digital

lab 8/e

software system

user's manual

digital equipment corporation

DEC-8E-ALUMA-B-D

L A B - 8 / E

S O F T W A R E   S Y S T E M

U S E R ' S   M A N U A L

For additional copies, order No. DEC-8E-ALUMA-B-D from Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

First Printing, August, 1972
First Revision, February, 1973

Your attention is invited to the last two pages of this document. The "How to Obtain Software Information" page explains how to keep up-to-date with DEC's software. The "Reader's Comments" page, when filled in and mailed, is beneficial to both you and DEC; all comments received are acknowledged and considered when documenting subsequent manuals.

PREFACE

This manual supersedes the following manuals:

Advanced Averager User's Manual
(DEC-LB-U61A-D)

Basic Averager User's Manual
(DEC-LB-U6ØB-D)

PST, TIH and Latency Histogram Programs User's Manual
(DEC-LB-U4ØB-D)

Auto- and Cross-Correlation User's Manual
(DEC-LB-U41B-D)

DAQUAN User's Manual
(DEC-LB-U8ØB-D)

BASIC/RT User's Manual
(DEC-LB-U7ØB-D)

NMR Simulator User's Manual
(DEC-LB-U53A-D)

NMR Averager User's Manual
(DEC-LB-U52A-D)

and includes the LAB8/E Data Conversion program.

Advanced versions of the Advanced Averager; Basic Averager; TIH, PST and Latency; Auto- and Cross-Correlation; and DAQUAN programs have been written to run under OS/8. These programs are referred to as the MS (Mass Storage) versions.

CONTENTS

BOOK ONE

ADVANCED AVERAGER

CONTENTS

BOOK ONE

TABLES

CHAPTER 1

ADVANCED AVERAGER FOR PAPER TAPE

1.0 GENERAL DESCRIPTION

The LAB8/E Advanced Averaging Program digitizes, displays, and averages analog signals at rates from 175 μsec to 4 seconds/point. Prestimulus averaging, dual resolution sweeps, on-line computation of 95 percent confidence limits and trend, signal editing and sorting, preset sweep count, as well as a conversational compiler for selecting the parameters of an average, are provided.

Before the user proceeds to the actual running of the Advanced Averager, he should try to gain an overall understanding of the program.

The Advanced Averager was written to be as flexible a signal averager as possible. Practically all of the parameters that define an average are specified by the user. For this reason, an entire program has been devoted to the setup phase of the averaging process. This is the first of four programs (or sections) that make up the Advanced Averager. Section I asks and records the answers to the parameter questions. When all of the questions pertaining to the average have been answered, either the answers are left in the memory of the computer for use by the remaining three sections, or they are punched on paper tape (a control tape) which may be used in place of Section I at a later time. The latter feature provides a fast means of entering the averaging parameters to the remaining three sections of the Advanced Averager.

Once the average has been defined by either Section I or a control tape, the program that performs the data acquisition is read in. This is Section II. Since the primary purpose of Section II is to acquire and sum the data, the command structure is limited to data acquisition and display commands, such as: start summing, stop summing, start again, display the input or the sum, and expand or contract the sum.

When the data has been collected, Section III is called in to process it. At this time the true average is calculated and the final computations for the statistics are performed. Because this section is devoted entirely to computation, there is no display and no keyboard interaction.

Upon the completion of the calculations, the output program (Section IV) is read. This program is designed to make the data available to the user in as many useful ways as possible. Display of each average and its associated statistics is provided for photography along with selective type-out and integration routines. Plotter calibration and plotter output routines are also provided.

When output has been completed, another average may be taken by reading in a control tape or Section I, followed by Section II.

It should be understood that while one section is present in core, other sections are not present and may not be referenced.

1.1 Program Tapes

There are seven binary tapes which make up the LAB8/E Advanced Averaging System. These are: DEC-8E-AAP1A-A-PB, DEC-8E-AAP2A-A-PB, DEC-8E AAPTA-A-PB1-PB4, and DEC-8E-AAAPA-A-PB. DEC-8E-AAP1A-A-PB contains all four sections of the Advanced Averager. It should be used if the system does not have disk or DECtape. Although the sections are contained on one tape, they are separated by "leader/trailer" code. This signals the loader program that the end of a section has been reached and reading should stop until the next section is required. DEC-8E-AAP2A-A-PB contains Sections II-IV and should always be used if a control tape is used instead of Section I and the system does not have disk or DECtape. DEC-8E-AAPTA-A-PB contains Sections I-IV and should not be used when the program is to be run from either disk or DECtape. Control tapes should be used in conjunction with DEC-8E-AAAPA-A-PB when disk or DECtape is used. If the system is run from disk or DECtape, refer to instructions in Appendix B for loading and storing the programs. The Averager can be run under OS/8 and data can be written to mass storage. Refer to Chapter 3 for program tapes and instructions.

1.2 Basic System Requirements

PDP 8/E Computer	with at least 4K of core. Additional core will be utilized automatically by the program.
AD8-E + AM8-EA + AM8-EC	10 bit analog to digital (A/D) converter with pre-amplifiers and eight channel multiplexer.
DK8-ES	Real time programmable clock.
VC8-E	10 bit display controller.

H945AA	Mounting panel and power supplies for A/D, clock, and display controller.
VR14, Tektronix 602,603,604,RM503 or other acceptable scope	Display oscilloscope.
ASR33	Teletype[1].
High-speed reader	Strongly recommended.

The OS/8 version requires the standard OS/8 configuration.

[1]Teletype is a registered trademark of the Teletype Corporation.

CHAPTER 2

RUNNING THE ADVANCED AVERAGER

2.0 SYSTEM INITIALIZATION AND PROGRAM LOADING

For the LAB8/E programs to run properly, the PDP-8/E must have its knob and switches set in certain positions. Before running any programs, the user should check to see that the knob and switches are set in their correct positions.

The user starts the system by turning the key on the left-hand side of the PDP-8/E console to POWER. The key must not be turned as far as PANEL LOCK. Next, the user must ensure that the HALT and SING STEP keys are in the up position. Then, the status knob in the right-hand center of the console is turned to AC. This step enables the contents of the accumulator (AC) to be seen in the lower of the two rows of lights. Next, the Teletype switch is turned to LINE, and, finally, the scope is turned on. The LAB8/E system is now ready to receive programs.

All of the LAB8/E programs are loaded with the Binary Loader program (BIN). If the BIN Loader is not in core (field Ø), it must be loaded with the RIM Loader. See Appendix A for details of loading BIN and RIM.

To load the Advanced Averager program from paper tape:

a. Place the tape (DEC-8E-AAP1A-A-PB, Sections I-IV) in the tape reader so that the leader/trailer (a single channel 8 punch) is on top of the reading mechanism.

b. Turn the Teletype control knob to LINE.

c. Turn the scope to ON.

d. Set the Switch Register (SR) to 7777_8. (All switches in the up position.)

e. Press the ADDR LOAD key.

f. If high-speed reader is used, turn on reader and set SR to 3777 (bit Ø in down position).

g. If a low-speed reader is used, set the reader switch to START.

h. Press the CLEAR and CONT switches on the computer console.

i. The tape should start moving through the reader. If it does not, BIN is not present and must be loaded per the instructions in Appendix A.

j. When the tape stops, check that the AC lights are all off (to see the accumulator, turn the status knob to AC). If they are not all off, a read error has occurred and steps a. through j. should be repeated.

k. When loading is complete (AC≠Ø), press CONT.

The following message appears on the scope:

LAB8/E IS READY

HIT RETURN TO PROCEED

RETURN is the button on the far right of the Teletype keyboard. Press this key to go to the next display. RETURN is used throughout the LAB8/E program to go on to the next step or to "execute" an answer which has just been typed.

2.1 The LAB8/E Compiler (Section I)

Section I defines the average to be taken. The answers the program receives are used throughout the remainder of the Advanced Averager.

1. Typing the RETURN key after the introductory display causes the following question to appear on the oscilloscope:

   DIGITAL I/O? (Y OR N)
   (Y FOR YES, N FOR NO)

   This question asks if the digital I/O option (12 bits in, 12 bits out) is part of the system configuration. If N for NO is typed, no further questions are asked regarding digital I/O. If the option is part of the system, but the contingency feature is not to be used in a particular experimental run, N can be answered to this question. RETURN must be typed after either Y or N to execute the answer. If Y is answered, then question 10 is asked later.

   NOTE

   a. If YES is answered to the above question, the digital I/O must be the unmodified DR8-EA for the program to function properly.

   b. In all conversational mode YES or NO questions, SPACE, RETURN, or any character other than Y is interpreted as NO.

2. After the RETURN key is typed, the following will appear regardless of the answer to question 1.

   SYNC ON INPUT: S___

The blank can be filled in with Ø, 1, 2, or 4. This is the number of the phone jack connector on the front of the laboratory panel to which the synchronization cable is attached. Attach the lead now. S1, S2 and S4 are inputs for either fast pulses or slowly varying analog signals (a coupling can be used to lead a signal to both a sync input and one of the main analog signal inputs, allowing the LAB8/E to synchronize on the signal itself). The knobs above S1, S2, and S4 control the levels at which the input pulse (or signal) causes the Schmitt triggers to fire. For details of Schmitt trigger firing, see Appendix C.

If SØ is chosen for the sync pulse, no sync lead is required. The program will automatically trigger itself every Least Sweep Interval (for definition, see question 17). Simultaneously with start of the sweep, a +3 to Ø volt pulse, the duration of a sample (see sampling rate, question 9) will be presented to bit Ø of the digital I/O. This enables the LAB8/E to control the stimulation in this particular trial.

NOTE

The user should always verify that the scope displays the intended answer before typing the RETURN key. The user can correct errors by using the RUBOUT key.

3. The following display appears next:

   STANDARD RESOLUTION:

   - - - - DATA POINTS

   Standard resolution refers to the main signal epoch which this and the following two displays define. It takes the following three parameters to define a signal epoch for the LAB8/E.

   a. Number of data points into which the waveform is to be digitized,

   b. Length (in milliseconds or seconds), and,

   c. Delay (amount of time either before or after the sync pulse that the signal epoch begins).

   Any number from 65 to 1Ø24 (using the basic system requirements) can be typed. However, standard resolution is only one of the two basic building blocks of the LAB8/E system. It may be associated with specific analog input and sort code (with digital I/O) before signals defined by this signal epoch can be averaged. Therefore, it should be remembered that (in the basic configuration) if 1Ø24 data points are used to define the standard resolution epoch, there will be no room for high resolution, standard deviation, or trend. A reasonable choice might be 2ØØ data points for the standard resolution epoch.

4. After entering the number of data points, followed by a carriage RETURN, the following will be displayed:

   LENGTH:_____ _SEC

This asks the user to define the length of the signal epoch he wishes to look at or analyze. Note that there is a space between the fifth and sixth blanks. This indicates that the first five blanks are for one to five numbers (or one to four plus decimal point) which must be followed by a carriage RETURN, and the sixth is for either an M to indicate milliseconds or SPACE or RETURN to indicate seconds.

Numbers may be entered in any position (leading or trailing blanks and leading Ø's are not required). The sampling rate (epoch length/number of data points) can be between 175 µsec/point and 4 sec/point. For faster sampling rates the Basic Averager must be used (refer to Book Two of this manual).

