    :                                                                :
0754     C    ..................................................................
0755          *  Closes scanner and reads DVM, counter.  *
0756        101 FORMAT (A2)
0757        801 FORMAT ("C")
0758       1001 FORMAT ("T3T3")
0759       1201 FORMAT ("T")
0760       1501 FORMAT ("C")
0761
0762            WRITE ( 8, 801)
0763            WRITE (15,1501)
0764            IC = ICON(ICHAN,0)
0765            WRITE (LU, 101) IC
0766            GO TO (01,02) K
0767
0768         01 CALL TRIGR (10)
0769            READ  (10,   *) DUM
0770            CALL TRIGR (10)
0771            READ  (10,   *) SCANR
0772            GO TO 03
0773
0774         02 WRITE (12,1201)
0775            READ  (12,   *) SCANR
0776
0777         03 WRITE (LU, 801)
0778            RETURN
0779            END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00104 COMMON = 00000

```
0780          SUBROUTINE TIME (IMON,IDAY,IYEAR,IHOUR,IMIN,ISEC)
0781    C     ..................................................................
0782    C     :                                                                :
0783    C     :  Get date and time and convert the variables to ASCII           :
0784    C     :                                                                :
0785    C     ..................................................................
0786         *  Gets date and time   ASCII string.      *
0787          COMMON / CONTR / CNTRL
0788          INTEGER ITIME(5)
0789          INTEGER CNTRL(256)
0790      901 FORMAT (" ERROR DETECTED IN PROGRAM TIME"/
0791         *        " CALL EXEC (11,ITIME)"/)
0792
0793          IMON = 2H$$
0794          IDAY = 2H$$
0795          IYEAR= 2H$$
0796          IHOUR= 2H$$
0797          IMIN = 2H$$
0798          ISEC = 2H$$
0799          CALL EXEC (11+100000B,ITIME)
0800          GO TO 02
0801       01 GO TO 03
0802       02 CALL ABREG (IA,IB)
0803          GO TO 04
0804       03 IMON  = ICON(CNTRL(1),0)
0805          IDAY  = ICON(CNTRL(2),0)
0806          IYEAR = ICON(CNTRL(3),0)
0807          IHOUR = ICON(ITIME(4),0)
0808          IMIN  = ICON(ITIME(3),0)
0809          ISEC  = ICON(ITIME(2),0)
0810          RETURN
0811       04 WRITE ( 6, 901) IA,IB
0812          RETURN
0813          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00146 COMMON = 00000

```
0814          SUBROUTINE WAIT (TWAIT)
0815    C     .....................................................................
0816    CCC   .                                                                   .
0817    CCC   .  Cause a defined time delay of TWAIT*10 milliseconds.             .
0818    CCC   .  Author:     Hans M. Zebner                                       .
0819    CCC   .  Date:       February 13, 1980                                    .
0820    CCC   .  Because of the simplicity of the program the program             .
0821    CCC   .  description is included in this box.                             .
0822    CCC   .  TWAIT  ...   Desired time delay is (TWAIT*10) milliseconds.      .
0823    CCC   .  TNOW   ...   Present time.                                       .
0824    CCC   .  TM(5)  ...   Input time buffer (required for EXEC call).        .
0825    CCC   .  TSTOP  ...   Final time.                                         .
0826    CCC   .                                                                   .
0827    C     .....................................................................
0828         *, Causes a defined time delay.              Geopfarth, Zebner
0829          IMPLICIT INTEGER (T)
0830          INTEGER TM(5)
0831
0832       01 CALL EXEC (11,TM)
0833          THOUR = TM(4)
0834          TSTOP = TM(1) + TM(2)*100 + TM(3)*6000 + TWAIT
0835
0836       02 CALL EXEC (11,TM)
0837          TNOW  = TM(1) + TM(2)*100 + TM(3)*6000
0838          IF ( TM(4)   .NE.  THOUR ) GO TO 01
0839          IF ( TNOW    .LT.  TSTOP ) GO TO 02
0840
0841          RETURN
0842          END
```

FTN4 COMPILER: HP92060-16092 REV. 1926 (790430)

** NO WARNINGS ** NO ERRORS ** PROGRAM = 00087 COMMON = 00000

8. DATA REDUCTION PROGRAMS

Three data reduction programs can be initiated from within the TXCO system. They are the following:

(i) Program REDAB (Enter 9)

This program was written to reduce data from the A-B Kulite probe system following the method given in Ref 2, and outputs distributions of velocity magnitude and flow angles.

(ii) Program REDCO (Enter 10)

This program reduces survey data taken with the combination temperature-pneumatic probe and outputs distributions of pressure rise, temperature rise, Mach number, flow angle and losses.

