

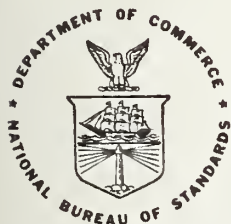
NBSIR 77-1383

Interactive Computer Program for the Determination of Reverberation Time

Thomas W. Bartel

Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

December 1977



U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NBSIR 77-1383

**INTERACTIVE COMPUTER PROGRAM
FOR THE DETERMINATION OF
REVERBERATION TIME**

Thomas W. Bartel

Institute for Basic Standards
National Bureau of Standards
Washington, D.C. 20234

December 1977

U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, *Secretary*

Dr. Sidney Harman, *Under Secretary*

Jordan J. Baruch, *Assistant Secretary for Science and Technology*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Acting Director*

ABSTRACT

A description of the computer program used to measure the reverberation time in a reverberation room is presented. The program controls the operation of a real-time analyzer, a random noise generator, and a microphone multiplexer. The reverberation time for each digitally recorded decay curve is determined from a straight line least-squares fit. The program is written in FORTRAN V and requires approximately 35,000 eight-bit bytes of core memory. Flow charts, source listings, and sample printouts are included.

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. INSTRUMENTATION	1
III. DIGITAL PROCEDURE TO DETERMINE THE REVERBERATION TIME . . .	2
IV. PROGRAM SEQUENCE	4
V. SOFTWARE DETAILS	7
VI. INPUT/OUTPUT LOGICAL UNITS	8
APPENDIX A - Abbreviated Flow Chart of Reverberation Time Program	14
APPENDIX B - Program REVERB Flow Chart, Terminal Messages, and Listings	18
APPENDIX C - Subroutine SIGNAL Flow Chart, Terminal Messages, Printouts, and Listings	31
APPENDIX D - Subroutine CURVFT Flow Chart, Terminal Messages, Printout, and Listings	48
APPENDIX E - Subroutine DSDATA Flow Chart, Terminal Messages, Printouts, and Listings	59
APPENDIX F - Subroutine AVRGE Flow Chart, Terminal Messages, Printout, and Listings	68
APPENDIX G - Subroutine RESULTS Flow Chart, Terminal Messages, Printouts and Listings	74
APPENDIX H - Subroutine HUMID Flow Chart, Terminal Messages, and Listings	84
APPENDIX I - Subroutine SAMPLE Flow Chart and Listings	91
APPENDIX J - Subroutine BAND Flow Chart and Listings	94
APPENDIX K - Subroutine SUBMUX Flow Chart and Listings	97
APPENDIX L - Listings for Function Subprograms TINORM and STUDIM	100
APPENDIX M - Listings for Assembly Language Subroutines MUX, RTA, SOUND, and DELAY	103
APPENDIX N - Listings for Program GEN1 and its Function Subprograms STUDIN, FISHIN, and FISH	111

LIST OF TABLES AND FIGURES

	Page
TABLE 1. Reverberation Time Program Elements	10
FIGURE 1. Simplified Schematic of Instrumentation System Used to Determine the Reverberation Time	12
FIGURE 2. Typical Plot of a Digitally-Obtained Decay Curve and the Definitions Used to Determine the Reverberation Time	13

I. INTRODUCTION

This report describes the computer program used to measure the reverberation time in a reverberation room according to the Standard Method of Test for Sound Absorption of Acoustical Materials in Reverberation Rooms, ASTM C423-66. The program is written to control the instrumentation system used for the NBS reverberation room. It should, however, be suitable for use in any laboratory with a similar instrumentation system.

In this program, the reverberation decays are digitally recorded simultaneously in the 30 1/3-octave bands from 25 to 20 000 Hz. (The program allows the operator to limit the processing of the data to a narrower frequency range, which may be appropriate for the characteristics of the reverberation room.) Typically 20 s are allowed for sampling the 30 filtered decaying sounds and 15 s for the sound field to reach steady state. (The sound buildup times are typically less than 2.5 s.) During the 15-second buildup the decay rates of the preceding decay are computed from a least-squares curve fit in each of the 1/3-octave bands in the selected frequency range. This procedure permits the decay rates to be recorded and processed as fast as the decays can be physically performed, about 100 decays per hour. The program allows a detailed examination of the decay curves to determine whether or not they can be approximated by single straight lines.

II. INSTRUMENTATION

The program assumes that several elements of the instrumentation system are under computer control. These elements, shown in Fig. 1, are a real-time analyzer, a random noise generator, and a microphone multiplexer. The real-time analyzer filters the input signal through 30 contiguous

1/3-octave band-pass filter channels, samples each channel simultaneously for an integration period, determines the rms level for each channel, and transmits the digital results to the computer. For the NBS system, an integration time of 0.1 s is used for reverberation time measurements. The time required to transmit the digitized output level from all 30 1/3-octave filters to the computer is approximately 1 ms.

The output of the random noise generator is turned on and off by the computer. For most measurements made with the NBS system, unfiltered pink noise is sent to the amplifier and loudspeaker, resulting in sufficient dynamic range to measure the reverberation time simultaneously in the 1/3-octave bands from 80 Hz to 10 kHz.

The computer controls the channel selection of a microphone multiplexer. In the NBS system, the multiplexer has 16 channels, of which 12 are used with the 12-microphone array installed in the reverberation room. In the reverberation time program, the multiplexer is used, at the operator's option, to select the next microphone channel after each decay.

III. DIGITAL PROCEDURE TO DETERMINE THE REVERBERATION TIME

The reverberation time for each decay curve is determined from a straight line least-squares fit to the digitized output of each 1/3-octave filter in the following manner. First the digitized output of each filter is converted by the real-time analyzer to an rms value using an averaging time of 0.1 s. This rms value for each filter is read by the computer in less than 1.0 ms and another 0.1-second averaged rms value is obtained, and so on. These rms values are depicted by the solid circles in Fig. 2.

Before the actual decay curve can be defined two steady-state values for each 1/3-octave band are determined, the long-term average signal level prior to the abrupt cessation of the sound in the room and the long-term average system noise level (both acoustical and electrical) of each microphone. The average system noise level in each 1/3-octave band is determined by sampling the signal when the sound source is turned off. This is denoted the average noise level in Fig. 2. The averaging time, determined by the number of consecutive 0.1-s samples taken, is chosen by the operator. These noise levels are assumed to remain constant throughout the entire set of measurements. The long-term average signal level, denoted the average signal level in Fig. 2, is obtained just prior to each decay by sampling the signal for a short time before the sound source is turned off. The operator chooses the number of consecutive 0.1-s samples taken before and after the sound cutoff (maximum total time of 20 s). Typical values are 4 s and 16 s, respectively.

Since the point at which each decay curve starts is often poorly defined, the upper limit for the least-squares curve fitting procedure is arbitrarily set x dB below the average signal level. A typical value for x is 5 dB. Similarly the effects of the system noise must not influence the least squares fit and, therefore, the lower limit is arbitrarily set y dB above the average noise level. A typical value for y is 10 dB. Both x and y are chosen by the operator. The limits for the curve fitting procedure are denoted the upper and lower amplitude limits in Fig. 2.

Since the distribution of the 0.1-second averaged rms values about the fitted line defining the decay curve can be relatively large, an iterative procedure is used to determine the first and last data points to

be used by the least squares procedure. The initial decay curve is determined from a least squares fit starting with the first point after the sound has been turned off and ending with the first point to reach the noise level. The intersection of this initial decay curve with the upper and lower amplitude limits is then determined. The data points nearest these intersections are chosen as the new upper and lower points for the curve fitting procedure and a new decay curve is determined. This process is repeated until the intersections of the upper and lower amplitude limits with the decay curve are within 0.1 s of the points previously used to determine the decay curve. This procedure generally requires fewer than five iterations and the decay times for all 22 1/3-octave bands from 80 and 10 000 Hz can be determined in less than 15 s.

When one wants to determine if a decay curve contains a "break point" resulting from two distinct decay times, the program may first be run with the value of y increased such that the decay curve is computed using data that only span 15 or 20 dB. Then the program may be rerun with the value of x increased to 15 or 20 dB and the value of y decreased to its original value. If these two measurements result in statistically distinguishable reverberation times in any 1/3-octave band, then the room has two distinct decay times in that band.

IV. PROGRAM SEQUENCE

The manner in which the reverberation time measurement is performed is summarized in the abbreviated flowchart given in Appendix A. At the start of the program the operator enters the desired confidence interval and level of the average reverberation times, the 1/3-octave frequency range, and the time between samples, which is approximately 0.1 s but is accurately determined in advance with a time-interval counter.

The average signal and ambient noise levels are then determined. This section of the program enables the operator to adjust the loudspeaker amplifier, microphone and real-time analyzer gains to achieve maximum dynamic range while remaining below the signal level overload region of the real-time analyzer (see Printout 1, Appendix C). Note that the signal and noise levels in the printouts are the averages of several successive 0.1-s samples obtained from the real-time analyzer. In the NBS system, the resolution of the real-time analyzer is 0.25 dB. The levels are printed to the nearest 0.01 dB in order to aid in tracing the computations performed on the data, rather than to represent the uncertainty of the data.

To allow for different noise levels among the microphone channels used in the measurement, the operator may conduct a separate ambient noise measurement for each channel (see Printout 2, Appendix C). After these adjustments the ambient noise levels are obtained and stored for determining the lower curve fitting limits of the decay curves.

The operator then enters the number of samples to be taken before and after the sound is cut off, and the values x and y , discussed in the previous section, to be used in determining the curve-fitting limits. One decay is sampled upon initiation by the operator, who may observe the curve-fitting parameters (see Printout 1, Appendix D) to ensure that the measurement is proceeding correctly. Following this the operator makes a decision of "YES," "NO," or "AUTO" in response to the question "Do you want to conduct another decay?" The results of these responses are as follows:

YES: Conduct and analyze one more decay. Operator controls the microphone selection and initiates the sampling of the decay.

AUTO: Conduct and analyze several decays automatically. Operator first chooses the number of decays to be taken, then the operation of the microphone multiplexer (step or not step between decays), and finally the insertion of any delay time for sound buildup that is desired in addition to the time required to compute the reverberation times, which is a maximum of 15 s but shorter if a narrower frequency range has been chosen.

NO: Do not conduct any more decays at this point.

If the operator entered "NO" to the above question, he may then examine the decay curves of the last decay to inspect their linearity and the superposition of the computed least-squares fit. Such a display is shown in Printouts 1 and 2, Appendix E. Printout 1 displays, in tabular form, the level transmitted by the real-time analyzer for each 0.1-s integration period for a specified 1/3-octave band. The levels are printed in horizontal rows of 10. Printout 2 is a graphical display of the same data. Following this, he may request a display of the average reverberation time for each 1/3-octave frequency band of all of the decays conducted up to this point (see Printout 1, Appendix F). Also displayed are the confidence intervals achieved at this point, and the estimated number of decays remaining in each band to attain the confidence interval specified at the beginning of the measurement.

The operator may then choose to conduct more decays using the same "YES," "NO," or "AUTO" choices described previously. If "NO" is entered, the measurement is terminated following a printout of the results (see Printouts 1 and 2, Appendix G).

V. SOFTWARE DETAILS

The programming language used is the Interdata FORTRAN V Level 1 Software System.* It is a superset of ANSI Standard X3.9-1966 FORTRAN and should be compatible or nearly compatible with the FORTRAN V used on other computer systems. Users of this program on other systems should check the compatibility of FORMAT statements using the A Format and INTEGER*2 statements which define the storage size of integers to be two 8-bit bytes.

The program for measuring reverberation time consists of a main program, REVERB, written in FORTRAN V, several FORTRAN subroutines, and several assembly language subroutines written for use on an Interdata Model 70 minicomputer. These elements are listed in Table 1. The corresponding flow charts, sample printouts, and listings are given in the appendices. The last appendix, Appendix N, gives the listings of a separate FORTRAN program, GENI, and three of its subprograms, STUDIN, FISHIN, and FISH, that are not a part of the program REVERB. They are used to generate the t-distribution table read from logical unit 4 by the function subprogram STUDIM, called by REVERB.

*The commercial computer products utilized are identified in order to adequately describe the program elements discussed in this report. In no case does such identification imply recommendation or endorsement by the National Bureau of Standards, nor does it imply that these products are necessarily the best available for the purpose.

In the NBS system, the subroutines SIGNAL, CURVFT, DSDATA, AVRGE, and RESULTS are loaded as overlays called by the main program, REVERB. Each call to these subroutines is therefore preceded by the instructions REWIND 0 and CALL IFETCH, which cause the overlay to be loaded. In addition, there is a branch to "End of Job" if there is an overlay load error. These instructions should be deleted or bypassed if this program is used in a system not requiring overlays. The subroutines will then be loaded and called as regular subroutines.

In the NBS system the complete task occupies 35,452 eight-bit bytes of memory, of which 12,000 bytes are used for storing the real-time analyzer data for each decay (200 samples/1/3-octave band x 30 bands = 6000 samples, with each sample being stored as a 2-byte integer).

The flow charts, given in the appendices, conform to the American National Standard Flowchart Symbols and Their Usage in Information Processing, X3.5-1970. In these flow charts, the small numbers enclosed in parentheses refer to FORTRAN statement labels in the program listings.

In the program listings, the first three statements of every program or subprogram are control statements used to insert the program name, in the form of a binary label, into the compiled object code.

VI. INPUT/OUTPUT LOGICAL UNITS

The following logical units are used for input/output operations:

- LU0 - File from which overlays are loaded (see Section V).
- LU1 - Scratch file for accumulating a table of the reverberation times for each decay; this file is read when computing the average reverberation times.
- LU2 - Output file for storing the final table of averaged reverberation times; can be used as input to other programs, such as the sound power program.

- LU3 - Hard copy printout device. Should be assigned to a 132 character line printer. During execution of the program, the operator may choose the printout device logical unit to obtain either a hard copy printout or a view of the results on the CRT terminal.
- LU4 - File containing a table of the t-distribution for the chosen confidence level. This table is used by the function subprogram STUDIM, and must be created by another program, GEN1, given in Appendix N. This file must be a random record access file.
- LU5 - Operator communication device. Should be assigned to a CRT terminal. Through this device the operator controls the course of the measurement, making the decisions that are called for and entering certain measurement parameters, which include:
1. The desired confidence interval and confidence level.
 2. The desired 1/3-octave frequency bands.
 3. The number of samples to be taken for each decay, both before and after the sound source is cutoff.
 4. The upper and lower curve-fitting limits.
 5. The number of decays to be conducted automatically.
 6. Whether or not to step the microphone multiplexer between decays.

Transmission of data to and from the real-time analyzer, microphone multiplexer, and random noise generator is not handled through logical units, but through separate subroutines written in the assembly language for the Interdata Model 70 minicomputer. These subroutines are included in TABLE 1; their listings are given in Appendix M.

TABLE 1. REVERBERATION TIME PROGRAM ELEMENTS

PROGRAM NAME	TYPE OF PROGRAM	FUNCTION	SUBROUTINES CALLED
REVERB	FORTRAN Program	Performs reverberation room decay measurements and computes reverberation time.	SIGNAL, CURVFT, DSDATA, AVRGE, RESULTS, SAMPLE, SOUND, DELAY, MUX, BAND, RTA, HUMID, STUDIM, TINORM, SUBMUX
SIGNAL	FORTRAN Subroutine	Determines signal-to-noise levels. Allows operator to adjust speaker amplifier gain and measurement system gains to achieve optimum signal-to-noise levels before beginning decay measurements.	SAMPLE, BAND, SOUND, SUBMUX, MUX, DELAY
CURVFT	FORTRAN Subroutine	Performs least-squares curve fit on decay data.	BAND
DSDATA	FORTRAN Subroutine	Displays real-time analyzer data for one decay, in both tabular form and as a graphic plot. The plot also shows the least-squares curve fit superimposed on the data points.	BAND
AVRGE	FORTRAN	Averages the results of many decays. Prints a table indicating, band by band, whether the desired confidence interval has been achieved. For those bands that do not achieve it, the approximate number of decays still necessary to achieve the desired confidence interval is indicated.	TINORM, STUDIM
RESULTS	FORTRAN Subroutine	Displays the results after final decay has been taken, which includes graphic plot of total room absorption versus frequency.	HUMID

TABLE 1. (Continued)

PROGRAM NAME	TYPE OF PROGRAM	FUNCTION	SUBROUTINE CALLED
HUMID	FORTRAN Subroutine	Computes relative humidity from wet and dry bulb temperatures. Also computes speed of sound.	None
SAMPLE	FORTRAN Subroutine	Takes a specified number of real-time analyzer samples and checks for signal levels that exceed its maximum input level.	RTA
BAND	FORTRAN Subroutine	Converts one band of real-time analyzer samples from fixed-point to floating-point numbers.	None
SUBMUX	FORTRAN Subroutine	Steps the microphone multiplexer to a given microphone channel.	MUX
TINORM	FORTRAN Function Subprogram	Computes value of the inverse of the standard normal distribution function for a specified probability.	None
STUDIM	FORTRAN Function Subprogram	Finds value of inverse of student's t-distribution for a specified argument by reading from a table generated on a storage file by the program GENI	None
MUX	Assembly Language Subroutine, FORTRAN Callable	Steps and reads multiplexer	None
RTA	Assembly Language Subroutine, FORTRAN Callable	Controls and reads real-time analyzer. Also shuts off sound after the specified number of samples have been taken.	None
SOUND	Assembly Language Subroutine, FORTRAN Callable	Turns random noise generator on and off.	None
DELAY	Assembly Language Subroutine, FORTRAN Callable	Provides a delay of 0 to 32.767 s in approximate 1-ms increments.	None

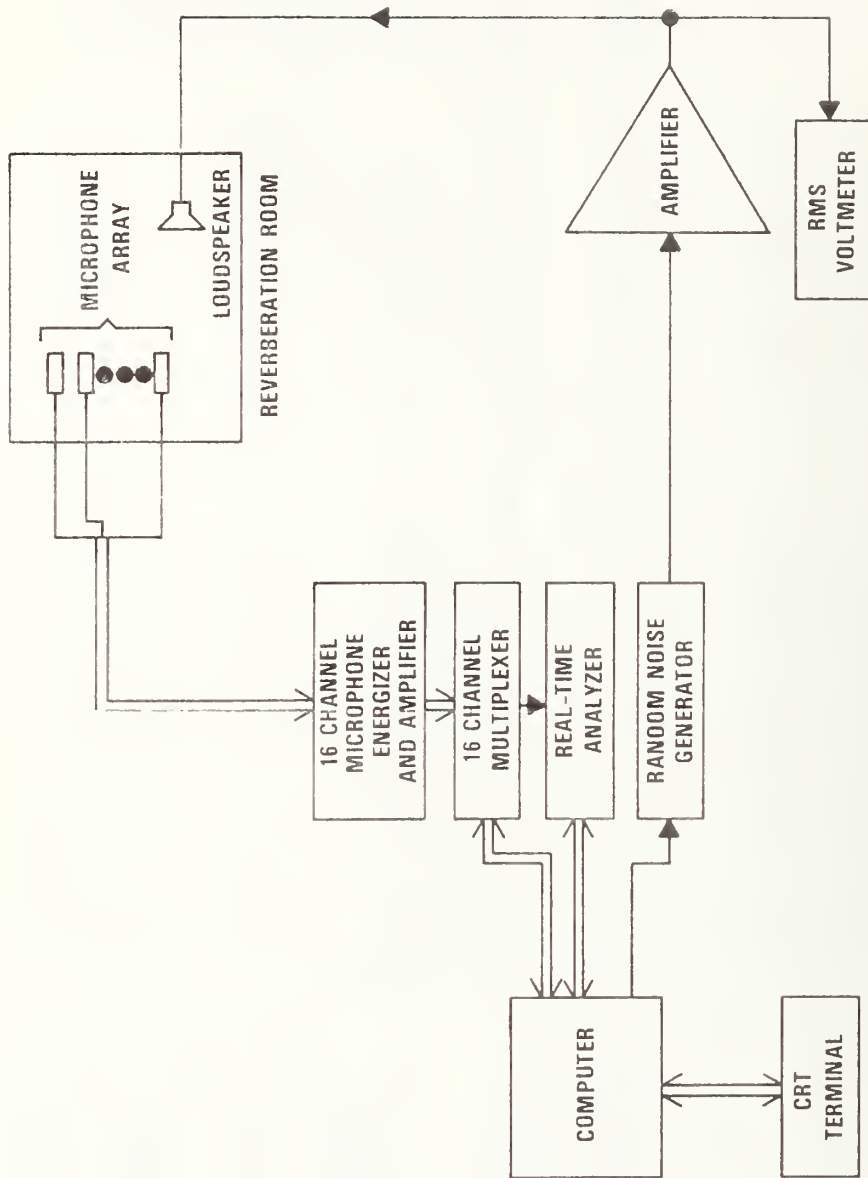


FIGURE 1. SIMPLIFIED SCHEMATIC OF INSTRUMENTATION SYSTEM USED TO DETERMINE THE REVERBERATION TIME.

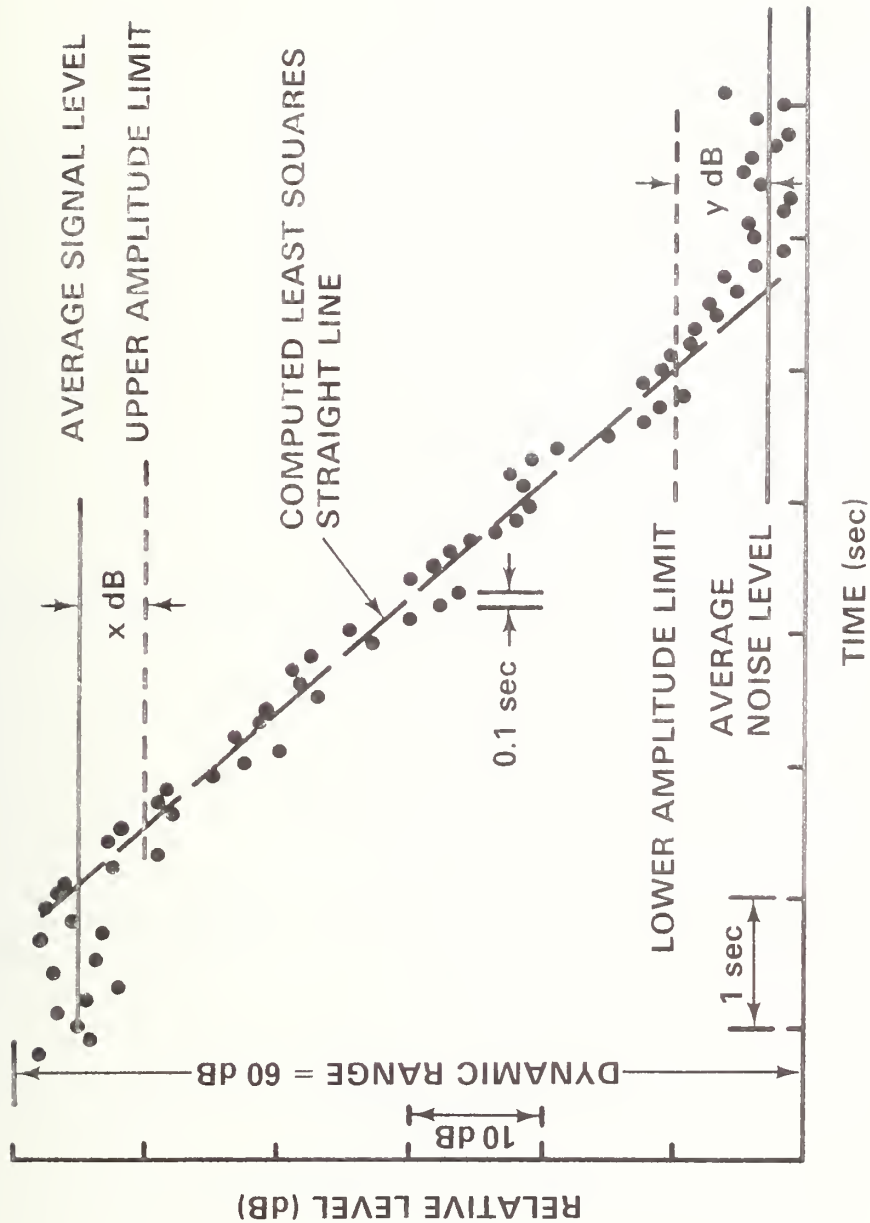
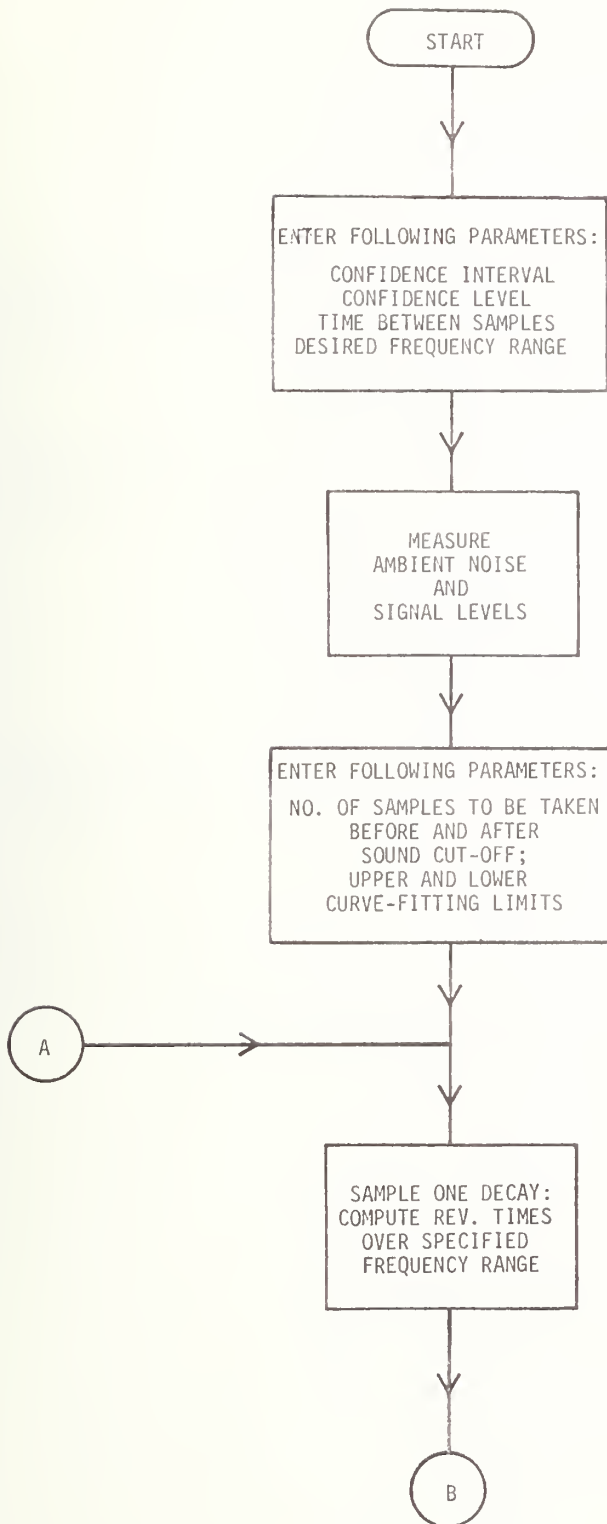


FIGURE 2. TYPICAL PLOT OF A DIGITALLY-OBTAINED DECAY CURVE AND THE DEFINITIONS USED TO DETERMINE THE REVERBERATION TIME.

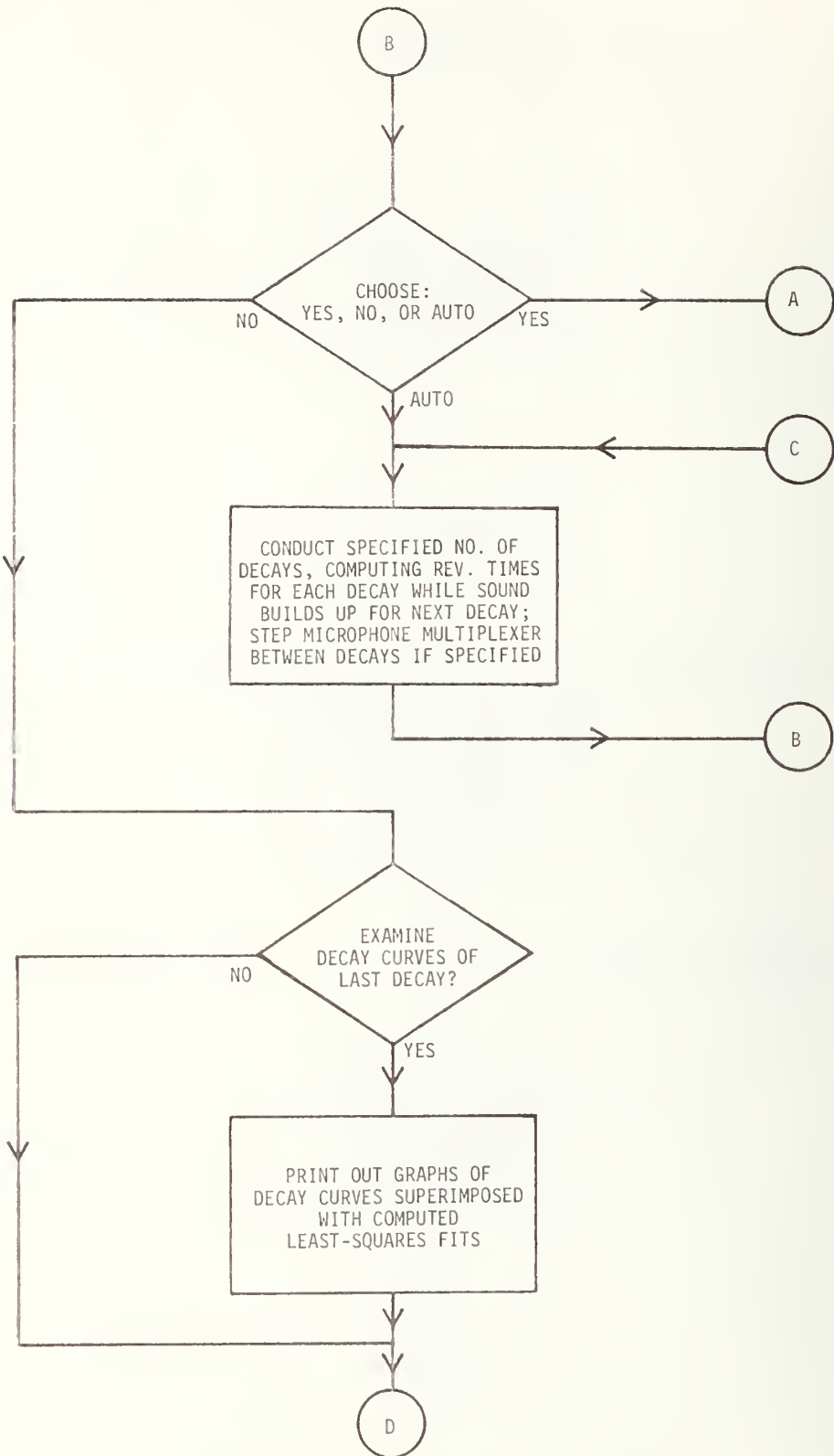
APPENDIX A

Abbreviated Flow Chart of Reverberation Time Program

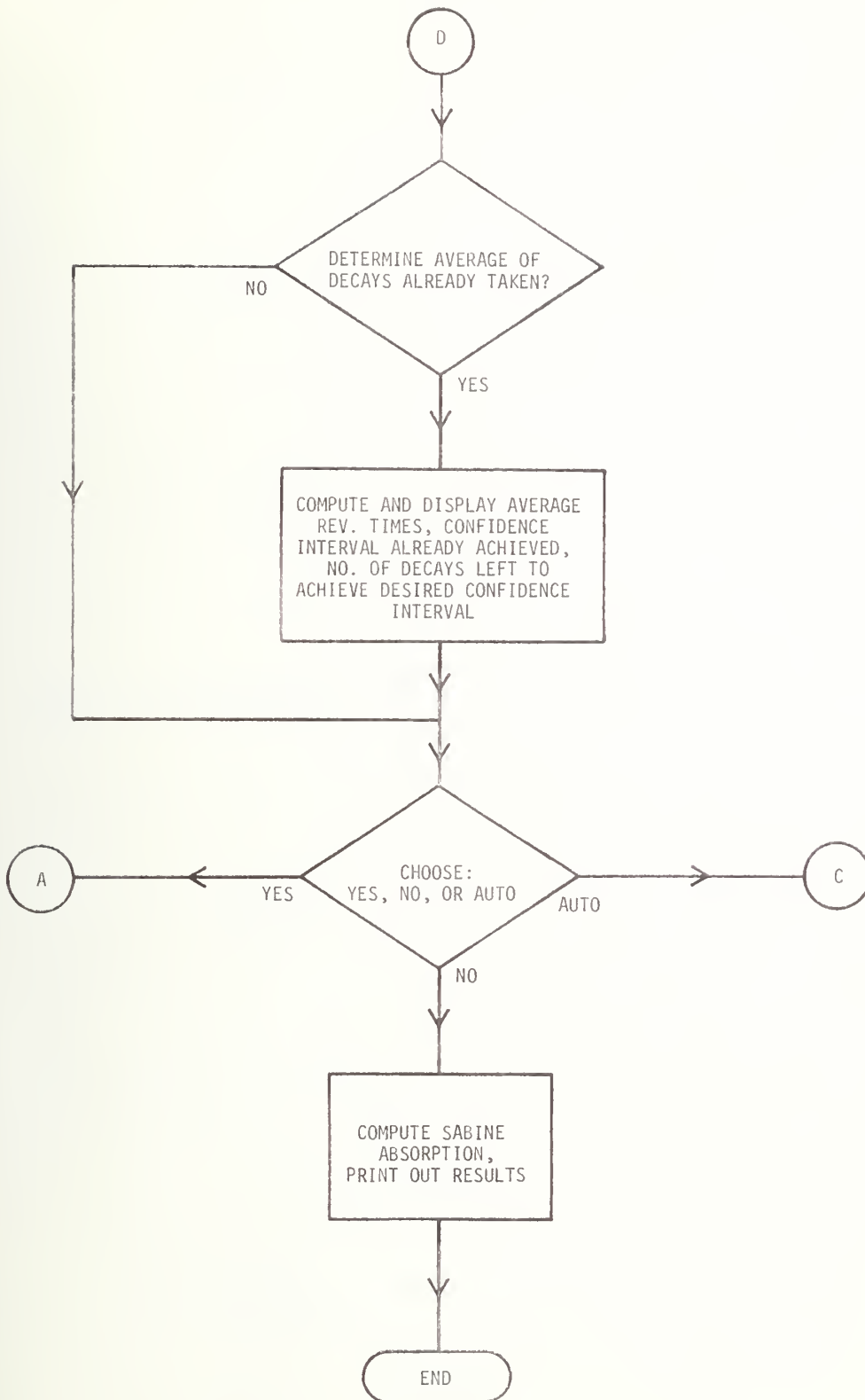
ABBREVIATED FLOW CHART: REVERB



ABBREVIATED REVERB-2



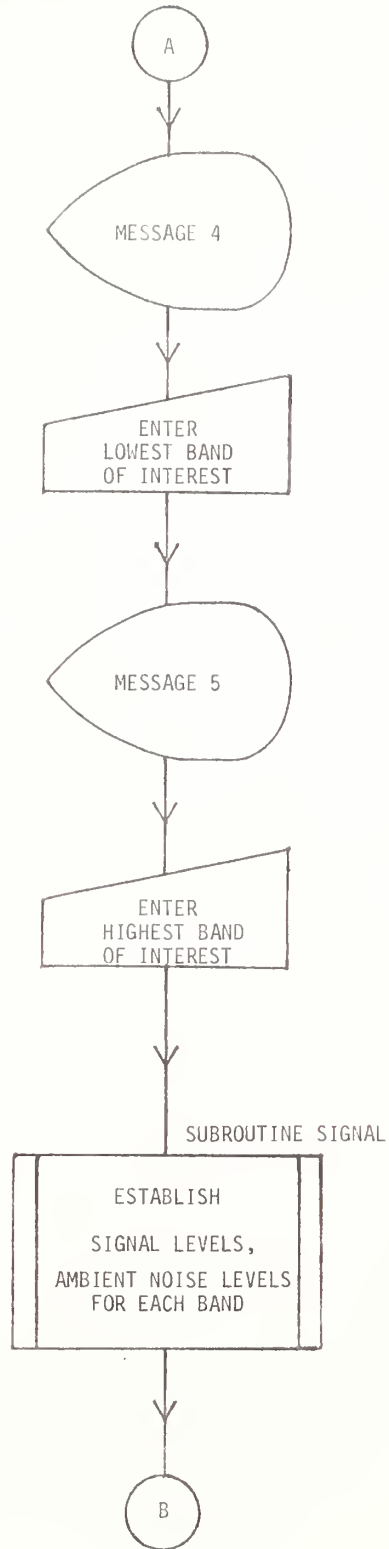
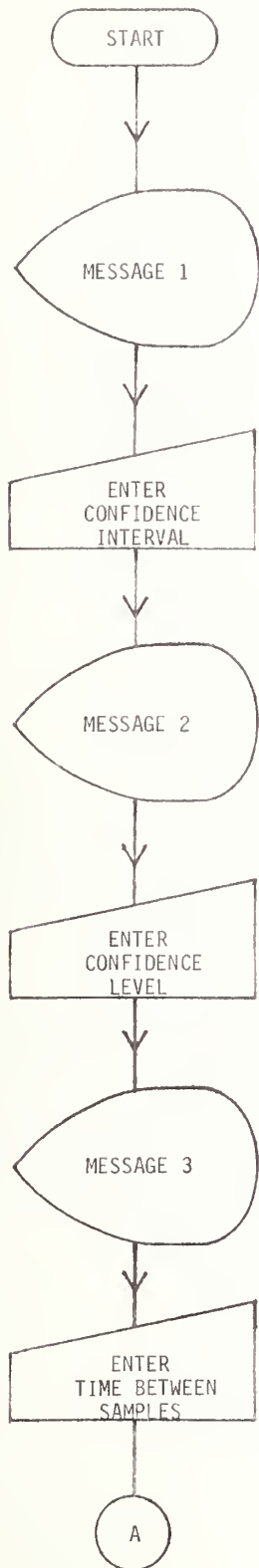
ABBREVIATED REVERB-3