5. The only parameter yet to be specified for the first basic signal epoch is the time at which the epoch begins (with reference to the sync pulse).

    DELAY:_____ _SEC

Note that this display is similar to the preceding; however, a minus sign (hyphen on keyboard) is allowed to indicate a signal epoch beginning before the sync pulse. If a negative delay is used, the sync pulse must arrive no later than the time equivalent of the last data point. In other words, a negative delay cannot be longer than the length of the signal epoch. Otherwise, the following error message will appear:

    ? SWEEP END EARLY

indicating that the sweep would end before sync pulse arrived. Type the RETURN key to respecify the delay. Type LINE FEED to change both length and delay.

A negative delay is of use if the user suspects his system of anticipating the stimulus and reacting before the stimulus actually arrives, or if the user is interested in detecting any CNV that may be present. Specifying a negative delay will enable him to view the state of his system before the stimulus arrives. Thus, he sees any pre-stimulus response which otherwise would have remained obscure and the base line before the onset of response.

A positive delay would be used if the elicited response did not occur immediately following the stimulus, but at a later time (i.e., the time until response was long in comparison to the length of the sweep). Using a positive delay would yield a sweep whose focus was on the area of interest and not over the entire time range.

NOTE

> In any display requiring a numerical answer, typing the RETURN key only is interpreted as an answer of Ø.

If only one resolution is given, it is referred to as the high resolution; however, to change it refer to the standard resolution input.

Defining the Second (or High Resolution) Basic Signal Epoch

6. The next display is:

HIGH RESOLUTION EPOCH

_ _ _ _ DATA POINTS

Defining two epochs (standard and high resolution) permits simultaneous averaging of signals in two different time intervals at two different sampling rates on the same or different signals. Averaging is done at the standard resolution only if this question is answered with Ø or RETURN.

If the high resolution epoch is used, at least 65 points must be requested. The maximum number of points that can be specified by the combined epochs is 1Ø24; if 4ØØ points were used in the standard resolution, up to 624 points could be specified here.

NOTE

There are $3Ø72_{1Ø}$ core locations available for storing data. It takes three locations to take an average of one point (one for temporary storage and two for the average). Therefore, if only an average is taken, the maximum number of points that may be averaged is 1Ø24 (3Ø72/3). If more core is available, the maximum possible number of points is increased.

7. Typing RETURN will now bring up the following display:

HI LENGTH:_____ _SEC

If the high resolution epoch length typed results in a per-point sampling rate which is slower than that resulting from the definition of the standard resolution epoch, the following error message appears:

? BAD SAMPLING RATIO

This means that either the number of data points in the high resolution epoch or the length must be redefined so as to give a new sampling rate faster than or equal to the standard resolution sampling rate. Type RETURN to change the high resolution parameters. Type LINE FEED to change both high and standard (or low).

8. After typing RETURN, the following is displayed:

HI DELAY:_____ _SEC

Again a delay (positive or negative) is permitted. This allows the user to position a high resolution window about an area of particular interest. RETURN must be typed after the numerical part of the answer and again after the unit portion (M for milliseconds or SPACE for seconds).

If the epoch lengths and delays selected for standard and high resolution are such that there is no overlap between them, the following error message appears:

? NO COMMON INTERVAL

Type RETURN to respecify both high and standard (low) resolution epoch lengths and delays. They must be respecified so that the epochs overlap by at least one point. Type LINE FEED to change only the high resolution parameters.

As noted, the number of high and low resolution data points must exceed 64. If 64 or less points are specified for the standard resolution, the question is reasked. If 64 or less points are specified for the high resolution, a straight-line display results.

9. After successfully defining the three parameters for both the standard and high resolution signal epochs, the following sweep summary appears (these figures are examples only):

BEGINS	AT RATE	ENDS
Ø	5ØØU	99.5ØM
-19.5ØM	1.ØØØM	179.5M
-: CHANGE (H,L__)		

In the above example, the standard resolution epoch samples the first point at 19.5Ø ms before the sync pulse, the sampling rate is 1.ØØØ ms per point, and the last data point is 179.5 ms after the sync pulse.

The high resolution epoch begins at the sync pulse, the sampling rate is 5ØØ µs/point (twice the rate of the standard resolution epoch) and the epoch ends 99.5Ø ms after the sync pulse.

When the high resolution is displayed, it is shown to have the same length as the standard resolution, so it is effectively blown up in the horizontal direction.

NOTE

a. All measurements (such as DELAY, SORT AT, etc.) in the use of the LAB8/E are referenced to the leading edge of the sync pulse.

b. Both standard resolution and high resolution epoch lengths and delays should be integer multiples of the fastest sweep rate selected. If the user selects fractional multiples, the LAB8/E calculates and displays the closest approximation. This is the sweep summary display.

c. There is an uncertainty of 1 standard resolution sampling interval in the sweep start time. This is expressed in the sweep summary by not rounding the low resolution begin and end times.

d. If only a standard (low) resolution is specified along with a <u>positive</u> delay time, then the high resolution becomes the standard (low) resolution and the BEGIN and END values of the low resolution sweep summary equal the positive delay time. (A zero or negative delay appears as zero in the low resolution sweep summary.) These low resolution values are ignored by the remaining sections of the program. Only the high resolution figures in future calculations.

There are only two data rates in the system, the high and low resolution rates. Then for all channels used, there can be the high resolution or the low resolution, or both rates on each channel. The system requires that the low resolution rate be a multiple of the high resolution rate. Namely, if the H.R.R. is R microseconds between points, then the L.R.R will be M*R microseconds between points where M=1,2,3. Hence, if the H.R.R. is 2ØØ microseconds between points then the L.R.R. must be 2ØØ or 4ØØ or 6ØØ, etc. microseconds between points. If the user puts in a high and low resolution length and number of points such that the L.R.R. is not a multiple of the H.R.R., then the low resolution rate will be adjusted to the nearest larger rate so that the rates are multiples. For example:

LOW RES.	300 points 200 M seconds	RATE=666μ
HIGH RES.	200 points 100 M seconds	RATE=500μ

Then the rate for the low resolution will be set to 1 millisecond.

10. If YES was typed in answer to DIGITAL I/O? (Question 1) the following will appear, otherwise Question 11 will appear:

SORT AT_____SEC

This refers to the contingency feature and specifies the sort time with reference to the sync pulse. The sort time is the instant at which the computer will look at the states of the rightmost eight bits of the digital contingency inputs; the states of any of these inputs can have changed at any time during the interval between the sync pulse and the sort time. At sort time the computer stores the pattern of Ø's and 1's indicating the states of the eight digital contingency inputs. When the averages are updated with the current sweep, this pattern is compared with the sort code specified for each discrete average waveform. (The sort code will be specified later at the same time that individual analog inputs are matched up with the standard and high resolution epochs.) If the pattern contains 1's in the same positions as in the corresponding sort code, the current sweep will be added to the average corresponding to this sort code; if not, the sweep will not be used in the average.

NOTE

a. For the program to read a "1" from any of the digital inputs at sort time, the digital input must have seen a ground at any time from the sync time to the sort time. If no ground is seen during this time period, Ø's will be read.

b. Since the computer will not look for the next sync pulse until after sort time, this feature can be used to artificially lengthen the sweep for purposes of disregarding sync pulses occurring too soon after the end of the previous signal epoch.

11. After typing RETURN, the following scope display will appear:

    AVG #1

    ANALOG INPUT__

In this display, ANALOG INPUT__ asks which analog input will be used for Average Number 1. If the first input is used, Ø can be typed, followed by a carriage RETURN.

Analog inputs can be associated with discrete averages in any order.

Example:

    AVG #1
    ANALOG INPUT 3

    AVG #2
    ANALOG INPUT 1

    AVG #3
    ANALOG INPUT 2

Since octal notation is used to specify the inputs, there are no inputs 8 and 9. In the basic configuration only inputs Ø-7 can be specified. If more inputs are implemented, inputs Ø-17 may be specified.

NOTE

For further explanation of octal and binary numbers, refer to Chapter 1 of Introduction to Programming, Volume 1.

12. The next display (if both high and standard resolutions have been defined) is:

    RESOLUTION:__(H,L)

which asks which of the two basic signal epochs (windows) is to be used to define average #1. H indicates high resolution; L indicates standard or low resolution. All averages using high resolution should be specified before low resolution averages so that the positions of the averages on the screen are in the same order as for their corresponding raw input (Monitor) displays.

13. If YES was answered to the original DIGITAL I/O? question, the following is displayed after typing RETURN, otherwise question 14 is displayed.

SORT CODE:___

NOTE

Analog input numbers and sort codes are the only parameters which must be specified in octal notation. All others are in the familiar decimal notation.

SORT AT_____ and SORT CODE:___ are the two displays which specify the parameters of the contingency condition. With the digital I/O option the contents of the temporary storage buffer are not automatically emptied into the series of memory locations allocated for the accumulation of a certain average. This is done only if certain conditions, or contingencies, are fulfilled. The sort code is what specifies the condition for the addition to memory of the most recently digitized data. For example, if a sort code of ØØ1 were specified for average #1, data acquired during each sweep would only be added to those memory locations allocated to average #1, if, at sort time, the status of the inputs in the contingency register were as follows:

C4	C5	C6	C7	C8	C9	C10	C11	
Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	(binary)
Ø			Ø			1		(octal)

a. C4-C11 refer to the rightmost 8 bits of the digital input register.

b. A logical AND is performed between the sort code and the contents of the contingency register. If the result of the AND procedure is equal to the sort code, then the contingency condition is satisfied.

c. In the above example, therefore, the contingency condition is also met when the contingency register contains a ØØ ØØØ 111, or a 1Ø 11Ø 111, or, in fact, whenever C11 contains a 1 regardless of the contents of the remaining bits in the register.

The contingency feature can be used to generate a wide series of different average waveforms from the same analog input based on changes in experimental conditions. The changes in experimental conditions, in turn, cause changes in the contingency register. This feature could be used to sort responses to different stimuli so that several different stimuli could be presented in random order. This feature could also be used to edit abnormal responses. If an artifact were present in the response, the contingency register could be set so that the particular response would be discarded or saved in a separate average (the signal would be discarded if the contingency register were set so that it did not correspond to any sort code). If the contingency feature is not to be used for a certain average, Ø or RETURN should be typed for the sort code.

14. After the RETURN key is typed, the following is displayed:

    CONFIDENCE LIMITS?:___

    (Y: YES)

    asking if confidence limits are desired for each point on this average.

NOTE

a. A single confidence limit data point requires three memory locations, whereas an average data point requires only two. In order to conserve memory locations, confidence limits should be requested for averages consisting of as few locations as possible. Relatively few data points could be allocated to the standard resolution epoch, for example, and confidence limits could be requested for this. Twice or three times the data points could then be allocated to the high resolution epoch, without confidence limits. If the lengths and delays were identical, this arrangement would produce one confidence limits point for every two or three average points, a ratio which is usually acceptable.

b. Confidence limits and trend points are displayed in Section IV (Output Section).

c. See Appendix D for confidence limit equation.

15. If Y (YES) is answered in response to the above display and the RETURN key is typed, the following appears:

    COMPUTE TREND ?: ___

    (Y: YES)

    The significance of trend will be explained in the description of Section IV of the program.