(iii) Program REDST (Enter 11)

This program reduces data taken from fixed instrumentation and outputs the steady-state performance of the compressor.

The above programs are documented separately.

9. CONCLUSIONS AND RECOMMENDATIONS

The large quantity and variety of instrumentation used in the transonic compressor test facility required that data acquisition programs be provided for the different types of data. This was achieved using a particular program structure. Data reduction programs were strictly separated but geared to the acquisition modules through the use of standard data arrays. The data acquisition programs TXCO1, TXCO2 and TXCO3 have been described in detail and the operator commands are explained. The reduction programs are to be documented separately.

The need for easy-to-understand program control leads to a conflict. If interactive messages which explain the program flow and offer menus giving a selection of next logical steps are included, this introduces extended I/O operations and leads to long programs whose speed in execution is slowed considerably by the I/O's. On the other hand the I/O's may be kept to a minimum, which speeds up execution, but this may also lead to communication gaps between the program and the operator. Since the research on the transonic compressor test rig is carried out in large part by visiting researchers and postgraduate students, it was decided to program closer to the first alternative. However, a very useful compromise was achieved through the introduction of the program control array. Should experience in using the TXCO-system show that

the interactive messages, error processing, or the checking for erroneous operator input are too extensive, then the programs should certainly be trimmed.

At the time the programs were written, a graphic software package was not present in the operating system and therefore original plotter software was generated. The switch from "home made" to HP-supported graphics is recommended.

Finally, if the instrumentation system is changed, corresponding changes can be introduced into the appropriate program module, or a new one can be added. Also, the same or a similar program system can easily be adapted for use on any other test rig or calibration apparatus in the laboratory.

APPENDIX A. DATA ACQUISITION WORK SHEETS

A.1. Data Locations

A.2. Steady State Data Array

A.3. Program Control Array (CNTRL)

A.4. Paced Data Array

A.1. DATA LOCATIONS

WORK SHEET: DATA LOCATIONS

Port	Scanivalve #1	Scanivalve #2	Scanner #1	Scanner #2	Ch
			Advance S/V #1	T1 A/C nozzle	0