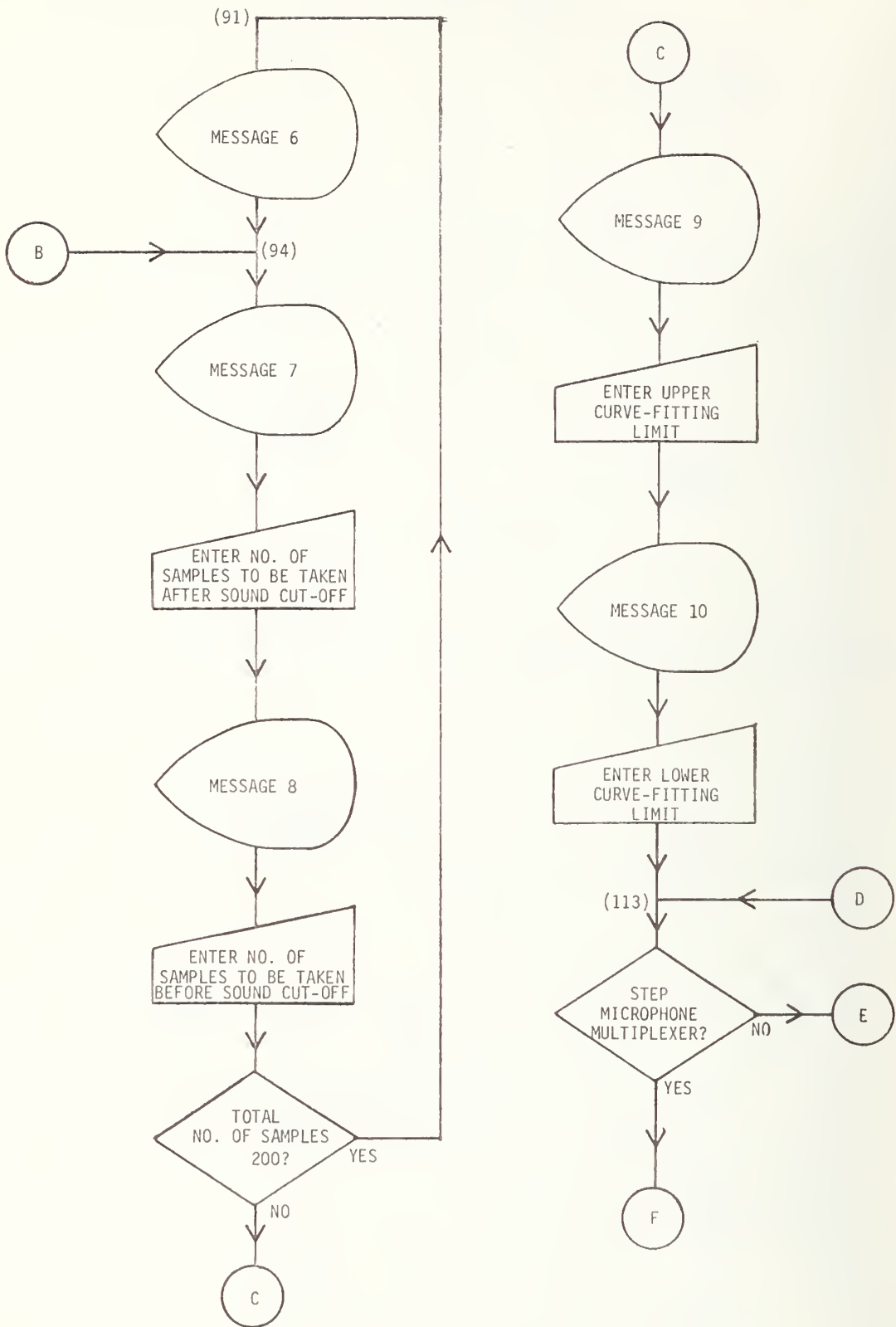


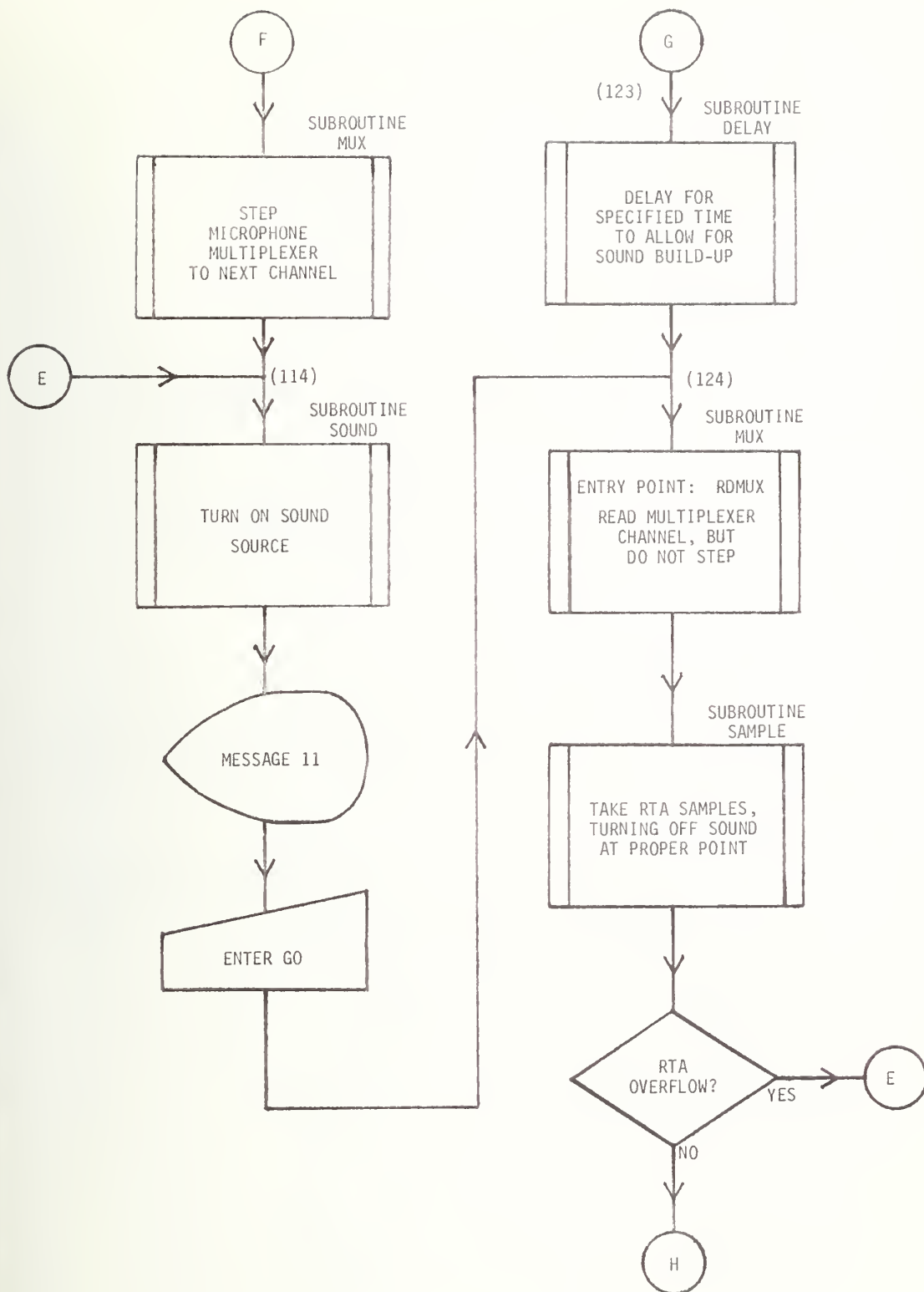
APPENDIX B

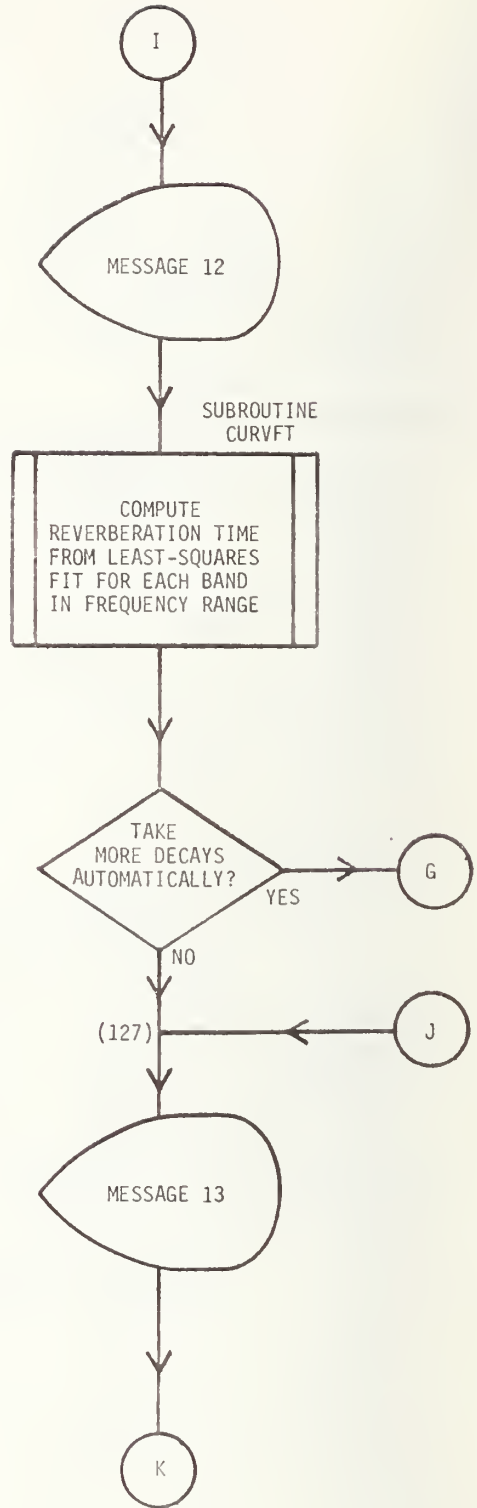
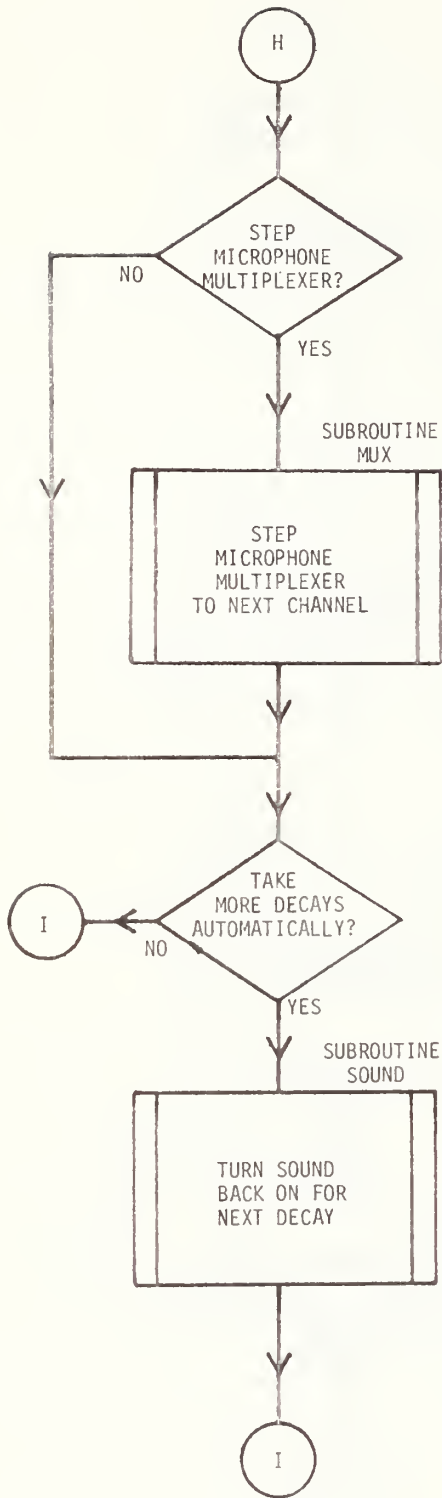
Program REVERB Flow Chart, Terminal Messages, and Listings

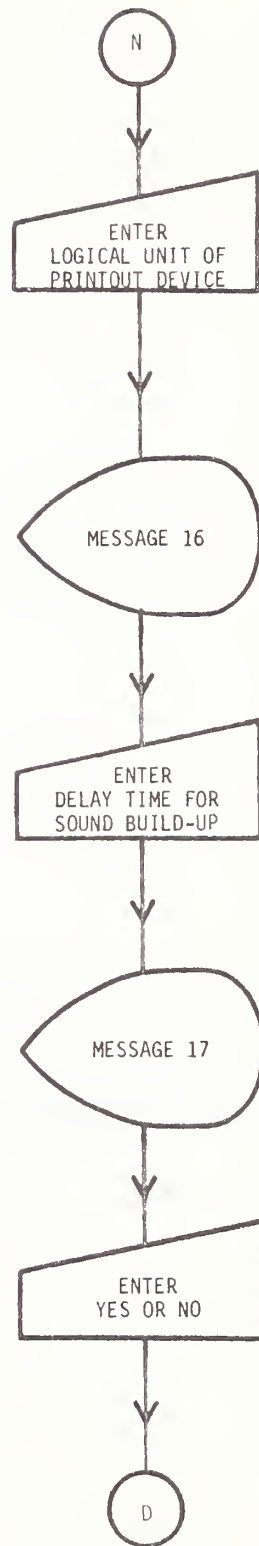
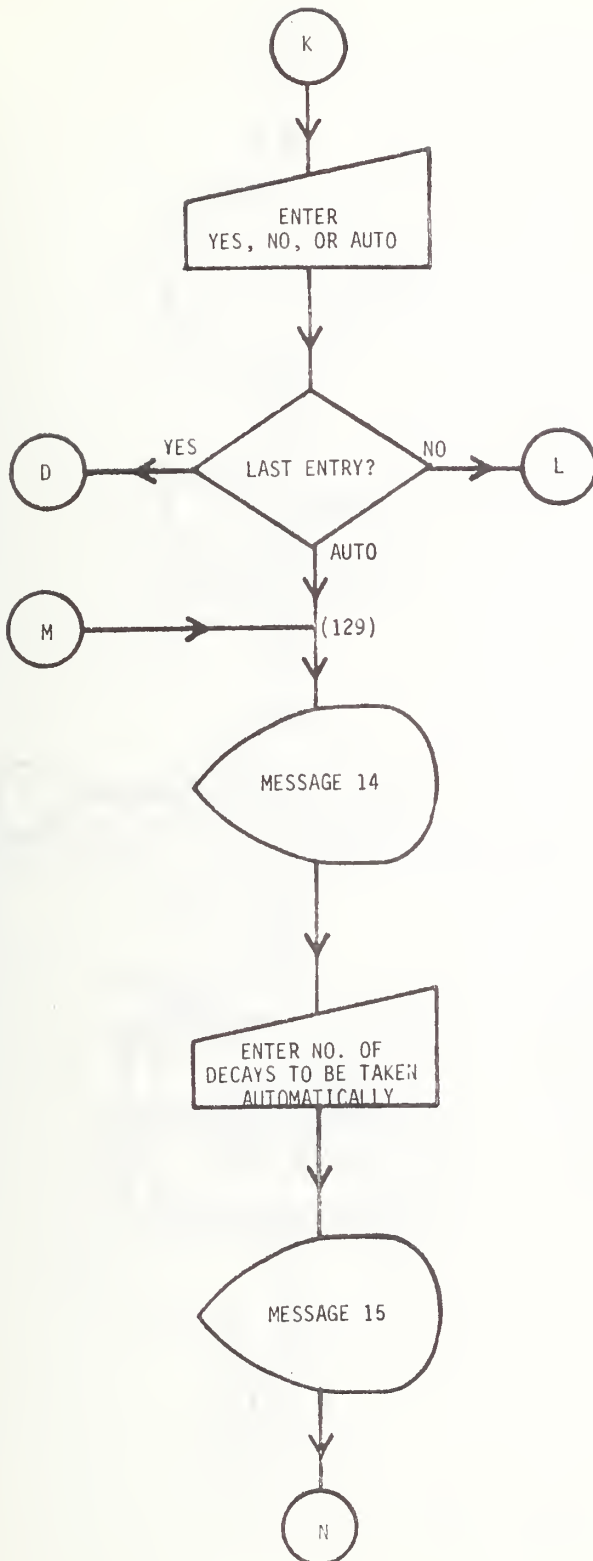
FLOW CHART: REVERB

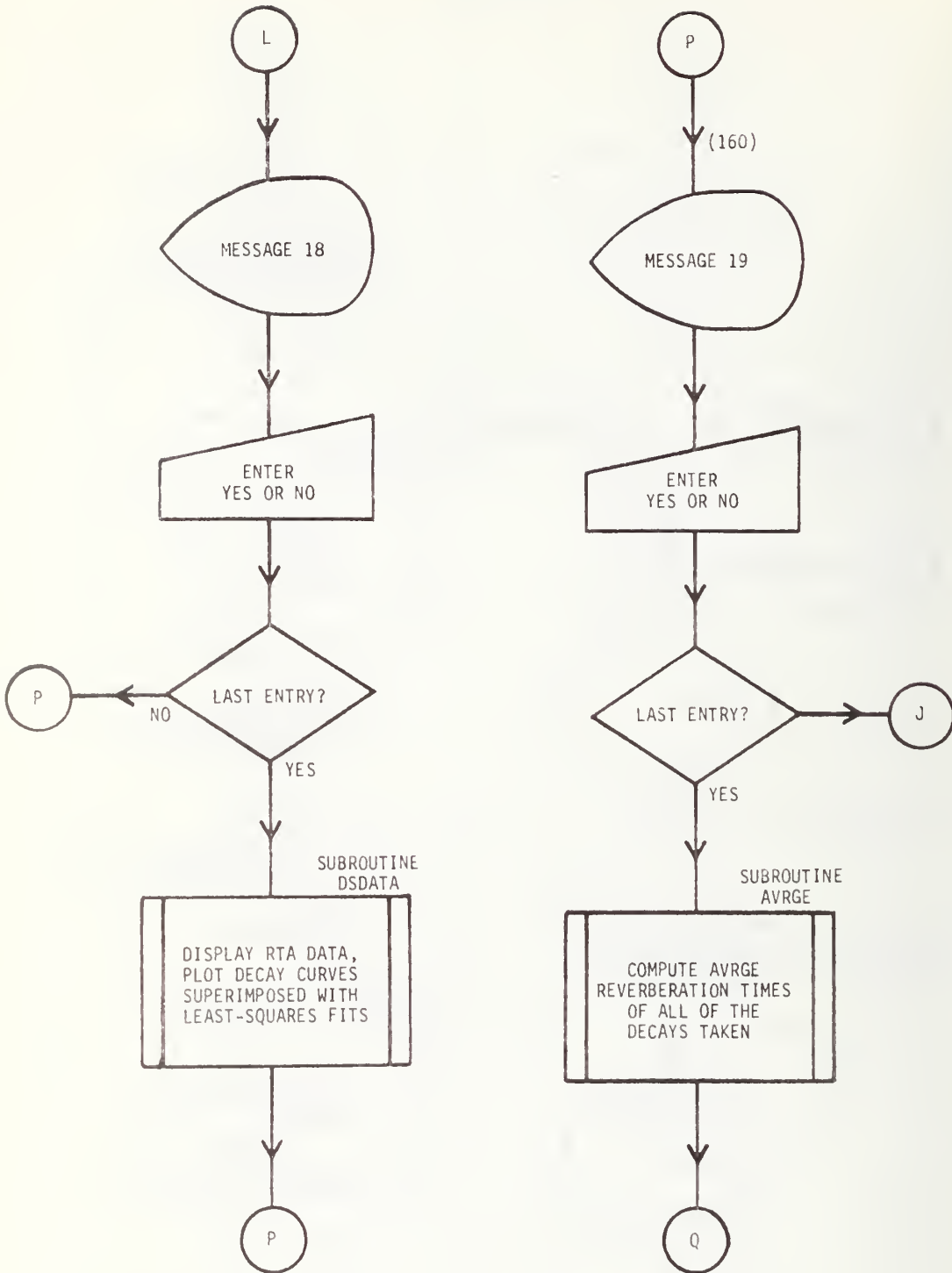


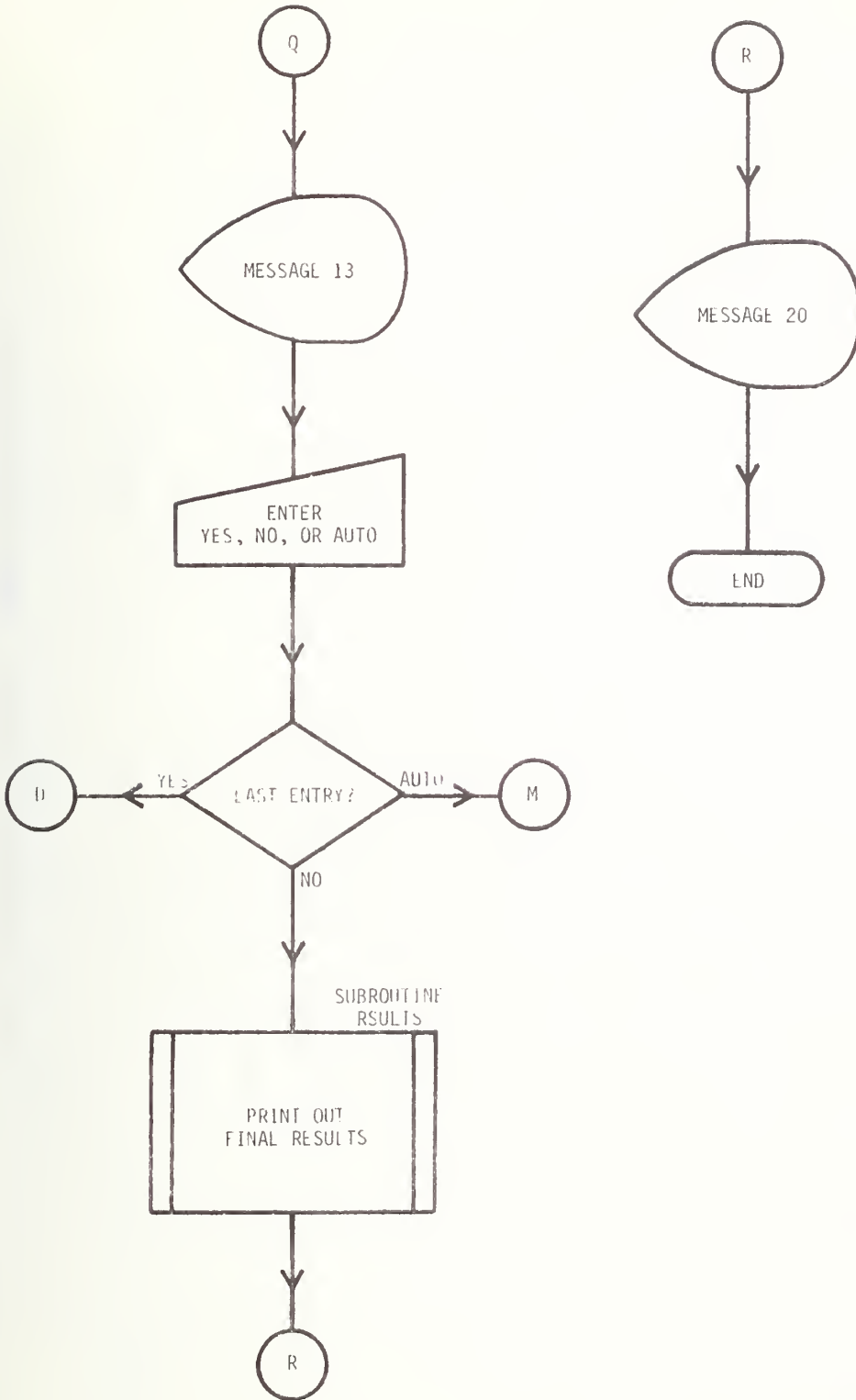












CRT TERMINAL MESSAGES: PROGRAM REVERB

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	<p>THIS PROGRAM RUNS AND ANALYZES REV ROOM DECAY MEASUREMENTS</p> <p>THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:</p> <p>LU0 -- FILE FROM WHICH OVERLAYS ARE LOADED LU1 -- SCRATCH FILE LU2 -- STORAGE FILE FOR STORING REVERBERATION TIME DATA LU3 -- HIGH SPEED PRINTER LU4 -- STORAGE FILE CONTAINING TABLE OF T-DISTRIBUTION LU5 -- CRT TERMINAL</p> <p>ENTER THE DESIRED CONFIDENCE INTERVAL TO BE ATTAINED IN TERMS OF PERCENT OF THE MEAN REV. TIME (X.X)</p> <p>ENTER THE DESIRED CONFIDENCE LEVEL IN PERCENT (XX.XX)</p> <p>ENTER THE TIME BETWEEN SAMPLES IN SECONDS (X.XXXXX)</p> <p>ENTER LOWEST BAND OF INTEREST (MIN=14)</p> <p>ENTER HIGHEST BAND OF INTEREST (MAX=43)</p> <p>YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES</p> <p>ENTER NO. OF SAMPLES REQUIRED FOR DECAY MEASUREMENT (3 DIGITS)</p> <p>ENTER NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF (3 DIGITS)</p>
2	30	
3	42	
4	45	
5	55	
6	92	
7	95	
8	100	

CRT TERMINAL MESSAGES: PROGRAM REVERB (page 2)

MESSAGE NO.	FORMAT NO.	MESSAGE
9	105	ENTER UPPER CURVE FITTING LIMIT IN DB DOWN FROM SIGNAL LEVEL (NORMALLY 5.0)
10	106	ENTER LOWER CURVE FITTING LIMIT IN DB UP FROM NOISE LEVEL (NORMALLY 10.0)
11	115	THE SOUND HAS BEEN TURNED ON ENTER "GO" TO START DECAY MEASUREMENT
12	125	DECAY NO. 1
13	128	DO YOU WANT TO CONDUCT ANOTHER DECAY? (ENTER "AUTO" FOR AUTOMATIC MODE)
14	130	ENTER NUMBER OF DECAYS TO BE CONDUCTED AUTOMATICALLY (4 DIGITS)
15	133	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-FITTING PARAMETERS
16	136	ENTER NO. OF SECONDS OF DELAY REQUIRED FOR SOUND BUILD-UP (XX.X) (MAX = 32) (THIS DELAY IS IN ADDITION TO THE COMPUTING AND PRINTOUT TIME)
17	138	DO YOU WANT TO STEP THE MICROPHONE MULTIPLEXER BEFORE EACH DECAY?
18	150	DO YOU WANT TO SEE THE DATA POINTS OF THE LAST DECAY?
19	170	DO YOU WANT TO SEE THE AVERAGES?
20	230	END OF JOB

PROGRAM REVERB

```
1 $ASSM
2 REVERB PROG REVERB - PROGRAM TO MEASURE REVERBERATION TIME
3 $FORT
4 C
5 C OVERLAYS CALLED: SIGNAL,CURVFT,DSDATA,AVRGE,RESULTS
6 C SUBPROGRAMS CALLED: SAMPLE,SOUND,DELAY,MUX,
7 C [INDIRECTLY]: HUMID,BAND,RTA,STUDIM,TINORM,SUBMUX
8 C
9 C -----
10 C INTEGER*2 B,NSIG,N,X0(6060),FREQ(30),LINE(121),
11 C 1BLOW,BHIGH,OVFLOW,AUTO,MM1(30),MM2(30)
12 C DIMENSION X(200),S0(30),S1(30),ARRAY1(30),ARRAY2(30)
13 C DATA FREQ/25,32,40,50,63,80,100,125,160,200,250,315,400,500,
14 C 1630,800,1000,1250,1600,2000,2500,3150,4000,5000,6300,8000,
15 C 210000,12500,16000,20000/
16 C WRITE (5,10)
17 C 10 FORMAT (XX/'THIS PROGRAM RUNS AND ANALYZES REV ROOM DECAY
18 C 1 MEASUREMENTS'/'XX/
19 C 2'THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:'/'XX/
20 C 35%.'LU0 -- FILE FROM WHICH OVERLAYS ARE LOADED'/'
21 C 45%.'LU1 -- SCRATCH FILE'/'
22 C 55%.'LU2 -- STORAGE FILE FOR STORING REVERBERATION TIME DATA'/'
23 C 65%.'LU3 -- HIGH SPEED PRINTER'/'
24 C 75%.'LU4 -- STORAGE FILE CONTAINING TABLE OF T-DISTRIBUTION'/'
25 C 85%.'LU5 -- CRT TERMINAL'/'XX/XX/
26 C 9'ENTER THE DESIRED CONFIDENCE INTERVAL TO BE ATTAINED
27 C 1 IN TERMS OF'/'PERCENT OF THE MEAN REV. TIME (X.XY)'
28 C READ (5,40) CINTER
29 C WRITE (5,30)
30 C 30 FORMAT ('ENTER THE DESIRED CONFIDENCE LEVEL IN PERCENT
31 C 1 (XX.XX)')
32 C READ (5,40) CLEVEL
33 C 40 FORMAT (F10.5)
34 C WRITE (5,42)
35 C 42 FORMAT ('ENTER THE TIME BETWEEN SAMPLES IN SECONDS (X.XXXXX)')
36 C READ (5,40) DELTA
37 C WRITE (5,45)
38 C 45 FORMAT ('ENTER LOWEST BAND OF INTEREST (MIN=14)')
39 C READ (5,50) BLOW
40 C 50 FORMAT (I2)
41 C WRITE (5,55)
42 C 55 FORMAT ('ENTER HIGHEST BAND OF INTEREST (MAX=43)')
43 C READ (5,50) BHIGH
44 C REWIND 0
45 C CALL IPETCH('SIGNAL',0,1STAT)
46 C IF (1STAT.NE.0) GO TO 200
47 C CALL SIGNAL (%,40,50,S1,FREQ,LINE,BLOW,BHIGH)
48 C GO TO 94
49 C 91 WRITE (5,92)
50 C 92 FORMAT ('YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES')
51 C 94 WRITE (5,95)
52 C 95 FORMAT ('ENTER NO. OF SAMPLES REQUIRED FOR DECAY MEASUREMENT',
53 C 14%,'(3 DIGITS)')
54 C READ (5,98)NDEC
55 C 98 FORMAT (I3)
56 C WRITE (5,100)
57 C 100 FORMAT ('ENTER NO. OF SAMPLES TO BE TAKEN BEFORE SOUND
58 C 1 CUT-OFF (3 DIGITS)')
59 C READ (5,98)NSIG
60 C N=NSIG+NDEC
61 C IF (N.GT.200) GO TO 91
```

REVERB-2

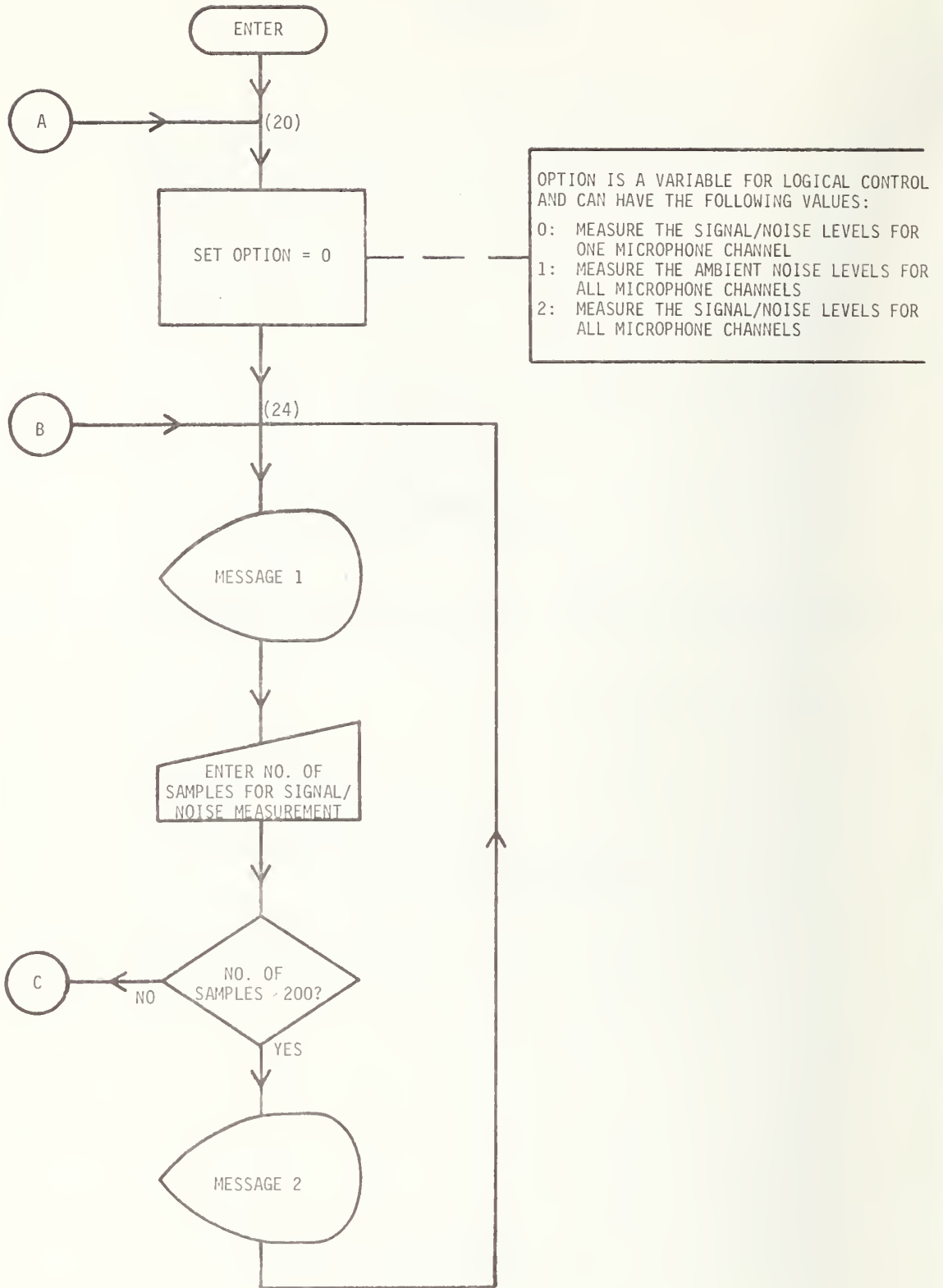
```
61      WRITE (5,105)
62 105  FORMAT ('ENTER UPPER CURVE FITTING LIMIT IN DB DOWN FROM
63      1 SIGNAL LEVEL'/'(NORMALLY 5.0)')
64      READ (5,40) CUPPER
65      WRITE (5,106)
66 106  FORMAT ('ENTER LOWER CURVE FITTING LIMIT IN DB UP FROM
67      1 NOISE LEVEL'/'(NORMALLY 10.0)')
68      READ (5,40) CLOWER
69      REWIND 1
70      NDCAYS=0
71      ICRVFT=0
72 110  AUTO=0
73      MPLEX=0
74 113  IF (MPLEX.EQ.1) CALL MUX(1)
75 114  CALL SOUND(1)
76      WRITE (5,115)
77 115  FORMAT ('THE SOUND HAS BEEN TURNED ON'/'
78      1 ENTER "GO" TO START DECAY MEASUREMENT')
79      READ (5,120)GS
80 120  FORMAT (A4)
81      GO TO 124
82 123  INDEX=INDEX+1
83      IF (INDEX.EQ.NAUTO) AUTO=2
84      CALL DELAY(NDELAY)
85 124  CALL RDMUX(MXRDDG)
86      CALL SAMPLE(NSIG,N,10,OVFLOW,X0,6060)
87      IF (OVFLOW.EQ.1) GO TO 114
88      IF ((MPLEX.EQ.1).AND.(AUTO.EQ.1)) CALL MUX(1)
89      IF (AUTO.EQ.1) CALL SOUND(1)
90      NDCAYS=NDCAYS+1
91      WRITE (5,125) NDCAYS
92 125  FORMAT ('DECAY NO. ',I4)
93      IF (ICRVFT.EQ.1) GO TO 126
94      REWIND 0
95      CALL IFETCH('CURVFT',0,ISTAT)
96      IF (ISTAT.NE.0) GO TO 200
97      ICRVFT=1
98 126  CALL CURVFT(N,NSIG,X,X0,S0,S1,ARRAY1,ARRAY2,DELTA,FREQ,
99      INDCAYS,AUTO,LU,BLOW,BHIGH,CUPPER,CLOWER,MM1,MM2,MXRDDG)
100     IF (AUTO.EQ.1) GO TO 123
101 127  WRITE (5,128)
102 128  FORMAT ('DO YOU WANT TO CONDUCT ANOTHER DECAY?
103      1 (ENTER "AUTO" FOR AUTOMATIC MODE)')
104     READ (5,120)GS
105     IF (GS.EQ.'NO')GO TO 140
106     IF (GS.NE.'AUTO') GO TO 110
107 129  WRITE (5,130)
108 130  FORMAT ('ENTER NUMBER OF DECAYS TO BE CONDUCTED
109      1 AUTOMATICALLY'/'(4 DIGITS)')
110     READ (5,131) NAUTO
111 131  FORMAT (I4)
112     IF (NAUTO.LT.2) GO TO 110
113     WRITE (5,133)
114 133  FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
115      1 FOR CURVE-FITTING PARAMETERS')
116     READ (5,135) LU
117 135  FORMAT (I1)
118     WRITE (5,136)
119 136  FORMAT ('ENTER NO. OF SECONDS OF DELAY REQUIRED FOR SOUND
120      1 BUILD-UP (XX.X)'/'(MAX = 32)'/'(THIS DELAY IS IN
```