NOTE

a. Each trend data point requires four memory locations in addition to the three for confidence limits and two for its associated average. Trend cannot be calculated without confidence limits.

b. See Appendix D for trend equation.

c. To optimize the power of LAB8, a few minutes might be spent calculating the maximum number of points available to the user in various situations. The following equation allows simple determination of this. Keep in mind that the number of memory locations used must not exceed 3Ø72 + n X 4Ø96, where n = the number of additional memory stacks.

Number of memory locations used =

x	+	2x	+	3x	+	4x	+	y	+2y	+	3y	+	4y+..
TEMP BUFFER		AVG. TERM		CONF. TERM		TREND TERM		TEMP BUFFER	AVG.		CONF.		TREND

where x = the number of standard resolution points and y = the number of high resolution points.

If some of the statistics are not used, their respective terms should be dropped from the equation.

16. When the above question is answered and carriage RETURN is typed, the following display appears:

AVG #2

ANALOG INPUT__

This can be answered in the same manner as in steps 11-15. If two resolutions are being used and average #1 is reserved for the high resolution, average #2 could be associated with the same input signal and question 12 should be answered with an L for low resolution. Additional averages can be set up as required until the following display appears (only after the last question for an individual average):

? INSUFFICIENT MEMORY

If an average with confidence limits resulted in this error message, or if trend was specified as well, it might be possible to set up just the average, and possibly confidence limits without exceeding the memory available.

Type the RETURN key to respecify this average. Type the LINE FEED key to respecify all averages. In certain cases, where a sampling rate close to the maximum possible was specified, the following error message may appear:

? TOO MANY INPUTS
FOR SWEEP RATE

It takes a finite amount of time for the computer to switch back and forth between a series of inputs, and this time may make it no longer possible to maintain the sampling rate. To remedy this, either a slower sampling rate should be specified or fewer inputs should be requested. Type the RETURN key to delete the last average. Type the LINE FEED key to delete all averages. Type CTRL/Z to start from the beginning.

NOTE

a. Under most circumstances when in the Conversational Mode Compiler section (Section I), it is possible to go back to the previous display by typing the LINE FEED key, in which case the previously-typed answer will be discarded and must be respecified.

b. In case it is desired to respecify the parameters of the averages, type the CTRL/Z keys (CTRL key depressed at the same time as the letter Z) to return to the display:

LAB8/E IS READY

HIT RETURN TO PROCEED

in which case all parameters can be redefined.

17. When sufficient averages have been set up, typing the LINE FEED key discontinues this section of the compiler and brings up the following display:

LEAST SWEEP INTERVAL

_____ _S (>238.Ø MS)

(238.Ø MS is an example only)

This indicates that to do the calculations which the LAB8/E makes between sweeps, the sync pulses can be no closer together than 238.Ø ms (in this example). External sync pulses can thus be at regular 1/4, 1/2, 1 second intervals, etc., or they can be at irregular intervals the shortest of which is 239.Ø ms (in this example). If the SØ sync ouptut is used, the LAB8/E supplies pulses at exactly the interval specified in answer to the above question. Sync pulses occurring at a shorter interval from the last acknowledged sync will be ignored (and not acknowledged).

NOTE

a. Any time between the least sweep interval and the user-specified minimum interval typed above is used by the LAB8/E to stabilize the display of inputs. The shortest interval desired or expected under experimental conditions and not the shortest interval allowed by the LAB8/E should normally be selected.

b. Because of differences between individual machines, trouble may develop if the user specifies a least sweep interval that is very close to the shortest interval allowed. This is especially so in a case of multiple inputs. The symptoms of the trouble will be a halt of the averaging process before the preset number of sweeps is reached. This is remedied by increasing the least sweep interval.

18. Typing the RETURN key brings up the following Preset Sweep Counter display:

AVERAGE____SWEEPS

The largest number that can be typed as an answer to this display is 4Ø95. However, answering Ø or RETURN is equivalent to requesting 4Ø96 sweeps. In the Signal Averaging section (Section II), typing the CTRL/R keys allows repetition of the present number of sweep counts.

19. The next display shows the positions the actual inputs occupy on the oscilloscope screen during the View Input mode of the on-line operation. As an example:

INPUTS

```
.          .          .
1:Ø, H     5:Ø, L
.          .          .
2:5, H     6:3, L
.          .          .
3:2, H     7:4, L
.          .          .
4:1, H
.          .          .
```

The numbers 1-7 indicate that, in this example, there will be seven different inputs and seven different oscilloscope traces in the View Input mode. The middle numbers indicate the input channel numbers associated with each trace, and the letters H or L represent high or low resolution signal epochs. Note that all high resolution traces precede low resolution traces in this INPUTS display.

20. Typing the RETURN key brings up the following display:

AVERAGES

```
.          .          .
1          5
.          .          .
2          6
.          .          .
3          7
.          .          .
4
.          .          .
```

If no more than one contingency condition for the same signal epoch (either high or standard resolution) on the same analog input has been selected, the number of inputs will be the same as the number of averages. The numbers in the above display correspond to AVG 1, AVG 2, AVG3, etc.

Typing the LINE FEED key causes a return to the previous question between the Sync question and the rate display (shown in step 9). To change answers after a RETURN is typed to question 9, CTRL/Z should be typed to return to question 1. CTRL/Z does not work while displays 19 and 20 are shown, and once the sweep summary has started to print. If any key is hit once the sweep summary starts to type, the "TRIGGER" section is skipped, and Section II of the tape is read.

21. Typing the RETURN key brings up the following display:

PUNCH CONTROL TAPE?___

(Y: YES)

Punching a control tape allows the user to:

a. Establish a precisely repeatable setup for averaging.

b. Avoid reading in the compiler (Section I) each time a particular set of parameters chosen is desired for an average.

c. Vary certain parameters during the averaging section (Section II) by reading in different control tapes and restarting Section II at 653Ø (set 653Ø in the switches, press ADDR LOAD and then CLEAR, CONT).

If a control tape is not desired, type N or the RETURN key and a summary of the averaging parameters will be typed. At the end of the summary a line similar to the following is printed[1]:

(VAP,Ø, 23Ø-6277

VAP indicates the version being used - in this case, version A.

Ø indicates the number of data fields (blocks of 4K of core) minus 1. Ø means that the machine is a 4K machine. 23Ø-6277 spots the area of core that is available for the storage of data.

Trigger Adjustment

After this message, the following is typed[2]:

TRIGGER

The operator should now consult the sweep summary typeout which corresponds to the parameters now in memory. Included will be the statement SYNC ON INPUT S___, followed by Ø, 1, 2, or 4. If SØ is being used, this section of the program actually has no function, since the sweep will begin as the result of an internal, computer-generated pulse which needs no adjustment. Depending on which of the numbers 1, 2, or 4 was selected, the operator should make sure that a cable carrying the sync pulse is connected to the proper phone connector on the front panel of the laboratory peripheral: S1, S2, or S4, respectively. The operator should then make sure that the input pulse, or analog signal, is in the range of ±5V at the time it is desired that the Schmitt trigger fire, and that this voltage is actually being delivered by the cable. An oscilloscope could be used for this purpose.

[1]This message is meaningless when output by the Advanced Averager MS program.

[2]If the number of channels and size of the channel buffers are such that the end of buffer link is stored in locations 4272_8-4365_8, the trigger section is destroyed and the results are unpredictable. If this situation occurs, create a control tape of the Q & A section and run the Averager with that tape.

An analog signal in the range of ±1V should now be connected to Analog Input Ø. If the Schmitt trigger is firing, this signal is displayed on the scope every time a sync pulse is received. For those acquainted with the wired-program averaging devices, the characteristic sweep of the signal from left to right beginning at the sync pulse will be familiar.

NOTE

> This is true, however, only in this section of the program. In the on-line averaging section, the movement of the illuminated part of the display from left to right across the waveform has nothing to do with the arrival of a sync pulse or the actual beginning of the sweep; it is merely the beginning of another display cycle.

The clock knob with the same number as the Schmitt trigger being used should now be adjusted to ensure reliable, regular firing of the Schmitt trigger every time a sync pulse is received. If displays appear on the scope periodically regardless of the knob (threshold) setting, the sync pulse apparently goes beyond ±5V each time. If it is important to synchronize the sweep with a particular point on the waveform arriving at the sync connector, the amplitude of this point should first be measured and then the Schmitt trigger threshold adjusted to this value. Once the synchronization has been satisfactorily adjusted, type any key to go on to the on-line averaging section (Section II). (Refer to Section 2.2.)

Ignore the dot on the X axis of the scope while the TRIGGER and channel Ø are being viewed in the TRIGGER section of Section I. It is not part of the input signal.

NOTE

> To reset the timing before proceeding to Section II, stop the computer and reload Section I via the Binary Loader.

d. An ! and a [(on the next line) are printed to request the loading of Section II. Turn the reader ON and Section II will begin loading. Proceed to the description of Section II.[1]

22. If Y for YES is typed followed by the RETURN key, the following display appears:

    TITLE

Note that there are twenty-one blanks which can be filled in. In addition to letters, numbers and normal punctuation, the following characters can be used:

[1]Not done in MS version.

!	Exclamation Mark	#	Number Sign
%	Percent Sign	/	Division Sign
_	Space	( )	Parentheses
<	Angle Brackets	↑	Up Arrow
>	(both directions)	←	Back Arrow
x	Multiplication Sign (*)	=	Equal Sign

After 21 (or fewer) characters are entered, type the RETURN key to indicate the end of the title identification. At this point the oscilloscope screen will go blank and ↑ will be printed.

To punch out a control or parameter tape:[1]

1. Push the button marked ON on the box at the upper left of the ASR33 Teletype, making sure that the leftmost switch (bit Ø) of the Switch Register is up (a binary 1), or,

2. With the high-speed paper tape punch, make sure that the leftmost switch (bit Ø) of the Switch Register is down (a binary Ø).

Now type CTRL/P (hold down the CTRL key while typing P). This causes the punch to begin punching out the control (parameter) tape. The first part of the tape to be punched is the 21 (or fewer) character title in hand-readable form (holes are punched in the tape in the same configuration as dots for the corresponding characters displayed on the scope). The strip of tape immediately following the hand-readable title will contain the ASCII (Teletype) characters making up the sweep summary. At the same time that the control tape is being punched, the teleprinter types the sweep summary corresponding to the parameters contained on this tape. This sweep summary can be torn off and attached to the control tape for future reference. As many control tapes as required can be generated with one loading of the compiler (Section I). The compiler returns to display 1 to permit the user to specify a new set of parameters if Y was answered to PUNCH CONTROL TAPE?___. (To determine what parameters are contained in a particular control tape, switch the Teletype to LOCAL and read in this first section of the tape. The punched tape characters immediately following the title should produce the sweep summary.) The last section (separated by a length of tape with continuous holes on the right side) contains the actual parameters required by the computer in concentrated form. Only this last section should be read followed immediately by the shorter of the two LAB8/E program tapes marked Use with Control Tapes Only.

NOTE

The user may return to display 1 at any time during Section I (except between questions 19 and 20) by executing the CTRL/Z command.

Reading a Control Tape for Paper Tape

The condensed information of a control tape (described above) is used to drive the averaging process according to the responses the user made in Section I of the averaging program.