1	PA-PA	PA-PA	" " #2	T turb in	1
2	PCAL-PA	PCAL-PA	" " #3	T turb out L	2
3	P1 nozzle-PA	P1 comb pr-PA	" " #4	T turb out R	3
4	P1 noz th-PA	P23 comb pr-PA	" " #5	T1 comp noz D	4
5	P1 noz f1-PA	P4 comb pr-PA	Reset S/V #1	T1 comp noz W	5
6	PBM-PA	PT2-PA	" " #2	T in sta 00	6
7	PT00-PA	PT1-PA	" " #3	T A4	7
8	S1-PA	PA-PA	" " #4	T B4	8
9	S2-PA	K eq-PA	" " #5	T C4	9
10	S3-PA	P alpha-PA	Transducer S/V #1	T cell	10
11	S4-PA	C7-PA	" " #2		11
12	S5-PA	A1-PA	" " #3	ΔT turb L	12
13	S6-PA	B1-PA	" " #4	ΔT turb R	13
14	S7-PA	C1-PA	" " #5	ΔT A4	14
15	S8-PA	A2-PA	RPM A/C	ΔT B4	15
16	S9-PA	B2-PA	RPM TTR	ΔT C4	16
17	S10-PA	C2-PA	RPM TCR		17
18	S11-PA	A3-PA	RPM TTR	T in ref pr	18
19	S12-PA	B3-PA	Blade pass frequ	T comb ref	19
20	S13-PA	C3-PA	TTR AXF		20
21	S14-PA	A4-PA	TTR CLAF		21
22	S15-PA	B4-PA	TTR N-Mv		22
23	S16-PA	C4-PA	TTR DyTQ		23
24	S17-PA	A5-PA	TTR StTQ		24
25	PA-PA	B5-PA	P barometric		25
26	PCAL-PA	C5-PA	P1 nozzle comp		26
27	S18-PA	A6-PA	P nozzle comp		27
28	S19-PA	B6-PA	P1 nozzle turb		28
29	H1-PA	C6-PA	P nozzle turb		29
30	H2-PA	A7-PA	rad pos comb pr		30
31	H3-PA	P bearing-PA	yaw comb pr		31
32	H4-PA	P thrust-PA	rad pos 'A' pr		32
33	H5-PA	PT turb in-PA	yaw 'A' pr		33
34	H6-PA	P st out L-PA	rad pos 'B' pr		34
35	H7-PA	P st out R-PA	yaw 'B' pr		35
36	H8-PA	PT ro out L-PA	Torque TCR		36
37	H9-PA	PT ro out R-PA	KUL ref pres		37
38	H10-PA	P ro out L-PA			38
39	H11-PA	P ro out R-PA			39
40	Diff T1-PA	PA-PA		wall KUL K6.	40
Port	Scanivalve #1	Scanivalve #2	Scanner #1	Scanner #2	Ch

Port	Scanivalve #1	Scanivalve #2	Scanner #1	Scanner #2	Ch
46	Diff T7-PA			wall KUL K10.	46
47	Diff T8-PA			" " K10.5	47
48	Diff T9-PA			" " K11.	48
49				" " K12.	49
50					50
51					51
52				'A' KUL pr	52
53				'B' KUL pr	53
54					54
55					55
56					56
57					57
58					58
59					59
60					60
61					61
62					62
63					63
64					64
65					65
66					66
67					67
68					68
69					69
70					70
71					71
72					72
73					73
74					74
75					75
76					76
77					77
78					78
79					79

A.2. STEADY STATE DATA ARRAY

WORK SHEET TO DECODE THE DATA

ARRAY FOR THE STEADY STATE DATA

File name convention: raw steady state data T4RRSS
reduced steady state data T9RRSS

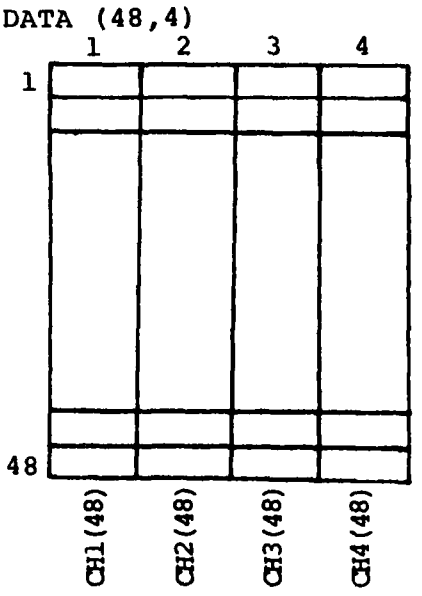
RR = ASCII converted run #
SS = file sequential # from this run

Note: In some cases the letter T (which stands for Transonic compressor) may be changed to any other character in order to prevent overwriting an existing data file.

Example: T40503 is the third data file from test run #5, that contains steady state data. The data reduction program REDST (Reduction steady state data) creates the file T90403 to take the reduced data from this run.