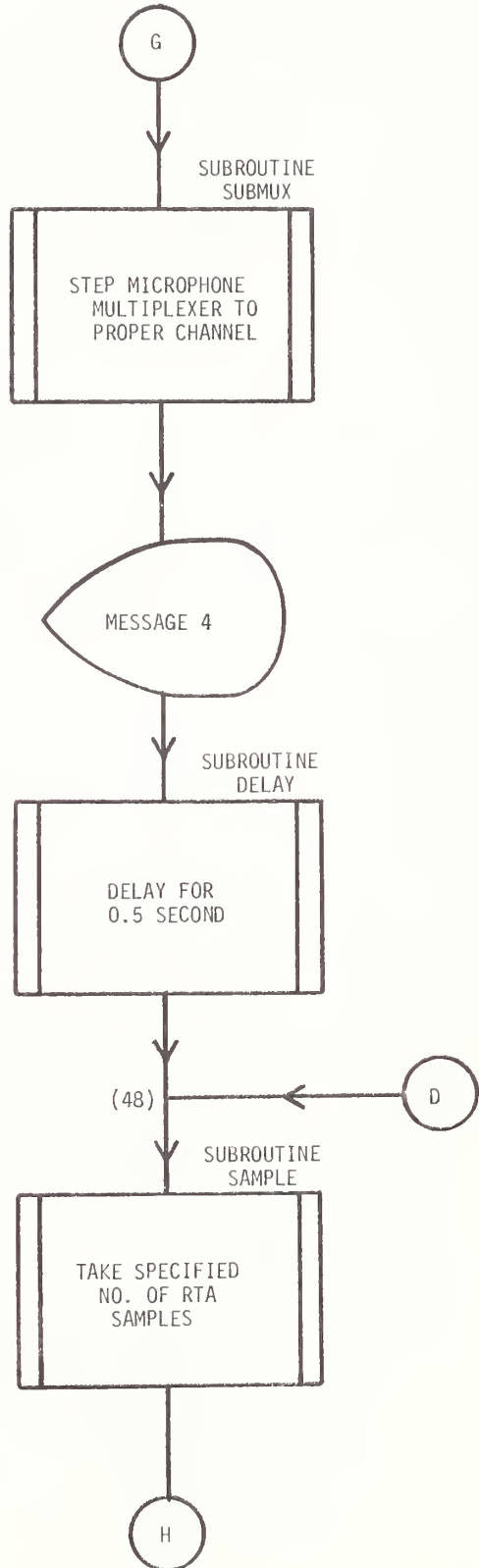
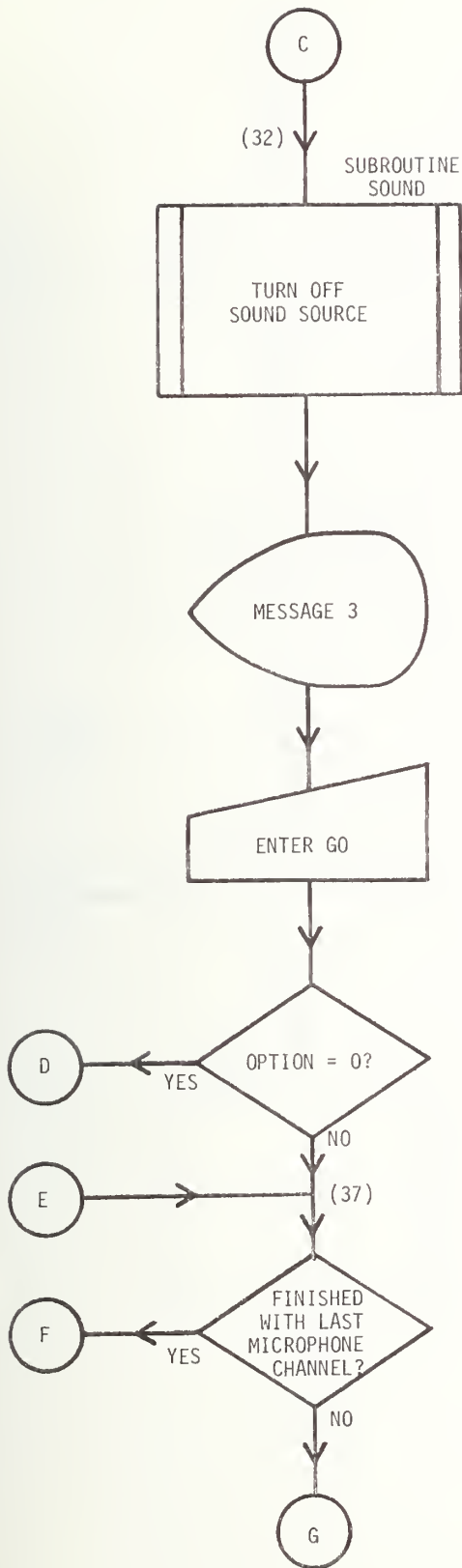
REVERB-3

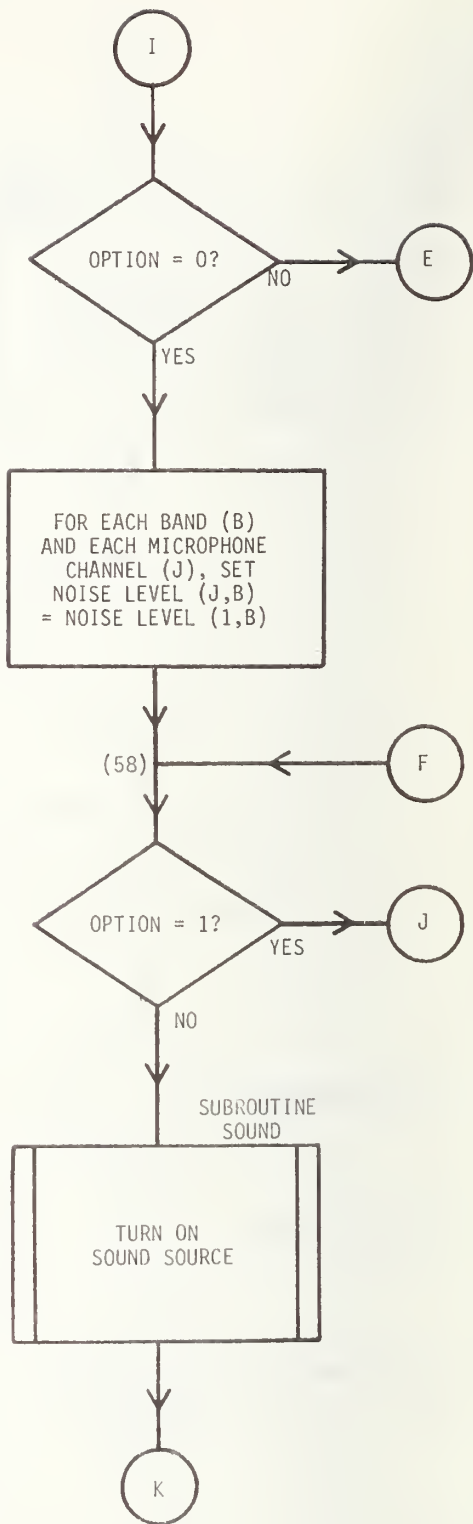
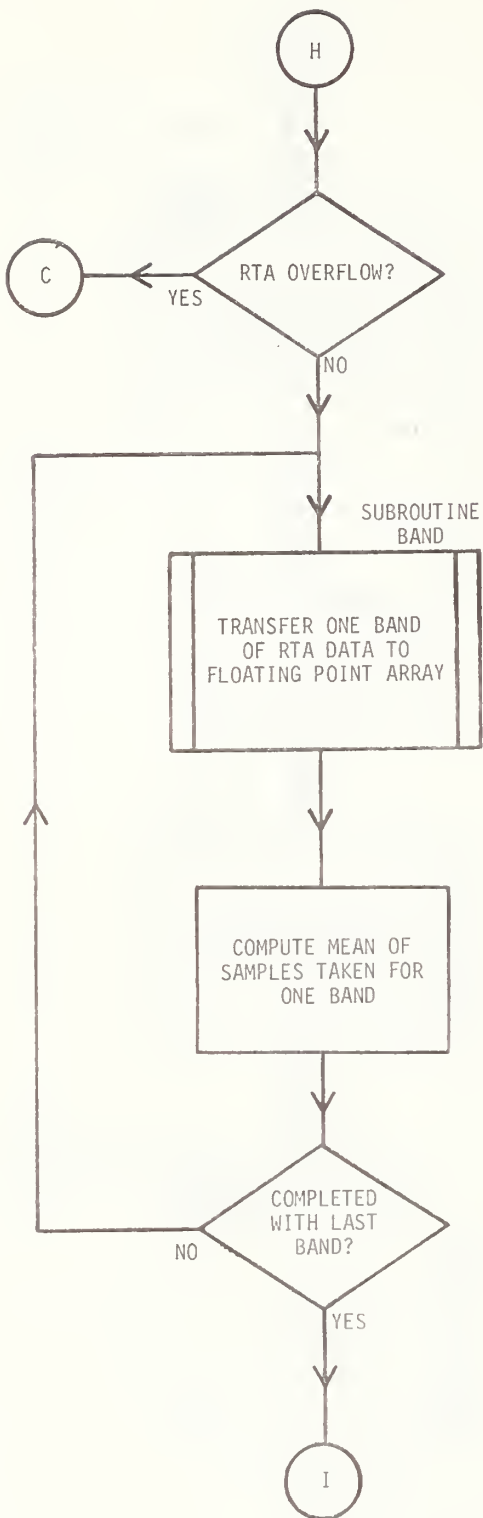
```
121      2 ADDITION TO THE COMPUTING AND PRINTOUT TIME)')
122      READ (5,40) TDELAY
123      NDELAY=TDELAY*1000.+5
124      WRITE (5,138)
125 138   FORMAT ('DO YOU WANT TO STEP THE MICROPHONE MULTIPLEXER
126       1 BEFORE EACH DECAY?')
127      READ (5,120) GS
128      MPLEX=1
129      IF (GS.EQ.'NO') MPLEX=0
130      AUTO=1.
131      INDEX=1
132      GO TO 113
133 140   ICRVET=0
134      WRITE (5,150)
135 150   FORMAT ('DO YOU WANT TO SEE THE DATA POINTS OF THE LAST DECAY?')
136      READ (5,120)GS
137      IF (GS.EQ.'NO')GO TO 160
138      REWIND 0
139      CALL IFETCH('DSDATA',0,ISTAT)
140      IF (ISTAT.NE.0) GO TO 200
141      CALL DSDATA(X,X0,ARRAY1,ARRAY2,FREQ,N,NDAYS,LINE,DELTA,
142 150,S1,CUPPER,CLOWER,MM1,MM2,MXRDG)
143 160   WRITE (5,170)
144 170   FORMAT ('DO YOU WANT TO SEE THE AVERAGES?')
145      READ (5,120)GS
146      IF (GS.EQ.'NO')GO TO 127
147      REWIND 0
148      CALL IFETCH('AVRGE',0,ISTAT)
149      IF (ISTAT.NE.0) GO TO 200
150      CALL AVRGE(X,X0,APPAY1,ARRAY2,NDAYS,CINTER,CLEVEL,FREQ,
151 180,BLOW,BHIGH)
152 180   WRITE (5,138)
153      READ (5,120)GS
154      IF (GS.EQ.'AUTO') GO TO 129
155      IF (GS.NE.'NO')GO TO 110
156      REWIND 0
157      CALL IFETCH('RESULTS',0,ISTAT)
158      IF (ISTAT.NE.0) GO TO 200
159      CALL RESULTS(X,X0,APPAY1,ARRAY2,FREQ,NDAYS,CLEVEL,NSIG,
160 200,INDEC,LINE,DELTA,BLOW,BHIGH,CUPPER,CLOWER)
161      GO TO 220
162 200   WRITE (5,210) ISTAT
163 210   FORMAT ('OVERLAY LOAD ERROR ',I4,' -- TASK ABORTED')
164 220   WRITE (5,230)
165 230   FORMAT ('X' 'END OF JOB')
166      END
```

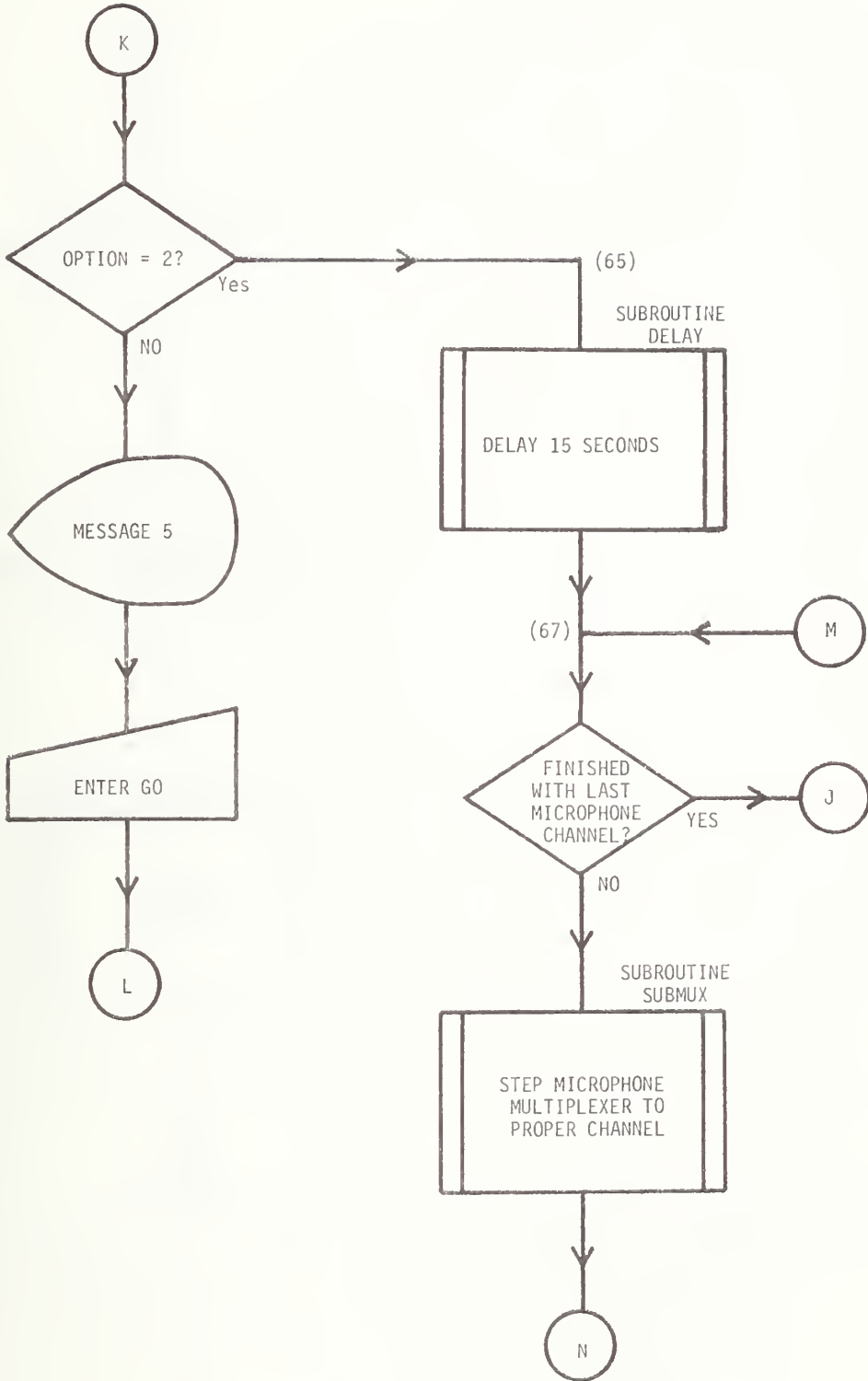
APPENDIX C

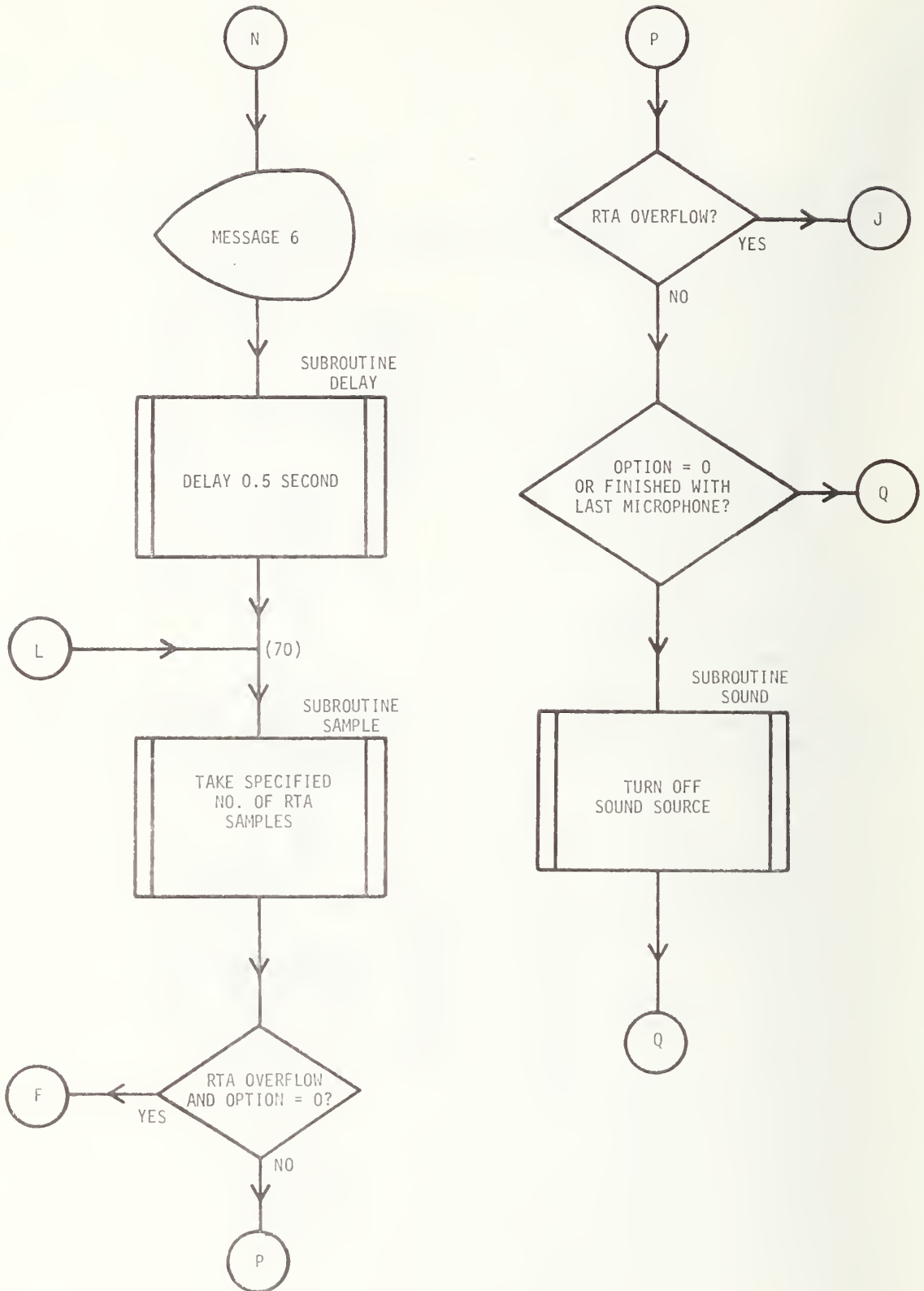
Subroutine SIGNAL Flow Chart, Terminal Messages, Printouts, and Listings

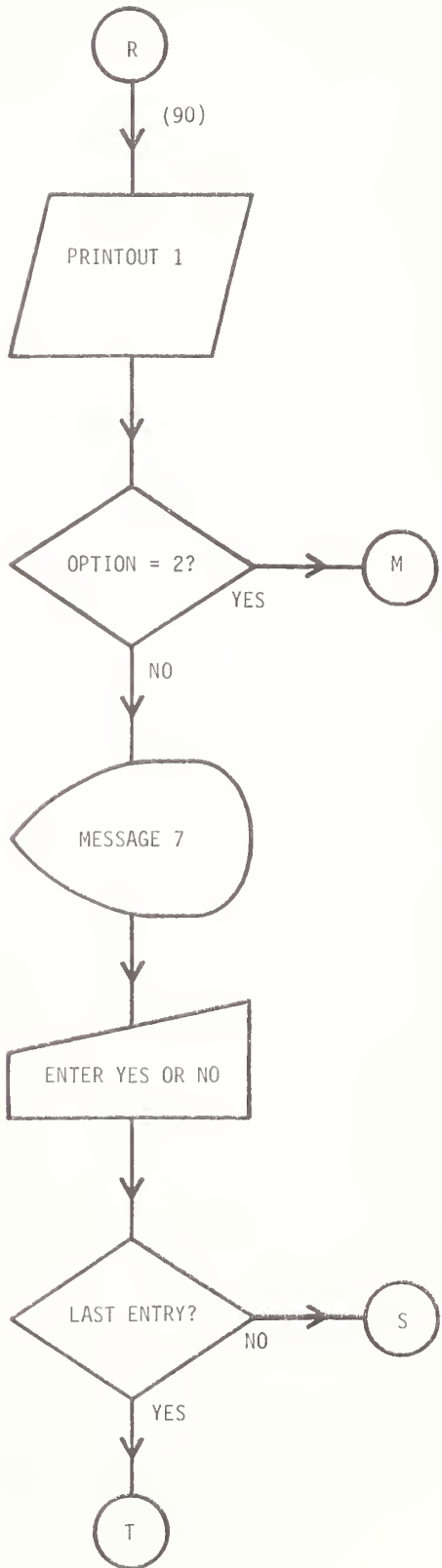
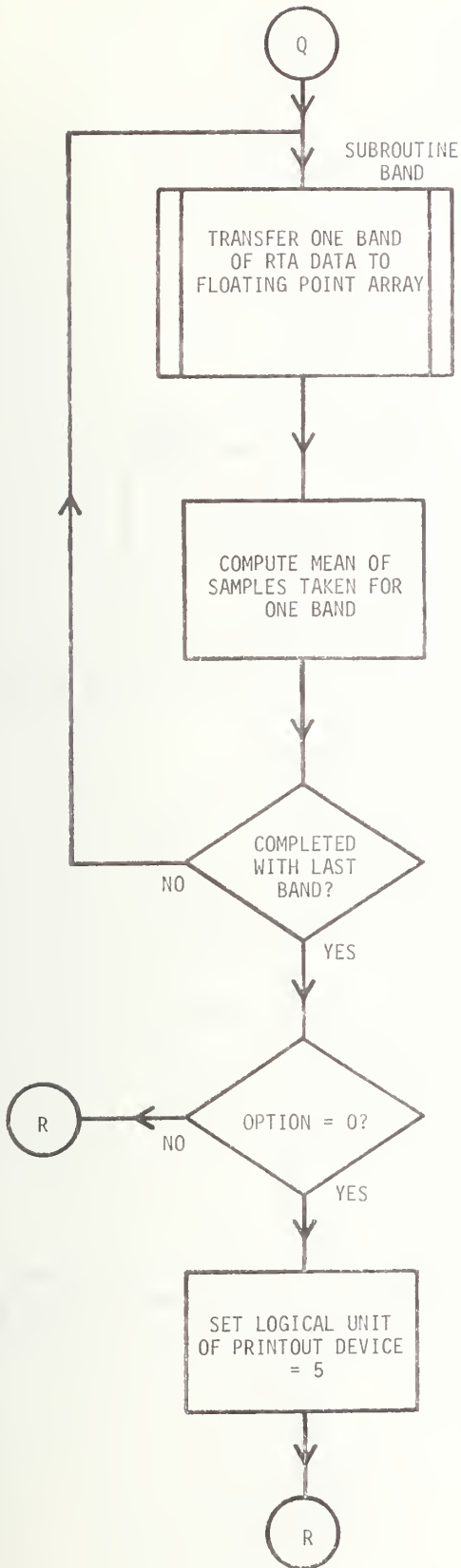


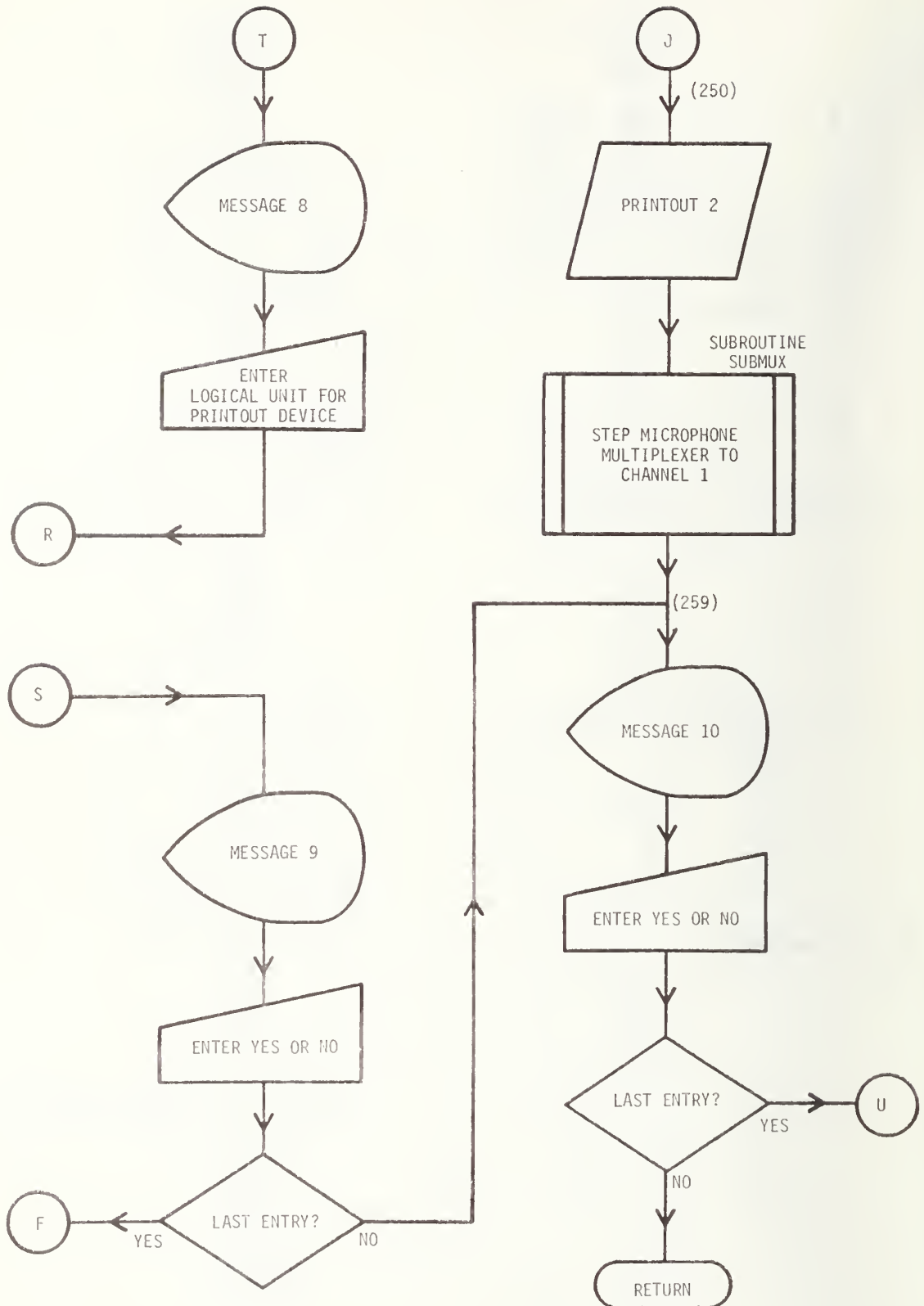


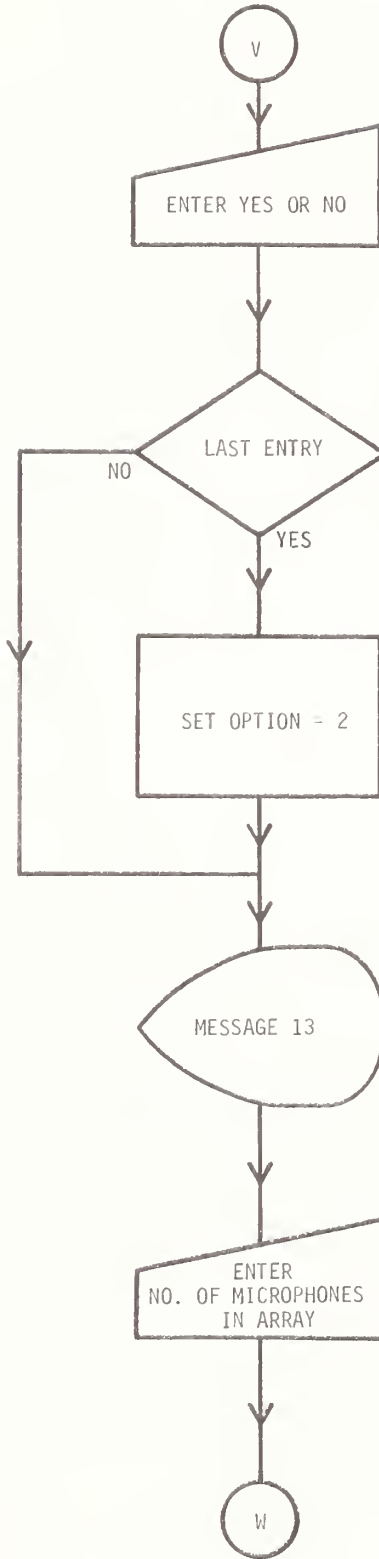
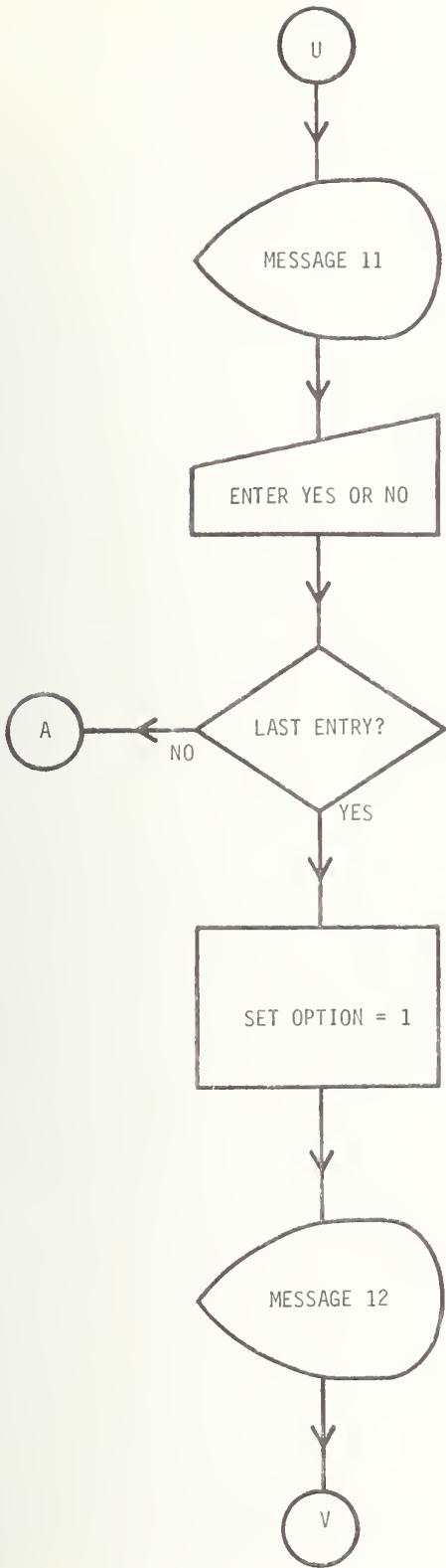


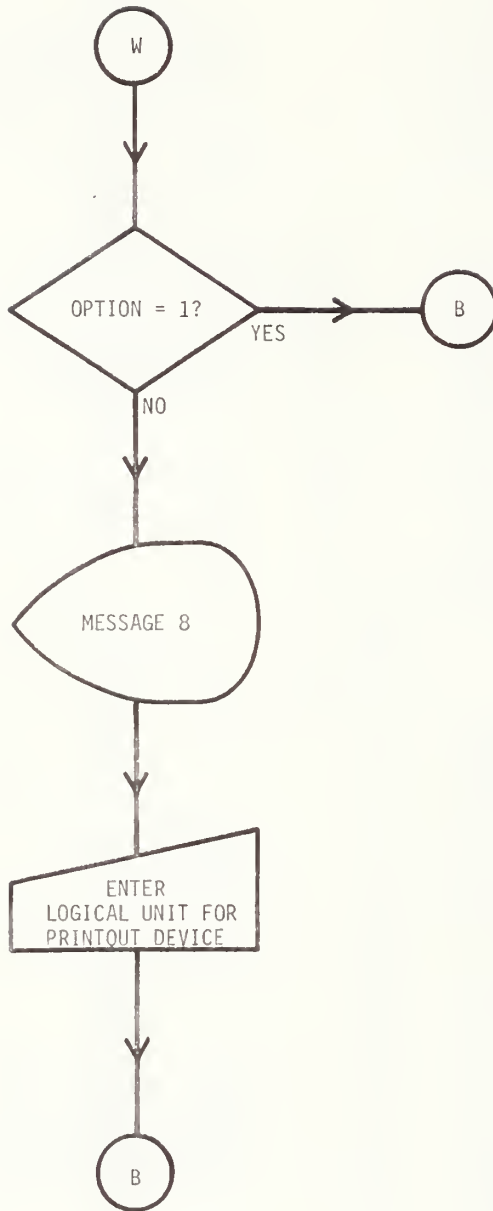












CRT TERMINAL MESSAGES: SUBROUTINE SIGNAL

MESSAGE NO.	FORMAT NO.	MESSAGE
1	25	ENTER NO. OF SAMPLES FOR SIGNAL/NOISE MEASUREMENT (3 DIGITS)
2	28	YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES
3	35	THE SOUND HAS BEEN TURNED OFF ENTER "GO" TO START AMBIENT NOISE MEASUREMENT
4	38	AMBIENT NOISE MEASUREMENT, MICROPHONE 1
5	60	THE SOUND HAS BEEN TURNED ON ENTER "GO" TO START SIGNAL MEASUREMENT
6	68	SIGNAL LEVEL MEASUREMENT, MICROPHONE 1
7	200	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
8	210	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
9	240	DO YOU WANT TO REPEAT THE SIGNAL MEASUREMENT?
10	260	DO YOU WANT TO REPEAT THE SIGNAL AND NOISE MEASUREMENT?
11	270	DO YOU WANT A SEPARATE AMBIENT NOISE MEASUREMENT FOR EACH MICROPHONE CHANNEL?
12	275	DO YOU WANT TO INCLUDE A SIGNAL LEVEL MEASUREMENT FOR EACH CHANNEL?
13	280	ENTER NO. OF MICROPHONES IN THE MICROPHONE ARRAY (2 DIGITS)

PRINTOUT 1 - SUBROUTINE SIGNAL

SIGNAL/NOISE MEASUREMENT: NO. OF SAMPLES = 200

FREQUENCY (HZ)	NOISE LEVEL (DB)	SIGNAL LEVEL (DB)	DYNAMIC RANGE (DB)
25		42.82	31.79
32	+	46.58	38.21
40	+	47.46	42.27
50	+	49.90	41.90
63	+	46.35	39.72
80	+	48.44	41.31
100	+	48.63	42.00
125	+	49.98	44.30
160	+	49.26	48.58
200	+	48.69	48.45
250	*	47.11	46.86
315	*	47.59	47.37
400	*	48.15	47.95
500	*	47.65	47.43
630	*	48.20	47.96
800	*	48.21	47.99
1000	*	47.89	47.62
1250	*	48.48	48.02
1600	*	49.25	49.00
2000	*	48.64	48.38
2500	*	47.50	47.24
3150	*	48.31	48.10
4000	*	47.31	46.99
5000	*	47.28	47.12
6300	*	46.58	46.39
8000	*	47.18	46.85
10000	*	47.89	47.44
12500	*	47.35	46.90
16000	*	46.42	44.38
20000	*	33.98	31.13

BAND NO.