[1] Not applicable to MS version.

a. Place the control tape in the reader with the leader/ trailer codes over the reading mechanism.

b. Turn the Teletype control knob to LINE.

c. Turn the scope to ON.

d. Set the SR to 7777_8.

e. Press the STOP and ADDR LOAD keys.

f. If using a low-speed reader, set the reader switch to START.

g. If using a high-speed reader, set the SR to 3777_8 and turn on the high-speed reader.

h. Press CLEAR and CONT switches. When tape stops, verify that the accumulator displays Ø.

i. If error has occurred (AC≠Ø), return to step a. (If this fails, verify that the Binary Loader is in core. (Refer to Appendix A.)

j. Place tape DEC-8E-AAP2A-A-PB in the reader.

k. Repeat steps b. to i.

When loading is complete, check to see that the twelve lights of the accumulator are off (this corresponds to the condition AC=Ø). If this is not the case, a read error has occurred and the tape should be reloaded until such time that the AC does equal zero. When AC=Ø, press the CONTinue key on the computer console. Section II starts.

2.2 On-Line Signal Averaging, Confidence Limits and Trend Data Collection

After the tape has been read, turn the reader to STOP[1]. Verify that the accumulator is Ø; if not, reread Section II[1]. Connect the leads to the analog inputs if this has not already been done. Press CONTinue on the console to proceed with Section II[1]. The LAB8/E system types < and starts in the View Input Mode. This means that whatever is displayed on the scope initially is the actual analog input to the system and not an accumulating average. This is done so that the operator may adjust the gains on his pre-amplifiers to produce outputs as close to ±1V as possible without clipping (flattening) the highest amplitude peaks. This can be done before the actual averaging is begun.

The traces which are displayed when the on-line section of the program starts have the same heights and widths and are in the same positions as in the INPUTS display (the INPUTS display was one of the last displays in the compiler section of the program). If many inputs are being used, or if their positions on the scope are somewhat unusual, it may be advisable to take a Polaroid scope photograph of the INPUTS display during the compiler operation to use as a reference during on-line operation.

[1]Not applicable in MS version.

NOTE

> If a relatively fast sampling rate has been specified, a large number of inputs selected, confidence limits and trend requested, or relatively little time allowed between sync pulses, the LAB8/E may have so many operations to perform that it does not have time to display more than every fourth or fifth data point. Display is considered a low priority operation in the LAB8/E because A/D conversion, multiplexing, storing the results of A/D conversions, confidence limit and trend computation (if requested) are considered more important. For this reason, the quality of the display may be somewhat degraded under certain circumstances during on-line operation.

If the V command is now typed, the LAB8/E will switch the view mode to show the contents of the memory locations in which the average waveforms will be accumulated. (Confidence limits and trend are not displayed during the on-line operation.) The average is automatically zeroed so that all of the traces in this display should now appear as straight, horizontal lines. Typing V again will switch the view mode back to the digitized inputs.

Type RETURN to begin the averaging process. Waveforms should now appear to grow where the straight lines had been in the previous display. If the straight lines do not change, the slops and threshold adjustments on the Schmitt trigger should be altered until the straight lines do change, see Appendix C for adjusting the Schmitt trigger. Typing V will change the view mode to View Input.

Typing V again will change it back to view averages. If a very fast sampling rate was specified, only a flash of light at the beginning of the display cycle may be visible. When the display appears to go off scale, type the command C for contract, which will divide the vertical display scale by a factor of two. This may be necessary since the data is saved as a 24-bit sum. X can also be typed to multiply the vertical display scale by a factor of two (X stands for eXpand).

The character > is printed when the preset number of sweeps has been attained. This display is the non-normalized sum of the data taken from each separate sweep. To erase this data and begin a new average, type CTRL/Z followed by the RETURN key. The computer will stop averaging and display the new waveform after the same preset number of sweeps.

If it is desired to take another group of sweeps, type CTRL/R and the RETURN key and the preset number of sweeps will be taken and added to the first group.

Table 1-1 is a summary of the control characters used in the on-line section of the LAB8/E Advanced Averager Program.

Table 1-1

On-Line Control Characters

Character	Explanation
CTRL/Z	Zeros (reinitialized) data points used for accumulation of average waveforms, sets view mode to View Input, prints <.
RETURN	Starts or resumes averaging process, sets view mode to View Average, types carriage RETURN and LINE FEED; returns display to zero scale.
LINE FEED	Interrupts averaging process (without initializing average to zero), sets view mode to View Input, types two LINE FEEDs.
V	Changes view mode (either from View Input to View Average, or vice versa).
X	Expands display of average by a factor of 2.
C	Contracts display of average by a factor of 2.
CTRL/R	Resets sweep counter to begin preset number of sweeps again. Previous average is not erased so that new sweeps are added to old after RETURN is typed. CTRL/R or CTRL/Z is required to resume averaging after > has been printed.
CTRL/P	Pulls in next section of the LAB8/E Advanced Averager Program tape.
The following characters are printed only:	
>	Indicates average is done (preset number of sweeps has been attained).
!	Indicates a requirement to read more tape. Turn Teletype reader to START if the low-speed reader is used. Does not apply to MS version.
[	Requests next section to be read.

NOTE

> Although confidence limits and trend can be computed during Section II, the subroutines for displaying confidence limits and trend data points (made up of three and four memory locations, respectively) are not included in the on-line averaging section. These are contained in the Output Section (Section IV).

2.3 Completion of Confidence Limits and Trend Computations (Section III)

In Section II the LAB8/E stores the accumulating sum in the core locations reserved for the average, the sums of the squares in the core locations reserved for confidence intervals, and the sums of the squares of successive differences in the core locations reserved for trend. Section III completes the calculations necessary to convert the sums of the squares into confidence intervals (95%), and the sum of the squares of successive differences into the trend function (see Appendix D). The user may proceed from Section II by typing CTRL/P (holding down the CTRL key and typing the P key).

When Section III has been read in, check the accumulator lights on the console to make sure that they are all out all Ø's)[2]. If this is not the case, i.e., if a reading error is indicated, reposition the tape to the beginning of Section III and press the CONTinue key on the console twice.[2] If Section III has been read properly, press CONTinue once to start Section III[2]. To compute standard deviation instead of confidence limits, modify the contents of locations PATCH 1 and PATCH 2 as follows before pressing CONTinue (which will start the section)[1]. After modification, start the program at location 76Ø3.

Location	Is	Change to
PATCH 1 (73Ø4)	45Ø6	741Ø
PATCH 2 (736Ø)	1265	1264

Section III will take between 20 and 50 seconds to complete the confidence interval and trend calculations (with 4K memory) depending on the number of data points involved. The only indication that the computer is calculating will be that the computer console lights will be flashing. When the calculation is complete, an ! and a [(on the next line) will be typed on the Teletype[2]. At this point, the reader

[1]For the Advanced Averager MS refer to section 3.2 for patch instructions.

[2]Not applicable to MS version.

switch on the Teletype should be pushed to START if the low-speed reader is used and Section IV will be read[2]. In the case of the high-speed paper tape reader, Section IV will be read automatically[2].

2.4 Output (Analog X-Y Recorder or Scope Photo; Digital Values in Actual Input Signal Units) (Section IV)[1]

When Section IV has been read in, verify that the accumulator lights are all out (all Ø's), i.e., that there has been no error during read.[2] (If an error is indicated, reposition the tape to the beginning of Section IV and press CONTinue twice)[2]. When the accumulator indicates all zeros, press CONTinue to start up Section IV[2].

As soon as Section IV is started, disconnect the leads to the analog inputs. One to four traces should appear on the oscilloscope. Only one trace will be displayed if N was answered to the following question (during Section I):

CONFIDENCE LIMITS?:__

If Y is answered to this question, but N is answered to the following question:

COMPUTE TREND?___

three waveforms are displayed. At first, it may appear that only one trace is visible even if confidence limits have been requested. If this is the case, turn up intensity on the oscilloscope to increase the persistence of the image and type X (for eXpand), if necessary, to separate the three curves to a greater degree. The Teletype prints a record of the times X has been typed so that C can be typed the same number of times later to restore calibration of the digital typeout. Typing RETURN will also restore the display to zero scale.

If confidence limits were requested during the preparation of the particular control tape being used, the scope intensity was turned up, and X typed as needed, three curves should be clearly visible in the upper part of the screen. These curves may appear to be one curve which jumps up and down between three different positions. These positions represent the average waveform (in the middle), the upper (95%) confidence limit, and the lower confidence limit. The

[1]Refer to Appendix F, How to Access Data, for more detailed information.
[2]Not applicable to MS version.

confidence limits indicate the range within which 95 percent of the data falls. These limits are accurate to +1 percent. (Strictly speaking, the probability is 95% that all the data fall within this range.)

A figure (┘) will appear on the left side of the screen. The horizontal bar represents ten sampling intervals; the vertical bar represents 1ØØ millivolts as seen by the A/D converter.

Two bright spots will appear on each curve. The positions of these spots (cursors) can be controlled by moving analog input knobs Ø and 1. The case of seemingly nonexistent cursors can be solved by a combination of two methods.

1. Turn the intensity down so that the waveform almost disappears. The cursors should be visible if they are on the trace.

2. Type the command I. Data will be printed on the Teletype in the following format:

   # Sweeps Scale Factor Lo Lim Hi Lim XXXX XXXX

   The significance of this will be explained later. Of interest now is Lo Lim and Hi Lim. These are the point positions of the cursors on the waveform. If ØØØØ is typed for a limit, the cursor is off the left-hand end of the trace and the knob must be turned counterclockwise. If both cursors are lost off the right-hand end of the trace, only the number of sweeps and the scale factor are typed.

Table 1-2 shows the different correction factors which the user can apply to the confidence limits calculated by the LAB8/E Advanced Averager Program (which are accurate to ±1%, assuming 40 to 128 sweeps) to obtain the absolute 95 percent confidence limits.

Table 1-2

Correction Factors for Confidence Limits

Number of Sweeps	P
8	1.15
16	1.Ø6
32	1.Ø2
4Ø	1.Ø1
6Ø	1.ØØ
64	1.ØØ
12Ø	Ø.99
128	Ø.99
256	Ø.98
512	Ø.98

True 95 percent Confidence Limits = m±pc

where

m	=	mean
p	=	correction factor
c	=	confidence limits calculated by LAB8/E (see Appendix D).

As indicated above, the LAB8/E goes several steps beyond the variance calculations. In all but cases involving a very small number of sweeps, the confidence limits computed by the LAB8/E can be used in lieu of the true confidence without applying the correction factor P.

If the trend function is requested, one additional waveform appears on the bottom of the screen. This is the trend function which indicates (a) if the successive sweeps which are added together to make up the average waveform are steadily drifting in amplitude, (b) if the successive sweeps are alternately increasing and then decreasing (oscillating), or (c) if there is no measurable consistent change between the sweeps.

Table 1-3 indicates the threshold values of the trend function for several values of N (number of sweeps). The lower trend threshold, t_d, is the value above which one can be 95 percent confident that there was no drift, or below which one can be 95 percent confident that there was consistent change (in one direction) in the amplitude from one sweep to the next, or from one group of sweeps to the next. The upper trend threshold, t_o, is the value above which one can be 95 percent confident that there was a measurable, repeatable increase followed by a decrease in amplitude from one sweep or group of sweeps to the next. The grid calibration routine in Section IV draws lines indicating trend = 2.ØØØ, trend = 2.5ØØ, and trend = 1.5ØØ, values which approximate the significant ranges shown in Table 1-3.