Due to interface bus problems the data acquisition program STDY sometimes has to be aborted, but may already have produced some valid raw data files. When the program is restarted, it tries to write to data files, whose names already exist. To avoid purging these files, the operator then interactively changes the first character of the data file name to U40503, e.g. The reduced data of course are in file U90503.

By the way, T40503 contains good data. You will find this file in cartridge 29 and can use it to get acquainted with REDST.



I	CH1(I)	CH2(I)	CH3(I)	CH4(I)	I
1	PA-PA	PA-PA	P barometric	n/a	1
2	PCAL-PA	PCAL-PA	P1 nozzle compr	n/a	2
3	P1 nozzle-PA	P1 comb pr-PA	P nozzle compr	PRTT	3
4	P1 noz th-PA	P23 comb pr-PA	P1 nozzle turb	TT1T	4
5	P1 noz f1-PA	P4 comb pr-PA	P nozzle turb	TT3T	5
6	PBM-PA	PT2-PA	n/a	DTTT	6
7	PT00-PA	PT1-PA	rad pos comb pr	MFLT	7
8	S1-PA	PA-PA	yaw comb pr	HPT	8
9	S2-PA	K eq-PA	rad pos 'A' pr	HPM	9
10	S3-PA	P alpha-PA	yaw 'A' pr	PRCT	10
11	S4-PA	C7-PA	rad pos 'B' pr	TT1C	11
12	S5-PA	A1-PA	yaw 'B' pr	TT3C	12
13	S6-PA	B1-PA	n/a	DTTC	13
14	S7-PA	C1-PA	T1 A/C nozzle	MFLC	14
15	S8-PA	A2-PA	T turb in	HPC	15
16	S9-PA	B2-PA	T turb out L	PRCTR	16
17	S10-PA	C2-PA	T turb out R	RPMCR	17
18	S11-PA	A3-PA	n/a	MFLCR	18
19	S12-PA	B3-PA	T1 comp noz D	TORQCR	19
20	S13-PA	C3-PA	T1 comp noz W	HPMR	20
21	S14-PA	A4-PA	T in sta 00	HPCR	21
22	S15-PA	B4-PA	T out A4	HPTR	22
23	S16-PA	C4-PA	T out B4	EFF0	23
24	S17-PA	A5-PA	T out C4	EFF1	24
25	PA-PA	B5-PA	T cell	EFF2	25
26	PCAL-PA	C5-PA	n/a	EFF3	26
27	S18-PA	A6-PA	ΔT turb L	n/a	27
28	S19-PA	B6-PA	ΔT turb R	n/a	28
29	H1-PA	C6-PA	ΔT A4	n/a	29
30	H2-PA	A7-PA	ΔT B4	T in ref pr	30
31	H3-PA	P bearing-PA	ΔT C4	T comb ref	31
32	H4-PA	P thrust-PA	n/a	n/a	32
33	H5-PA	PT turb in-PA	KUL ref pres	run #	33
34	H6-PA	P st out L-PA	wall KUL K6.	test #	34
35	H7-PA	P st out R-PA	" " K7.	point #	35
36	H8-PA	PT ro out L-PA	" " K8.	day	36
37	H9-PA	PT ro out R-PA	" " K8.5	month	37
38	H10-PA	P ro out L-PA	" " K9.	year	38
39	H11-PA	P ro out R-PA	" " K9.5	machine code	39
40	Diff T1-PA	PA-PA	" " K10.	n/a	40
41	Diff T2-PA	P diff 1-PA	wall KUL K10.5	n/a	41
42	Diff T3-PA	P diff 2-PA	" " K11.	n/a	42
43	Diff T4-PA	P diff 3-PA	" " K12.	case angle	43
44	Diff T5-PA	P diff 4-PA	" " K13.	n/a	44
45	Diff T6-PA	P diff 5-PA	" " K14.	n/a	45
46	Diff T7-PA	n/a	n/a	RPM	46
47	Diff T8-PA	n/a	'A' KUL pr	Torque	47
48	Diff T9-PA	n/a	'B' KUL pr	n/a	48
I	CH1(I)	CH2(I)	CH3(I)	CH4(I)	I

A.3. PROGRAM CONTROL ARRAY (CNTRL)

WORK SHEET TO DECODE/ENCODE THE CONTROL ARRAY CNTRL:

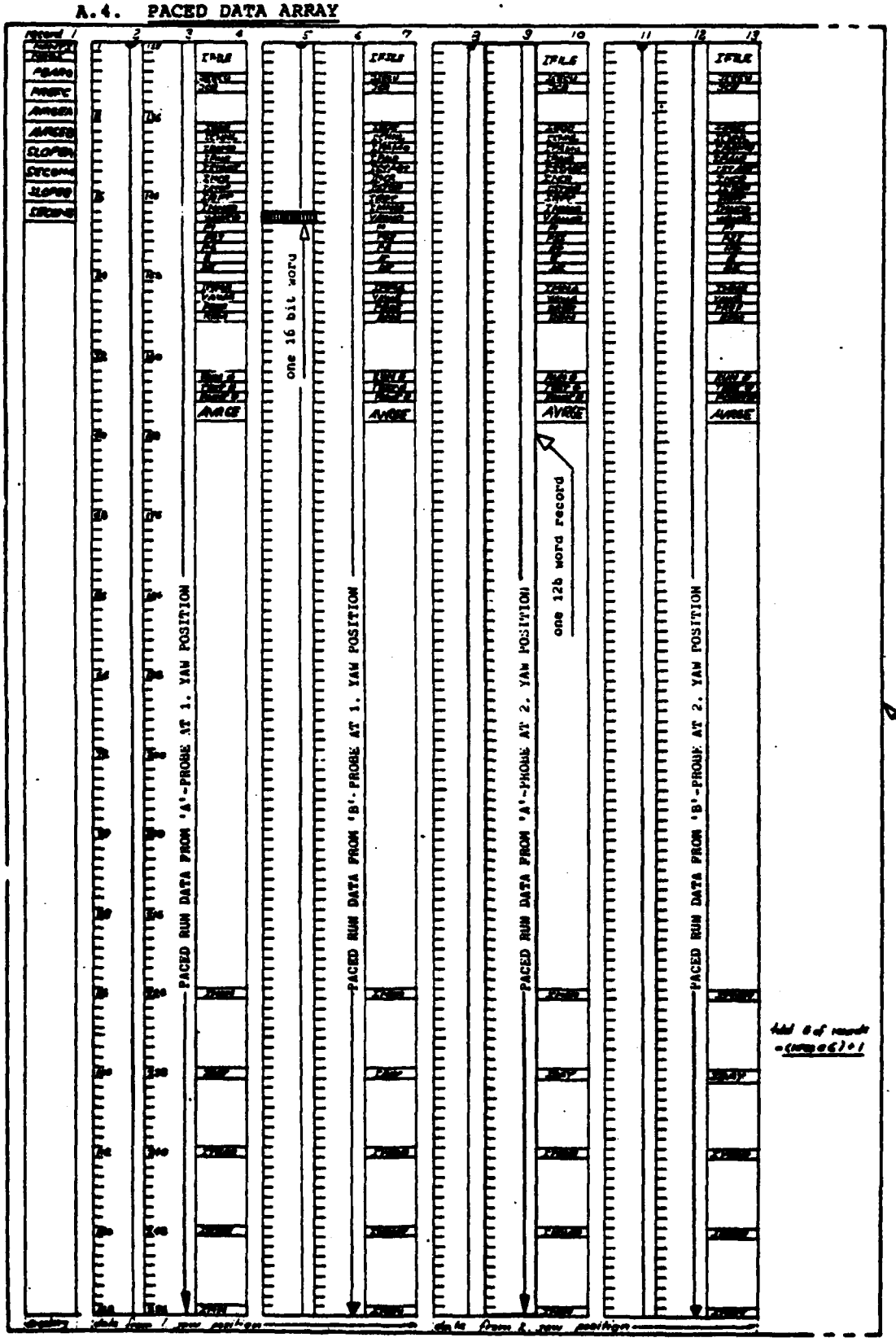
CNTRL(1) : Month of the test run.