FREQUENCY (HZ)

PRINTOUT 2 - SUBROUTINE SIGNAL

AMBIENT NOISE LEVELS

BAND NO.	FREQ.	MIC 1	MIC 2	MIC 3	MIC 4	MIC 5	MIC 6	MIC 7	MIC 8	MIC 9	MIC 10	MIC 11	MIC 12
14	25	10.11	15.04	8.81	3.74	13.49	4.44	19.76	9.36	23.85	14.95	19.54	20.69
15	32	8.29	12.42	6.99	3.76	12.15	8.38	19.16	10.82	22.69	12.50	17.63	21.97
16	40	5.87	13.76	4.98	1.40	12.05	0.21	17.77	6.80	20.75	11.78	18.30	20.08
17	50	8.88	15.82	8.78	7.11	11.85	2.10	15.99	7.32	19.05	12.54	15.19	18.09
18	63	6.57	11.21	5.76	3.95	9.41	3.66	16.26	6.89	16.43	8.55	12.70	15.84
19	80	8.71	11.83	4.36	6.48	9.43	5.71	12.34	5.24	13.06	7.17	10.55	12.74
20	100	7.80	7.99	6.96	7.15	7.06	5.14	9.98	5.31	9.65	6.52	8.89	10.17
21	125	6.01	8.57	7.07	7.46	8.25	4.00	7.47	4.65	5.68	5.44	8.93	6.97
22	160	0.61	1.38	0.07	0.64	1.13	0.11	1.96	0.18	1.49	0.31	0.95	1.59
23	200	0.09	0.25	0.00	0.05	0.04	0.04	0.56	0.01	0.18	0.02	0.27	0.16
24	250	0.06	0.02	0.00	0.00	0.00	0.00	0.01	0.00	0.02	0.00	0.05	0.00
25	315	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
26	400	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
27	500	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
28	630	0.39	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
29	800	0.45	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
30	1000	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
31	1250	0.48	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
32	1600	0.61	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
33	2000	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.00
34	2500	0.36	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
35	3150	0.26	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00
36	4000	0.22	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
37	5000	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.00	0.00
38	6300	0.24	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00
39	8000	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.05	0.00	0.00	0.00
40	10000	0.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07	0.00	0.00	0.00
41	12500	2.38	1.55	1.44	1.22	0.02	0.00	0.01	0.00	0.19	0.00	0.00	0.10
42	16000	2.51	2.51	2.55	3.00	1.85	1.17	1.16	1.11	2.36	1.39	1.47	1.86
43	20000					2.65	2.79	0.00	1.92	3.94	2.60	1.97	0.00

NO. OF SAMPLES = 200
NO. OF MICROPHONES = 12

SUBROUTINE SIGNAL

```
1 $ASSM
2 SIGNAL PROG SIGNAL - SUBROUTINE TO DETERMINE SIGNAL/NOISE LEVELS
3 $FORT
4 C
5 C -----
6 C OTHER SUBROUTINES CALLED:  SAMPLE,BAND,SOUND,SUBMUX,MUX,DELAY
7 C -----
8 SUBROUTINE SIGNAL(X,X0,S0,S1,FREQ,LINE,BLOW,BHIGH)
9 INTEGER*2 B,N,X0(6060),FREQ(30),LU,BELL,FF,OPTION,
10 BLOW,BHIGH,B1,B2,OVFLOW,LINE(61),BLANK,STAR,PLUS
11 DIMENSION X(200),S0(30,12),S1(30)
12 DATA BLANK,STAR,PLUS,FF/'X'2000',X'2A00',X'2B00',
13 1X'0C00'/'
14 20 OPTION=0
15 24 WRITE (5,25)
16 25 FORMAT ('ENTER NO. OF SAMPLES FOR SIGNAL/NOISE MEASUREMENT')
17 26 12X,'(3 DIGITS)')
18 26 READ (5,26)N
19 26 FORMAT (I3)
20 26 IF (N.LE.200)GO TO 32
21 28 WRITE (5,28)
22 28 FORMAT ('YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF 200 SAMPLES')
23 32 GO TO 24
24 32 CALL SOUND(0)
25 34 WRITE (5,35)
26 35 FORMAT ('THE SOUND HAS BEEN TURNED OFF')
27 35 1'ENTER "GO" TO START AMBIENT NOISE MEASUREMENT')
28 36 READ (5,36)GS
29 36 FORMAT (A4)
30 36 J=1
31 36 IF (OPTION.EQ.0) GO TO 48
32 37 J=J+1
33 37 IF (J.GT.NMIC) GO TO 58
34 37 CALL SUBMUX(J)
35 37 WRITE (5,38) J
36 38 FORMAT ('AMBIENT NOISE MEASUREMENT. MICROPHONE ',I2)
37 38 CALL DELAY(.500)
38 48 CALL SAMPLE(0,N,10,OVFLOW,X0,6060)
39 48 IF (OVFLOW.EQ.1) GO TO 32
40 48 DO 50 B=BLOW,BHIGH
41 48 IB=B-13
42 48 CALL BAND(B,N,X0,6060)
43 48 C=0.
44 48 DO 49 I=1,N
45 48 C=C+10.*X(I)/10.
46 49 CONTINUE
47 48 C=C/N
48 48 S0(IB,J)=10.*ALOG10(C)
49 50 CONTINUE
50 50 IF (OPTION.NE.0) GO TO 37
51 51 DO 55 J=2,12
52 51 DO 55 B=BLOW,BHIGH
53 51 IB=B-13
54 51 S0(IB,J)=S0(IB,1)
55 55 CONTINUE
56 58 IF (OPTION.EQ.1) GO TO 250
57 58 CALL SOUND(1)
58 58 IF (OPTION.EQ.2) GO TO 65
59 58 WRITE (5,60)
60 60 FORMAT ('THE SOUND HAS BEEN TURNED ON')
```

SIGNAL-2

```

61      1'ENTER "GO" TO START SIGNAL MEASUREMENT')
62      READ (5,36)GS
63      GO TO 70
64 65    CALL DELAY(15000)
65      J=0
66 67    J=J+1
67      IF (J.GT.NM10) GO TO 250
68      CALL SUBMUX(J)
69      WRITE (5,68) J
70 68    FORMAT ('SIGNAL LEVEL MEASUREMENT, MICROPHONE ',I2)
71      CALL DELAY(500)
72 70    CALL SAMPLE(N,H,10,OVFLOW,X0,6060)
73      IF ((OVFLOW.EQ.1).AND.(OPTION.EQ.0)) GO TO 58
74      IF (OVFLOW.EQ.1) GO TO 250
75      IF ((OPTION.EQ.0).OR.(J.EQ.NM10)) CALL SOUND(0)
76      DO 80 B=BLOW,BHIGH
77      IB=B-13
78      CALL BAND(B,N,X,X0,6060)
79      C=0.
80      DO 75 I=1,N
81      C=C+10.**((X(I)/10.))
82 75    CONTINUE
83      C=C/N
84      S1(IB)=10.*ALOG10(C)
85 80    CONTINUE
86      IF (OPTION.EQ.0) LU=5
87 90    WRITE (LU,92) N
88 92    FORMAT ('SIGNAL NOISE MEASUREMENT:',
89            1T30,'NO. OF SAMPLES = ',I3)
90      IF (OPTION.EQ.0) GO TO 98
91      WRITE (LU,94) J
92 94    FORMAT (T30,'MICROPHONE NO. ',I2)
93 98    WRITE (LU,100)
94 100   FORMAT (5(X/),13X,'NOISE LEVEL-----SIGNAL LEVEL (DB)')
95      IF (LU.EQ.5) GO TO 105
96      WRITE (LU,102)
97 102   FORMAT (X/X/6X,'0'.9X,'10'.8X,'20'.8X,'30'.8X,'40'.
98      18X,'50'.8X,'60'.77X,'NOISE',T86,'SIGNAL',T96,'DYNAMIC'/
99      26X,13(' ',4X),T77,'LEVEL',T87,'LEVEL',T97,'RANGE'/
100     3T78,'(DB)',T88,'(DB)',T98,'(DB)'/X)
101     GO TO 108
102 105   WRITE (5,106)
103 106   FORMAT (X/X/6X,'0'.9X,'10'.8X,'20'.8X,'30'.8X,'40'.
104     18X,'50'.8X,'60'/6X,13(' ',4X)/X/X)
105 108   DO 110 I=1,61
106     LINE(I)=BLANK
107 110   CONTINUE
108     DO 130 B=BLOW,BHIGH
109     IB=B-13
110     J1=S0(IB,J)+1.5
111     J2=S1(IB)+1.5
112     LINE(J1)=PLUS
113     LINE(J2)=STAR
114     IF (LU.EQ.5) GO TO 125
115     C=S1(IB)-S0(IB,J)
116     WRITE (LU,120)FREQ(IB),(LINE(I),I=1,61),B,S0(IB,J),S1(IB),C
117 120   FORMAT (15,X,61A1,X,I2,T76,F6.2,T86,F6.2,T96,F6.2)
118     GO TO 128
119 125   WRITE (5,126) FREQ(IB),(LINE(I),I=1,61),B
120 126   FORMAT (15,X,61A1,X,I2)

```

SIGNAL-3

SIGNAL-3

```

121 128 LINE(J1)=BLANK
122 LINE(J2)=BLANK
123 130 CONTINUE
124 WRITE (LU,140)FF
125 140 FORMAT ('X'FREQUENCY',36X,'BAND NO.',2X,'(HZ)',F1)
126 IF (OPTION.EQ.2) GO TO 67
127 WRITE (5,200)
128 200 FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
129 READ (5,36)GS
130 IF (GS.EQ.'NO')GO TO 230
131 WRITE (5,210)
132 210 FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
133 READ (5,200)LU
134 350 FORMAT ('I')
135 GO TO 30
136 230 WRITE (5,240)
137 240 FORMAT ('DO YOU WANT TO REPEAT THE SIGNAL MEASUREMENT?')
138 READ (5,36)GS
139 IF (GS.EQ.'NO') GO TO 58
140 GO TO 359
141 380 WRITE (3,252) (I,I=1,12)
142 252 FORMAT ('X'X'THE AMBIENT NOISE LEVELS',3X/X/X/X-
143 '1'BAND',2X,'NO.',178,'FREQ.',1716.9,'(HZ)',11.3X),
144 25 'MIC',1716.27,'(HZ)')
145 GO TO 257
146 257 B=BL00LENSH
147 I=10+13
148 JI=0+110+1
149 DO 259 I=1,12
150 GO TO 254
151 254 CONTINUE
152 WRITE (3,253) (I,I=1,12) (30019.10,I=1,12)
153 253 FORMAT ('1',12X,'13.715',10166.24,3X)
154 CONTINUE
155 WRITE (3,258) (I,I=1,12)FF
156 258 FORMAT ('X'X'NO. OF SAMPLES =',173,
157 '100. OF MICROPHONES =',112,'(I)')
158 CALL SUBROUTINE
159 259 WRITE (5,260)
160 260 FORMAT ('DO YOU WANT TO REPEAT THE SIGNAL AND NOISE
161 MEASUREMENT?')
162 READ (5,36)GS
163 IF (GS.EQ.'NO') GO TO 400
164 WRITE (5,200)
165 270 FORMAT ('DO YOU WANT A SEPARATE AMBIENT NOISE MEASUREMENT?')
166 READ (5,36)GS
167 IF (GS.EQ.'NO') GO TO 20
168 OPTION=1
169 GO TO 253
170 275 FORMAT ('DO YOU WANT TO MEASURE A SIGNAL LEVEL MEASUREMENT?')
171 READ (5,36)GS
172 IF (GS.EQ.'NO') GO TO 400-2
173 WRITE (5,200)
174 280 FORMAT ('ENTER NO. OF MICROPHONES IN THE MICROPHONE ARRAY')
175 READ (5,36)N1
176 CALL SUBROUTINE
177 285 FORMAT ('X')
178 IF (OPTION.EQ.1) GO TO 20
179 WRITE (5,200)

```

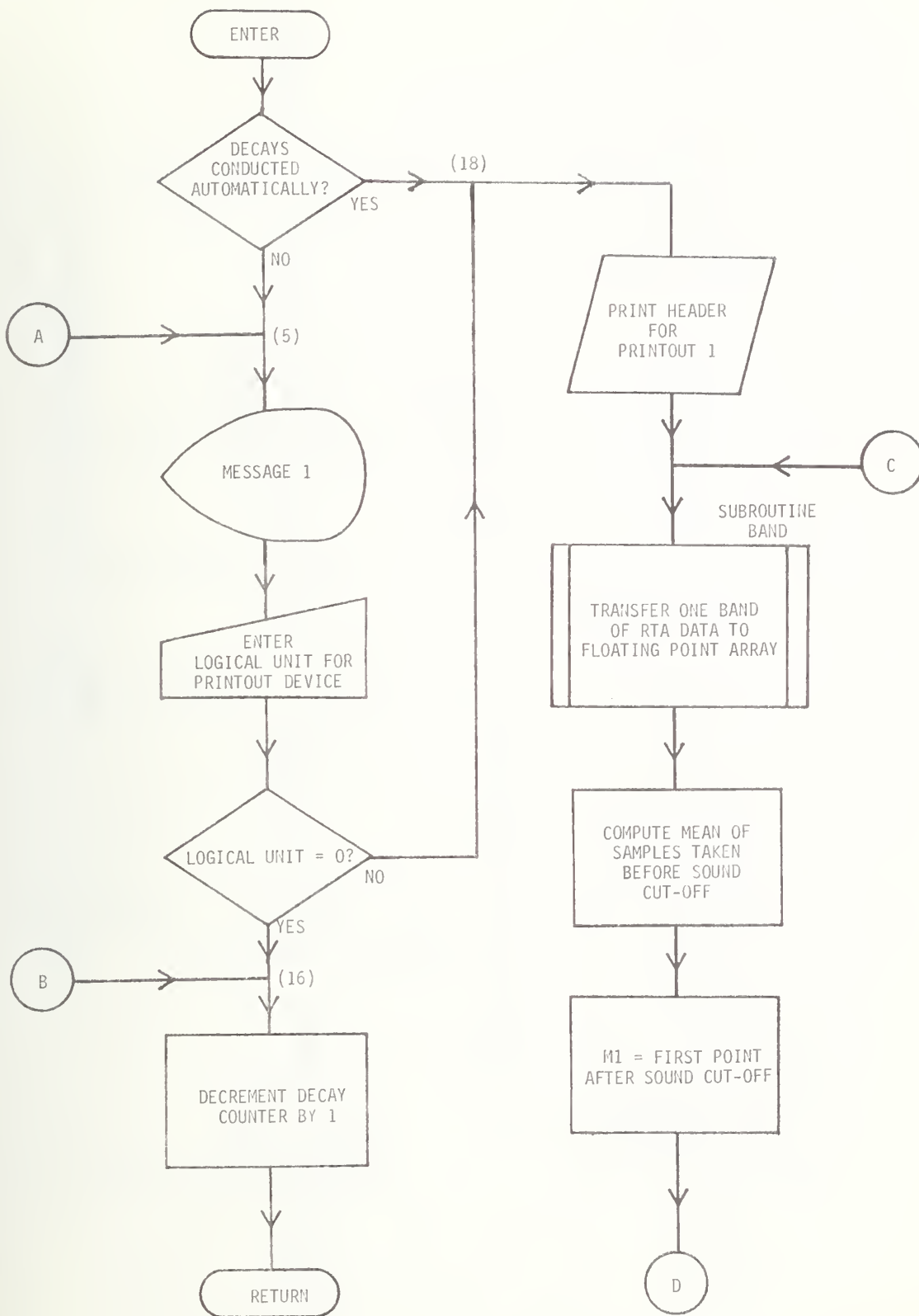
SIGNAL-4

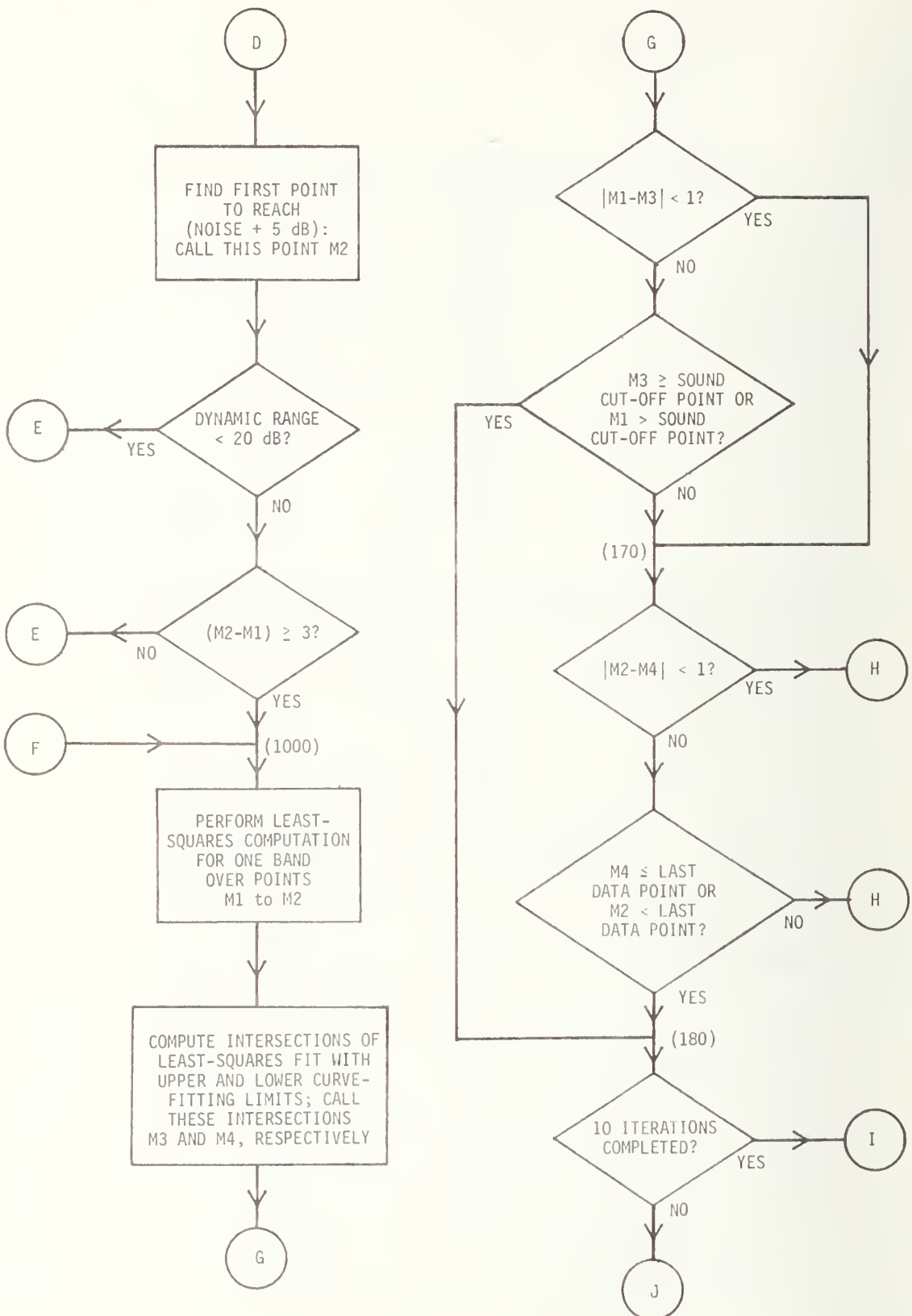
```
181      READ (5,220) LU  
182      GO TO 24  
183 400  RETURN  
184      END
```

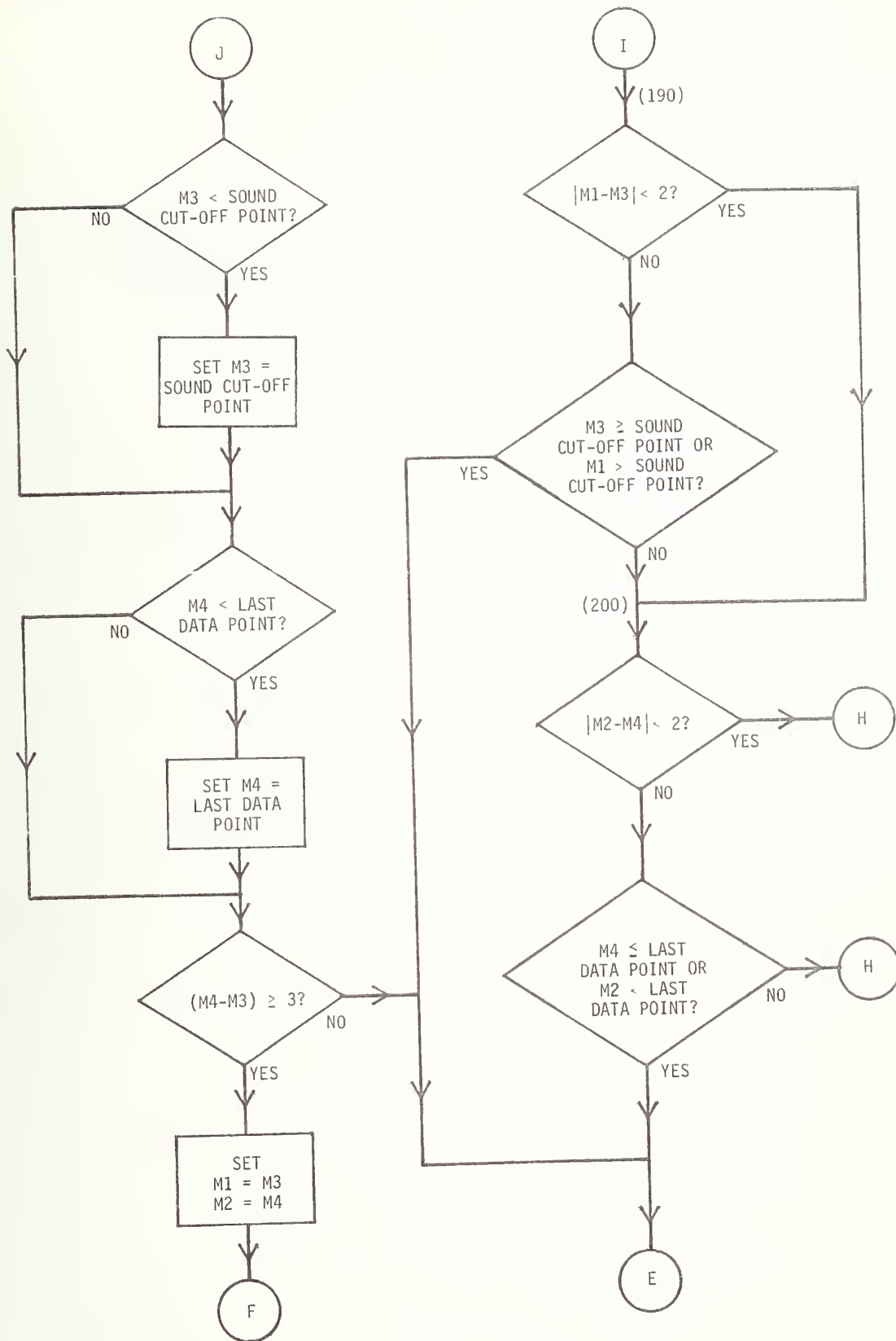
APPENDIX D

Subroutine CURVFT Flow Chart, Terminal Messages, Printout, and Listings

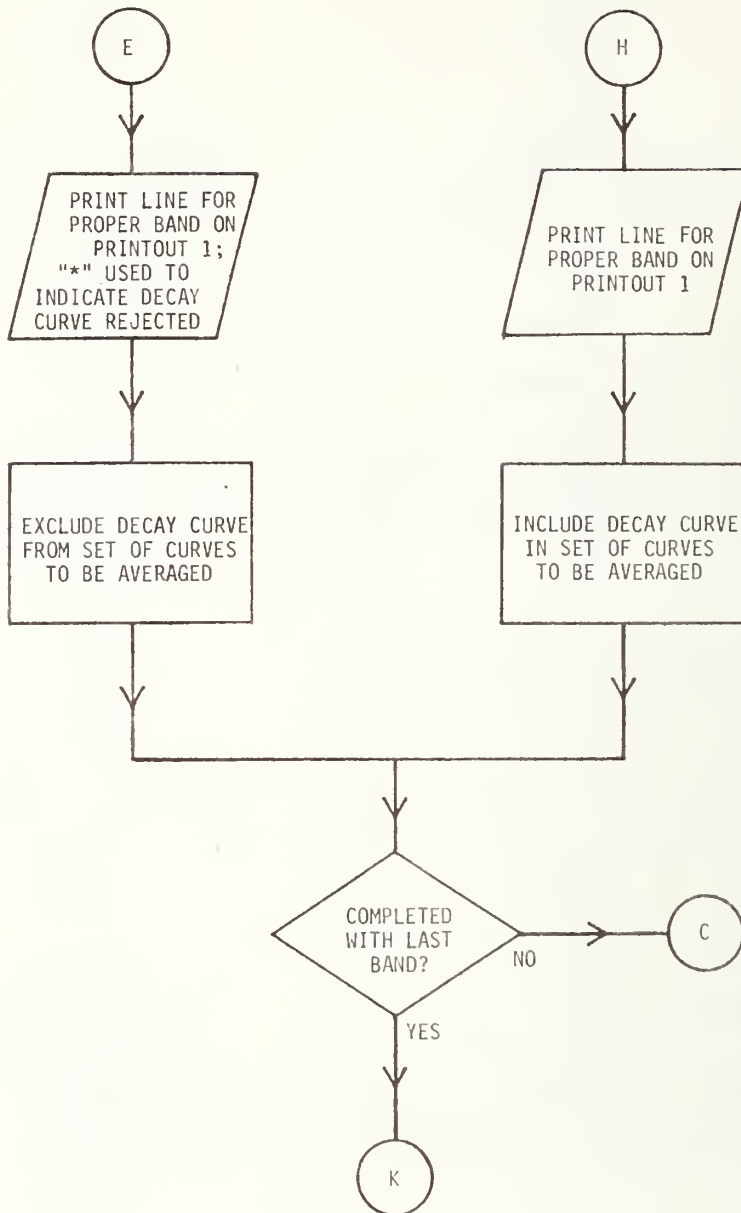
FLOW CHART: SUBROUTINE CURVFT

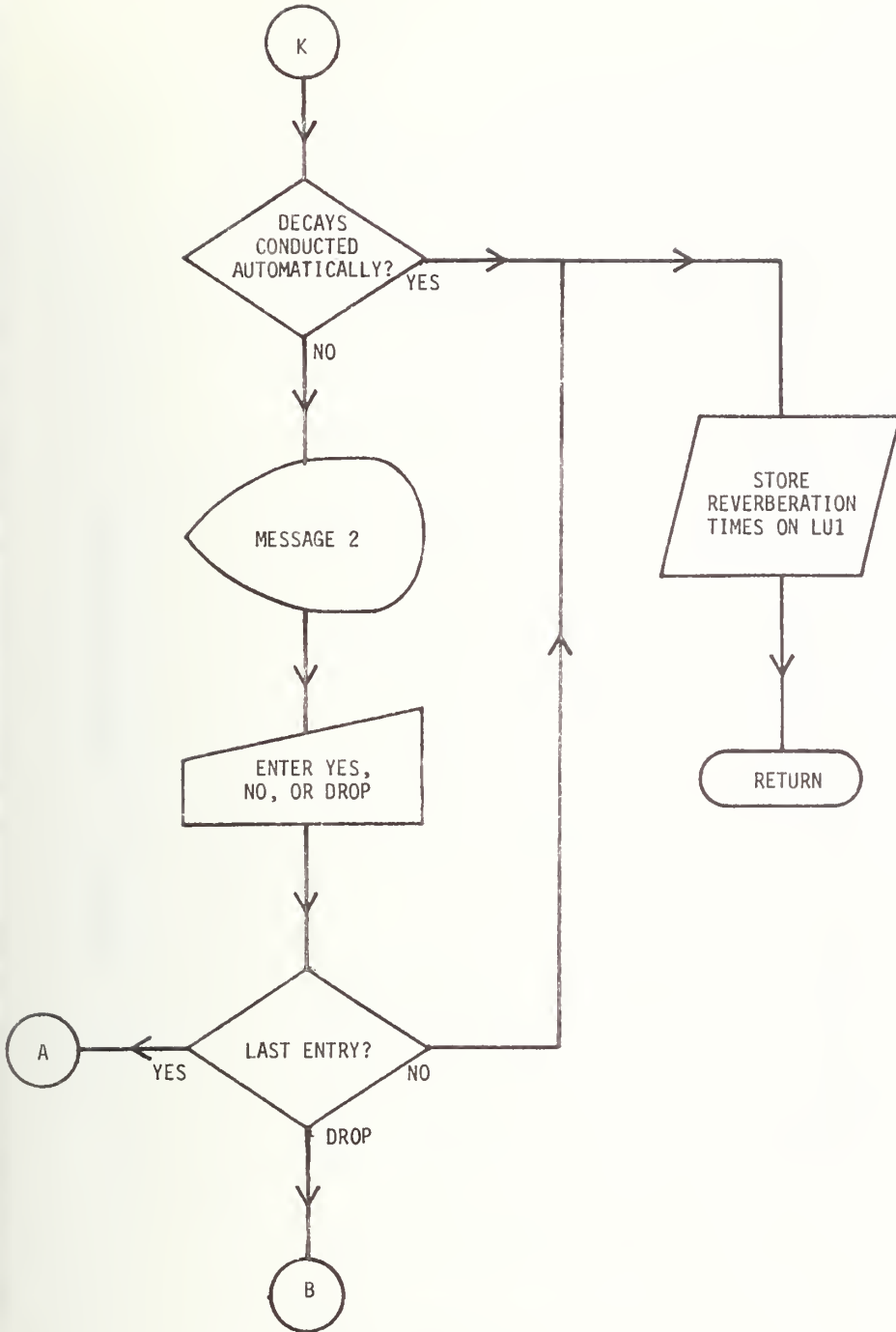






CURVFT-4





CRT TERMINAL MESSAGES: SUBROUTINE CURVFT

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-FITTING PARAMETERS (ENTER "0" IF YOU DO NOT WANT TO PROCESS THIS DECAY)
2	910	DO YOU WANT ANOTHER PRINT OF THESE RESULTS? (ENTER "DROP" IF YOU WANT TO DROP THIS DECAY FROM THE AVERAGES)

PRINTOUT 1 - SUBROUTINE CURVFT

DECAY NO. 1

BAND NO.	FREQ.	REVERB. TIME	SIGMA	SIGNAL LEVEL	NOISE LEVEL	NO. PTS.	M1	M2	NEW M1	NEW M2	DECAY RATE	INTCEPT	RMS OF RESIDUALS	NO. OF REPETITIONS
	(HZ)	(SEC)	(SEC)	(DB)	(DB)				(DB)	(DB)	(DB/SEC)	(DB)	(DB)	
14	25	23.41	0.22	44.72	10.11	78	45	122	45.53	121.95	2.56	51.15	0.49	2
15	32	18.80	0.65	45.97	8.29	72	48	119	47.43	118.37	3.19	55.81	1.94	2
16	40	12.35	0.73	45.20	5.87	48	41	88	38.64	88.63	4.86	58.51	2.76	2
17	50	9.17	0.52	46.93	6.88	36	51	86	50.32	85.48	6.55	74.27	2.33	2
18	63	5.13	0.49	44.67	6.57	22	42	63	42.66	62.37	11.71	88.52	3.35	2
19	80	3.77	0.38	47.83	8.31	17	43	59	43.33	58.70	15.93	110.38	3.24	2
20	100	4.51	0.41	47.65	7.80	17	41	57	38.56	57.19	13.32	92.75	2.44	2
21	125	4.10	0.23	49.87	6.01	20	42	61	42.06	61.76	14.63	105.05	2.10	2
22	160	4.32	0.17	48.19	6.61	25	45	69	45.61	69.01	13.89	105.27	1.92	2
23	200	5.39	0.19	48.01	6.09	31	44	74	44.06	73.61	11.12	91.00	2.00	2
24	250	6.52	0.18	45.85	6.06	35	45	79	45.05	78.43	9.21	81.48	1.52	2
25	315	6.63	0.20	46.61	6.02	36	45	81	45.91	80.77	9.05	82.32	1.68	2
26	400	6.80	0.15	47.35	6.08	38	46	83	46.27	82.76	8.83	82.37	1.28	2
27	500	6.67	0.13	46.68	6.07	37	46	82	46.39	81.49	9.00	82.59	1.10	2
28	630	6.08	0.11	47.00	6.30	33	46	78	46.00	78.05	9.87	86.51	1.00	2
29	800	5.93	0.09	46.84	6.39	32	45	77	45.89	76.94	10.11	87.32	0.81	2
30	1000	5.80	0.10	46.96	6.45	33	45	76	45.49	75.90	10.35	88.07	0.89	2
31	1250	5.00	0.11	47.56	6.34	35	45	75	45.09	77.23	10.01	86.76	1.01	2
32	1600	5.71	0.08	48.34	6.48	31	45	75	44.44	75.67	10.51	89.06	0.71	2
33	2000	5.34	0.10	47.48	6.61	29	45	73	44.72	73.04	11.24	91.69	0.91	2
34	2500	4.66	0.08	46.09	6.34	25	44	68	44.29	68.14	12.87	96.88	0.77	2
35	3150	4.12	0.05	47.09	6.38	22	44	65	43.50	65.01	14.58	103.87	0.51	2
36	4000	3.77	0.05	45.92	6.26	18	43	60	42.99	60.18	17.81	115.83	0.53	2
37	5000	2.69	0.04	45.95	6.22	15	42	56	42.45	56.33	22.27	133.41	0.60	2
38	6300	2.31	0.03	45.10	6.18	12	42	53	41.82	52.82	27.15	151.10	0.50	2
39	8000	1.57	0.03	46.05	6.24	9	42	48	41.89	49.96	38.10	197.11	0.54	2
40	10000	1.15	0.02	47.02	6.34	7	42	48	41.50	47.58	51.97	253.82	0.53	2
41	12500	0.82	0.02	46.27	6.51	6	41	46	41.45	45.61	73.52	339.02	0.72	2
42	16000	0.55	0.01	45.35	2.28	4	41	44	41.20	43.77	109.06	479.58	0.40	2
*43	20000			29.65	2.51	3	41	43						