The trend function provides an important means for evaluating the validity of the data represented by the average waveform. An average waveform accompanied by fairly wide confidence limits can indicate any of three radically different conditions which can only be distinguished by means of the trend function. These conditions are:

1. A signal of interest oscillating up and down in amplitude between fairly wide limits.

Table 1-3

Threshold Values of the Trend Function

N	t_d	t_o
8	Ø.98	3.Ø2
16	1.23	2.77
32	1.44	2.56
4Ø	1.49	2.51
6Ø	1.58	2.41
64	1.58	2.41
12Ø	1.62	2.38
128	1.62	2.38
256	1.62	2.38
512	1.62	2.38

2. A signal of interest which has increased or decreased significantly in amplitude during the averaging run.

3. A signal of interest which has not changed at all during the averaging run but which happens to be very small in amplitude compared to the noise in which it is buried.

Trend between t_o and 4.ØØ indicates Condition 1, trend between Ø.Ø1 and t_d indicates Condition 2, and trend between t_d and t_o indicates Condition 3. Condition 1 suggests that the averaging run be broken up into a series of shorter runs to try to establish the period of oscillation. Condition 2 also suggests that the averaging run be broken up into a series of shorter runs to establish the period of time in which the greatest change occurred. Condition 3 suggests that the averaging run be made even longer to narrow the change between the confidence limits and further improve the signal to noise ratio. An average waveform accompanied by a fairly narrow confidence limit and a trend function between t_o and t_d indicates that sufficient sweeps have already been taken.

All of the curves displayed on the oscilloscope at one time in Section IV belong to the same average (AVG #1, AVG #2, or AVG #3, etc.). This is in contrast to the display in Section II in which all the averages (but not confidence limits or trend) were displayed on the oscilloscope at one time. The displays in Section IV are like pages in a book. Pages might be thought of as labeled AVG #1, AVG #2, etc. Each contains its own discrete average and all the waveforms (either 2 or 3) associated with it. To display, plot, or print any average, turn to the correct page and proceed with the desired routine. Pages may be turned forward by typing the RETURN key (a RETURN on the last page will bring up the first page; CTRL/Z will return the display to the first page).

Plotter Calibration Routine

Section IV includes calibration and adjustment routines for the X-Y recorder. Appendix H describes how to connect the plotter to the LAB8/E. To calibrate the X-Y recorder, type K (calibrate). The following message is typed:

BIAS

and the maximum voltage appears at both the X and Y outputs of the oscilloscope connections. The bias controls on the X-Y recorder should now be adjusted until the pen is at the appropriate corner of the plotter field for maximum voltage (usually the upper right-hand corner). Type the RETURN key. The following message is typed:

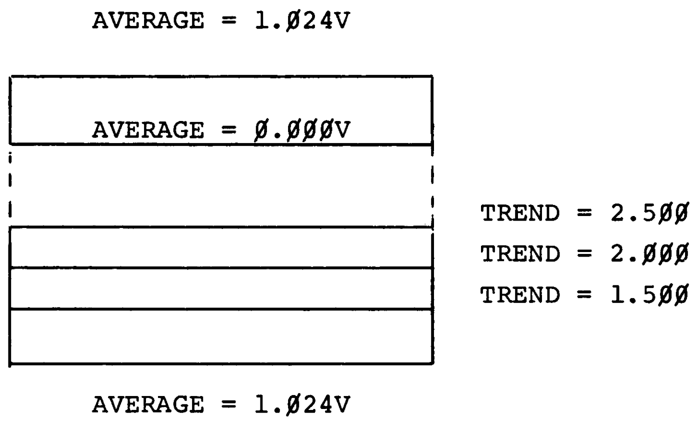
GAIN

Minimum voltage appears at both the X and Y outputs of the oscilloscope. The gain controls on the X-Y recorder should now be adjusted so that the pen is either at the lower left-hand corner of the printed section of X-Y recorder paper (or at some multiple of ten chart divisions from the right-hand edge in the X direction and some multiple of eight chart divisions from the upper border in the Y direction).

Because adjusting the gain controls of most X-Y recorders affects the bias setting (and vice versa), it will usually be necessary to type RETURN several more times (repeating the "BIAS" and "GAIN" adjustment cycle) until the cycle can be repeated with the pen alternately at the upper right-hand corner and then at the lower left position without readjustment of the controls. To EXIT from the calibrate routine type CTRL/Q.

Drawing a Grid on the X-Y Chart Paper

In order to draw in the borders on the X-Y chart paper indicating full scale positive, Ø, and full scale for negative for the average and confidence limits as well as lines indicating trend = 2.5ØØ, trend = 2.ØØØ and trend = 1.5ØØ (see diagram below), type the command G on the Teletype.



The following message is typed:

LIFT

Turn the X-Y recorder on and type the RETURN key on the Teletype. The following message is printed:

LOWER

Follow this instruction by lowering the pen against the X-Y chart paper, turning analog knob 3 fully counterclockwise, and typing the RETURN key. This causes the grid to be plotted. The speed of the plot can be made slower by turning knob 3 clockwise. When the grid is completed, the following message is printed:

LIFT

Lift the pen away from the chart paper before any other character is typed on the Teletype, turn off the X-Y recorder, and type CTRL/Q to resume the display of the average.

NOTE

If the X-Y recorder is left on while waveforms are being displayed on the scope, the X-Y recorder pen will shake furiously as it attempts to move at the very rapid rate of the oscilloscope display. If the pen is against the chart paper at this time, the chart will be ruined.

Plotting the Average, Confidence Limits and Trend

The plotting of curves can be started by typing P (for plot) on the Teletype. This stops the oscilloscope display and causes the following message to be printed:

LIFT

Turn on the plotter and lift the pen away from the chart paper. Type the RETURN key, and the following message is printed:

LOWER

Disregard this message the first time. Keep the pen raised away from the paper and type the RETURN key. The pen begins tracing the average waveform except that the pen is not touching the paper. Quickly adjust analog knob 3 for the fastest slew rate possible without running the risk of rounding off the steeply rising or falling parts of the waveform. When this waveform is completely traced, the following message is printed:

LIFT

(If only the average was computed, OFF is printed.) If the slew rate is adjusted for the X-Y recorder in use, type CTRL/Q followed by a P command. Answer LIFT by typing the RETURN key. The following is printed:

LOWER

If analog knob 3 is set for the proper slew rate, lower the pen against the paper and type the RETURN key. The average is traced on the paper.

After this waveform is completed, the following is printed:

LIFT

If any further adjustment of the slew rate is desired, knob 3 can be adjusted at this time (clockwise to make the pen move slower; counterclockwise to make it move faster). Type RETURN and the following is printed:

LOWER

Lower the pen against the paper and type the RETURN key to begin plotting the upper confidence limits on the average. Repeat the process for the lower confidence limits and trend waveform (if specified).

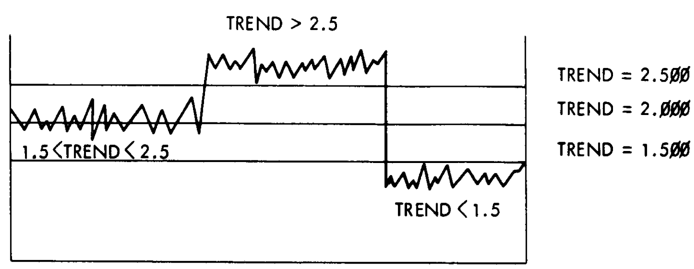
When the last waveform specified in this average is written, the following is printed:

OFF

Lift the pen and turn off the X-Y recorder. Type the RETURN key to start the display.

Interpreting the Result of the X-Y Recorder Trend Readout

The following is an example of a typical trend readout at the bottom of the X-Y recorder chart paper (it is assumed that there were more than 4Ø sweeps in the average).



In this example (with 95% confidence) the signal remained constant during the first one-third of each sweep. The trend = 3 in the second one-third of the sweep indicates (with 95% confidence) an oscillating condition in this part of the waveform. The signal was alternately higher and then lower in amplitude from one sweep or group of sweeps to the next sweep or group of sweeps in this section of the waveform. The last section of the trend function, in this example, indicates (with 95% certainty) a steady drift in amplitude from one sweep or group of sweeps to the next sweep or group of sweeps. Trends of 4 and Ø instead of approximately 3 and 1 as in this example, would have indicated a greater certainty of oscillation, and a greater certainty of drift, respectively.

Digital Typeout and Punched Paper Tape Output on ASR33 Teletype

To begin the typeout of the data corresponding to the average, confidence limits, and trend for the average being displayed, type the command T on the Teletype. For punched paper tape output, merely push the button marked ON on the control box on the left-hand side of the Teletype before typing T. The average confidence limits and trend for the data points between and including the two cursors are printed. The same data printed is also punched on paper tape.

The format of the data printed on the teleprinter is similar to the following:

ØØ61,	ØØØØ		
ØØ1Ø,	-Ø344,	ØØ22,	2Ø75,
ØØ11,	-Ø338,	ØØ29,	1Ø93,
ØØ12,	-Ø327,	ØØ23,	2311,
	etc....	....	....

The first number in the upper left-hand corner is n, the number of sweeps included in this average. The second number on the same line is k, the scale factor (expressed as a power of 2). In order for the following numbers in the left-hand column to be in actual calibrated input signal units, the scale factor must be Ø. The scale factor may be decreased or increased by typing the commands C or X respectively.

The numbers in the four columns beginning at the second line are the data point number, the mean, the confidence interval, and 1ØØØ times the trend, from left to right respectively.

The confidence limits are the mean ± the scaled confidence interval. In the example:

-Ø344 + ØØ22 = -Ø322 = Upper 95 Percent Confidence Limit

-Ø344 - ØØ22 = -Ø366 = Lower 95 Percent Confidemce Limit

Confidence limits for each of the other average points can be calculated in similar fashion.

For its interpretation, trend merely requires a decimal point between the first and the next three digits in each four-digit group in the third column. In the example, trend for the first data point is 2.Ø75, for the second data point is 1.Ø93, and for the third is 2.311, etc.

If trend was not computed for this particular average, there will be only three columns in the printout. If confidence limits were not computed, there will be only two columns (the data points making up the average).

NOTE

1. Only the number of sweeps and the scale factor will be typed out if both cursors are off the right-hand end of the trace.

2. If the input is constant and there is no variance, the computation of trend will involve division by zero. This returns a zero answer. If this happens, the trend value is set to 2, which indicates no trend has taken place.

3. Commas are provided to make any output tape compatible with BASIC/RT (refer to Book Seven).

When the last requested point in this particular average has been typed, the display of this average is resumed. To discontinue digital readout before all the data points are read, type the CTRL/Q command, which also resumes the display of this average.

Finding the Area Under the Curve

Type I to print the sum of the data points between the two cursors. The format of the printout is:

```
#sweeps, scale factor, lo lim, hi lim, sum avg's, sum conf. lims.,
```

The high and low limits identify the bounds on the data points included in the integration. The sum of the averages and sum of confidence limits are suffixed with ↑ if overflow occurs. To get the true summation (supposing overflow occurred) scale the average down by typing the command C and take the new scale factor into account when noting the result is within scale. To get the true integral, multiply the summation by the clock rate. The sum of the confidence limits is not reported when none has been computed.