CNTRL(2) : Day of the test run.
CNTRL(3) : Year of the test run.
CNTRL(4) : Test run #.
CNTRL(5) : Test # of this run.
CNTRL(6) : Point # of this test.
CNTRL(15) : Machine Code.
CNTRL(19) : LU# of the standard interactive input device.
CNTRL(20) : LU# of the standard output device.
CNTRL(21) : LU# of the optional output device.
CNTRL(22) : LU# of the plotter.
CNTRL(30) : Cartridge reference # for data files.
CNTRL(31) : Security code for data files.
CNTRL(32) : First and second character of data file name; IFILE(1)
CNTRL(33) : Third and fourth character of data file name; IFILE(2)
CNTRL(34) : Fifth and sixth character of data file name; IFILE(3)
CNTRL(36) : Initializes fast steady state data reduction run, if set to 1.
CNTRL(37) : Suppresses printing of heading in subroutines FREER and PACER, if set to 1.
CNTRL(39) : Suppresses creating/opening and closing of files in subroutines FREER and PACER, if set to 1.
CNTRL(40) : Suppresses analog output of just acquired paced run data to terminal, if set to 1.
CNTRL(41) : 100*Factor to vary size in X-direction of a drawing.
CNTRL(42) : 100*Factor to vary size in Y-direction of a drawing.
CNTRL(50) : Indicates the son program to be scheduled and the subroutine to be called therefrom.

```
01  ...  Schedule TXC01 and call ABSRV
02  ...       "      "    "    "  CALIB
03  ...       "      "    "    "  FREER
04  ...       "      "    "    "  PACER
05  ...  Schedule TXC02 and call COMB
06  ...       "      "    "    "  STDY
07  ...  Schedule TXC03 and call CHECK
08  ...       "      "    "    "  CHNGE
09  ...  Schedule REDAB
10  ...  Schedule REDCO
11  ...  Schedule REDST.
```

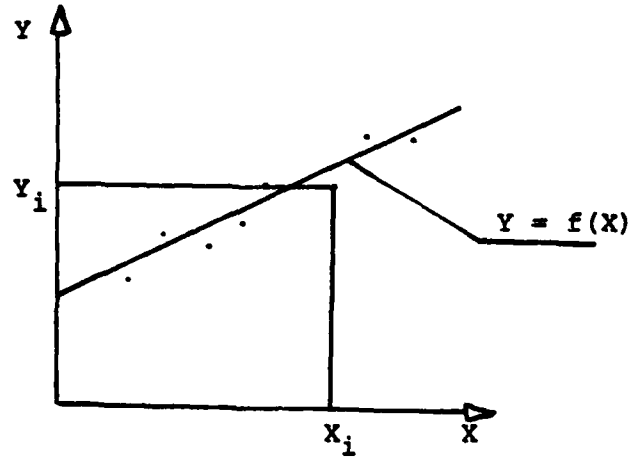
CNTRL(61) : Number of S/V controller #I.
CNTRL(62) : Number of S/V controller #II.
CNTRL(63) : Number of S/V controller #III.
CNTRL(64) : Number of S/V controller #IV.
CNTRL(65) : Number of S/V controller #V.
CNTRL(71) : LU# of scanner #1.
CNTRL(72) : LU# of scanner #2.

CNTRL(212)	:	Accounting variable subroutine ABSRV: output page #.
CNTRL(213)	:	" " " ABSRV: current file #.
CNTRL(214)	:	" " " CALIB: output page #.
CNTRL(215)	:	" " " CALIB: current file #.
CNTRL(216)	:	" " " FREER: output page #.
CNTRL(217)	:	" " " FREER: current file #.
CNTRL(218)	:	" " " PACER: output page #.
CNTRL(219)	:	" " " PACER: current file #.
CNTRL(221)	:	Blade pair (1 - 9), if Pacer is operated in Mode 2.