SUBROUTINE CURVFT

```

1 $ASSM
2 CURVFT PROG CURVFT - SUBROUTINE FOR LEAST-SQUARES ANALYSIS OF DECAYS
3 $FORT
4 C
5 C -----
6 C OTHER SUBROUTINES CALLED: BAND
7 C -----
8 SUBROUTINE CURVFT(I,NSIG,X,X0,S0,S1,ARRAY1,ARRAY2,DELTA,
9 IFREQ,NDCAYS,AUTO,LU,BLOW,BHIGH,CUPPER,CLOWER,MM1,MM2,MXRDG)
10 INTEGER*2 B,NSIG,N,X0(6060),FREQ(30),LU,FF,AUTO,
11 IBLOW,BHIGH,MM1(30),MM2(30)
12 INTEGER*4 RETRN
13 DIMENSION X(200),S0(30,12),S1(30),ARRAY1(30),ARRAY2(30)
14 DATA FF/X'0000'
15 DO 2 I=1,30
16   ARRAY1(I)=0.
17 2 CONTINUE
18   IF (AUTO.NE.0) GO TO 18
19 5 WRITE (5,10)
20   FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE FOR CURVE-
21   FITTING PARAMETERS'/'(ENTER "0" IF YOU DO NOT WANT TO
22   2 PROCESS THIS DECAY)')
23   READ (5,15)LU
24   15 FORMAT (I1)
25   IF (LU.GT.0) GO TO 18
26   NDCAYS=NDCAYS-1
27   GO TO 2020
28 18 ASSIGN 150 TO RETRN
29   WRITE (LU,19)NDCAYS
30   19 FORMAT (X//X//45X,'DECAY NO.',I4)
31   IF (LU.EQ.5) GO TO 22
32   WRITE (LU,20)
33   20 FORMAT (X//X//'BAND',9X,'REVERB.',8X,'SIGNAL',3X,'NOISE',3X,
34   1'NO.',13X,'NEW',5X,'NEW',4X,'DECAY',12X,'RMS OF',6X,'NO. OF'
35   2X,'NO.',2X,'FREQ.',3X,'TIME',3X,'SIGMA',2(3X,'LEVEL'),2X,'PTS.',
36   32X,'M1',5X,'M2',5X,'M1',6X,'M2',5X,'RATE',2X,'INTCEPT',2X,
37   4'RESIDUALS',2X,'REPETITIONS',X)
38   GO TO 25
39 22 WRITE (LU,23)
40   23 FORMAT (X//X//'BAND',9X,'REVERB.',8X,'SIGNAL',3X,'NOISE',
41   13X,'NO.',13X,'NEW',5X,'NEW'
42   2X,'NO.',2X,'FREQ.',3X,'TIME',3X,'SIGMA',2(3X,'LEVEL'),
43   32X,'PTS.',2X,'M1',3X,'M2',5X,'M1',6X,'M2',X)
44 25 DO 900 B=BLOW,BHIGH
45   IB=B-13
46   CALL BAND(B,N,X,X0,6060)
47   SUM1=0
48   DO 45 I=1,NSIG
49     SUM1=SUM1+10.*(X(I)/I0)
50 45 CONTINUE
51   SUM1=SUM1/NSIG
52   S1(IB)=10*ALOG10(SUM1)
53   NREP=0
54   M1=NSIG+1
55   DO 130 I=M1,N
56     IF (X(I)-S0(IB,MXRDG)-5.)131,131,130
57 130 CONTINUE
58   M2=I
59   IF ((S1(IB)-S0(IB,MXRDG)).LT.20.) GO TO 850
60 150 IF (M2-M1-3)850,1000,1000
      NREP=NREP+1

```

CURVFT-2

```
61      AXIS1=(SLOPE*(M1-1)*DELTA)+AXIS0
62      RM3=1+(AXIS1-S1(IB)+CUPPER)/(SLOPE*DELTA)
63      RM4=1+(AXIS1-S0(IB,MXRDG)-CLOWER)/(SLOPE*DELTA)
64      M3=INT(RM3+.5)
65      M4=INT(RM4+.5)
66      IF (ABS(M1-RM3).LT.1.) GO TO 170
67      IF ((M3.GE.NSIG+1).OR.(M1.GT.NSIG+1)) GO TO 180
68 170   IF (ABS(M2-RM4).LT.1.)GO TO 880
69      IF ((M4.LE.N).OR.(M2.LT.N)) GO TO 180
70      GO TO 880
71 180   IF (NREP.EQ.10) GO TO 190
72      IF (M3.LT.NSIG+1) M3=NSIG+1
73      IF (M4.GT.N) M4=N
74      IF (M4.LT.M3+3) GO TO 860
75      M1=M3
76      M2=M4
77      GO TO 1000
78 190   IF (ABS(M1-RM3).LT.2.) GO TO 200
79      IF ((M3.GE.NSIG+1).OR.(M1.GT.NSIG+1)) GO TO 860
80 200   IF (ABS(M2-RM4).LT.2.) GO TO 880
81      IF ((M4.LE.N).OR.(M2.LT.N)) GO TO 860
82      GO TO 880
83 850   M=M2-M1+1
84      IF (LU.EQ.5) GO TO 854
85      WRITE (LU,852)B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2,NREP
86 852   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X),48X,I2)
87      GO TO 900
88 854   WRITE (LU,856) B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2
89 856   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X))
90      GO TO 900
91 860   IF (LU.EQ.5) GO TO 864
92      WRITE (LU,862)B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
93      IM,M1,M2,RM3,RM4,SLOPE,AXIS1,VRMS,NREP
94 862   FORMAT ('*',I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),4(F6.2,2X),X,
95      1F6.2,9X,I2)
96      GO TO 900
97 864   WRITE (LU,866) B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
98      IM,M1,M2,RM3,RM4
99 866   FORMAT ('*',I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),2(F6.2,2X))
100     GO TO 900
101 870   IF (LU.EQ.5) GO TO 854
102     WRITE (LU,872)B,FREQ(IB),S1(IB),S0(IB,MXRDG),M,M1,M2,
103     1SLOPE,NREP
104 872   FORMAT ('*',I2,2X,I5,18X,2(F6.2,2X),3(I3,2X),16X,F6.2,18X,I2)
105     GO TO 900
106 880   IF (LU.EQ.5) GO TO 884
107     WRITE (LU,882)B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
108     IM,M1,M2,RM3,RM4,SLOPE,AXIS1,VRMS,NREP
109 882   FORMAT (X,I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),4(F6.2,2X),X,
110     1F6.2,9X,I2)
111     GO TO 888
112 884   WRITE (LU,886) B,FREQ(IB),T60,VT60,S1(IB),S0(IB,MXRDG),
113     IM,M1,M2,RM3,RM4
114 886   FORMAT (X,I2,2X,I5,2X,4(F6.2,2X),3(I3,2X),2(F6.2,2X))
115 888   ARRAY1(IB)=T60
116     ARRAY2(IB)=AXIS1
117     MM1(IB)=M1
118     MM2(IB)=M2
119 900   CONTINUE
120     IF (LU.EQ.5) GO TO 906
```

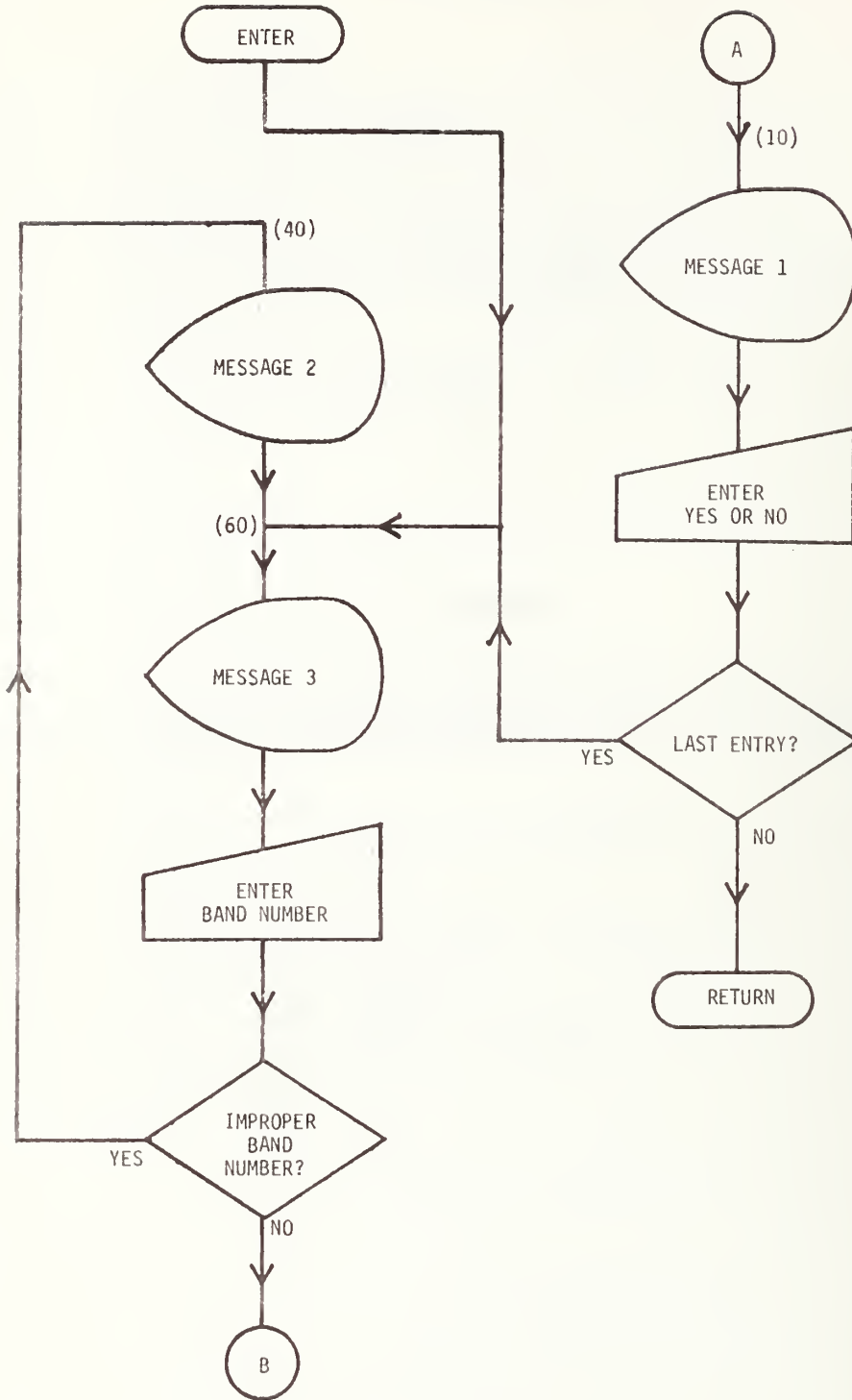
CURVFT-3

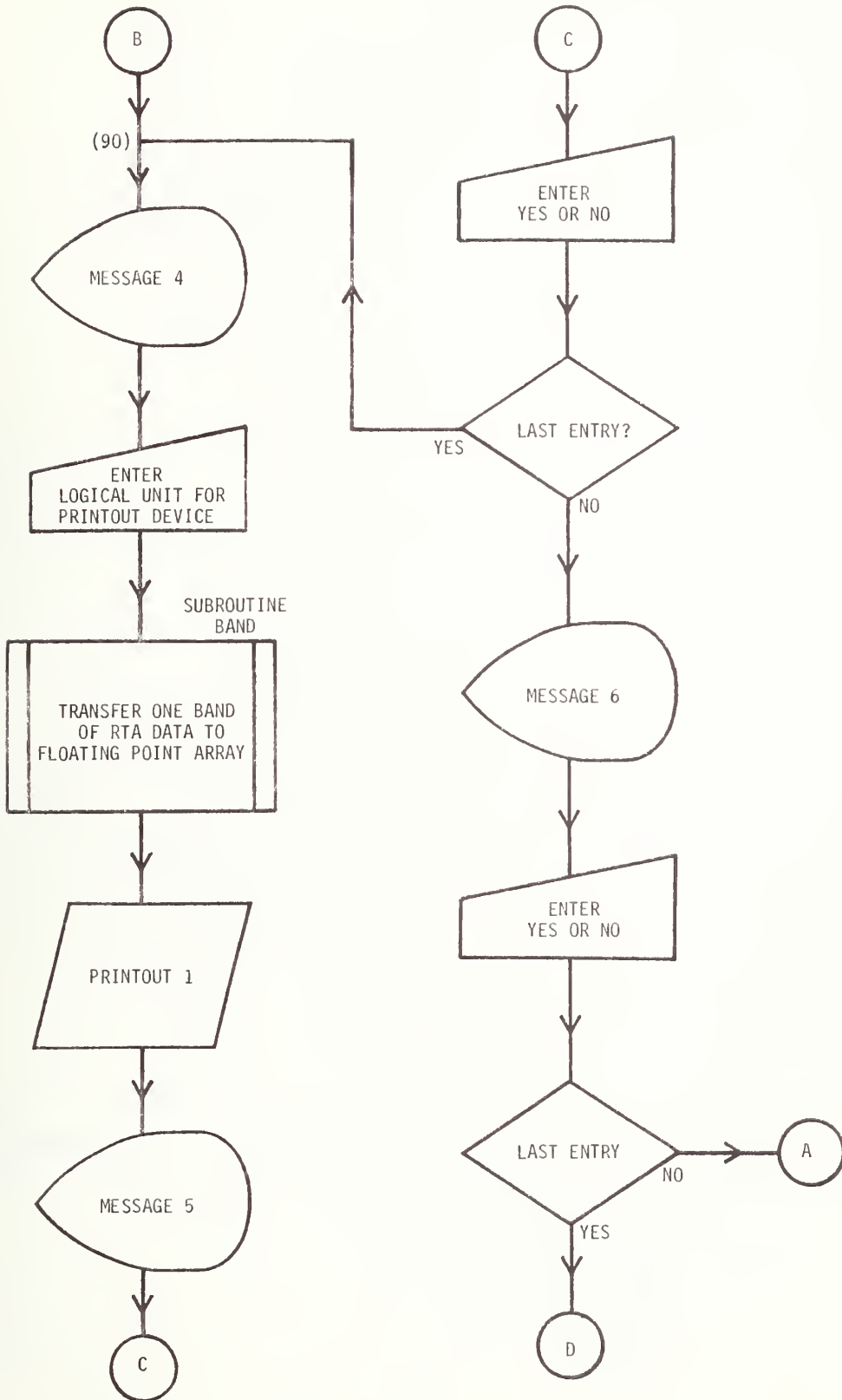
```
121      WRITE (LU,905)FF
122 905   FORMAT (X/6X,'(HZ)'.2(3X,'(SEC)'.2(4X,'(DB)'.32X,'(DB/SEC)'.
123      13X,'(DB)'.5X,'(DB)'/A1)
124      GO TO 908
125 906   WRITE (LU,907) FF
126 907   FORMAT (X/6X,'(HZ)'.2(3X,'(SEC)'.2(4X,'(DB)'/A1)
127 908   IF (AUTO.NE.0) GO TO 2000
128      WRITE (S,910)
129 910   FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?'/
130      1'(ENTER "DROP" IF YOU WANT TO DROP THIS DECAY FROM THE
131      2 AVERAGES)')
132      READ (S,915)GS
133 915   FORMAT (A4)
134      IF (GS.EQ.'DROP') GO TO 16
135      IF (GS.EQ.'NO')GO TO 2000
136      GO TO 5
137 1000  M=M2-M1+1
138      SUM1=0
139      SUM2=0
140      DO 1050 I=1,M
141      J=M1+I-1
142      SUM1=SUM1+X(J)
143      SUM2=SUM2+(I-1)*X(J)
144 1050  CONTINUE
145      SUM2=SUM2*DELTA
146      CON1=DELTA*M*(M-1)/2
147      CON2=DELTA*DELTA*M*(M-1)*((2*M)-1)/6
148      AXIS0=(SUM2-(SUM1*CON2/CON1))/(CON1-(M*CON2/CON1))
149      SLOPE=((M*AXIS0)-SUM1)/CON1
150      IF (SLOPE.LT..1)GO TO 870
151      T60=60./SLOPE
152      SUM1=0
153      DO 1100 I=1,M
154      J=M1+I-1
155      CON3=X(J)+SLOPE*(I-1)*DELTA-AXIS0
156      SUM1=SUM1+(CON3*CON3)
157 1100  CONTINUE
158      VRMS=SQRT(SUM1/N)
159      VSLOPE=VRMS*SQRT(M/((M*CON2)-(CON1*CON1)))
160      VAXIS0=VRMS*SQRT(CON2/M)
161      VT60=T60*VSLOPE/SLOPE
162      GO TO RETRN
163 2000  WRITE (L,2010)(ARRAY1(I),I=1,15)
164      WRITE (L,2010)(ARRAY1(I),I=16,30)
165 2010  FORMAT (15F8.4)
166 2020  RETURN
167      END
```

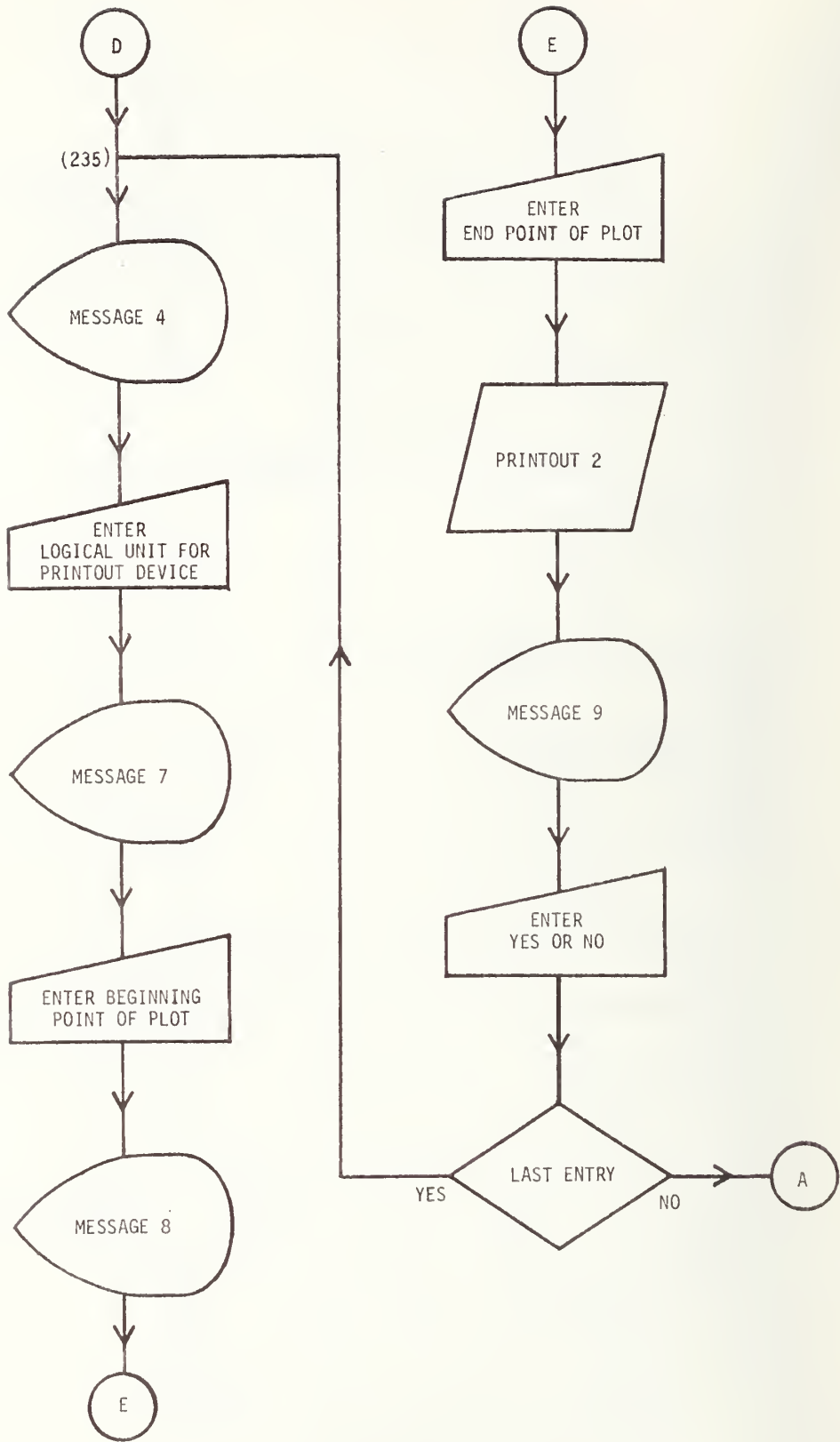

APPENDIX E

Subroutine DSDATA Flow Chart, Terminal Messages, Printouts, and Listings

FLOW CHART: SUBROUTINE DSDATA







CRT TERMINAL MESSAGES: SUBROUTINE DSDATA

MESSAGE NO.	FORMAT NO.	MESSAGE
1	20	DO YOU WANT TO SEE ANOTHER BAND OF DATA?
2	50	YOU HAVE ENTERED AN IMPROPER BAND NUMBER
3	70	ENTER BAND NUMBER
4	100	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
5	200	DO YOU WANT ANOTHER PRINT OF THIS TABLE?
6	230	DO YOU WANT A PLOT OF BAND 30?
7	238	ENTER BEGINNING POINT OF DESIRED PLOT (3 DIGITS)
8	240	ENTER END POINT OF DESIRED PLOT (3 DIGITS)
9	260	DO YOU WANT ANOTHER PLOT OF BAND 30?

PRINTOUT 1 - SUBROUTINE DSDATA

BAND 36										4000 HZ	DECAY NO.	1
1	2	3	4	5	6	7	8	9	10			
46.00	46.25	46.25	45.25	46.50	46.25	45.75	46.00	46.25	45.75		1	
45.50	47.25	46.75	45.25	46.00	47.00	45.25	46.00	45.25	45.75		2	
46.00	44.75	44.75	45.50	45.25	46.75	46.25	45.50	45.75	46.25		3	
44.75	47.00	45.00	45.00	43.75	47.25	46.50	46.75	46.25	45.00		4	
44.75	42.25	40.75	40.25	37.00	35.25	34.25	31.25	29.75	28.00		5	
26.75	25.25	22.50	21.50	19.75	17.75	16.75	13.50	13.00	10.00		6	
9.00	5.25	6.25	4.25	3.50	0.50	0.00	0.00	0.00	0.00		7	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		8	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		9	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		10	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		11	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		12	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		13	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		14	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		15	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		16	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		17	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		18	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		19	
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00		20	

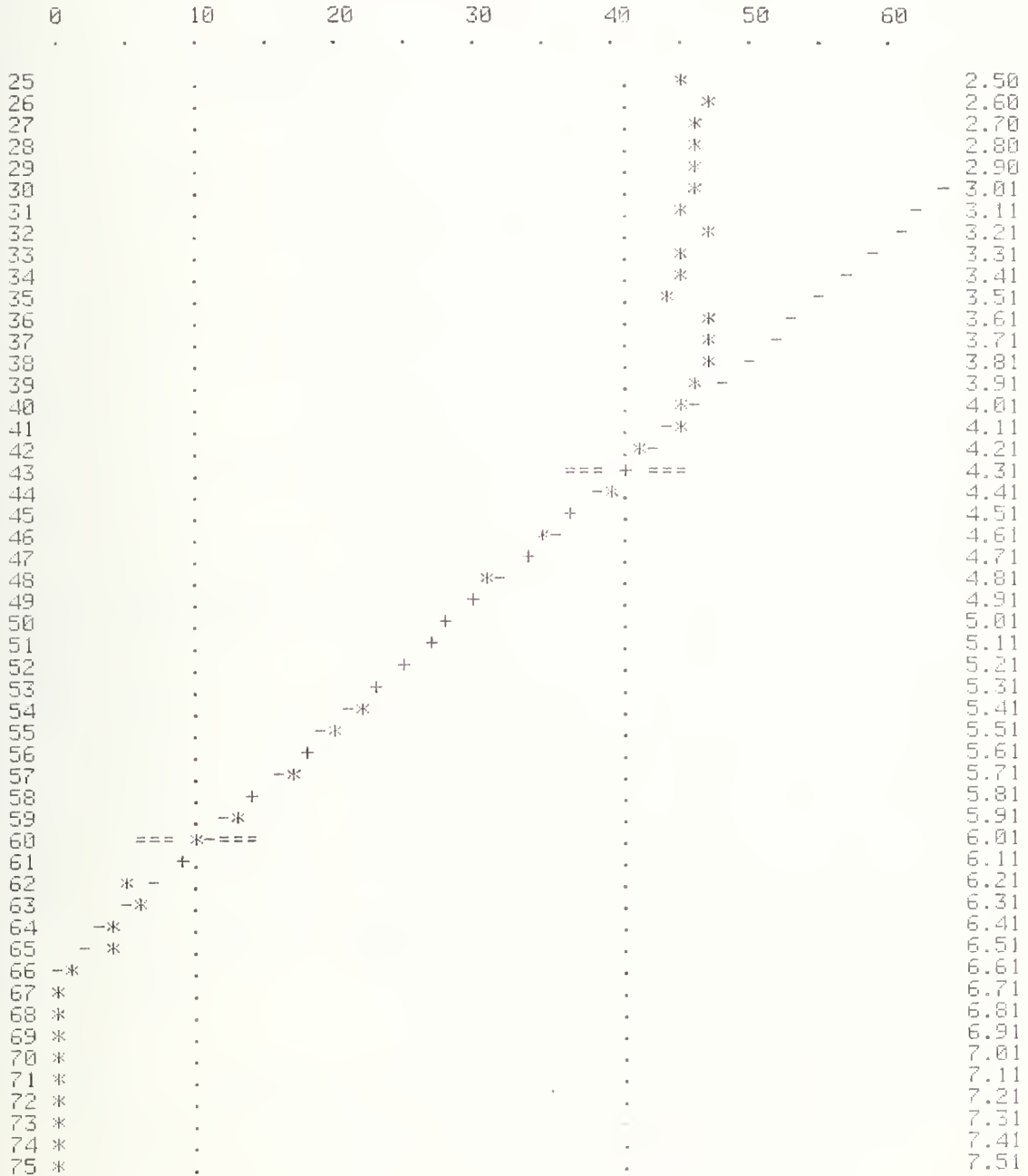
PRINTOUT 2 - SUBROUTINE DSDATA

BAND 36

4000 HZ

DECAY NO. 1

SOUND LEVEL (DB)



POINT NO.

REVERB. TIME = 3.37 SECONDS

TIME (SEC)

SUBROUTINE DSDATA

```

1 $ASSM
2 DSDATA PROG DSDATA - SUBROUTINE FOR DISPLAYING DECAY DATA POINTS
3 $FORT
4 C
5 C -----
6 C OTHER SUBROUTINES CALLED: BAND
7 C -----
8 SUBROUTINE DSDATA(X,X0,ARRAY1,ARRAY2,FREQ,N,NDAYS,LINE,DELTA,
9 150,S1,CUPPER,CLOWER,MM1,MM2,INPRDG)
10 INTEGER*2 B,LU,N,X0(6060),FREQ(30),LINE(65),MM1(30),MM2(30)
11 INTEGER*2 BLANK,STAR,PLUS,MINUS,DOT,EQUALS,FF
12 DIMENSION X(200),ARRAY1(30),ARRAY2(30),S0(30,12),S1(30)
13 DATA BLANK,STAR,PLUS,MINUS,DOT,EQUALS,FF/'X'2000',X'2000',X'2E00',X'3D00',X'0C00' /
14 GO TO 60
15 10 WRITE (5,20)
16 20 FORMAT ('DO YOU WANT TO SEE ANOTHER BAND OF DATA?')
17 READ (5,30) GS
18 30 FORMAT (A1)
19 IF (GS.F0.'NO')GO TO 900
20 GO TO 60
21 40 WRITE (5,50)
22 50 FORMAT ('YOU HAVE ENTERED AN IMPROPER BAND NUMBER')
23 60 WRITE (5,70)
24 70 FORMAT ('ENTER BAND NUMBER')
25 READ (5,80)B
26 80 FORMAT (I2)
27 IF ((B.LT.14).OR.(B.GT.43)) GO TO 40
28 90 WRITE (5,100)
29 100 FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
30 READ (5,110)LU
31 110 FORMAT (I1)
32 IB=B-13
33 CALL BAND(B,N,X,X0,6060)
34 WRITE (LU,130)B,FREQ(IB),NDAYS,(I,I=1,10)
35 130 FORMAT ('X/X/X-10X.'BAND ',I2,10X,15.'HZ',10X,'DECAY NO.',14/
36 1X)X2,10(12,5X)X)
37 NLLINES=N*10
38 DO 150 I=1,NLLINES
39 J1=(I-1)/10+1
40 J2=(I-1)/10+10
41 WRITE (LU,140)(X(I),I=J1,J2),I
42 140 FORMAT (10(F6,2X),3X,12)
43 150 CONTINUE
44 WRITE (LU,160)FF
45 160 FORMAT (A1)
46 WRITE (5,200)
47 200 FORMAT ('DO YOU WANT ANOTHER PRINT OF THIS TABLE?')
48 READ (5,30) GS
49 IF (GS.NE.'NO')GO TO 90
50 WRITE (5,230)B
51 230 FORMAT ('DO YOU WANT A PLOT OF BAND ',I2,' ?')
52 READ (5,30)GS
53 IF (GS.EQ.'NO') GO TO 10
54 235 WRITE (5,100)
55 READ (5,110)LU
56 WRITE (5,238)
57 238 FORMAT ('ENTER BEGINNING POINT OF DESIRED PLOT (3 DIGITS)')
58 READ (5,250)H1
59 WRITE (5,240)
60 240 FORMAT ('ENTER END POINT OF DESIRED PLOT (3 DIGITS)')

```

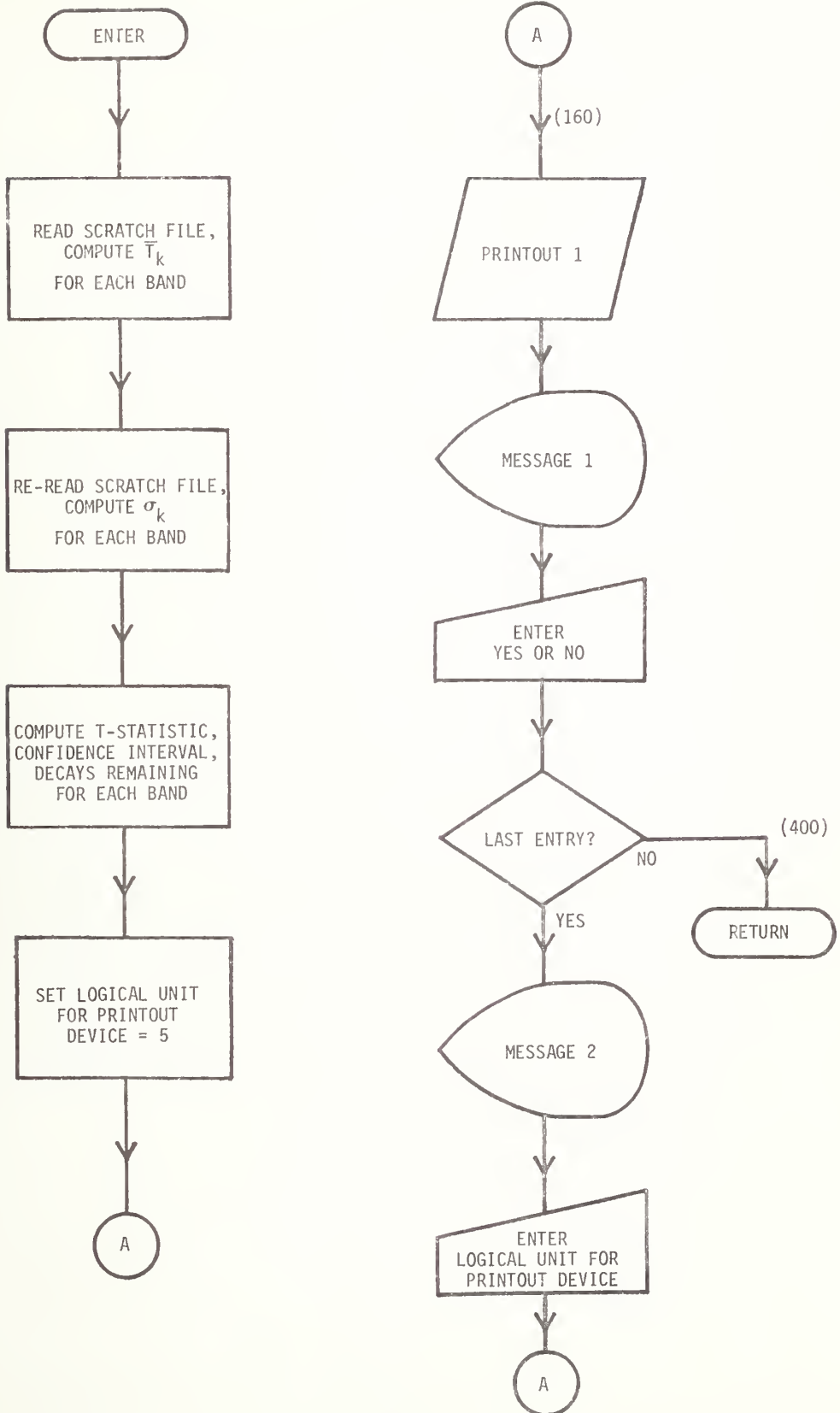

DSDATA-2

```
61 READ (5,250)N2
62 250 FORMAT (I3)
63 IF (ARRAY1(IB).LT.0.01) GO TO 370
64 SLOPE=60./ARRAY1(IB)
65 370 WRITE (LU,380)B,FREQ(IB),NDCAYS
66 380 FORMAT (I0,' BAND ',I2,10%,I5,' HZ',10%,*DECAY NO.',I4,4%,
67 1,25%,*SOUND LEVEL (DB)',X,4%,*0',9%,*10',8%,*20',8%,*30',8%,
68 2',40',8%,*50',8%,*60',4%,I3(' ',4X)/X)
69 J3=S1(IB)-CUPPER+1.5
70 J4=S0(IB,MNRIG)+CLOWER+1.5
71 390 DO 500 I=H1,N2
72 DO 400 I=1,65
73 LINE(J)=BLANK
74 400 CONTINUE
75 TIME=I*DELTA
76 J1=X(I)+1.5
77 IF (ARRAY1(IB).LT.0.01) GO TO 405
78 LINE(J3)=DOT
79 LINE(J4)=DOT
80 IF ((I.NE.MN1(IB)).AND.(I.NE.MN2(IB))) GO TO 404
81 J5=J1-4
82 J6=J1-2
83 DO 402 J=J5,J6
84 LINE(J)=EQUALS
85 402 CONTINUE
86 J5=J1+2
87 J6=J1+4
88 DO 403 J=J5,J6
89 LINE(J)=EQUALS
90 403 CONTINUE
91 404 J2=1.5+ARRAY2(IB)-SLOPE*(I-1)*DELTA
92 IF ((J2.LT.1).OR.(J2.GT.65)) GO TO 405
93 IF (I2.EQ.J1) GO TO 410
94 LINE(J2)=MINUS
95 405 LINE(J1)=STAR
96 GO TO 420
97 410 LINE(J1)=PLUS
98 420 WRITE (LU,430)I,(LINE(J),J=1,65),TIME
99 430 FORMAT (I3,X,65A1,F5,2)
100 500 CONTINUE
101 IF (ARRAY1(IB).LT.0.01) GO TO 520
102 WRITE (LU,510) ARRAY1(IB),FF
103 510 FORMAT (X/'POINT NO.',T24,'REVERB. TIME = ',F6,2,
104 1' SECONDS',T68,'TIME (SEC)'/A1)
105 GO TO 600
106 520 WRITE (LU,530) FF
107 530 FORMAT (X/'POINT NO.',T68,'TIME (SEC)'/A1)
108 600 WRITE (5,610) B
109 610 FORMAT ('DO YOU WANT ANOTHER PLOT OF BAND ',I2,' ?')
110 READ (5,30)GS
111 IF (GS.EQ.'NO') GO TO 10
112 GO TO 235
113 900 RETURN
114 END
```

APPENDIX F

Subroutine AVRGE Flow Chart, Terminal Messages, Printout, and Listings

FLOW CHART: SUBROUTINE AVRGE



CRT TERMINAL MESSAGES: SUBROUTINE AVRGE

MESSAGE NO.	FORMAT NO.	MESSAGE
1	320	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
2	340	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE

PRINTOUT 1 - SUBROUTINE AVRGE

NUMBER OF DECAYS CONDUCTED = 24

BAND NO.	FREQ. (HZ)	REVERB TIME (SEC)	STD. DEV. (SEC)	CONFIDENCE INTERVAL (% +OR-)	NO. OF DECAYS	DECAYS REMAINING
14	25	22.11	3.66	7.7	20	262
15	32	17.82	4.02	10.3	21	500
16	40	14.85	3.53	10.0	24	550
17	50	9.51	2.55	11.6	23	709
18	63	4.14	0.57	5.8	24	166
19	80	4.57	0.58	5.3	24	137
20	100	3.02	0.45	6.2	24	197
21	125	3.71	0.35	4.0	24	67
22	160	4.51	0.26	2.4	24	9
23	200	5.19	0.21	1.7	24	0
24	250	6.31	0.17	1.2	24	0
25	315	6.80	0.15	1.0	24	0
26	400	6.97	0.17	1.0	24	0
27	500	6.76	0.15	0.9	24	0
28	630	5.10	0.16	1.1	24	0
29	800	5.51	0.11	0.8	24	0
30	1000	5.94	0.08	0.6	24	0
31	1250	5.92	0.07	0.5	24	0
32	1600	5.86	0.07	0.5	24	0
33	2000	5.19	0.09	0.8	24	0
34	2500	4.63	0.07	0.6	24	0
35	3150	4.01	0.06	0.7	24	0
36	4000	3.94	0.05	0.6	24	0
37	5000	3.70	0.04	0.6	24	0
38	6300	2.12	0.05	0.9	24	0
39	8000	1.58	0.02	0.7	24	0
40	10000	1.16	0.03	1.0	24	0
41	12500	0.84	0.03	1.3	24	0
42	16000	0.61	0.04	2.6	24	19
43	20000				0	

SUBROUTINE AVRGE

```

1 $ASSM
2 AVRGE  PROG AVRGE — SUBROUTINE FOR AVERAGING THE RESULTS OF MANY DECAYS
3 $FORT
4 C
5 C  SUBPROGRAMS CALLED:  TINORM,STUDIN
6 C
7 C  SUBROUTINE AVRGE(X,X0,ARRAY1,ARRAY2,NDAYS,CINTER,CLEVEL,
8 IFREQ,BLOW,BHIGH)
9 INTEGER*2 X0(60),FREQ(30),B,BLOW,BHIGH,B1,B2,LU,FF
10 DIMENSION X(200),ARRAY1(30),ARRAY2(30)
11 DATA FF/X*0C00*
12 ALPHA=1.-CLEVEL*100.
13 DO 10 I=1,30
14 X(I)=0.
15 X0(I)=0
16 10 CONTINUE
17 REWIND 1
18 DO 20 J=1,NDAYS
19 READ (1,15)(ARRAY1(I),I=1,15)
20 READ (1,15)(ARRAY1(I),I=16,30)
21 15 FORMAT (15F8.4)
22 DO 20 I=1,30
23 IF (ARRAY1(I).EQ.0.) GO TO 20
24 X(I)=X(I)+ARRAY1(I)
25 X0(I)=X0(I)+1
26 20 CONTINUE
27 DO 30 I=1,30
28 IF (X0(I).EQ.0) GO TO 30
29 ARRAY1(I)=X(I)/X0(I)
30 30 CONTINUE
31 DO 40 I=1,30
32 X(I)=0.
33 40 CONTINUE
34 REWIND 1
35 DO 50 J=1,NDAYS
36 READ (1,15)(ARRAY2(I),I=1,15)
37 READ (1,15)(ARRAY2(I),I=16,30)
38 DO 50 I=1,30
39 IF (ARRAY2(I).EQ.0.) GO TO 50
40 X(I)=X(I)+ARRAY1(I)-ARRAY2(I)*X(ARRAY1(I)-ARRAY2(I))
41 50 CONTINUE
42 OPDIN=TINORM(1.-ALPHA/2.)
43 HDF=0
44 DO 100 I=1,30
45 IF (X0(I).LT.2) GO TO 100
46 IF (HDF.EQ.(X0(I)-1)) GO TO 70
47 HDF=X0(I)-1
48 TALPHA=STUDIN(CALPHA,HDF,4,5)
49 70 ARRAY2(I)=SORT(X(I),HDF)
50 X(I+30)=100.*ARRAY2(I)*TALPHA+(ARRAY1(I)*SORT(1.,X0(I)))
51 IF (X(I+30).LT.CINTER) GO TO 80
52 X0(I+30)=0
53 GO TO 100
54 80 C=(TALPHA+OPDIN)*2.
55 C=100.*(X(ARRAY2(I))+ARRAY1(I)+CINTER)
56 C=C+C*.5-X0(I)
57 IF (C.LE.32767.) GO TO 90
58 X0(I+30)=32,67
59 GO TO 100
60 90 X0(I+30)=C

```

AVRGE-2

```

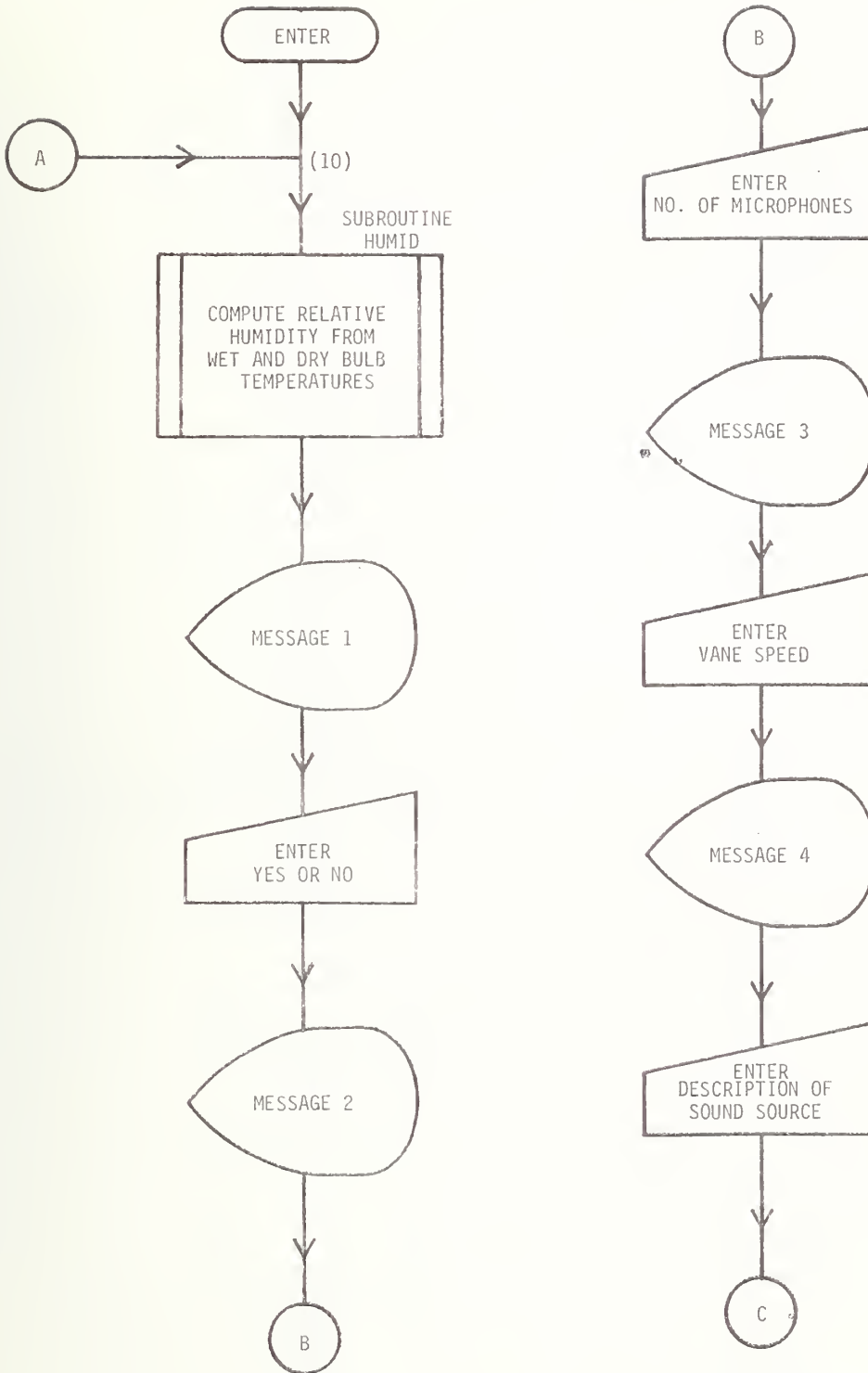
61      IF (X0(1+30).LT.1) X0(1+30)=1
62 100   CONTINUE
63      LU=5
64 160   WRITE (LU,170)MDCAYS
65 170   FORMAT (1,4X,T1,'NUMBER OF DECAYS CONDUCTED =',14/X,X,X,X
66      1,'BAND',T14,'REVERB',T31,'CONFIDENCE',T43,
67      2'NO.',AF',T52,'DECAYS',
68      3'2.',HO.',T7,'FRF0.',T15,'TIME',T22,'STD.DEV.',T32,'INTERVAL',
69      4T43,'DECAYS',T51,'REMAINING',
70      5B3,'H2',T7,'S60',T4X,'(SEC',T32,'(2,+OR-)',X)
71 180   DO 300 B=BLOW,BHIGH
72      I=B-13
73      IF (X0(I).EQ.0) GO TO 220
74      IF (X0(I).EQ.1) GO TO 240
75      WRITE (LU,200)B,FREQ(I),ARRAY1(I),ARRAY2(I),X(I+30),X0(I),X0(I+30)
76 200   FORMAT (7,12,2X,15,2X,F6.2,3X,F6.2,5X,F5.1,5X,14,5X,15)
77      GO TO 300
78 220   ARRAY1(I)=0.
79      ARRAY2(I)=0.
80      WRITE (LU,230)B,FREQ(I)
81 230   FORMAT (7,12,2X,15,35X,'0')
82      GO TO 300
83 240   ARRAY2(I)=0.
84      WRITE (LU,250)B,FREQ(I),ARRAY1(I)
85 250   FORMAT (7X,12,2X,15,2X,F6.2,37X,'1')
86 300   CONTINUE
87      WRITE (LU,310) FF
88 310   FORMAT (A1)
89      WRITE (5,320)
90 320   FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
91      READ (5,330)GS
92 330   FORMAT (A4)
93      IF (GS.EQ.'NO') GO TO 400
94      WRITE (5,340)
95 340   FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
96      READ (5,350)LU
97 350   FORMAT (I1)
98      GO TO 160
99 400   RETURN
100    END

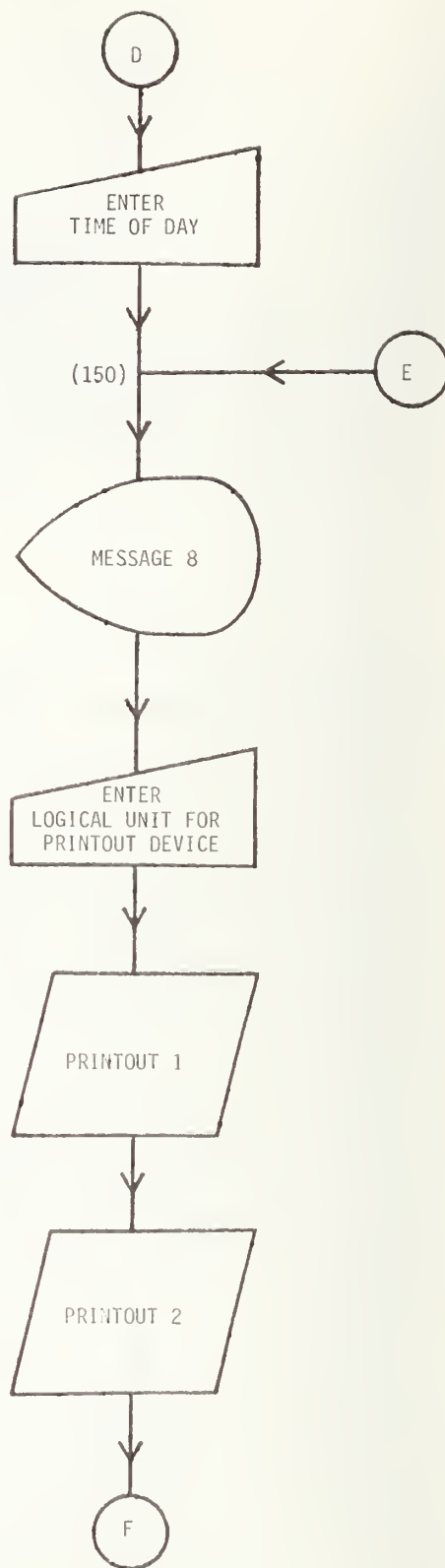
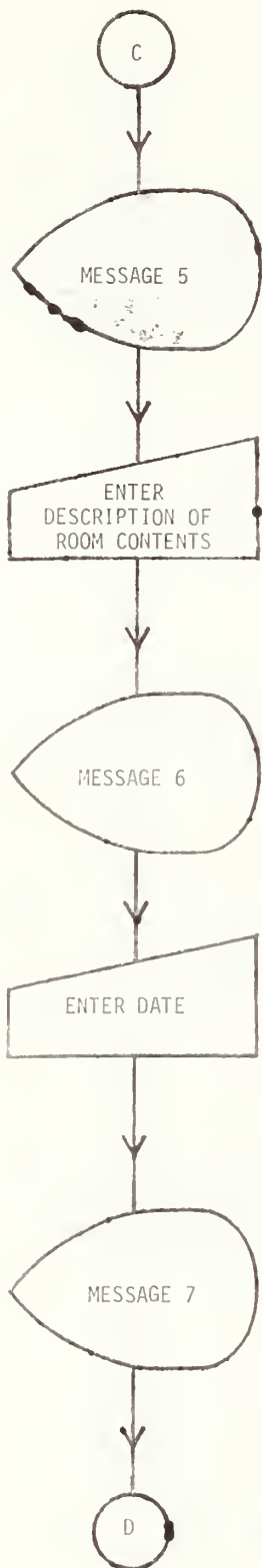
```

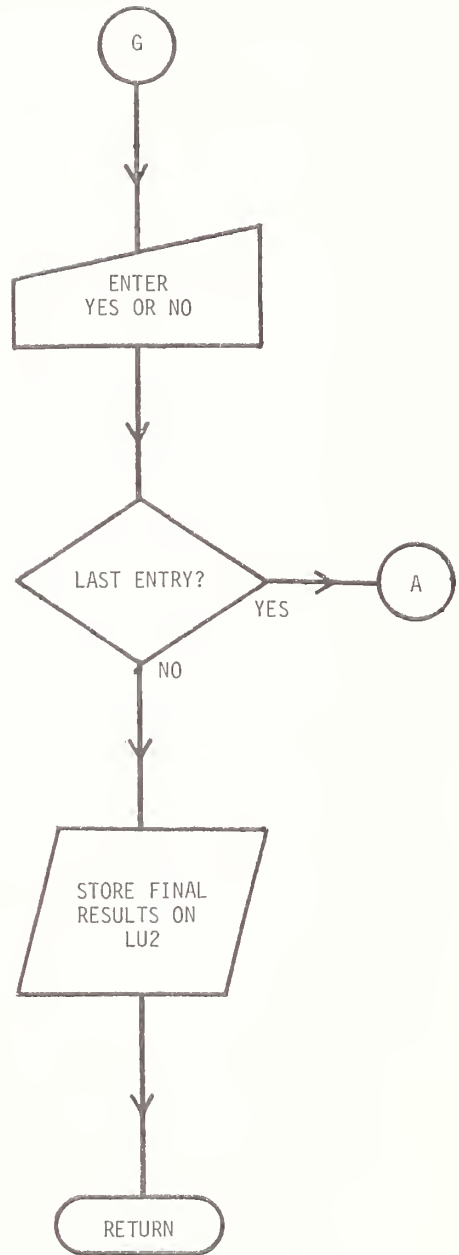
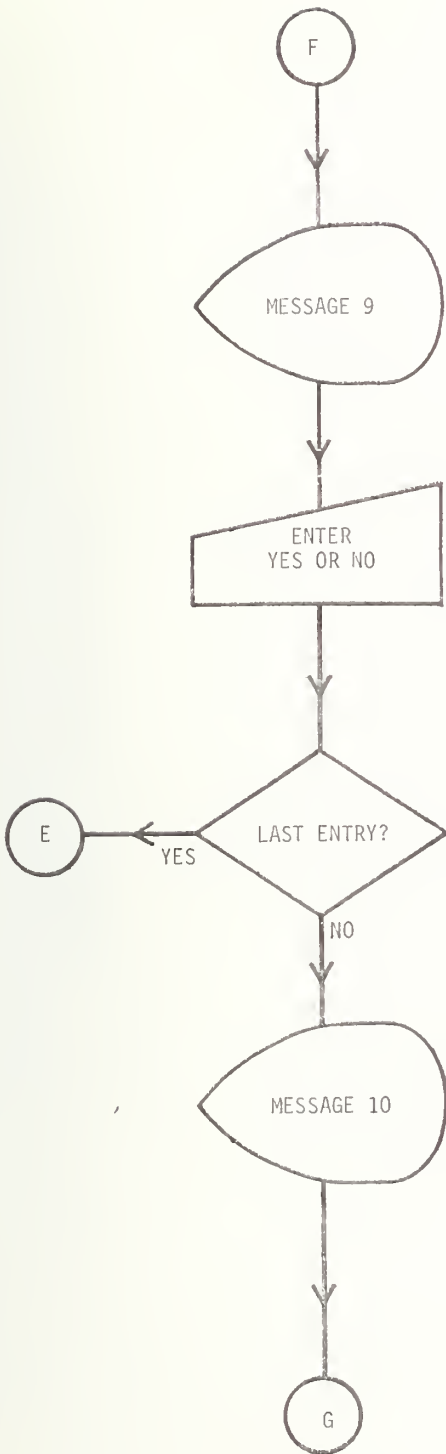
APPENDIX G

Subroutine RESULTS Flow Chart, Terminal Messages, Printouts and Listings

FLOW CHART: SUBROUTINE RESULTS



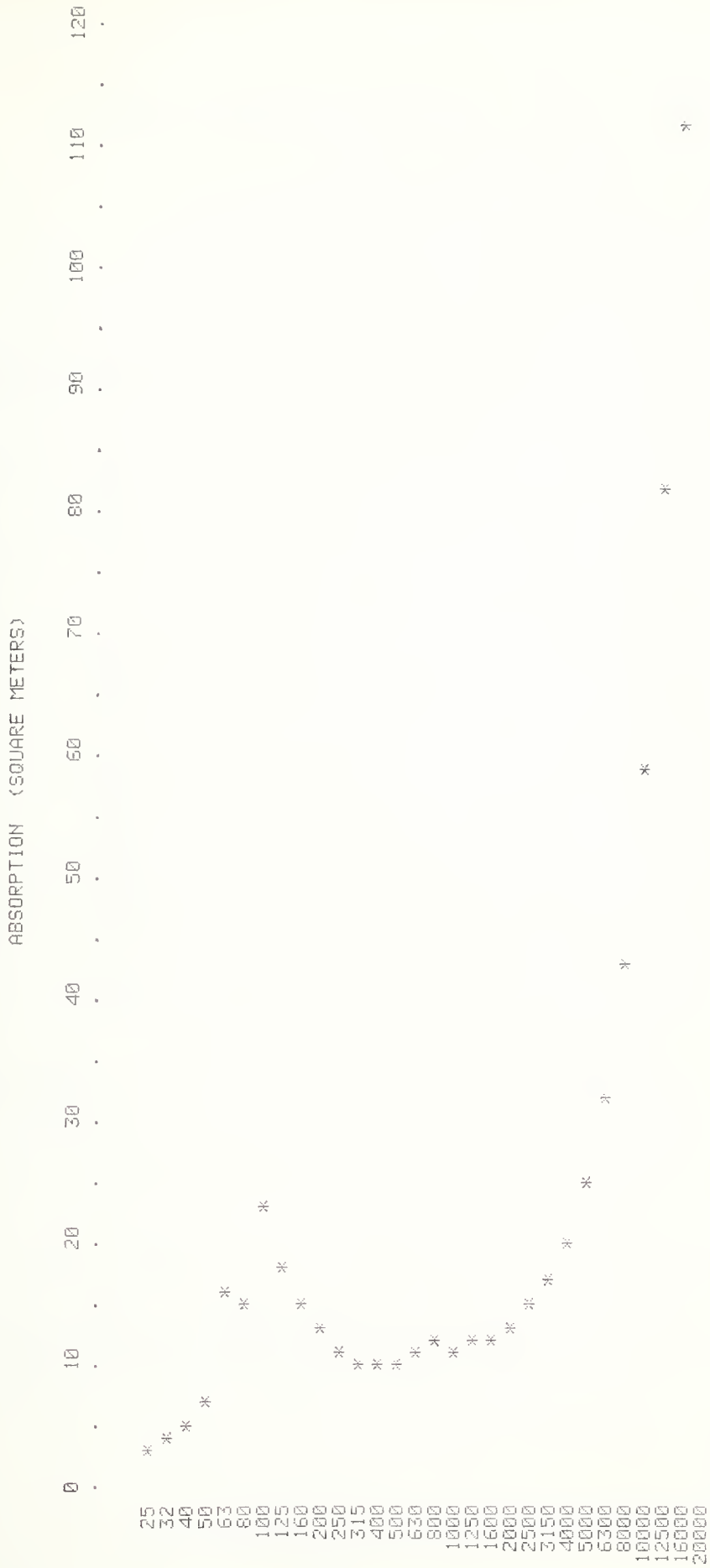




CRT TERMINAL MESSAGES: SUBROUTINE RESULTS

MESSAGE NO.	FORMAT NO.	MESSAGE
1	20	WERE THESE DECAYS PERFORMED IN THE MODEL REVERBERATION ROOM?
2	22	ENTER NO. OF MICROPHONES USED IN THIS MEASUREMENT (2 DIGITS)
3	30	ENTER VANE SPEED IN REV/MIN (XX.X)
4	50	DESCRIBE SOUND SOURCE (1 LINE)
5	60	DESCRIBE ROOM CONTENTS (1 LINE)
6	80	ENTER DATE
7	100	ENTER TIME OF DAY
8	160	ENTER LOGICAL UNIT FOR PRINTOUT DEVICE
9	900	DO YOU WANT ANOTHER PRINT OF THESE RESULTS?
10	920	DO YOU WANT TO CORRECT ANY TYPING MISTAKES IN THE PRINTOUT?

PRINTOUT 1 - SUBROUTINE RESULTS



PRINTOUT 2 - SUBROUTINE RESULTS

REVERBERATION DECAY MEASUREMENTS

BAND NO.	FREQ. (HZ)	ABSORPTION (METERS ²)	REVERB. TIME (SEC)	STD. DEV. (SEC)	RELATIVE STD. DEV. (%)	CONFIDENCE INTERVAL (% +OR-)	NO. OF DECAYS
14	25	3.08	22.11	3.66	16.56	7.7	20
15	32	3.83	17.82	4.02	22.57	10.3	21
16	40	4.59	14.85	3.53	23.78	10.0	24
17	50	7.17	9.51	2.55	26.82	11.6	23
18	63	16.46	4.14	0.57	13.68	5.8	24
19	80	14.92	4.57	0.58	12.59	5.3	24
20	100	22.58	3.02	0.45	14.75	6.2	24
21	125	18.38	3.71	0.35	9.45	4.0	24
22	160	15.12	4.51	0.26	5.68	2.4	24
23	200	13.14	5.19	0.21	3.96	1.7	24
24	250	10.80	6.31	0.17	2.72	1.2	24
25	315	10.03	6.80	0.15	2.26	1.0	24
26	400	9.79	6.97	0.17	2.38	1.0	24
27	500	10.09	6.76	0.15	2.21	0.9	24
28	630	11.04	6.18	0.16	2.60	1.1	24
29	800	11.53	5.91	0.11	1.89	0.8	24
30	1000	11.49	5.94	0.08	1.35	0.6	24
31	1250	11.52	5.92	0.07	1.26	0.5	24
32	1600	12.04	5.66	0.07	1.26	0.5	24
33	2000	13.13	5.19	0.09	1.83	0.8	24
34	2500	14.71	4.63	0.07	1.41	0.6	24
35	3150	17.02	4.01	0.06	1.56	0.7	24
36	4000	20.40	3.34	0.05	1.48	0.6	24
37	5000	25.26	2.70	0.04	1.41	0.6	24
38	6300	32.09	2.12	0.05	2.12	0.9	24
39	8000	43.05	1.58	0.02	1.57	0.7	24
40	10000	58.69	1.10	0.03	2.35	1.0	24
41	12500	81.50	0.84	0.03	3.02	1.3	24
42	16000	111.98	0.61	0.04	6.52	2.8	24
43	20000						0

TIME BETWEEN SAMPLES = 0.10017 SECONDS
 NO. OF SAMPLES IN SIGNAL MEASUREMENT = 40
 NO. OF SAMPLES IN DECAY MEASUREMENT = 160
 NO. OF DECAYS CONDUCTED = 24
 NO. OF MICROPHONES = 12

CURVE FITTING LIMITS:
 5.0 DB DOWN FROM SIGNAL LEVEL
 10.0 DB UP FROM NOISE LEVEL

CONFIDENCE LEVEL = 95.00 %

DRY BULB TEMPERATURE = 21.8 DEG. C (71.2 DEG. F)
 WET BULB TEMPERATURE = 16.7 DEG. C (62.0 DEG. F)
 RELATIVE HUMIDITY = 59.7 %
 BAROMETRIC PRESSURE = 980.8 MILLIBARS (735.7 MM HG)

WIND SPEED: 7.5 MPH
 SOUND SOURCE: MITSUBISHI 100W TL-10 SPEAKER IN (0,0,1) CORNER; PINK NOISE; 10 VRMS
 ROOM CONTENTS: 12 LOW FREQUENCY ABSORBERS

DATE: 3-18-77
 TIME: 7:30 A.M.

SUBROUTINE RESULTS

```

1 $ASSM
2 RESULTS PROG RESULTS - SUBROUTINE FOR DISPLAYING RESULTS OF MANY DECAYS
3 $FORT
4 C -----
5 C OTHER SUBROUTINES CALLED: HUMID
6 C -----
7 SUBROUTINE RESULTS(X,X0,ARRAY1,ARRAY2,FREQ,NDCAYS,CLEVEL,NSIG,
8 INDEC,LINE,DELTA,BLOW,BHIGH,CUPPER,CLOWER)
9 INTEGER*2 B,NSIG,NDEC,FREQ(30),LU,FF,BLANK,STAR,PLUS,
10 ILINE(12),X0(30),BLOW,BHIGH
11 DIMENSION X(60),ARRAY1(30),ARRAY2(30),DATE(5),TIME(5),
12 ISTATUS(18),SPEAKR(18)
13 DATA FF,BLANK,STAR,PLUS/'0000','2000','2400','2800'/
14 10 CALL HUMID(HUMIDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,PRESS1,PRESS2,
15 IVSOUND)
16 WRITE (5,20)
17 20 FORMAT ('WERE THESE DECAYS PERFORMED IN THE MODEL REVERBERATION
18 ROOM?')
19 READ (5,910) GS
20 MODEL=1
21 IF (GS.EQ.'NO') MODEL=0
22 WRITE (5,22)
23 22 FORMAT ('ENTER NO. OF MICROPHONES USED IN THIS MEASUREMENT
24 (2 DIGITS)')
25 READ (5,24) NMIC
26 24 FORMAT (I2)
27 WRITE (5,30)
28 30 FORMAT ('ENTER VANE SPEED IN REV. MIN (XX.X)')
29 READ (5,40) VANE
30 40 FORMAT (F10.5)
31 WRITE (5,50)
32 50 FORMAT ('DESCRIBE SOUND SOURCE (1 LINE)')
33 READ (5,70) (SPEAKR(I),I=1,18)
34 WRITE (5,60)
35 60 FORMAT ('DESCRIBE ROOM CONTENTS (1 LINE)')
36 READ (5,70) (STATUS(I),I=1,18)
37 70 FORMAT (18A4)
38 WRITE (5,80)
39 80 FORMAT ('ENTER DATE')
40 READ (5,90) (DATE(I),I=1,5)
41 90 FORMAT (5A4)
42 WRITE (5,100)
43 100 FORMAT ('ENTER TIME OF DAY')
44 READ (5,90) (TIME(I),I=1,5)
45 VOLUME=424.8
46 IF (MODEL.EQ.1) VOLUME = 6.6375
47 DO 140 I=1,30
48 IF (X0(I).EQ.0) GO TO 140
49 X(I)=.921*60*VOLUME/(ARRAY1(I)*VSOUND)
50 140 CONTINUE
51 150 WRITE (5,160)
52 160 FORMAT ('ENTER LOGICAL UNIT FOR PRINTOUT DEVICE')
53 READ (5,170) LU
54 170 FORMAT (I1)
55 WRITE (LU,300)
56 300 FORMAT ('X'//51X,'ABSORPTION (SQUARE METERS)'//X/6X,'0',
57 19X,'10',8X,'20',8X,'30',8X,'40',8X,'50',8X,'60',8X,'70',8X,
58 2'80',8X,'90',7X,'100',7X,'110',7X,'120'//6X,25(' ',4X)//X/X)
59 DO 350 B=BLOW,BHIGH
60 I=B-13

```

RESULTS-2

```

61      DO 310 J=1,121
62      LINE(J)=BLANK
63 310   CONTINUE
64      IF (X0(I).EQ.0) GO TO 330
65      J=X(I)+1.5
66      IF (J.LT.1) GO TO 330
67      IF (J.GT.121) GO TO 320
68      LINE(J)=STAR
69      GO TO 330
70 320   LINE(121)=PLUS
71 330   WRITE(LU,340)FREQ(I), (LINE(J),J=1,121)
72 340   FORMAT(15,X,121A1)
73 350   CONTINUE
74      WRITE (LU,360)FF
75 360   FORMAT (X//FREQUENCY//2X,*(HZ)//A1)
76      WRITE (LU,500)
77 500   FORMAT (T21,'REVERBERATION DECAY MEASUREMENTS'//X//X//
78 1'BAND',T28,'REVERB.',T47,'RELATIVE',T57,'CONFIDENCE',
79 2T69,'NO. OF'
80 3T2,'NO.',T8,'FREQ.',T15,'ABSORPTION',T29,'TIME',T37,'STD.DEV.',
81 4T47,'STD.DEV.',T58,'INTERVAL',T69,'DECAYS'/
82 5T8,'(HZ)',T15,'(METERS*2)',T28,'(SEC)',T38,'(SEC)',
83 6T49,'(%)',T58,'(%,+OR-)'//)
84      DO 600 B=BLOW,BHIGH
85      I=B-13
86      IF (X0(I).EQ.0)GO TO 550
87      IF (X0(I).EQ.1)GO TO 570
88      C=100.*(ARRAY2(I)-ARRAY1(I))
89      WRITE (LU,520)E,FREQ(I),X(I),ARRAY1(I),ARRAY2(I),C,X(I+30),X0(I)
90 520   FORMAT (T2,I2,T7,I5,T15,F7.2,T27,F6.2,T37,F6.2,T46,F7.2,
91 1T59,F5.1,T69,I4)
92      GO TO 600
93 550   WRITE (LU,560)B,FREQ(I)
94 560   FORMAT (I2,I3),I5,T72,'0')
95      GO TO 600
96 570   WRITE (LU,580)B,FREQ(I),X(I),ARRAY1(I)
97 580   FORMAT (T2,I2,T7,I5,T15,F7.2,T27,F6.2,T72,'1')
98 600   CONTINUE
99      WRITE (LU,630) DELTA,NSIG,CURPEP,HDEC,CLOWER,NDAYS,
100     INIC,CLEVEL,DTEMP,C,TEMPF
101 630   FORMAT (I2,I3,' TIME BETWEEN SAMPLES = ',F7.5,' SECONDS',
102 1T54,'CURVE FITTING LIMITS:'
103 2'NO. OF SAMPLES IN SIGNAL MEASUREMENT = ',I3,
104 3T48,F5.1,' DB DOWN FROM SIGNAL LEVEL',
105 4'NO. OF SAMPLES IN DECAY MEASUREMENT = ',I3,
106 5T48,F5.1,' DB UP FROM NOISE LEVEL',
107 6'NO. OF DECAYS CONDUCTED = ',I4,
108 7'NO. OF MICROPHONES = ',I2,
109 8'CONFIDENCE LEVEL = ',F5.2,' %',
110 9'IF BULB TEMPERATURE = ',F4.1,' DEG. C (',F4.1,' DEG. F)')
111     IF (NT,IPC,EQ.=1000.) GO TO 640
112     WRITE (LU,650) WTEMP,C,TEMPF
113 650   FORMAT ('MET BULB TEMPERATURE = ',F4.1,' DEG. C
114 1',F4.1,' DEG. F)')
115 640   WRITE (LU,650) HUMDT5,PRESS2,PRESS1
116 650   FORMAT ('RELATIVE HUMIDITY = ',F4.1,' %',
117 1'BAROMETRIC PRESSURE = ',F6.1,' MILLIBARS (',
118 2F5.1,' IN HG)')
119     WRITE (LU,660) VANE,(SPEHR(I),I=1,18),(STATUS(I),I=1,18)
120 660   FORMAT ('VANE SPEED = ',F4.1,' RPM')