To find the latency of a particular part of the waveform displayed from the beginning of the waveform, position the left-hand cursor on the point in question and type the I command. The third number printed is the number of the point covered by the cursor. To find the latency of that point, multiply the point number by the sampling rate.

It should be noted that there is a certain amount of error inherent in the sum of the averages printed by the I command. The sum of the averages is done as single precision (12 bits) add on scaled numbers. If the sum exceeds plus or minus 2048 during the summing the up arrow is printed, indicating an overflow. The sum is obtained as follows. The double precision data point is first scaled by the scale factor then added to the single precision sum. This type of adding creates the sum's error. As a result the difference between two identical sums of N numbers taken at a scale factor differing by 1 can be as much as N. For example if the first sum of N numbers is at a scale of -3 and the resultant scaled values are all even positive values, then the resultant N numbers at a scale of -4 will be exactly 1/2 the number at the scale of -3. The sum at a scale of -4 will be exactly 1/2 the sum at a scale of -3, and no error will be seen between the two sums. If the N numbers at a scale of -3 are all positive odd numbers then double the sum of the N numbers at a scale of -4 will differ from the sum at a scale of -3 by N. Hence, if the result of the sum is small in comparison to the number of values summed, the comparison of two sums taken with different scale factors can be relatively large. For example if the values are both positive and negative, the sum can be small in comparison to the number of values summed. If 200 points were summed, the sum on identical data could be 1ØØ for the scale of -3 and Ø for a scale of -4. Hence it is important to keep the sums as large as possible (small scale factor) but this is not always possible. When positive and negative values are added, the resultant sum can be small no matter what is done.

For the above reasons, comparisons between I values of different averages should be made at the same scale value. Two averages can be almost identical and one will overflow at a scale of -3 for example and the other one will not overflow. Hence it is important to assure that all values being compared be calculated at a scale that will not overflow following averages.

In Section IV of the averager the cursors may shift between two points. This is due to a one bit jitter in reading the pots. It will not affect the T or I commands since the limits are output. If a large jitter is seen, check if all inputs are disconnected, and when only the sync is in, the sync input may be miswired, causing a ground loop.

Proceeding to the Next Display of an Average

Type the RETURN key to proceed from one display of an average and associated statistics to the next. Type CTRL/Z to return to the first average. If all averages computed have been displayed, the display returns to the first average. To begin computing new averages, remount either a control tape followed by DEC-8E-AAP2A-A-PB or DEC-8E-AAP1A-A-PB and type CTRL/P.

Table 1-4 summarizes the Teletype keyboard commands during Section IV operation.

Table 1-4

Section IV Commands

Command	Explanation
C	Contract display; divide data to be printed, punched or written by a factor of 2.
X	Expand display; multiply data to be printed, punched or written by a factor of 2.
K	Calibrate ("Gain...", "Bias...", etc.) analog plotter.
G	Plot grid on analog plotter (average and confidence limits = Ø + 1.Ø24V, -1.Ø24V; trend = 2.ØØØ, 2.5ØØ, and 1.5ØØ).
P	Plot data on analog plotter.
	NOTE For commands G and P above, analog knob 3 should be turned fully counterclockwise then clockwise a bit before executing these commands.
T	Type (and punch, if Teletype punch is on) on teleprinter the digital data between cursors.
I	Sum of the data between the cursors.
RETURN	Proceed to the next step or next average; return display to zero scale.
CTRL/Q	Quit current operation, return to display of current average.
CTRL/Z	Return to the display of the first average.
CTRL/P	Read in next section of tape.

CHAPTER 3

ADVANCED AVERAGER MS

3.0. GENERAL DESCRIPTION

The Advanced Averager MS (Mass Storage) for OS/8 will write data on mass storage and may operate from control tapes stored on mass storage. Otherwise, the operation is the same as paper tape system. The binaries of the Advanced Averager MS come in five sections[1]:

	Papertape	DECtape (DEC-8E-ALMSA-A-UB)
Section 1	DEC-8E-AAA1A-A-PB	AAVG1.BN
Section 2	DEC-8E-AAA2A-A-PB	AAVG2.BN
Section 3	DEC-8E-AAA3A-A-PB	AAVG3.BN
Section 4	DEC-8E-AAA4A-A-PB	AAVG4.BN
Section 5	DEC-8E-AAA5A-A-PB	AAVG5.BN

All five sections are "CHAINED" together under OS/8 so that each section calls the next from the system device on command or automatically.

3.1 Using Advanced Averager MS

Loading the Programs

The programs are distributed on paper tape or DECtape in binary format. If desired, the files can be PIPed (with /B) onto the system. (Note that the files are binary and a .BN extension must be assigned to the files.) The binaries must be saved on the OS/8 system device as follows:

Load each of the overlays in turn via the ABS Loader and save it with the SAVE command. The SAVE commands for the overlays are:

Section 2	-	SAVE SYS	AAVG2.SV	64ØØ-7577;Ø6527=Ø
Section 3	-	SAVE SYS	AAVG3.SV	64ØØ-7577;Ø6777=Ø
Section 4	-	SAVE SYS	AAVG4.SV	Ø-177,64ØØ-7577;ØØØ17=Ø
Section 5	-	SAVE SYS	AAVG5.SV	64ØØ-7177;Ø6377=Ø

Section 1 of the Advanced Averager has two starting locations 4ØØ and 4Ø3. Start at 4ØØ to do parameter initialization of Section 1; start at 4Ø3 when a control tape is being used to run Section 2.

[1]Sections 1, 2, 3 and 4 are the same as I, II, III and IV of the Advanced Averager.

Section 1 can be run after being loaded with ABS Loader or after being SAVEd. If Section 1 is to be SAVEd, use the SAVE command:

To start at 4ØØ,

```
SAVE SYS    AAVG1.SV  Ø-7577,12ØØØ-13177;ØØ4ØØ=Ø
```

or to start at 4Ø3,

```
SAVE SYS    AAVG1.SV  Ø-7577,12ØØØ-13177;ØØ4Ø3=Ø
```

Then to start the program respond to the dot with

```
R  AAVG1
```

STARTING LOCATIONS:

Section 1	4ØØ 4Ø3	to run Section 1 to use control tapes
Sections 2, 3, 4	have the same starting addresses as the paper tape system.	
Section 5	64ØØ.	

NOTE

> When a SAVEd module is being chained under OS/8, the starting address in the SAVE command is its true starting address -1.

CONTROL FILES:

Control files replace control tapes of the paper tape system. When a control file is desired, YES is answered to the question "PUNCH CONTROL TAPE". The program prints an asterisk to which the user gives the output file name in OS/8 format:

*dev:FILE.5<

The file is written to the device specified as file FILE.5 and control is given to OS/8. When producing a control file or running the Averager from Section 1, the starting address is 4ØØ.

To run the Averager from a control file, Section 1 of the Averager is started at 4Ø3. The program then asks for the input control file by typing an asterisk. The file is given in OS/8 format:

* dev:FILE.5

This would load FILE.5 from the device specified. The TRIGGER section would then start.

3.2 New Commands

The following paragraphs explain the features of Advanced Averager MS which differ from the Advanced Averager.

Section 1

1. Typing CTRL/C returns control to OS/8.
2. Typing the RETURN key in the TRIGGER section loads Section 2.
3. "PUNCH A CONTROL TAPE" refers to a control file, answer Y for YES, N for NO.

   If a control file is to be output to mass storage, an asterisk is printed. In response, the OS/8 file definition is given as shown above.

4. If the program is started at 403, an asterisk is printed and the control file is given as shown above.

Section 2

1. CTRL/C returns control to OS/8 and replaces the CTRL/U, user command.

2. CTRL/P loads Section 3.

   If a user command is desired, the CTRL/C can be replaced or adjusted as desired.

Section 3

Section 3 runs by itself after being loaded. When it is completed, Section 4 is loaded.

The following procedure must be used to patch the program to compute standard deviation instead of confidence limits. When section 3 (AAVG3.SV) is to be saved, load AAVG3.BN via the Absolute Loader. Call ODT via the OD command to OS/8 and change the two locations as shown.

Location	IS	Change to
PATCH 1 (7304)	4506	7410
PATCH 2 (7360)	1265	1264

SAVE this module as explained in section 3.1.

Section 4

CTRL/C returns the user to OS/8.

CTRL/W replaces CTRL/P. CTRL/W loads Section 5 which outputs the data collected to a mass storage device.

Section 5

An asterisk is printed requesting the output file and device. For example,

*dev:FILE.1<

outputs the averaged data on the device specified in file FILE.1.

3.3 Errors

Most errors are I/O errors, assuming the five sections were saved correctly. For more information on errors, refer to the Software Support Manual for OS/8.

Section 1

If an error is detected in Section 1 while linking the chains of the program, the Program prints USER ERROR 1, and returns to OS/8. This will happen if Sections 2-5 were not SAVEd correctly.
An error when reading or writing a control file returns the program to OS/8. This can be caused by giving an output file when an input file is requested or an input file when an output file is requested. In addition, if the device does not exist or the file is not found, an error occurs.

Note that if a file exists with the same file name given the new file, the old file is deleted.

Sections 2, 3, 4

If the programs were not saved correctly, the program crashes. This is true for Sections 1 and 5 also.

Section 5

The following errors can occur:

USER ERROR 1 - the output device is not found.

USER ERROR 2 - an output file cannot be created.

USER ERROR 3 - a write error occurs while writing the output data.

USER ERROR 4 - the file cannot be closed or the device is full.

Other errors result in Monitor Error messages. (Refer to Chapter 9 (OS/8) of Introduction to Programming.)

3.4 Restart Program

To run the Averager again, type CTRL/C then R AAVG1 when the dot is printed.

3.5 Data Format of Output File

The output file created by the CTRL/W command is in OS/8 format as follows:

```
BLOCK 1   WORD 1 = 1
          WORD 2 = Ø  if no more Job list blocks
                 = 1  if the next block is a Job list block.
          WORD 3 - 256 = Job lists from Advanced Averager.
                         A Ø first word of the Job list is
                         the end of the Job lists.
BLOCK 2   If WORD 2 of previous block = 1, then the format is
          the same as for Block 1.
          If WORD 2 of previous block = Ø, this block starts
          data.

1st DATA BLOCK  WORD 1 = number of sweeps
                WORD 2 and on = data as stored in core of
                       Advanced Averager.
```

3.6 Types of Data Blocks on Mass Storage

1. START OF ADVANCED AVERAGER BLOCK DATA

   WORD 1 = number of sweeps
   WORD 2 and on = data as stored in core

2. CONTINUATION OF DATA INTO NEXT MASS STORAGE BLOCK

WORD 1 and on = data continues from previous block as stored in core.

3. LINKED ADVANCED AVERAGER DATA BLOCK

WORD 1 and on = data starts with the average value of a data point as stored in core. The first data of a linked data block in core always starts a new output block.

Data blocks or linked data blocks as written by the Advanced Averager are written on mass storage starting with a new block, in a contiguous fashion. The data is ended with the length of the next linked data block or a Ø.