CNTRL(222)	:	Start count for data acquisition using Pacer encode.
CNTRL(223)	:	Increment for data acquisition using Pacer encode.
CNTRL(224)	:	Stop count for data acquisition using Pacer encode.
CNTRL(225)	:	# of repetitions at each location in blade passage.
CNTRL(230)	:	Total # of high speed data acquisitions either in free or in paced run mode to be taken.
CNTRL(231)	:	A/D input channel for KULITE type 'A' probe.
CNTRL(232)	:	" " " " " " 'B' probe.
CNTRL(235)	:	" " " " wall KULITE K6.
CNTRL(236)	:	" " " " " " K7.
CNTRL(237)	:	" " " " " " K8.
CNTRL(238)	:	" " " " " " K8.5.
CNTRL(239)	:	" " " " " " K9.
CNTRL(240)	:	" " " " " " K9.5.
CNTRL(241)	:	" " " " " " K10.
CNTRL(242)	:	" " " " " " K10.5.
CNTRL(243)	:	" " " " " " K11.
CNTRL(244)	:	" " " " " " K12.
CNTRL(245)	:	" " " " " " K13.
CNTRL(246)	:	" " " " " " K14.
CNTRL(249)	:	Character used for analog display in subroutine PICTR.
CNTRL(250)	:	# of multiples of 10ms for S/V controller time delay.
CNTRL(251)	:	Total # of free run measurements (max. 1664).

A.4. PACED DATA ARRAY



APPENDIX B. LINEAR APPROXIMATION BY METHOD OF LEAST SQUARES



Data: X_i and Y_i ; i = 1,...,NPNTSI

Equation: $Y = C_1 + C_2 \cdot X$

Difference for Each Data Point: $R_i = Y_i - f(X_i)$; i=1,...,NPNTSI

Sum of Squares of Differences: $$R = \sum_{i=1}^{NPNTSI} R_i^2 = \left[\sum_{i=1}^{NPNTSI} Y_i - (C_1+C_2 \cdot X_i)\right]^2$$

The value of R depends on the values of the coefficients C_1 and C_2. In order to determine a minimum value for R, the expression for R is partially differentiated with respect to C_1 and C_2 and the two derivatives are equated to zero. Differentiating,

$$\frac{\partial R}{\partial C_1} = \sum_{i=1}^{NPNTSI} 2 \cdot \left[Y_i - (C_1 + C_2 \cdot X_i)\right] \cdot (-1)$$

and

$$\frac{\partial R}{\partial C_2} = \sum_{i=1}^{NPNTSI} 2 \cdot \left[Y_i - (C_1 + C_2 \cdot X_i)\right] \cdot (-X_i)$$

Setting each expression to zero,

$$\sum_{i=1}^{NPNTSI} (Y_i - C_1 - C_2 \cdot X_i) = 0$$

$$\sum_{i=1}^{NPNTSI} (Y_i \cdot X_i - C_1 \cdot X_i - C_2 \cdot X_i^2) = 0$$

This gives two equations in which C_1 and C_2 are the only unknowns. Omitting the limits of summation for simplicity,

$$\sum C_1 + \sum C_2 X_i = \sum Y_i$$

$$\sum C_1 X_i + \sum C_2 X_i^2 = \sum Y_i \cdot X_i$$

or, in matrix notation (note that C_1 and C_2 are constants)

$$\begin{vmatrix} NPNTSI & \sum X_i \\ \sum X_i & \sum X_i^2 \end{vmatrix} \cdot \begin{vmatrix} C_1 \\ C_2 \end{vmatrix} = \begin{vmatrix} \sum Y_i \\ \sum Y_i X_i \end{vmatrix}$$

or

$$A \cdot C = B$$

The components of the matrix C are obtained using

$$a_{11} = NPNTSI$$

$$a_{12} = a_{21} = \sum X_i$$

$$a_{22} = \sum X_i^2$$

$$b_1 = \sum Y_i$$

$$b_2 = \sum Y_i \cdot X_i$$

$$C_1 = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix}} = \frac{a_{22}b_1 - a_{12}b_2}{a_{11}a_{22} - a_{12}^2}$$

$$C_2 = \frac{a_{11}b_2 - a_{12}b_1}{a_{11}a_{22} - a_{12}^2}$$

$$C_1 = \frac{\sum X_i^2 \cdot \sum Y_i - \sum X_i \cdot \sum (Y_i\ X_i)}{NPNTSI \cdot \sum X_i^2 - (\sum X_i)^2}$$

$$C_2 = \frac{NPNTSI \cdot \sum (Y_i \cdot X_i) - \sum X_i \cdot \sum Y_i}{NPNTSI \cdot \sum X_i^2 - (\sum X_i)^2}$$

REFERENCES

1. Shreeve, R.P., Simmons, J.M., Winters, K.A. and West, J.C. Jr., "Determination of Transonic Compressor Flow Field by Synchronized Sampling of Stationary Fast Response Transducers", Symposium on Non-Steady Fluid Dynamics, ASME 1978 Winter Annual Meeting, San Francisco, Ca., Dec. 1978.