```

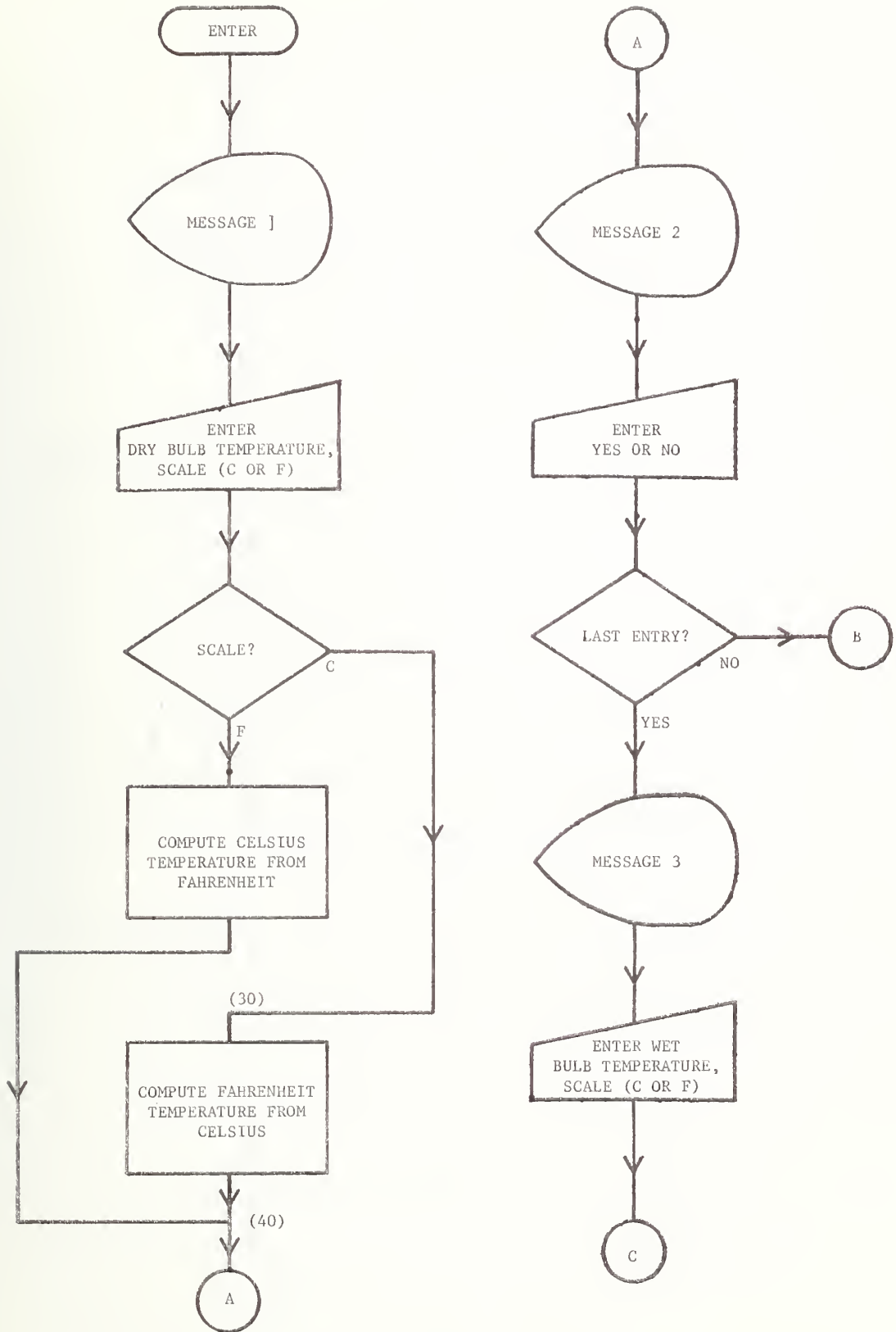

RESULTS-3

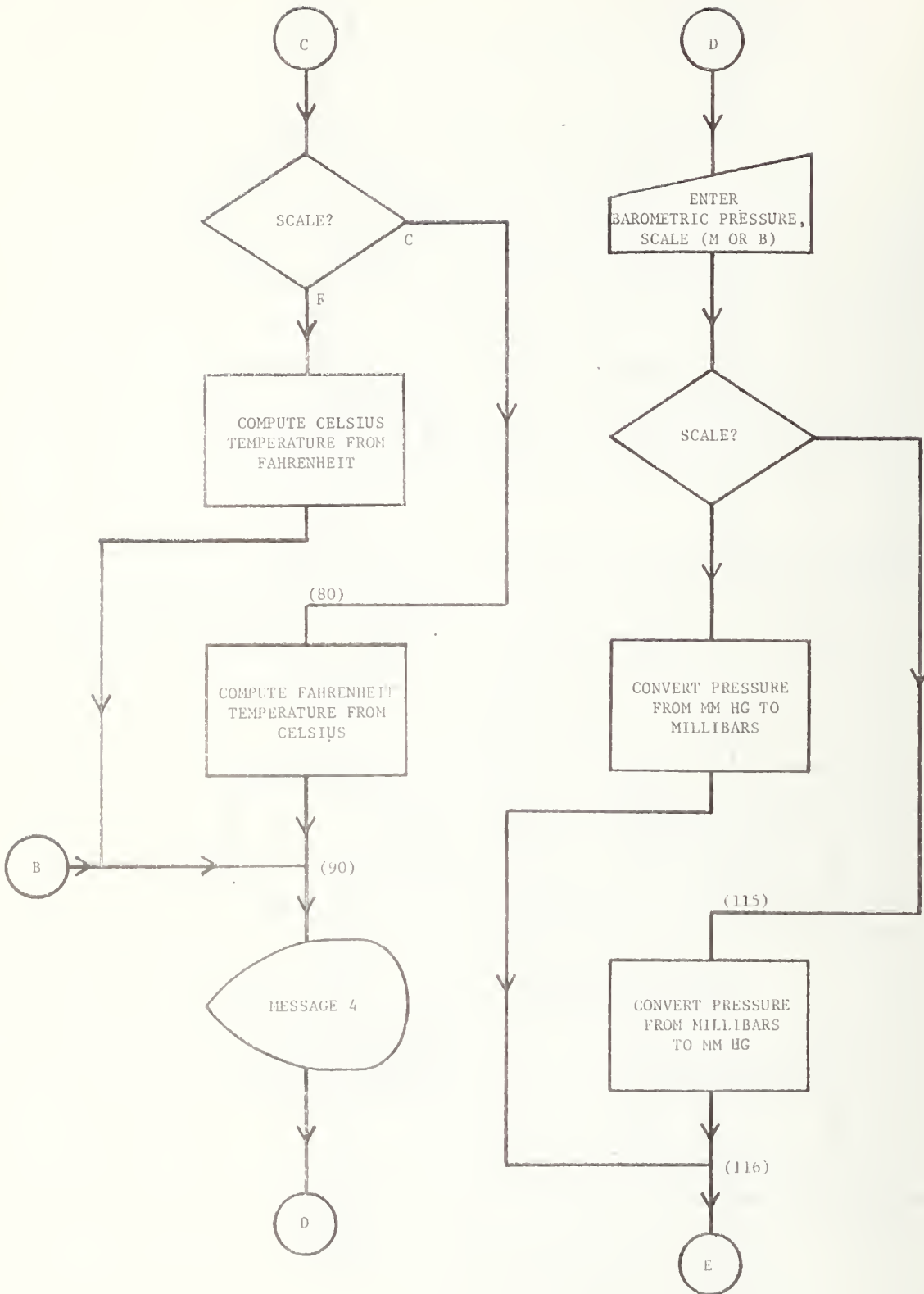
```
121      1'SOUND SOURCE:      *.18A4/'ROOM CONTENTS:  *.18A4)
122      IF (MODEL.EQ.8) GO TO 680
123      WRITE (LU,670)
124 670   FORMAT ('MODEL ROOM')
125 680   WRITE (LU,690) (DATE(I),I=1,5),(TIME(I),I=1,5),FF,FF
126 690   FORMAT ('%2' DATE:  *.5A4/'TIME:  *.5A4/'A1/'A1)
127      WRITE (5,900)
128 900   FORMAT ('DO YOU WANT ANOTHER PRINT OF THESE RESULTS?')
129      READ (5,910)GS
130 910   FORMAT (A4)
131      IF (GS.NE.'NO')GO TO 150
132      WRITE (5,920)
133 920   FORMAT ('DO YOU WANT TO CORRECT ANY TYPING MISTAKES IN THE
134      1 PRINTOUT?')
135      READ (5,910)GS
136      IF (GS.NE.'NO')GO TO 10
137      REWIND 2
138      WRITE (2,925) (DATE(I),I=1,5),(TIME(I),I=1,3)
139 925   FORMAT (5A4,3A4)
140      DO 950 I=1,30
141      B=I+13
142      WRITE (2,930) B,FREQ(I),X(I),ARRAY1(I),ARRAY2(I),X(I+30),X0(I)
143 930   FORMAT (I2,X, I5,X,F8.4,X,F8.4,X,F9.5,X,F6.3,X,I3)
144 950   CONTINUE
145      WRITE (2,960) DTEMPC,HUMDTY,PRESS2
146 960   FORMAT (3F6.1)
147      RETURN
148      END
```

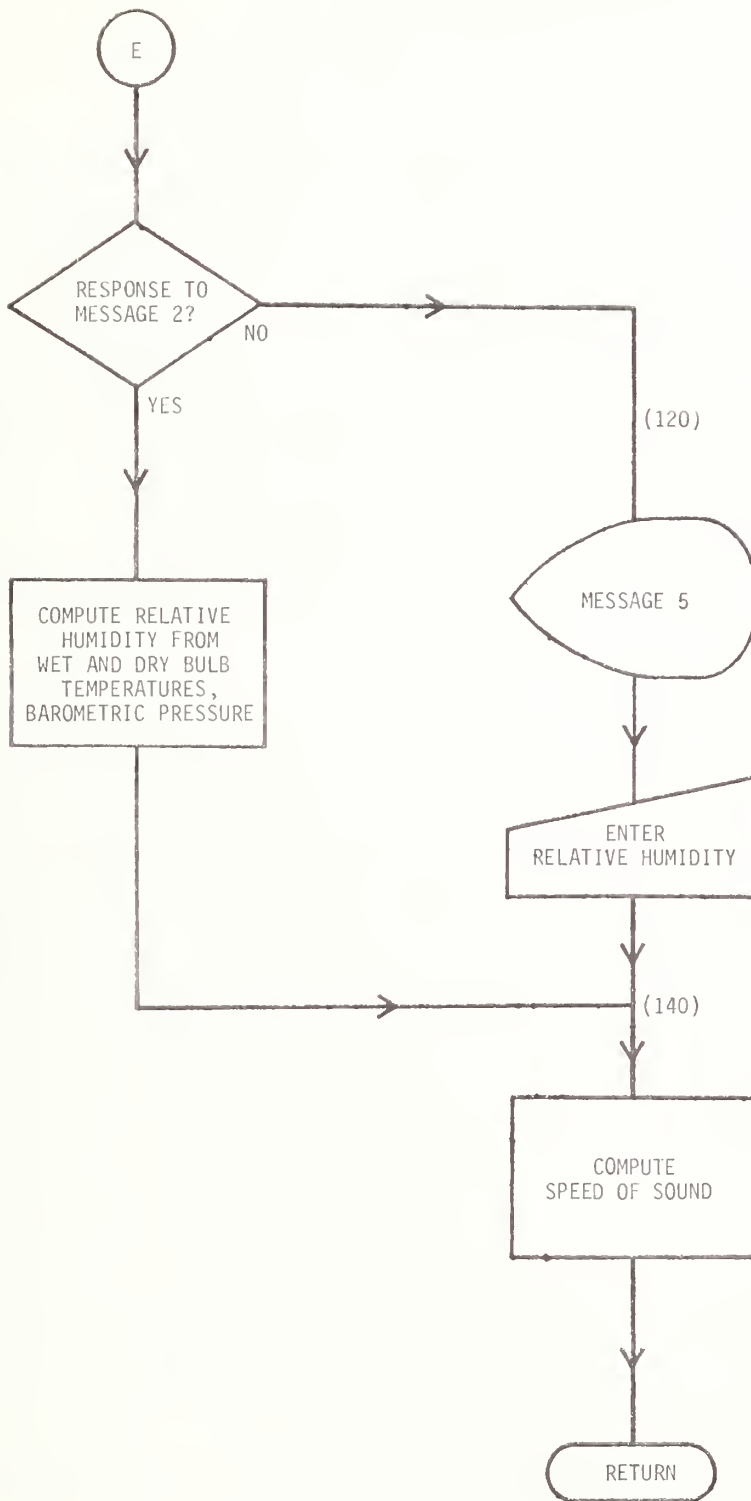
APPENDIX H

Subroutine HUMID Flow Chart, Terminal Messages, and Listings

FLOW CHART: SUBROUTINE HUMID







CRT TERMINAL MESSAGES: SUBROUTINE HUMID

MESSAGE NO.	FORMAT NO.	MESSAGE
1	10	ENTER DRY BULB TEMPERATURE (C OR FXX.X)
2	50	DO YOU WANT TO COMPUTE THE HUMIDITY FROM THE WET BULB TEMPERATURE?
3	70	ENTER WET BULB TEMPERATURE (C OR FXX.X)
4	100	ENTER BAROMETRIC PRESSURE (CHOOSE ONE OF THE FOLLOWING) MXXX.X (IN MILLIMETERS OF HG) BXXXX.X (IN MILLIBARS)
5	130	ENTER THE RELATIVE HUMIDITY IN PERCENT (XX.X)

SUBROUTINE HUMID

```

1 $ASSM
2 HUMID PROG HUMID - SUBROUTINE FOR COMPUTING HUMIDITY FROM WET BULB TEMP
3 $FORT
4 C
5 C -----
6 C THIS SUBROUTINE ALLOWS THE ENTERING OF ATMOSPHERIC FACTORS OF
7 C TEMPERATURE, RELATIVE HUMIDITY AND PRESSURE. HUMIDITY CAN BE
8 C COMPUTED FROM DRY AND WET BULB TEMPERATURE READINGS.
9 C THE SPEED OF SOUND IS ALSO COMPUTED.
10 C
11 C FORTRAN CALL STATEMENT:
12 C -CALL HUMID (HUMIDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,
13 C PRESS1,PRESS2,VSOUND)
14 C
15 C HUMDTY (REAL) = RELATIVE HUMIDITY (PERCENT)
16 C DTEMPF (REAL) = DRY BULB TEMPERATURE (FAHRENHEIT)
17 C DTEMPC (REAL) = DRY BULB TEMPERATURE (CELSIUS)
18 C WTEMPF (REAL) = WET BULB TEMPERATURE (FAHRENHEIT)
19 C WTEMPC (REAL) = WET BULB TEMPERATURE (CELSIUS)
20 C NOTE: WTEMPC IS SET = -1000, IF YOU CHOOSE
21 C NOT TO ENTER THE WET BULB TEMPERATURE.
22 C PRESS1 (REAL) = BAROMETRIC PRESSURE (MM HG)
23 C PRESS2 (REAL) = BAROMETRIC PRESSURE (MILLIBARS)
24 C VSOUND (REAL) = SPEED OF SOUND (METERS/SECOND)
25 C
26 C NO OTHER SUBROUTINES CALLED
27 C -----
28 C SUBROUTINE HUMID(HUMDTY,DTEMPF,DTEMPC,WTEMPF,WTEMPC,
29 C 1PRESS1,PRESS2,VSOUND)
30 10 WRITE (5,10)
31 10 FORMAT ('ENTER DRY BULB TEMPERATURE (C OR F(X).X)')
32 20 READ (5,20)SCALE,TEMP
33 20 FORMAT (A1,F4.1)
34 IF (SCALE.EQ.'C') GO TO 30
35 DTEMPF=TEMP
36 DTEMPC=5*(TEMP-32)/9
37 GO TO 40
38 DTEMPF=32+9*TEMP/5
39 DTEMPC=TEMP
40 50 WRITE (5,50)
41 50 FORMAT ('DO YOU WANT TO COMPUTE THE HUMIDITY FROM THE WET BULB
42 1 TEMPERATURE?')
43 60 READ (5,60)GS
44 60 FORMAT (A4)
45 IF (GS.EQ.'NO') GO TO 90
46 70 WRITE (5,70)
47 70 FORMAT ('ENTER WET BULB TEMPERATURE (C OR F(X).X)')
48 80 READ (5,80)SCALE,TEMP
49 IF (SCALE.EQ.'C') GO TO 80
50 WTEMPF=TEMP
51 WTEMPC=5*(TEMP-32)/9
52 GO TO 90
53 WTEMPF=32+9*TEMP/5
54 WTEMPC=TEMP
55 90 WRITE (5,100)
56 100 FORMAT ('ENTER BAROMETRIC PRESSURE (CHOOSE ONE OF
57 1 THE FOLLOWING)')
58 2' M(X).X (IN MILLIMETERS OF HG)'
59 3' B(X).X (IN MILLIBARS)'
60 110 READ (5,110)SCALE,PRESS
61 110 FORMAT (A1,F6.1)

```

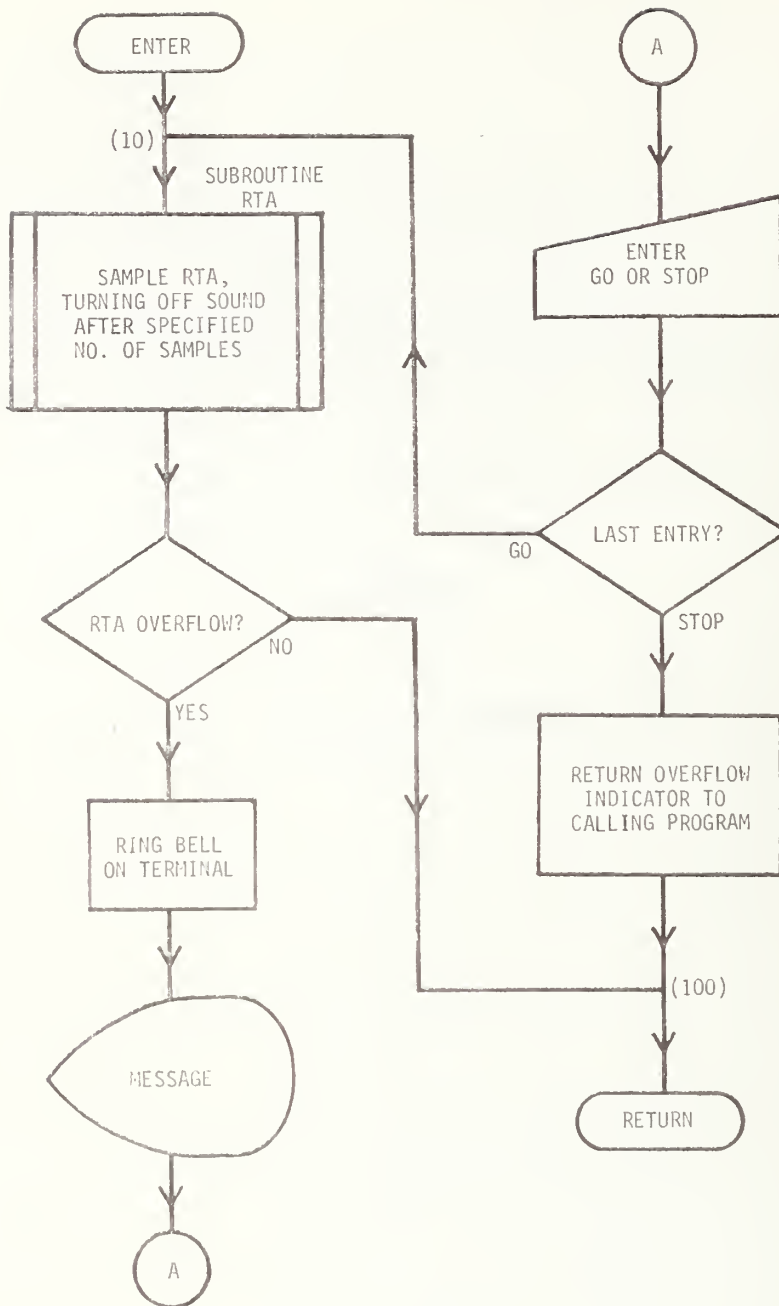
HUMID-2

```
61      IF (SCALE.EQ.'0') GO TO 115
62      PRESS1=PRESS
63      PRESS2=PRESS1*1.33322
64      GO TO 116
65 115   PRESS2=PRESS
66      PRESS1=PRESS2*.750064
67 116   IF (GS.EQ.'NO') GO TO 120
68      TEMP=DTEMPC+273.16
69      ED=23.5315-(2939/TEMP)-4.922*ALOG10(TEMP)
70      ED=(1013.25/1333)*10.**ED
71      TEMP=WTEMPC+273.16
72      EW=23.5315-(2939/TEMP)-4.922*ALOG10(TEMP)
73      EW=(1013.25/1333)*10.**EW
74      E=EW-.00066*PRESS1*(DTEMPC-WTEMPC)*(1+.00115*WTEMPC)
75      HUMDTY=100*E/ED
76      GO TO 140
77 120   WTEMPC=-1000.
78      WRITE (5,130)
79 130   FORMAT ('ENTER THE RELATIVE HUMIDITY IN PERCENT (XX.X)')
80      READ (5,135) HUMDTY
81 135   FORMAT (F5.2)
82 140   VSOUND=331.43+.59*DTEMPC
83      RETURN
84      END
```


APPENDIX I

Subroutine SAMPLE Flow Chart and Listings

FLOW CHART: SUBROUTINE SAMPLE



MESSAGE: YOU HAVE SIGNAL LEVEL OVERFLOW ON BAND XX, POINT XXX, LEVEL = XXX.XX
 ENTER "GO" TO REPEAT MEASUREMENT, OR "STOP" TO INTERRUPT PROGRAM

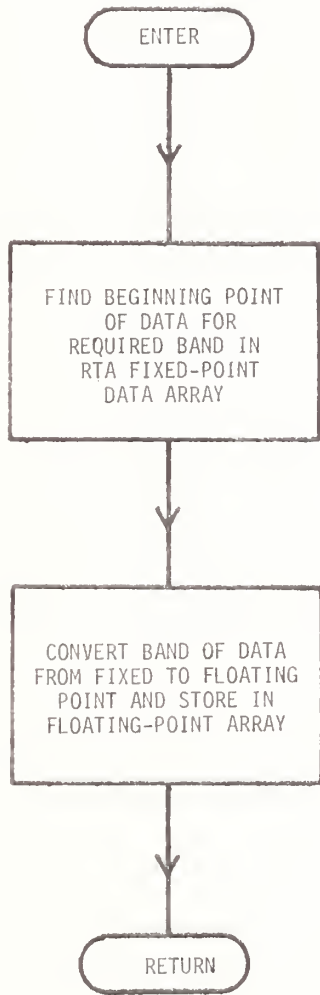
SUBROUTINE SAMPLE

```
1 $ASSM
2 SAMPLE PROG SAMPLE - SUBROUTINE FOR TAKING RTA SAMPLES
3 $FORT
4 C -----
5 C THIS SUBROUTINE TAKES A SPECIFIED NUMBER OF RTA SAMPLES AND
6 C CHECKS FOR OVERFLOW.
7 C FORTRAN CALL STATEMENT:
8 C -CALL SAMPLE (NSOUND,NSAMP, TX100,OVFLOW,X0,NX0)
9 C NSOUND (INT*2) = NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF
10 C NSAMP (INT*2) = TOTAL NO. OF SAMPLES TO BE TAKEN (MAX = 200)
11 C TX100 (INT*2) = INTEGRATION TIME IN SECONDS X 100
12 C OVFLOW (INT*2) = 1 IF AN OVERFLOW CONDITION EXISTS, = 0 OTHERWISE;
13 C (THIS IS AN OPTIONAL PARAMETER FOR THE FORTRAN
14 C CALLING PROGRAM)
15 C X0 (INT*2) = ARRAY CONTAINING THE FIXED-POINT RTA DATA
16 C NX0 (INT) = NO. OF ELEMENTS IN THE ARRAY X0:
17 C SHOULD BE CHOSEN SO THAT NX0=(30*NSAMP)+60
18 C
19 C OTHER SUBROUTINES CALLED: RTA
20 C -----
21 SUBROUTINE SAMPLE (NSOUND,NSAMP, TX100,OVFLOW,X0,NX0)
22 INTEGER*2 NSOUND,NSAMP, TX100,OVFLOW,X0(NX0), BELL, I
23 DATA BELL/'0700'/
24 OVFLOW=0
25 10 CALL RTA(NSOUND,NSAMP, TX100,X0)
26 IF (X0(1).NE.0) GO TO 100
27 I=X0(3)+1
28 X=X0(4)/100.
29 WRITE (5,20) BELL
30 20 FORMAT (A1)
31 WRITE (5,30) X0(2),I,X
32 30 FORMAT ('YOU HAVE SIGNAL LEVEL OVERFLOW ON BAND ',I2,
33 1', POINT ',I3,'. LEVEL = ',F8,2)
34 2'ENTER "GO" TO REPEAT MEASUREMENT, OR "STOP" TO INTERRUPT
35 3 PROGRAM')
36 READ (5,40) GS
37 40 FORMAT (A4)
38 IF (GS.NE.'STOP') GO TO 10
39 OVFLOW=1
40 100 RETURN
41 END
```

APPENDIX J

Subroutine BAND Flow Chart and Listings

FLOW CHART: SUBROUTINE BAND



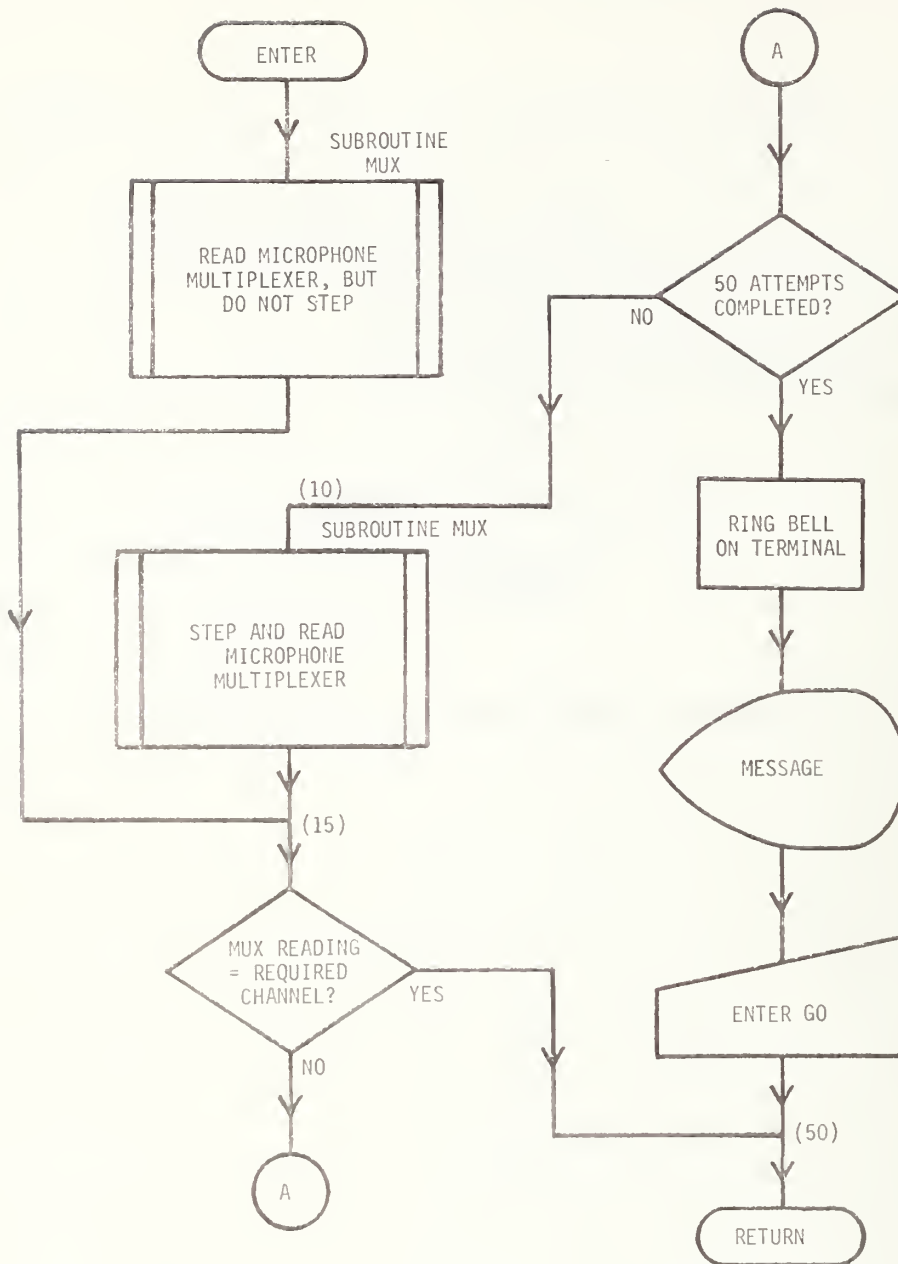
SUBROUTINE BAND

```
1 $ASSM
2 BAND PROG BAND = SUBROUTINE FOR CONVERTING ONE BAND OF RTA SAMPLES
3 $FORT
4 C -----
5 C THIS SUBROUTINE CONVERTS ONE BAND OF RTA SAMPLES TO FLOATING POINT
6 C FORTRAN CALL STATEMENT:
7 C -CALL BAND (BANDNO,NSAMP,X, X0,NX0)
8 C BANDNO (INT*2) = BAND NUMBER (BAND 14 = 25 HZ)
9 C NSAMP (INT) = NO. OF SAMPLES TAKEN BY RTA (MAX = 250)
10 C X (REAL) = ARRAY TO CONTAIN THE FLOATING POINT RTA DATA
11 C FOR THE SPECIFIED BAND
12 C X0 (INT*2) = ARRAY CONTAINING THE FIXED-POINT RTA DATA
13 C NX0 (INT) = NO. OF ELEMENTS IN THE ARRAY X0:
14 C SHOULD BE CHOSEN SO THAT NX0=(30*NSAMP)+60
15 C
16 C NO OTHER SUBROUTINES CALLED
17 C -----
18 SUBROUTINE BAND(BANDNO,NSAMP,X,X0,NX0)
19 DIMENSION X(NSAMP)
20 INTEGER*2 BANDNO,NX0(NX0),I,INDEX,ISTPT
21 ISTPT=(BANDNO-14)*(NSAMP+2)+2
22 DO 10 I=1,NSAMP
23 INDEX=ISTPT+I
24 X(I)=X0(INDEX)*100.
25 10 CONTINUE
26 RETURN
27 END
```

APPENDIX K

Subroutine SUBMUX Flow Chart and Listings

FLOW CHART: SUBROUTINE SUBMUX



MESSAGE: THE PROPER MICROPHONE CANNOT BE SELECTED AUTOMATICALLY
 MANUALLY STEP MULTIPLEXER TO MICROPHONE XX
 ENTER "GO" TO CONTINUE

SUBROUTINE SUBMUX

```
1 $ASSM
2 SUBMUX PRDG SUBMUX - SUBROUTINE TO SET MULTIPLEXER TO A GIVEN MICROPHONE
3 $FORT
4 C -----
5 C THIS SUBROUTINE STEPS THE MICROPHONE MULTIPLEXER TO A GIVEN MICROPHONE
6 C CHANNEL.
7 C
8 C FORTRAN CALL STATEMENT:
9 C -CALL SUBMUX(J)
10 C J (INTEGER) = THE MICROPHONE CHANNEL TO BE SELECTED
11 C
12 C OTHER SUBPROGRAMS CALLED: MUX
13 C -----
14 C SUBROUTINE SUBMUX(J)
15 C INTEGER*2 BELL
16 C DATA BELL/'X'0700'/
17 C I=0
18 C CALL RDMUX(MXRDG)
19 C GO TO 15
20 10 CALL MUX(MXRDG)
21 15 IF (MXRDG.EQ.J) GO TO 50
22 C I=I+1
23 C IF (I.LT.50) GO TO 10
24 C WRITE (5,20) BELL
25 20 FORMAT (A1)
26 C WRITE (5,30) J
27 30 FORMAT (X/'THE PROPER MICROPHONE CANNOT BE SELECTED
28 1 'AUTOMATICALLY'/'MANUALLY STEP MULTIPLEXER TO MICROPHONE ',I2/
29 2 'ENTER "GO" TO CONTINUE')
30 C READ (5,40) QS
31 40 FORMAT (A4)
32 50 RETURN
33 C END
```

APPENDIX L

Listings for Function Subprograms TINORM and STUDIM

FUNCTION SUBPROGRAM TINORM

```
1 $ASSM
2 TINORM PROG TINORM - SUBPROG. GIVING INVERSE OF NORMAL DISTRIBUTION
3 $FORT
4 C -----
5 C THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF THE STANDARD
6 C NORMAL DISTRIBUTION FUNCTION. IT IS A VERSION OF THE UNIVAC 1108
7 C STAT-PACK ROUTINE TINORM.
8 C
9 C FORTRAN CALLING SEQUENCE:
10 C > VARIABLE = TINORM(ALPHA)
11 C
12 C TINORM (REAL) = THE CALCULATED VALUE OF THE INVERSE OF THE NORMAL
13 C DISTRIBUTION: THIS IS THE ORDINATE, OR NUMBER OF
14 C STANDARD DEVIATIONS FROM THE MEAN CORRESPONDING TO
15 C THE GIVEN VALUE OF ALPHA.
16 C ALPHA (REAL) = THE VALUE OF THE PROBABILITY FOR WHICH THE ORDINATE
17 C IS TO BE CALCULATED; THIS IS EQUAL TO 1.0 MINUS
18 C THE AREA UNDER THE CURVE FROM TINORM TO INFINITY.
19 C (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
20 C
21 C NO OTHER SUBPROGRAMS CALLED
22 C -----
23 FUNCTION TINORM(ALPHA)
24 DIMENSION A(3),B(3)
25 DATA (A(I),I=1,3)/.010328,.802853,2.515517/,
26 1(B(I),I=1,3)/.001308,.189269,1.432788/
27 X=ALPHA
28 IF (X.GT..5) X=1.-X
29 X=SQRT(-2.*ALOG(X))
30 TINORM=X-(A(3)+X*(A(2)+X*A(1)))/(1.+X*(B(3)+X*(B(2)+X*B(1))))
31 IF (ALPHA.LT..5) TINORM=-TINORM
32 RETURN
33 END
```

FUNCTION SUBPROGRAM STUDIM

```
1 $ASSM
2 STUDIM PROG STUDIM - SUBPROG. SUBSTITUTING FOR STUDIN
3 $FORT
4 C -----
5 C THIS SUBPROGRAM IS A SUBSTITUTE FOR STUDIN, GIVING THE VALUE OF
6 C THE INVERSE OF STUDENT'S T-DISTRIBUTION BY READING FROM A TABLE
7 C GENERATED ON A STORAGE FILE BY THE PROGRAM GEN1.
8 C
9 C IT IS INTENDED FOR USE IN THOSE CASES WHEN THE MEMORY SPACE
10 C REQUIRED BY STUDIN IS NOT AVAILABLE.
11 C
12 C THE FILE MUST BE GENERATED FOR THE INTENDED VALUE OF ALPHA;
13 C ALPHA HERE HAS THE SAME MEANING AS IN THE PROGRAM STUDIN. AND
14 C THE VALUE STUDIM CORRESPONDS TO THE VALUE STUDIN.
15 C
16 C THE FILE MUST BE A RANDOM RECORD ACCESS FILE.
17 C
18 C FORTRAN CALLING SEQUENCE:
19 C > VARIABLE = STUDIM(ALPHA,N,LUFILE,LUERR)
20 C
21 C STUDIM (REAL) = THE CALCULATED VALUE OF T
22 C ALPHA (REAL) = THE CONFIDENCE COEFFICIENT
23 C LUFILE (INT) = LOGICAL UNIT OF THE STORAGE FILE
24 C LUERR (INT) = LOGICAL UNIT FOR ERROR MESSAGE PRINTOUT
25 C (USUALLY LUERR = 5)
26 C -----
27 C FUNCTION STUDIM(ALPHA,N,LUFILE,LUERR)
28 C DIMENSION T(50)
29 C INTEGER*2 BELL
30 C DATA BELL/X'0700'
31 10 READ (LUFILE,REC=0) ALPHA1
32 C IF (ABS(ALPHA-ALPHA1).LE..0001) GO TO 30
33 C WRITE (LUERR,20) BELL,LUFILE,ALPHA1,ALPHA,LUFILE
34 20 FORMAT (A1//X' THE STORAGE FILE ON LU',I2,' CONTAINS THE TABLE
35 1 FOR ALPHA = ',F6.4
36 2' THE REQUIRED T-DISTRIBUTION TABLE IS FOR ALPHA = ',F6.4//X'
37 3' REASSIGN LU',I2,' TO THE PROPER FILE'//
38 4' ENTER "CONTINUE" WHEN READY')
39 C PAUSE 1
40 C GO TO 10
41 30 N1=N
42 C IF (N1.GT.1000) N1=1000
43 C NSECTR=1+(N1-1)/50
44 C READ (LUFILE,REC=NSECTR) T
45 C J=N1-50*(NSECTR-1)
46 C STUDIM=T(J)
47 C RETURN
48 C END
```

APPENDIX M

Listings for Interdata Assembly Language Subroutines

MUX, RTA, SOUND, and DELAY

SUBROUTINE MUX

```

1 MUX      PROG  MUX      - PROGRAM FOR STEPPING AND READING MUX
2          WIDTH 132
3 * -----
4 * FORTRAN CALL STATEMENTS:
5 * -CALL MUX(X) --- STEP AND READ
6 * -CALL RDMUX(X) --- READ ONLY
7 * PARAMETER = MULTIPLEXER READING
8 * -----
9          ENTRY  MUX
10         ENTRY  RDMUX
11 UIM      EQU   12
12 WR       EQU   13
13 INDEX    EQU   14
14 R15      EQU   15
15 MUX      STM   12,RETURN      START HERE TO STEP MULTIPLEXER
16         LHI   UIM,X*8B
17         LIS   WR,10
18         OCR   UIM,WR
19         LIS   WR,1
20         WHR   UIM,WR
21         XHR   WR,WR
22         WHR   UIM,WR
23         B     CONT
24 RDMUX    STM   12,RETURN      START HERE TO READ MULTIPLEXER
25         LHI   UIM,X*8B
26 CONT     LHI   INDEX,1000     THESE NEXT 4 LINES PROVIDE A DELAY
27 DELAY   SLLS  WP,15
28         SIS   INDEX,1
29         BP   DELAY
30         LIS   WP,11
31         OCR   UIM,WR
32         WHR   UIM,WR
33         PHR   UIM,WP
34         SLLS  WP,12
35         SPLS  WP,12
36         LH   INDEX,0(R15)     FIND FORTRAN ADDRESS OF PARAMETER
37         STH  WP,0(INDEX)      STORE PARAMETER AT ITS FORTRAN ADDRESS
38         LM   12,RETURN
39         AIS  R15,4
40         BP   R15
41 RETURN  DS    8
42 END