ADVANCED AVG. CORE (as it exists in Section IV)

```
     Loc. 23Ø  Job List (7 words)  for AVERAGE 1
          237  Job List (7 words)  for AVERAGE 2
          246  Job List (7 words)  for AVERAGE 3

DATA START 1   # of Sweeps of Average 1
               DATA of average 1 in one field, 100 locations
                 :
                 Ø  end of data

DATA START 2   # of sweeps of average 2
               Data of average 2 in one field, 300 locations
                 :
                 Ø  end of data

DATA START 3   # of sweeps of Average 3
               Data of average 3 in multiple fields, 100 locations
                 :
               -# of points in linked block
               CDF
               Start of data -1

DATA CONTINUATION 3  Data of average 3 continued, 300 locations
                       :
                       Ø
```

Data as written to output device by CTRL/W command:

---

[1]Refer to Appendix I, for more information on data block structure.

Block 1

```
Word 1 = 1
     2 = Ø
     3 = Job list of average 1
    10 = Job list of average 2
    17 = Job list of average 3
```

Block 2

```
Word 1   # of sweeps of average 1
     2   Data
         :
```

Block 3

```
Word 1   # of sweeps of average 2
     2   Data
         :
   256   :
```

Block 4

```
Word 1     continuation of data
           :
     46    Ø
```

Block 5

```
Word 1     # of sweeps of average 3
     2     Data
           :
     102   -# of points in linked block = -300
```

Block 6

```
Word 1     continuation of data
           :
     256
```

Block 7

```
Word 1     continuation of data
           :
     45    Ø
```

BOOK TWO

BASIC AVERAGER

CONTENTS

BOOK TWO

FIGURES

CHAPTER 1

BASIC AVERAGER

1.0 INTRODUCTION

The LAB8/E Basic Averager Program digitizes, displays, and averages analog signals at rates from 33[1] μsec per point per channel. The program is entirely core resident and its output is compatible with the statistics calculations and output section (3 and 4) of the LAB8/E Advanced Averaging program (DEC-8E-AAP1A-A-PB). The user can select and adjust, on-line, the sampling rate, the number of sweeps, and the delay via the Teletype. By using control tapes (LB-UØ1B-PB through LB-UØ8B-PB), the user shapes LAB8/E to his particular needs and is able to perform sophisticated statistical manipulation on data. Refer to Chapter 4 for the OS/8 version of the Basic Averager.

1.1 Program Tapes

The Basic Averager program consists of one large paper tape and a set of eight small tapes. The large tape, DEC-8E-ACAØA-A-PB, consists of three programs separated from each other by leader/trailer code. This code signals the loader program that a program has been read and that reading should stop until the next program is required. The first program of the three is the Basic Averager. The other two programs are taken from Sections III and IV of the Advanced Averager program. These two programs are optional and are provided as a convenience to the user.

The eight short tapes are called "control tapes". One of these must be used when the Basic Averager is run. See Appendix J for details of control tapes.

1.2 Basic System Requirements

PDP8/E computer with 4K of core

AD8-ES single input A/D converter

or

AD8-E + AM8-EA + AM8-EC A/D converter with 8 channel multiplexer and preamps

DK8-ES real time programmable clock

[1]33 microseconds is the maximum rate for multiple channels. However, a single channel can be run at a maximum rate of 25 microseconds.

VC8-E 10 bit display controller

H945AA mounting panel and power supply

VR14 or Tektronix 602,603 604 or RM503 } display oscilloscope

ASR33 Teletype

High speed reader is used when available.

The OS/8 version requires the standard OS/8 configuration.

CHAPTER 2

RUNNING THE BASIC AVERAGER

2.1 Connecting Leads

The leads from the experiment to the LAB8/E should be connected to the proper inputs to assure proper functioning. The sync pulse should always be connected to S1. If only one signal is being averaged, analog input Ø is the only one used. If the eight channel multiplexer is implemented and more than one channel is being averaged, the additional channels should be connected to analog inputs 1, 2, etc. Consecutive inputs starting with input Ø must be used.

2.2 System Initialization and Program Loading

For the LAB8/E programs to run properly, the PDP-8/E must have its knob and switches set in certain positions. Before running any programs, check to see that the knob and switches are in their correct positions.

Turn the system on by turning the key on the lefthand side of the PDP8/E console to POWER. Do not turn as far as PANEL LOCK. Next, ensure that the HALT and SING STEP keys are in the up position. Turn the status knob in the right hand center of the console to AC. This step enables the contents of the accumulator to be seen in the lower of the two rows of lights. Turn the Teletype knob to LINE, and finally turn the scope on. The LAB8/E system is now ready to receive programs.

All of the LAB8/E programs are loaded with the Binary Loader (BIN) program. If BIN is not in core it must be loaded with the RIM Loader. See Appendix A for details of loading BIN and RIM.

Select a control tape from the following chart:

TAPE	# CHANNELS	STATISTICS	DATA POINTS
LB-UØ1A-PB	1	Average	1000
LB-UØ2A-PB	2	Average	500
LB-UØ3A-PB	4	Average	250

TAPE	# CHANNELS	STATISTICS	DATA POINTS
LB-UØ4A-PB	1	Average & Confidence Limits	500
LB-UØ5A-PB	2	Average & Confidence Limits	250
LB-UØ6A-PB	4	Average & Confidence Limits	125
LB-UØ7A-PB	1	Average, Confidence Limits & Trend	300
LB-UØ8A-PB	2	Average, Confidence Limits & Trend	150

NOTE

Only the first control tape can be used if the system is equipped with a single A/D input.

To modify any of the control tapes, the user must obtain the ASCII version of the tape and change it, using the Symbolic Tape Editor (DEC-Ø8-ESAB-PB). When the desired changes are made, the tape must be assembled using the PAL III Symbolic Assembler (DEC-Ø8-ASAC-PB) before it can be used as a control tape with the Basic Averager. Refer to Appendix J for detailed instructions.

2.3 Loading the Basic Averager

To run the Basic Averager both the Averager program itself and a control tape must be read. Using the BIN Loader, load the control tape first:

a. Place a control tape in the tape reader so that the leader/trailer code is over the read mechanism.

b. Turn the Teletype to LINE.

c. Turn the scope to ON.

d. Set the Switch Register (SR) to 7777_8 (all switches in the up position).

e. Press the ADDR LOAD key.

f. If using a low-speed reader, set the reader switch to START.

g. If using a high-speed reader, set the SR to 3777_8 and turn on the high-speed reader.

h. Press the CLEAR and CONT switches on the computer console.

i. The tape should move through the reader. If it does not, BIN is not present and must be loaded as described in Appendix A.

j. When the tape stops, verify that the accumulator lights are all off (to see the accumulator, turn the status knob to AC). If they are not all off, a read error has occurred and steps a. through j. should be repeated.

k. When the control tape has been correctly loaded, follow the above steps to load the Basic Averager, DEC-8E-ACAØA-A-PB.

l. When loading is complete, press CONT.

NOTE

When the LAB8/E Basic Averager is loaded, it modifies the Binary Loader to allow a program start by simply pressing CONTinue. This feature requires the user to start the Binary Loader at 7777_8 each time a tape is loaded.

See Figure 2-1 for a flow diagram of the operating procedure. The LAB8/E responds with

R:

Type the sampling rate to be used[1]. The number typed is considered by the program to be in units of microseconds. A number between 33[2] and 4096 should be typed, followed by a RETURN key. This sets the time between samples. If a slower sampling rate is required, the Advanced Averager should be used. The program types

N:

In response to N:, type the number of sweeps desired in the average followed by the RETURN key. If an error has been made in typing, type the CTRL/Q command and the question will be asked again. N=Ø is treated as 4096.

The program will now print

D:

Answer with the delay in terms of the clock rate. That is, if the sampling rate is 5Ø μsec and it is desired to delay 5ØØ μsec after the sync pulse before sampling begins, type 1Ø, followed by the RETURN key. Again, if a typing error is made, type CTRL/Q and the number of sweeps and delay can be respecified.

NOTE

When a sync pulse is sensed via the Schmitt trigger, the program waits one sample interval before the first A/D sample is taken. This timing wait is in addition to the delay specified in answer to D:.

It is possible to alter these parameters at any time during the experimental run by typing CTRL/A.

---

[1] The rate specified is the total rate on all channels, not the rate for each channel. For example, if the same rate is specified for Control Tape 1 and Control Tape 3, the sampling rate on channel Ø is four times faster when Control Tape 1 is used.

[2] 33 microseconds is the maximum rate for multiple channels. However, a single channel can be run at a maximum rate of 25 microseconds.

The program now begins showing the input signal on the CRT display. A colon and two LINE FEEDs are typed and the input signal appears on the oscilloscope. Adjust the threshold for the Schmitt trigger until a partial display is shown. The partial display assures that the trigger is firing. To begin averaging, type the RETURN key. The summation will then be shown as it accumulates.

The accumulated sum may be cleared at any time by typing CTRL/Z.

Type V at any time to switch from viewing the input to viewing the summation of the sweeps or vice versa. Type X to expand the viewing scale. Type C to contract the viewing scale of the summation. The input data cannot be scaled.

Type LINE FEED to pause averaging at any point, and then type the RETURN key to resume. Typing LINE FEED always forces the display to the view inputs mode. The sampling continues until the number of sweeps requested in the parameter setup have been taken. The average may be prematurely terminated by typing CTRL/Q.

If the average has been completed, it can be plotted by typing the command P to enter plot mode, turning on the X-Y plotter, and then typing the RETURN key to move the pen to the beginning of the plot. Lower the pen and type the RETURN key. Lift the pen and type the RETURN key again. The pen moves to the beginning of the plot. Lower the pen and type the RETURN key again to replot the data or plot the next of a series of curves. Terminate plot mode by typing P and turning off the plotter. Type the RETURN key. The display will again become active.

A completed average can be output on the Teletype by typing the command T. The message

```
T:
```

is typed. If it is desired to dump all of the accumulated averages, answer this with a Ø or the RETURN key. If it is desired to type selectively, one entire channel or selective portions of a channel, type 1, 2, 3, or 4 plus the RETURN key to type out the first, second, third, or fourth channel in the display. The message

```
L:
```

is typed. Answer this with the low limit in terms of the data point number in which you are interested and the RETURN key. The message

H:

is typed. Answer this by typing the high limit data point to be output and the RETURN key.

The first two numbers typed out represent the number of sweeps included in the average and the scale factor, respectively. The scale factor is the power of two by which the data has been scaled. If the number of sweeps is a power of two, then, by typing C and X while the display is active, the gain can be adjusted until the scale factor is appropriate for the number of sweeps. That is, 2 raised to the scale factor power is equal to the number of sweeps. The typeout of data points will then be in millivolts as seen at the input to the A/D Converter.

Following the typeout of the number of sweeps and the scale factor, the value of each point in the average is typed out in millivolts.

NOTE

If a single input A/D is used (input voltage ±5V), the point values should be multiplied by a factor of 5 to yield the true value.

To terminate typeout at any time, simply type CTRL/Q.

If another set of sweeps is to be added to the average already accumulated after a group of sweeps has been completed, type CTRL/R to retain the data and reinitialize for more averaging.

To begin averaging after reinitialization has taken place, type the RETURN key.

If it is desired to start averaging with a zero buffer, type CTRL/Z and the RETURN key.