2. Shreeve, R.P., McGuire, A.G. and Hammer, J.A., "Calibration of a Two Probe Synchronized Sampling Technique for Measuring Flows Behind Rotors", ICIASF '79 Record, IEEE Cat. No. 79CH1500-8AES, Proceedings of the International Congress on Instrumentation in Aerospace Simulation Facilities held at Monterey, Ca., Sept. 24th - 26th, 1979.

3. McCarville, P.A., "Hardware and Software Improvements to a Paced Data Acquisition System for Turbomachines", Naval Postgraduate School Master's Thesis, June 1981.

4. Adler, D. and Taylor, P.M., "A Procedure for Obtaining Velocity Vector from Two High Response Impact Pressure Probes", Naval Postgraduate School Technical Report NPS67-80-007, August 1980.

5. Shreeve, R.P., "Calibration of Flow Nozzles Using Traversing Pitot-Static Probes", Naval Postgraduate School, Department of Aeronautics, NPS-57Sf3071A, Monterey, Ca., July 1973.

6. Shreeve, R.P., "Report on the Testing of a Hybrid (Radial to Axial) Compressor," Naval Postgraduate School, Department of Aeronautics, NPS-57Sf3112A, Monterey, Ca., November 1973.

7. Sharma, S., "Transonic Compressor Steady State Data Reduction", Naval Postgraduate School, Turbopropulsion Laboratory, Technical Note 79-03, Monterey, Ca., September 1979.

DISTRIBUTION LIST

		No. of Copies
1.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	4
2.	Office of Research Administration Code 012A Naval Postgraduate School Monterey, California 93940	1
3.	Chairman Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940	1
4.	Director, Turbopropulsion Laboratory Department of Aeronautics Naval Postgraduate School Monterey, California 93940	30
5.	Dr. Gerhard Heiche Naval Air Systems Command Code AIR-310 Navy Department Washington, D.C. 20360	1
6.	Mr. Karl H. Guttmann Naval Air Systems Command Code AIR-330 Navy Department Washington, D.C. 20360	1
7.	Dr. A. D. Wood Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217	1
8.	Commanding Officer Naval Air Propulsion Test Center Attn: Mr. Vernon Lubosky Trenton, New Jersey 08628	1

9. National Aeronautics & Space Administration — 1
Lewis Research Center (Library)
2100 Brookpark Road
Cleveland, Ohio 44135

10. CAG Library — 1
The Boeing Company
Seattle, Washington 98124

11. Library — 1
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215

12. Library — 1
Pratt and Whitney Aircraft
Post Office Box 2691
West Palm Beach, Florida 33402

13. Library — 1
Pratt and Whitney Aircraft
East Hartford, Connecticut 06108

14. Chief, Fan and Compressor Branch — 1
Mail Stop 5-9
NASA Lewis Research Center
2100 Brookpark Road
Cleveland, Ohio 44135

15. Prof. D. Adler — 1
Technion Israel Institute of Technology
Department of Mechanical Engineering
Haifa 32000
ISRAEL

16. Director, Whittle Laboratory — 1
Department of Engineering
Cambridge University
ENGLAND

17. Prof. F. A. E. Breugelmans — 1
Institut von Karman de la Dynamique
des Fluides
72 Chausee de Waterloo
1640 Rhode-St. Genese
BELGIUM

18. Library — 1
Air Research Mfg. Corporation
Division of Garrett Corporation
402 South 36th Street
Phoenix, Arizona 85034

19. Prof. Jacques Chauvin 1
Universite D'Aix-Marseille
1 Rue Honnorat
Marseille, FRANCE

20. Mr. James V. Davis 1
Teledyne CAE
1330 Laskey Road
Toledo, Ohio 43601

21. Dr. Robert P. Dring 1
United Technologies Research Labs
400 Main Street
Hartford, Connecticut 06108

22. Mr. Jean Fabri 1
ONERA
29, Ave. de la Division Leclerc