```

SUBROUTINE RTA

```

1 RTA      PROG RTA      - PROGRAM FOR DRIVING RTA
2          WIDTH 132
3 *
4 * -----
5 * FORTRAN CALL STATEMENT:
6 * -CALL RTA(NSOUND,NSAMP,TX100,X0)
7 * NSOUND = NO. OF SAMPLES TO BE TAKEN BEFORE SOUND CUT-OFF
8 * NSAMP  = TOTAL NO. OF SAMPLES TO BE TAKEN
9 * TX100  = INTEGRATION TIME X 100
10 * X0    = ARRAY OF 2-BYTE INTEGERS TO CONTAIN THE
11 *        FIXED-POINT RTA DATA
12 * -----
13 UIM      EQU 0      ENTRY RTA
14 N1000    EQU 1      CONTAINS THE INTEGER 1000
15 N100     EQU 2      CONTAINS THE INTEGER 100
16 N25      EQU 3      CONTAINS THE INTEGER 25
17 WR       EQU 4
18 WR1      EQU 5
19 WR2      EQU 6
20 ACC      EQU 7
21 MR1      EQU 8      MULTIPLICATION REGISTERS
22 MR2      EQU 9
23 LR1      EQU 10     LOOP REGISTERS FOR BXLE LOOP
24 LR2      EQU 11
25 LP3      EQU 12
26 INDEX    EQU 13     NUMBER OF SAMPLES ALREADY TAKEN
27 R14      EQU 14
28 R15      EQU 15
29 RTA      STM 0,RETURN ENTRY POINT
30          LHI UIM,X'8B'
31          LH  WR1,6(R15) FIND FORTRAN ADDRESS OF TX100
32          LH  WR,0(WR1)  LOAD TX100 INTO WR
33          CHI WR,10     DETERMINE INTEGRATION TIME
34          BE  T10
35          CHI WR,25
36          BE  T25
37          CHI WR,50
38          BE  T50
39          CHI WR,100
40          BE  T100
41          CHI WR,200
42          BE  T200
43          CHI WR,400
44          BE  T400
45          CHI WR,800
46          BE  T800
47          CHI WR,1600
48          BE  T1600
49          CHI WR,3200
50          BE  T3200
51 T10      LIS WR,8     DETERMINE TIME SECTION OF
52          B   TIME     RTA CONTROL WORD
53 T25      LIS WR,2
54          B   TIME
55 T50      LIS WR,4
56          B   TIME
57 T100     LIS WR,6
58          B   TIME
59 T200     LIS WR,1
60          B   TIME

```

RTA-2

61	T400	LIS	WR.3	
62		B	TIME	
63	T800	LIS	WR.5	
64		B	TIME	
65	T1600	LIS	WR.7	
66		B	TIME	
67	T3200	LIS	WR.9	
68	TIME	SLLS	WR.12	
69		AHI	WR.X'08FF'	
70		STH	WR.PTACWD	SET RTA CONTROL WORD
71		LH	ACC.8(R15)	LOAD FORTRAN ADDRESS OF X0 INTO ACC
72		STH	ACC.STORE	STORE ADDRESS OF X0 IN STORE
73		LH	WR.2(R15)	FIND FORTRAN ADDRESS OF NSOUND
74		LH	WR1.0(WR)	LOAD NSOUND INTO WR1
75		STH	WR1.NSOUND	STORE NSOUND IN NSOUND
76		LH	WR.4(R15)	FIND FORTRAN ADDRESS ON NSAMP
77		LH	WR1.0(WR)	LOAD NSAMP INTO WR1
78		STH	WR1.NSAMP	STORE NSAMP IN NSAMP
79		LHR	WR.WR1	
80		SLLS	WR1.8	
81		AIS	WR.2	
82		LHR	WR2.WR	
83		AHR	WR.WP2	WR NOW CONTAINS 2(NSAMP+2), WHICH IS
84		XHR	WR2.WP2	THE TOTAL AMOUNT OF STORAGE OCCUPIED
85		LIS	LR1.1	BY EACH LIST: USED TO UPDATE ACC
86		LIS	LR2.1	
87		LHI	LP3.30	
88	LOOP	STH	WR1.0(ACC)	STORE LIST PARAMETER BLOCK AT
89		STH	WR2.2(ACC)	BEGINNING OF 30 CONSECUTIVE LISTS
90		AHP	ACC.WP	(THESE ARE CIRCULAR LISTS)
91		BXLE	LP1.LOOP	
92		LHI	N1000.1000	
93		LHI	N100.100	
94		LHI	N25.25	
95		XHR	INDEX:INDEX	SET INDEX=0 (NO. SAMPLES ALREADY TAKEN)
96	LOOP3	CH	INDEX.NSOUND	IF INDEX = NSOUND, THEN CONTINUE;
97		BTC	X'3'.BRNCH1	OTHERWISE, GO TO BRNCH1
98		LIS	WR.1	
99		OCR	UIM.WP	SELECT RTA CONTROL LINE
100		LHI	WR.X'10FF'	
101		WHR	UIM.WR	STOP RTA
102		XHR	WP.WP	
103		OCR	UIM.WP	SELECT SOUND CONTROL LINE
104		XHR	WP.WP	
105		WHR	UIM.WP	STOP SOUND
106		BFFS	0.5	BRANCH FORWARD 5 LOCATIONS
107	BRNCH1	CHI	INDEX.0	IF INDEX = 0, THEN CONTINUE;
108		BTC	X'3'.BRNCH2	OTHERWISE, GO TO BRNCH2
109		LIS	WR.1	
110		OCR	UIM.WP	SELECT RTA CONTROL LINE
111		WH	UIM.PTACWD	SEND RTA CONTROL WORD (SET UP RTA)
112		LIS	WR.4	
113		OCR	UIM.WP	SELECT START MEASUREMENT LINE
114		WHR	UIM.WP	START MEASUREMENT
115	BRNCH2	LIS	LR1.1	
116		LIS	LR3.6	
117	LOOP1	XHR	WR.WP	THROW OUT 1ST 6 READINGS
118		SSR	UIM.WP	(BANDS 12 AND 13)
119		BTBS	X'1'.1	
120		RHR	UIM.WP	

RTA-3

121		BXLE	LR1.LOOP1	
122		LH	ACC.STORE	
123		LH	WR2.NSAMP	
124		AI5	WR2.2	
125		LHR	WR.WR2	
126		AHR	WR2.WR	WR2 NOW CONTAINS 2(NSAMP+2),
127		LIS	LR1.1	WHICH IS USED TO UPDATE ACC
128		LHI	LR3.30	
129	LOOP2	SSR	UI1.WR	TEST FOR RTA BUSY
130		BTBS	X*4*.1	WAIT IF BUSY
131		RHR	UI1.WR	1ST READING (BAND NUMBER) THROWN OUT
132		SSR	UI1.WR	
133		BTBS	X*4*.1	
134		RHR	UI1.WR	2ND READING
135		LHR	WR2.WR	CONVERT 10'S DIGIT
136		SRLS	WR2.12	
137		MHR	MR1.N1000	WR1 USED TO ACCUMULATE RESULT OF
138		LHR	WR1.MR2	BCD TO BINARY CONVERSION
139		LHR	WR2.WR	CONVERT 1'S DIGIT
140		SLLS	WR2.4	
141		SRLS	WR2.12	
142		MHR	MR1.N100	
143		AHR	WR1.MR2	
144		SSR	UI1.WR	
145		BTBS	X*4*.1	
146		RHR	UI1.WR	3RD READING
147		LHR	WR2.WR	CONVERT 1/4 DB DIGIT
148		SLLS	WR2.4	
149		SRLS	WR2.12	
150		MHR	MR1.N25	
151		AHR	WR1.MR2	
152		SRLS	WR.12	CHECK 100'S DIGIT
153		LIS	MR1.1	(BIT 1 = VALUE OF 100'S DIGIT)
154		CHR	WR.MR1	(BIT 2 = RTA OVERFLOW BIT)
155		BTC	X*1*.ABL	IF 100'S DIGIT < 1. BRANCH TO ABL
156		BTC	X*2*.OVFLOW	IF 100'S DIGIT > 1. BRANCH TO OVFLOW
157		AHI	WR1.10000	IF 100'S DIGIT = 1. ADD 10000 TO WR1
158		B	ABL	
159	OVFLOW	SIS	WR.2	IN CASE OF OVERFLOW, REMOVE BIT 2
160		LHI	MR2.10000	FROM VALUE OF 100'S DIGIT
161		MHR	MR1.WR	
162		AHR	WR1.MR2	WR1 NOW CONTAINS VALUE OF OVERFLOW
163		LH	ACC.STORE	DATA POINT
164		ZHR	WR.WR	
165		STH	WR.0(ACC)	STORE "0" AT ADDRESS OF X0(1)
166		AI5	LR1.13	
167		STH	LR1.2(ACC)	STORE BAND NUMBER AT X0(2)
168		STH	INDEX.4(ACC)	STORE SAMPLE NUMBER AT X0(3)
169		STH	WR1.6(ACC)	STORE VALUE OF OVERFLOW POINT AT X0(4)
170		B	RTRN	STOP SAMPLING AND RETURN TO FORTRAN
171	ABL	ABL	WR1.0(ACC)	ADD FINAL NUMBER TO PROPER LIST
172		AHR	ACC.WR2	
173		BXLE	LR1.LOOP2	
174		AI5	INDEX.1	
175		CH	INDEX.NSAMP	
176		BTC	X*8*.LOOP3	IF INDEX < NSAMP. RETURN TO LOOP3;
177	RTRN	LM	0.RETURN	OTHERWISE. RETURN TO FORTRAN
178		AI5	R15.10	
179		BR	R15	
180	RETURN	DS	32	

RTA-4

```
181 STORE DS 2 ADDRESS OF X0(1) (START OF THE X0'S)
182 NSOUND DS 2
183 NSAMP DS 2
184 RTACWD DS 2 RTA CONTROL WORD
185 * -----
186 * LOOP: BXLE LOOP: STORES LIST PARAMETER BLOCK
187 * AT BEGINNING OF 30 CONSECUTIVE CIRCULAR LISTS
188 * LOOP1: THROWS OUT 1ST 6 READINGS (BANDS 12 & 13)
189 * LOOP2: BXLE LOOP: READS BANDS 14 TO 43. CONVERTS
190 * DATA FROM BCD TO BINARY. AND STORES THE
191 * FIXED-POINT DATA IN 30 CONSECUTIVE LISTS
192 * LOOP3: ENCLOSSES BOTH LOOP1 AND LOOP2:
193 * REPEATS MEASUREMENT UNTIL NSAMP SAMPLES
194 * HAVE BEEN TAKEN
195 * -----
196 END
```

SUBROUTINE SOUND

```
1 SOUND   PRG  SOUND
2         WIDTH 132
3 *
4 *   FORTRAN CALL STATEMENT:
5 *   -CALL SOUND(X)
6 *   X = SOUND CONTROL WORD
7 *   SOUND ON:  X = 1
8 *   SOUND OFF: X = 0
9 *
10        ENTRY  SOUND
11 UIM    EQU    13
12 WR     EQU    14
13 R15    EQU    15
14 SOUND  STM    13.RETURN
15        LHI    UIM,X*8B'
16        XHR    WR,WR
17        OCR    UIM,WR
18        LH     WR,2(R15)
19        LH     WR,0(WR)
20        SLLS   WR,8
21        WHR    UIM,WR
22        LM     13.RETURN
23        AIS    R15,4
24        BR     R15
25 RETURN  DS     6
26        END
```

- PROGRAM FOR TURNING ON SOUND

ENTRY POINT

SELECT SOUND CONTROL LINE
FIND FORTRAN ADDRESS OF X
LOAD X INTO WR

SEND SOUND CONTROL WORD

RETURN TO FORTRAN

SUBROUTINE DELAY

```
1 DELAY   PROG DELAY           - APPROXIMATELY 1 MILLISECOND INCREMENT
2         WIDTH 132
3 * -----
4 * FORTRAN CALL STATEMENT:
5 * -CALL DELAY(X)
6 * X = NUMBER OF 1 MILLISECOND INCREMENTS
7 * -----
8         ENTRY DELAY
9 WR      EQU 12
10 INDEX1 EQU 13
11 INDEX2 EQU 14
12 R15    EQU 15
13 DELAY  STM 12,RETURN          ENTRY POINT
14        LH  WR,2(R15)         FIND FORTRAN ADDRESS OF X
15        LH  INDEX2,0(WR)      LOAD X INTO INDEX2
16        BNP STOP
17 LOOP2  LHI INDEX1,100
18 LOOP1  SLLS WR,15
19        SIS INDEX1,1
20        BP  LOOP1
21        SIS INDEX2,1
22        BP  LOOP2
23 STOP   LM  12,RETURN
24        AIS R15,4
25        BR  R15
26 RETURN DS  8
27        END
```

APPENDIX N

Listings for Program GEN1 and its Function Subprograms

STUDIN, FISHIN, and FISH

This program is not called by the program REVERB. However, it is required to generate the t-distribution table read from logical unit 4 by the function subprogram STUDIM.

PROGRAM GEN1

```

1 $ASSM
2 GEN1 PROG GEN1 - PROGRAM TO GENERATE STORAGE FILE FROM STUDIN
3 $FORT
4 C
5 C -----
6 C THIS PROGRAM GENERATES A STORAGE FILE FROM THE SUBPROGRAM STUDIN,
7 C WHICH GIVES THE INVERSE OF STUDENT'S T-DISTRIBUTION. THIS FILE
8 C IS GENERATED FOR ONE SPECIFIED VALUE OF ALPHA, AND CONSISTS OF THE
9 C T-VALUES CORRESPONDING TO THE DEGREES OF FREEDOM FROM 1 TO 1000.
10 C THE FILE IS INTENDED FOR USE BY THE SUBPROGRAM STUDIN, WHICH IS A
11 C SUBSTITUTE FOR STUDIN FOR THOSE CASES WHEN THE MEMORY SPACE REQUIRED
12 C BY STUDIN IS NOT AVAILABLE.
13 C
14 C SUBPROGRAMS CALLED: STUDIN.
15 C [INDIRECTLY]: FISHIN,FISH,TINORM
16 C -----
17 DIMENSION T(50)
18 INTEGER*2 IT(50),FF
19 DATA FF/'X'0C00'/
20 WRITE (5,10)
21 10 FORMAT ('X/X'/'THIS PROGRAM GENERATES A STORAGE FILE OF THE T-VALUES
22 1 OF STUDENT'S T-DISTRIBUTION FOR A SPECIFIED VALUE OF ALPHA.'/
23 2'X/X'/'THE FOLLOWING LOGICAL UNIT ASSIGNMENTS ARE REQUIRED:'/'X/
24 35X'/'LU1 -- STORAGE FILE (22 SECTORS REQUIRED)'/
25 45X'/'LU2 -- SCRATCH FILE'/
26 55X'/'LU3 -- HIGH SPEED PRINTER'/
27 65X'/'LU5 -- CRT TERMINAL'/'X/X'/
28 7'DO YOU WANT TO CHANGE THE LOGICAL UNIT ASSIGNMENTS?')
29 READ (5,20) QS
30 20 FORMAT (A4)
31 IF (QS.EQ.'NO') GO TO 30
32 PAUSE 1
33 30 WRITE (5,35)
34 35 FORMAT ('DO YOU WANT A PRINTOUT OF THE FILE ONLY?')
35 READ (5,20) QS
36 IF (QS.NE.'NO') GO TO 140
37 WRITE (5,40)
38 40 FORMAT ('ENTER ALPHA (X.XX)')
39 IT(1)NOTE: THIS VALUE IS 2 TIMES THE VALUE BEING USED IN TABLE
40 2 3 T14. OF BOWLER & LIEBERMAN'S "ENGINEERING STATISTICS")
41 READ (5,50) ALPHA
42 50 FORMAT (F10,5)
43 WRITE (5,60)
44 60 FORMAT ('ENTER MAX. NO. OF ITERATIONS (3 DIGITS)')
45 READ (5,70) ICMAX
46 70 FORMAT (I3)
47 REMIND 1
48 REWIND 2
49 WRITE (1) ALPHA
50 N=0
51 DO 110 I=1,20
52 DO 100 I=1,50
53 N=N+1
54 T(I)=STUDIN(ALPHA,N,ICMAX,IC)
55 IT(I)=IC
56 WRITE (5,80) N,T(I),IC
57 80 FORMAT ('I = ',I4,7X,'T = ',F8,3,6X,'I3, ITERATIONS')
58 100 CONTINUE
59 WRITE (1) T
60 WRITE (2) IT

```

GEN1-2

```
61 110 CONTINUE
62 ENDFILE 1
63 WRITE (5,130)
64 130 FORMAT ('ENTER "CONTINUE" TO PRINT OUT TABLE')
65 PAUSE 1
66 140 REWIND 1
67 READ (1) ALPHA
68 WRITE (3,150) ALPHA,(1,1=1,10)
69 150 FORMAT (X//X//T38,'T-DISTRIBUTION TABLE',10X,'ALPHA = ',F6.4)
70 1X//X//5X,10(12.9X)//X//X)
71 N=0
72 DO 170 J=1,20
73 READ (1) T
74 DO 170 K=1,5
75 N=N+10
76 K2=10*K
77 K1=K2-9
78 WRITE (3,160) (T(I),I=K1,K2),N
79 160 FORMAT (10(F10.5,X),4X,14)
80 170 CONTINUE
81 WRITE (3,180) FF
82 180 FORMAT (A1)
83 REWIND 2
84 WRITE (3,200) (1,1=1,10)
85 200 FORMAT (X//X//T50,'NO. OF ITERATIONS'//X//X//
86 15X,10(12.9X)//X//X)
87 N=0
88 DO 220 J=1,20
89 READ (2) IT
90 DO 220 K=1,5
91 N=N+10
92 K2=10*K
93 K1=K2-9
94 WRITE (3,210) (IT(I),I=K1,K2),N
95 210 FORMAT (4X,10(13.8X),14)
96 220 CONTINUE
97 WRITE (3,180) FF
98 END
```

FUNCTION SUBPROGRAM STUDIN

```
1 #ASSM
2 STUDIN PROG STUDIN - SUBPROG. GIVING INVERSE OF STUDENT'S T-DISTRIBUTION
3 #FORT
4 C -----
5 C THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF STUDENT'S
6 C T-DISTRIBUTION. IT IS A VERSION OF THE UNIVAC 1108 STAT-PACK
7 C ROUTINE STUDIN.
8 C
9 C NOTE--THE VALUE (T) COMPUTED BY THIS PROGRAM RELATES TO THE VALUES
10 C GIVEN IN TABLE 3 OF THE APPENDIX OF BOWKER & LIEBERMAN'S
11 C "ENGINEERING STATISTICS" AS FOLLOWS:
12 C THE TABLE GIVES THE VALUES T(ALPHA;N)
13 C THIS PROGRAM COMPUTES STUDIN(ALPHA1;N), WHERE STUDIN = T, WHICH
14 C IS THE SAME VALUE GIVEN IN THE TABLE IF ALPHA1 = 2*ALPHA.
15 C THUS, TO REPRODUCE THE VALUES OF TABLE 3, USE THE RELATION
16 C T = STUDIN(2.*ALPHA,N,ICMAX,IC)
17 C (THERE SEEMS TO BE SOME VARIATION IN THE DEFINITION OF ALPHA
18 C AMONG DIFFERENT TEXTS. THE STAT-PACK ROUTINE STUDIN CALLS
19 C IT THE CONFIDENCE COEFFICIENT.)
20 C
21 C FORTRAN CALLING SEQUENCE:
22 C > VARIABLE = STUDIN(ALPHA,N,ICMAX,IC)
23 C
24 C STUDIN (REAL) = THE CALCULATED VALUE OF T
25 C ALPHA (REAL) = THE CONFIDENCE COEFFICIENT
26 C (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
27 C N (INT) = THE NO. OF DEGREES OF FREEDOM
28 C ICMAX (INT) = MAXIMUM LIMIT FOR ITERATION COUNTER
29 C (10 SEEMS TO BE SUFFICIENT; THE ITERATION APPLIES
30 C TO FISHIN, WHICH IS CALLED BY STUDIN.)
31 C IC (INT) = ITERATION COUNTER
32 C
33 C OTHER SUBPROGRAMS CALLED: FISHIN
34 C (INDIRECTLY): TINORM, FISH
35 C -----
36 C FUNCTION STUDIN(ALPHA,N,ICMAX,IC)
37 C BETA=1.-ALPHA
38 C STUDIN=1. /SORT(FISHIN(BETA,N,1,ICMAX,IC))
39 C RETURN
40 C END
```


FUNCTION SUBPROGRAM FISHIN

```

1  $ASSM
2  FISHIN PROG FISHIN - SUBPROG. GIVING INVERSE OF FISHER'S F-DISTRIBUTION
3  $FORT
4  C
5  C -----
6  C THIS SUBPROGRAM COMPUTES THE VALUE OF THE INVERSE OF FISHER'S
7  C F-DISTRIBUTION. IT IS A VERSION OF THE UNIVAC 1108 STAT-PACK
8  C ROUTINE FISHIN.
9  C
10 C NOTE--THE VALUE (F) COMPUTED BY THIS PROGRAM RELATES TO THE VALUES
11 C GIVEN IN TABLE 4 OF THE APPENDIX OF BOWKER & LIEBERMAN'S
12 C "ENGINEERING STATISTICS" AS FOLLOWS:
13 C THE TABLE GIVES THE VALUES F(ALPHA;N1,N2)
14 C THIS PROGRAM COMPUTES FISHIN(ALPHA,N1,N2), WHERE FISHIN = F, THE
15 C SAME VALUE GIVEN IN THE TABLE. (ALPHA IS CALLED THE CONFIDENCE
16 C COEFFICIENT.)
17 C
18 C FORTRAN CALLING SEQUENCE:
19 C > VARIABLE = FISHIN(ALPHA,N1,N2,ICMAX,IC)
20 C
21 C FISHIN (REAL) = THE CALCULATED VALUE OF F
22 C ALPHA (REAL) = THE CONFIDENCE COEFFICIENT,
23 C (ALPHA MUST LIE BETWEEN 0.0 AND 1.0)
24 C N1 (INT) = THE DEGREES OF FREEDOM OF THE FIRST SAMPLE
25 C N2 (INT) = THE DEGREES OF FREEDOM OF THE SECOND SAMPLE
26 C ICMAX (INT) = MAXIMUM LIMIT FOR ITERATION COUNTER
27 C (10 SEEMS TO BE SUFFICIENT; MOST OF THE VALUES IN
28 C THE COLUMN FOR N1=1 DO NOT CONVERGE TO THE CHOSEN
29 C LIMITS EVEN AFTER 100 ITERATIONS. HOWEVER, NO
30 C GAIN IN THE ACCURACY IS OBSERVED AFTER THE FIRST
31 C FEW ITERATIONS. 100 ITERATIONS REQUIRE AT LEAST
32 C 20 SECONDS. FOR ALL VALUES FOR N1 AND N2 GREATER
33 C THAN 1, CONVERGENCE SEEMS TO OCCUR IN LESS THAN
34 C 10 ITERATIONS, USUALLY 3.)
35 C IC (INT) = ITERATION COUNTER
36 C
37 C OTHER SUBPROGRAMS CALLED: TINORN, FISH
38 C -----
39 C FUNCTION FISHIN(ALPHA,N1,N2,ICMAX,IC)
40 C DOUBLE PRECISION FSHNDP,Y1,Y2,X,Y,H,G,GLOG,C
41 C Y1=N1
42 C Y2=N2
43 C -----
44 C ADJUST FOR DEGREES OF FREEDOM EQUAL TO 1
45 C -----
46 C IF (N1.EQ.1) Y1=2.
47 C IF (N2.EQ.1) Y2=2.
48 C -----
49 C CALL TINORN TO GET INVERSE NORMAL VALUE OF 1.-ALPHA
50 C -----
51 C X=TINORN(1.-ALPHA)
52 C -----
53 C COMPUTE LAMBDA VALUE
54 C -----
55 C Y=(X*X-3.)*6.
56 C IC=1
57 C -----
58 C COMPUTE INITIAL APPROXIMATION TO THE INVERSE 'F' FUNCTION
59 C -----
60 C Y1=1./Y1-1.)
61 C Y2=1./Y2-1.)

```

FISHIN-2

```

61      H=2./(Y1+Y2)
62      X=X*DSQRT(H+Y)/H-(Y1-Y2)*(Y+5./6.-2./(3.*H))
63      X=DEXP(2.*X)
64 C -----
65 C COMPUTE THE CONSTANT TO THE 'F' DISTRIBUTION.
66 C TESTING FOR N1 AND/OR N2 ODD OR EVEN
67 C -----
68      G=1.
69      IB1=2
70      IF (MOD(N1,2).EQ.0) GO TO 10
71      G=1.7724539
72      IB1=1
73 10    IB2=2
74      IF (MOD(N2,2).EQ.0) GO TO 20
75      G=G+1.7724539
76      IB2=1
77 20    IB3=2
78      IF (MOD(N1+N2,2).EQ.0) GO TO 30
79      G=G+1.7724539
80      IB3=1
81 30    IF ((IB1+IB2).NE.2) G=2.*G
82      GLOG=DLOG10(G)
83      IF ((N1+N2).LE.3) GO TO 50
84      ND=N1+N2-2-IB3+1
85      DO 40 I=1,ND,2
86          IF ((IB1+I-1).LE.(N1-2)) GLOG=GLOG+ALOG10(IB1+I-1.)
87          IF ((IB2+I-1).LE.(N2-2)) GLOG=GLOG+ALOG10(IB2+I-1.)
88 40    GLOG=GLOG-ALOG10(IB3+I-1.)
89 C -----
90 C COMPUTE THE VALUE OF FISHIN (DOUBLE PRECISION VALUE = FSHNDP)
91 C -----
92 50    Y2=N2/(N2+N1*X)
93      Y1=1.-Y2
94      XSP=X
95      K=1
96      Y=1.-ALPHA-FISH(XSP,N1,N2)
97      IF (Y.LT.0.) K=-1
98      IF (Y.EQ.0.) GO TO 60
99      Y=DLOG10(DABS(Y))
100     C=(N1*DLOG10(Y1)+N2*DLOG10(Y2))/2.
101     Y=GLOG+Y-C
102     IF (Y.LT.-75.) GO TO 55
103     Y=10.*Y
104     GO TO 60
105 55    Y=0.
106 60    Y=1.+K*Y
107     FSHNDP=X*Y
108 C -----
109 C IF FISHIN (FSHNDP) IS NEGATIVE, RESET TO .5*LAST APPROXIMATION(X)
110 C -----
111     IF (Y.LT.0.) FSHNDP=.5*X
112 C -----
113 C IF ABS. VALUE OF THE DIFFERENCE IS LESS THAN .5D-6, RETURN
114 C -----
115     IF (DABS(X-FSHNDP-1.)<.5D-6) GO TO 70
116 C -----
117 C IF REL. VALUE OF THE DIFFERENCE IS LESS THAN .5D-6, RETURN
118 C -----
119     IF (DABS(X-FSHNDP).LT.(.5D-6)) GO TO 70
120     IF (10.GE.10*H) GO TO 70

```

FISHIN-3

```
121      IC=IC+1
122 C -----
123 C SET APPROXIMATION EQUAL TO FISHIN (FSHNDP). CONTINUE TO ITERATE
124 C -----
125      X=FSHNDP
126      GO TO 50
127 70 FISHIN=FSHNDP
128      RETURN
129      END
```

FUNCTION SUBPROGRAM FISH

```
1 $ASSM
2 FISH PROG FISH - FCN SUBPROGRAM FOR EVALUATING FISHER'S F-DISTRIBUTION
3 $FORT
4 C -----
5 C THIS SUBPROGRAM COMPUTES THE VALUE OF FISHER'S F-DISTRIBUTION
6 C AT THE POINT F WITH N1,N2 DEGREES OF FREEDOM. IT IS A VERSION
7 C OF THE UNIVAC 1108 STAT-PACK ROUTINE FISH.
8 C
9 C THE VALUE OF FISHER'S DISTRIBUTION = THE PROBABILITY THAT A VARIABLE
10 C DISTRIBUTED WITH FISHER'S F-DISTRUBUTION WITH N1,N2 DEGREES OF
11 C FREEDOM IS LESS THAN OR EQUAL TO A GIVEN VALUE (F).
12 C
13 C NOTE--THIS PROBABILITY RELATES TO THE VALUES GIVEN IN TABLE 4 OF
14 C THE APPENDIX OF BOUKER & LIEBERMAN'S "ENGINEERING STATISTICS"
15 C AS FOLLOWS:
16 C THE TABLE GIVES THE VALUES F(ALPHA;N1,N2)
17 C THIS PROGRAM COMPUTES FISH(F,N1,N2) WHERE
18 C FISH = BETA = 1.-ALPHA
19 C BETA = PROBABILITY [ F(ALPHA;N1,N2) < OR = F ]
20 C (ALPHA IS CALLED THE CONFIDENCE COEFFICIENT IN THE STAT-PACK
21 C ROUTINE FISHIN.)
22 C
23 C FORTRAN CALLING SEQUENCE:
24 C > VARIABLE = FISH(F,N1,N2)
25 C
26 C FISH (REAL) = THE CALCULATED VALUE OF THE PROBABILITY
27 C F (REAL) = THE VALUE AT WHICH THE PROBABILITY IS CALCULATED
28 C (F CANNOT BE LESS THAN 0.0)
29 C N1 (INT) = THE DEGREEES OF FREEDOM OF THE FIRST SAMPLE
30 C N2 (INT) = THE DEGREEES OF FREEDOM OF THE SECOND SAMPLE
31 C
32 C NO OTHER SUBPROGRAMS CALLED
33 C -----
34 C FUNCTION FISH(F,N1,N2)
35 C DOUBLE PRECISION FISHP;:::Y,H,C,HLOG
36 C LOGICAL E1,E2,E3
37 C -----
38 C INITIALIZATION AND SETTING OF LOGICAL SWITCHES
39 C TO .TRUE. IF THE DEGREEES OF FREEDOM ARE EVEN
40 C -----
41 C E1=.FALSE.
42 C E2=.FALSE.
43 C E3=.FALSE.
44 C IF (MOD(N1,2).EQ.0) E1=.TRUE.
45 C IF (MOD(N2,2).EQ.0) E2=.TRUE.
46 C X=N2/(N2+N1+1)
47 C IF (.NOT.(E1.OR.E2)) GO TO 50
48 C IF (E1.AND..NOT.E2) GO TO 20
49 C IF (.NOT.E1.AND.F2) GO TO 10
50 C IF (N1.LE.N2) GO TO 20
51 C -----
52 C INITIALIZATION FOR SECOND DEGREE OF FREEDOM EVEN AND
53 C LESS THAN FIRST DEGREE OF FREEDOM IF IT, TOO, IS EVEN
54 C -----
55 C 10 I=N1
56 C N1=N2
57 C N2=1
58 C X=1.-X
59 C E3=.TRUE.
60 C -----
```

FISH-2

```

61 C      INITIALIZATION FOR FIRST DEGREE OF FREEDOM EVEN AND
62 C      LESS THAN SECOND DEGREE OF FREEDOM IF IT, TOO, IS EVEN
63 C      -----
64 C      20  Y=1.-X
65 C      -----
66 C      COMPUTE PROBABILITY FOR AT LEAST ONE DEGREE OF FREEDOM EVEN
67 C      -----
68 C      FISHDP=0.
69 C      HLOG=(N2/2.)*DLOG10(X)
70 C      I1=1
71 C      22  IF (HLOG.GT.-75.) GO TO 28
72 C      HLOG=HLOG+DLOG10(Y*(N2+2.*(I1-1))/(2.*I1))
73 C      I1=I1+1
74 C      GO TO 22
75 C      28  H=10.**HLOG
76 C      M=N1/2.
77 C      IF (I1.GT.M) GO TO 32
78 C      DO 30 I=1,M
79 C      FISHDP=FISHDP+H
80 C      30  H=(H**Y*(N2+2.*(I-1)))/(2.*I)
81 C      32  FISH=FISHDP
82 C      IF (E3) GO TO 40
83 C      -----
84 C      ADJUST CALCULATED PROBABILITY IF ITS
85 C      ONES COMPLEMENT WAS CALCULATED ORIGINALLY
86 C      -----
87 C      FISH=1.-FISH
88 C      RETURN
89 C      40  I=N1
90 C      N1=N2
91 C      N2=I
92 C      RETURN
93 C      -----
94 C      COMPUTE PROBABILITY FOR BOTH DEGREES OF FREEDOM ODD
95 C      -----
96 C      50  Y=1.-X
97 C      H=.6366197723*SQRT(X*Y)
98 C      C=DSQRT((1./X)-1.)
99 C      FISHDP=.6366197723*DATAN(C)
100 C      IF (N2.EQ.1) GO TO 70
101 C      M=N2-2
102 C      DO 60 I=1,M,2
103 C      FISHDP=FISHDP+H
104 C      H=H***(I+1)/(I+2)
105 C      IF (H.LT.1.D-75) GO TO 62
106 C      60  CONTINUE
107 C      GO TO 70
108 C      62  IF (I.EQ.1) GO TO 70
109 C      I1=I+2
110 C      HLOG=DLOG10(H)
111 C      DO 64 I=1,M,2
112 C      64  HLOG=HLOG+DLOG10(X*(I+1)/(I+2))
113 C      GO TO 72
114 C      70  HLOG=DLOG10(H)
115 C      72  IF (N1.EQ.1) GO TO 90
116 C      C=N2
117 C      HLOG=HLOG+DLOG10(C)
118 C      I1=1
119 C      74  IF (HLOG.GT.-75.) GO TO 78
120 C      HLOG=HLOG+DLOG10(Y*(N2+I1)/(I1+2))

```

FISH-3

```
121      I1=I1+2
122      GO TO 74
123  78   H=10.**HLOG
124      M=N1-2
125      IF (I1.GT.M) GO TO 90
126      DO 80 I=11,M,2
127      FISHDP=FISHDP-H
128  80   H=H*Y*(N2+1)/(I+2)
129  90   FISH=FISHDP
130      RETURN
131      END
```

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date	02. Summary prepared by (Name and Phone)	03. Summary action
Yr. Mo. Day	Thomas W. Bartel (301) 921-3783	New Replacement Deletion
7 7 1 0 2 8		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
04. Software date	05. Software title	Previous Internal Software ID
Yr. Mo. Day	Interactive Computer Program for the Determination of Reverberation Time	
7 6 1 2 3 0		
06. Short title	07. Internal Software ID	
REVERB	None	

08. Software type	09. Processing mode	10. General	Application area	Specific
<input type="checkbox"/> Automated Data System <input checked="" type="checkbox"/> Computer Program <input type="checkbox"/> Subroutine/Module	<input checked="" type="checkbox"/> Interactive <input type="checkbox"/> Batch <input type="checkbox"/> Combination	<input type="checkbox"/> Computer Systems Support/Utility <input checked="" type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Bibliographic/Textual	<input type="checkbox"/> Management/Business <input type="checkbox"/> Process Control <input type="checkbox"/> Other	Acoustics; reverberation room measurements

11. Submitting organization and address	12. Technical contact(s) and phone
Institute for Basic Standards National Bureau of Standards Washington, D.C. 20234	Thomas W. Bartel (301) 921-3783

13. Narrative

The program is used to measure the reverberation time in a reverberation room according to ASTM C423-66, Standard Method of Test for Sound Absorption of Acoustical Materials in Reverberation Rooms. The computer controls several elements of the instrumentation system through a special interface, acquires the digitized sound pressure levels of the decaying reverberant sound field, and determines the reverberation time through a linear least-squares computation. Transmission of data to and from the instrumentation system is handled through separate subroutines written in the computer's assembly language.

14. Keywords

Acoustics; computer-controlled instrumentation system; real-time analyzer; reverberation room; reverberation time.

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
Interdata Model 70	RTOS	Fortran V, Level 1, Interdata Assembly lang.	1436
19. Computer memory requirements	20. Tape drives	21. Disk/Drum units	22. Terminals
35,452 8-bit bytes	None	One/2.5 megabyte unit	One

23. Other operational requirements

132-character line printer
real-time analyzer, random noise generator, microphone multiplexer interfaced to computer

24. Software availability	25. Documentation availability
Available <input checked="" type="checkbox"/> Limited <input type="checkbox"/> In-house only <input type="checkbox"/>	Available <input checked="" type="checkbox"/> Inadequate <input type="checkbox"/> In-house only <input type="checkbox"/>
Paper or mag tape	Internal report

26. FOR SUBMITTING ORGANIZATION USE

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET		1. PUBLICATION OR REPORT NO. NBSIR 77-1383	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE Interactive Computer Program for the Determination of Reverberation Time			5. Publication Date December 1977	
			6. Performing Organization Code 200.03	
7. AUTHOR(S)			8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234			10. Project/Task/Work Unit No. 2003116	
			11. Contract/Grant No.	
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP) Same			13. Type of Report & Period Covered Final	
			14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES				
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) A description of the computer program used to measure the reverberation time in a reverberation room is presented. The program controls the operation of a real-time analyzer, a random noise generator, and a microphone multiplexer. The reverberation time for each digitally recorded decay curve is determined from a straight line least-squares fit. The program is written in FORTRAN V and requires approximately 35,000 eight-bit bytes of core memory. Flow charts, source listings, and sample printouts are included.				
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons) Automated data acquisition; computer; reverberation room, reverberation time; software.				
18. AVAILABILITY		19. SECURITY CLASS (THIS REPORT)		21. NO. OF PAGES
<input checked="" type="checkbox"/> Unlimited		UNCLASSIFIED		126
<input type="checkbox"/> For Official Distribution. Do Not Release to NTIS		20. SECURITY CLASS (THIS PAGE)		22. Price
<input type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13		UNCLASSIFIED		\$7.25
<input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS) Springfield, Virginia 22151				