92 Chatillon
FRANCE

23. Prof. Dr. Ing Heinz E. Gallus 1
Lehrstuhl und Institut fur Strahlantiebe
und Turbourbeitsmashinen
Rhein.-Westf. Techn. Hochschule Aachen
Templergraben 55
5100 Aachen, WEST GERMANY

24. Professor J. P. Gostelow 1
School of Mechanical Engineering
The New South Wales Institute of Technology
AUSTRALIA

25. Dr. Ing. Hans-J. Heinemann 1
DFVLR-AVA
Bunsenstrasse 10
3400 Gottingen, WEST GERMANY

26. Professor Ch. Hirsch 1
Vrije Universiteit Brussel
Pleinlaan 2
1050 Brussels, BELGIUM

27. Chairman 1
Aeronautics and Astronautics Department
31-265 Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

28. Dr. B. Lakshminarayana 1
Professor of Aerospace Engineering
The Pennsylvania State University
233 Hammond Building
University Park, Pennsylvania 16802

29. Mr. R. A. Langworthy — 1
Army Aviation Material Laboratories
Department of the Army
Fort Eustis, Virginia 23604

30. Prof. Dr. L. G. Napolitano — 1
Director, Institute of Aerodynamics
University of Naples
Viale C. Augusto
80125 Napoli
ITALY

31. Prof. Erik Nilsson — 1
Institutionen for Stromningsmaskinteknik
Chalmers Tekniska Hogskola
Fack, 402 20 Goteborg 5
SWEDEN

32. Prof. Gordon C. Oates — 1
Department of Aeronautics and Astronautics
University of Washington
Seattle, Washington 98105

33. Prof. Walter F. O'Brian — 1
Mechanical Engineering Department
Virginia Polytechnic Institute and
State University
Blacksburg, Virginia 24061

34. Dr. P. A. Paranjee — 1
Head, Propulsion Division
National Aeronautics Laboratory
Post Bag 1799
Bangalore - 17, INDIA

35. R. E. Peacock, Code 67Pc — 1
Department of Aeronautics
Naval Postgraduate School
Monterey, California 93940

36. Dr. W. Schlachter — 1
Brown, Boveri-Sulzer Turbomachinery Ltd.
Dept. TDE
Escher Wyss Platz
CH-8023 Zurich, SWITZERLAND

37. Prof. T. H. Okiishi — 1
Professor of Mechanical Engineering
208 Mechanical Engineering Building
Iowa State University
Ames, Iowa 50011

38. Dr. Fernando Sisto 1
Professor and Head of Mechanical
Engineering Department
Stevens Institute of Technology
Castle Point, Hoboken, New Jersey 07030

39. Dr. Leroy H. Smith, Jr. 1
Manager, Compressor and Fan
Technology Operation
General Electric Company
Aircraft Engine Technology Division
DTO Mail Drop H43
Cincinnati, Ohio 45215

40. Dr. W. Tabakoff 1
Professor, Department of Aerospace
Engineering
University of Cincinnati
Cincinnati, Ohio 45221

41. Mr. P. Tramm 1
Manager, Research Labs
Detroit Diesel Allison Division
General Motors
P.O. Box 894
Indianapolis, Indiana 46206

42. Prof. Dr. W. Traupel 1
Institut fur Thermische Turbomaschinen
Eidg. Technische Hochschule
Sonneggstrasse 3
8006 Zurich, SWITZERLAND

43. Dr. Arthur J. Wennerstrom 1
ARL/LF
Wright-Patterson AFB
Dayton, Ohio 45433

44. Dr. H. Weyer 1
DFVLR
Linder Hohe
505 Porz-Wahn
WEST GERMANY

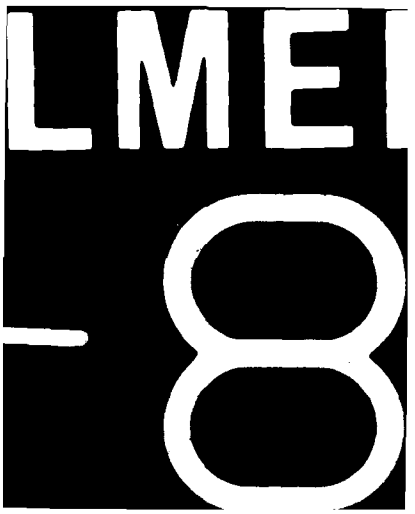
45. Mr. P. F. Yaggy 1
Director
U.S. Army Aeronautical Research Laboratory
AMES Research Center
Moffett Field, California 94035

46. Prof. C. H. Wu 1
P.O. Box 2706
Beijing 100080
CHINA

47. Director 1
Gas Turbine Establishment
P.O. Box 305
Jiangyou County
Sichuan Province
CHINA

48. Professor Leonhard Fottner 1
Department of Aeronautics and Astronautics
German Armed Forces University
Hochschule des Bundeswehr
Werner Heisenbergweg 39
8014 Neubiberg near Munich
WEST GERMANY

49. BDM Corporation 5
P.O. Box 2019
Monterey, Calaifornia 93940


LME
8