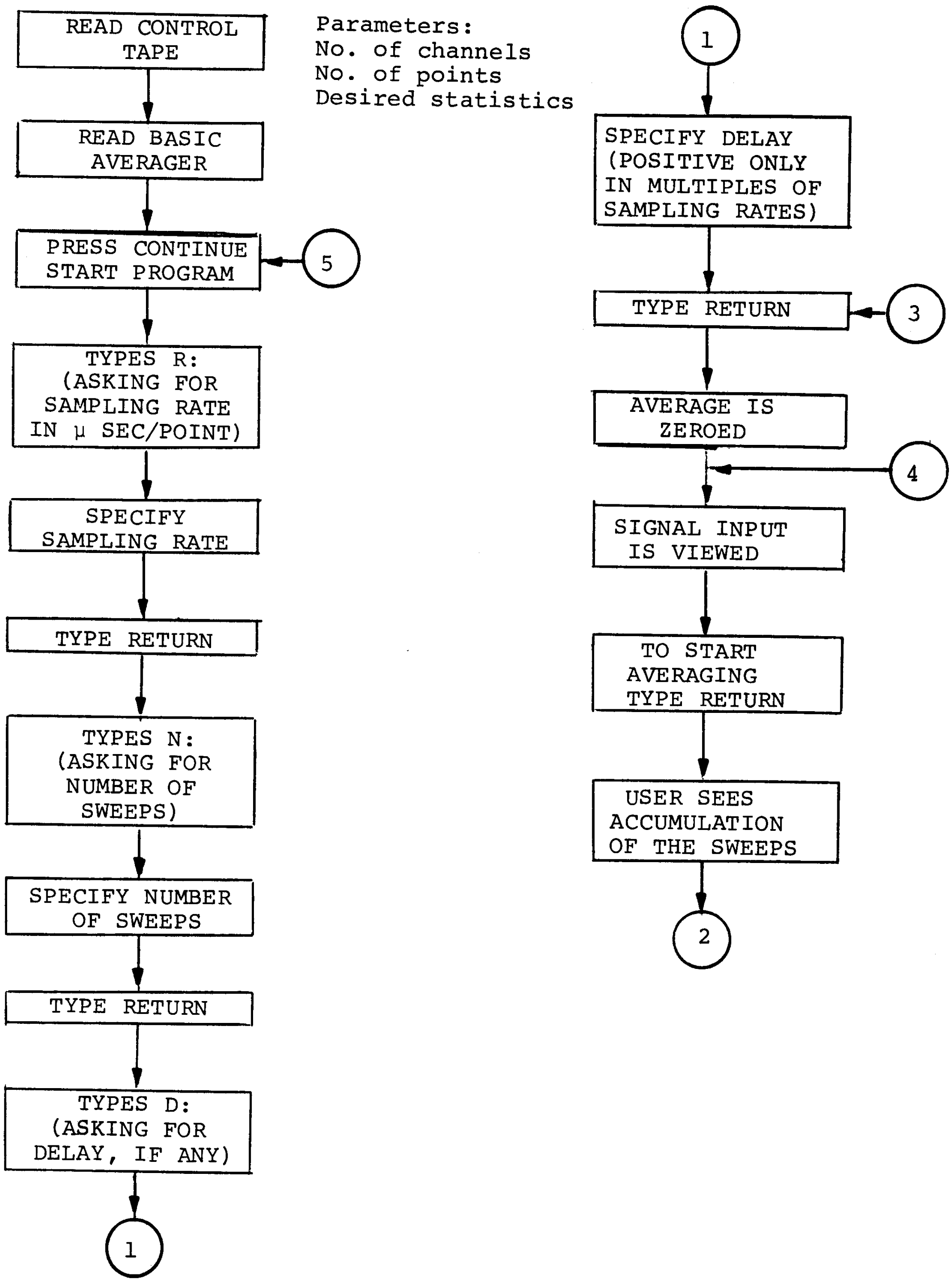


Figure 2-1 Basic Averager: User Flow Chart

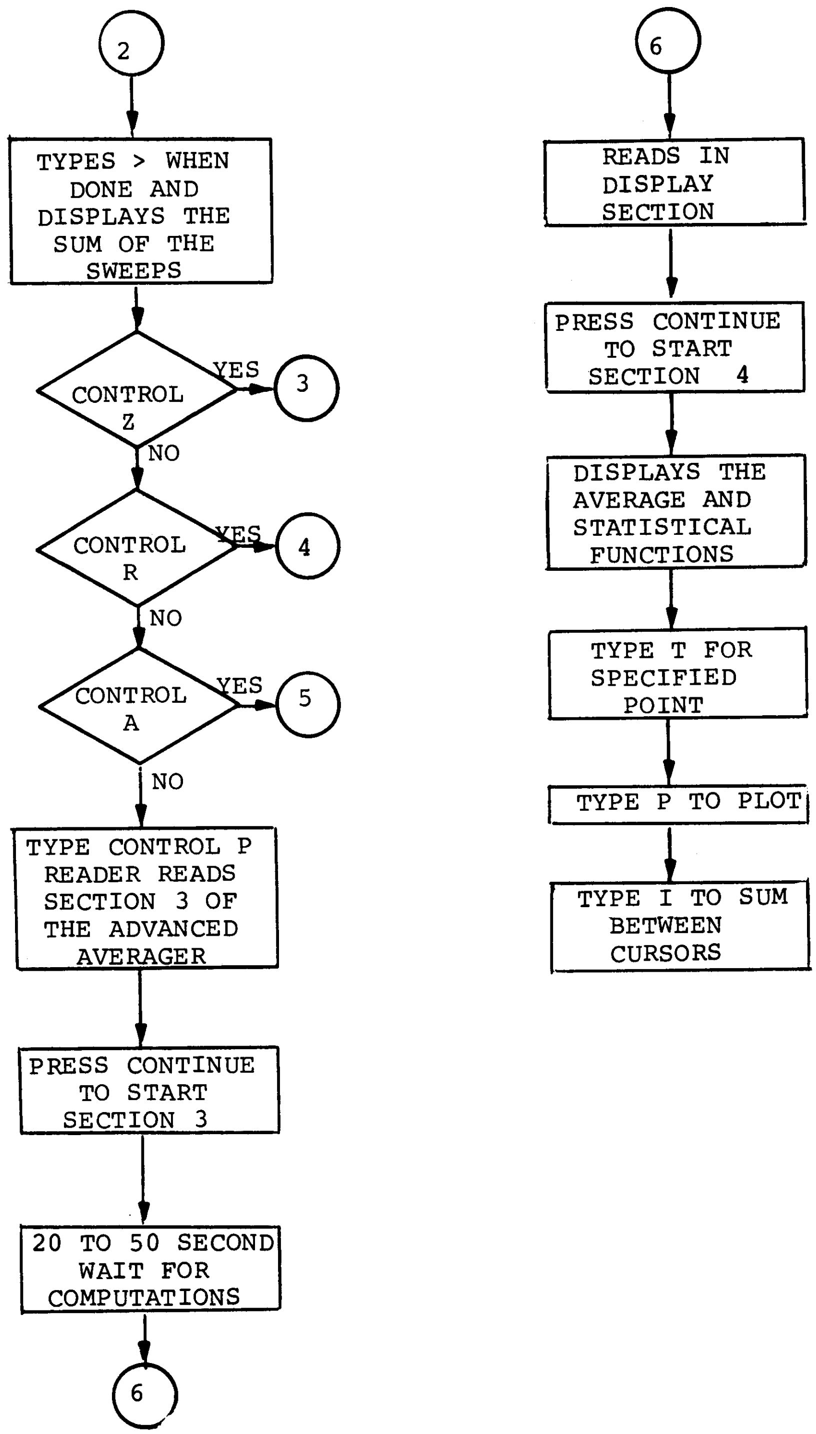


Figure 2-1 Basic Averager: User Flow Chart (Cont)

CHAPTER 3

COMPUTATIONS OF STATISTICS AND OUTPUT

3.0 COMPLETION OF CONFIDENCE LIMITS AND TREND COMPUTATION

Although not strictly a part of the Basic Averager system, statistical analysis of data can be accomplished, if the eight-channel multiplexer is part of the system. If the eight channel multiplexer is not implemented, the user is urged not to proceed, but he may do so at his own risk. Typing the CTRL/P command advances the user to two statistical sections following the Basic Averager on the paper tape provided.

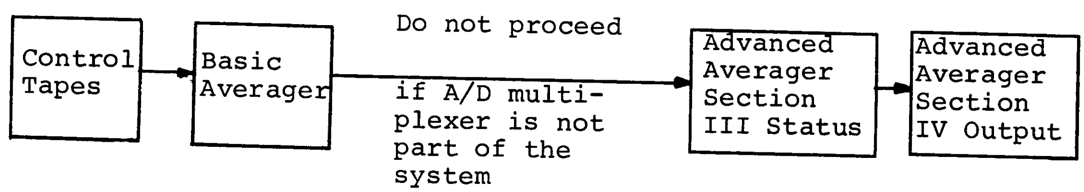


Figure 2-2 Block Diagram of the Basic Averager

After the first statistical section (Section III) has completed loading and the accumulator equals zero, press CONTinue. Computations will then be done on the raw statistics calculated while the averager was in operation (provided control tapes requesting confidence limits and trend were used).

If the accumulator does not equal zero after reading, an error has occurred. Reposition tape to the beginning of Section III and press the CONTinue key on the console twice. The rest of the Basic Averager is the same as the Advanced Averager (Book One, Section 2.3).

On output (Section IV) the number of traces (1-4) which appear on the oscilloscope depends on the control tape used.

To begin computing new averages, remount a control tape, type the CTRL/P command, and load the Basic Averager, DEC-8E-ACAØA-A-PB.

CHAPTER 4

BASIC AVERAGER MS

4.0 GENERAL DESCRIPTION

Basic Averager MS is the same as the papertape version except that it will chain in Sections 3, 4 and 5 of the Advanced Averager. When the Basic Averager is done and in output mode, Section 3 of the Advanced Averager can be loaded by typing a CTRL/P. The binary tape of the Basic Averager is DEC-8E-ABAØA-A-PB and the binaries of the control tape are

Control tape	1	DEC-8E-ABA1A-A-PB
	2	DEC-8E-ABA2A-A-PB
	3	DEC-8E-ABA3A-A-PB
	4	DEC-8E-ABA4A-A-PB
	5	DEC-8E-ABA5A-A-PB
	6	DEC-8E-ABA6A-A-PB
	7	DEC-8E-ABA7A-A-PB
	8	DEC-8E-ABA8A-A-PB

4.1 Using the Basic Averager MS

<u>Loading the Programs</u>

NOTE

Mass Storage output from BASIC Averager is done by Section 5 of the Advanced Averager. If no MS output is desired, the BASIC Averager can be run without Sections 3, 4, and 5 of the Advanced Averager.

If Sections 3, 4 and 5 of the Advanced Averager are to be used to output data as in the Advanced Averager, the following programs must be saved as SAVE files under OS/8. Sections 3, 4 and 5 of the Advanced Averager must be saved after being loaded with the Absolute Loader. (Refer to Book One.)

```
Section 3 - SAVE SYS AAVG3.SV 64ØØ-7577;Ø6777=Ø
Section 4 - SAVE SYS AAVG4.SV Ø-177,64ØØ-7577;ØØØ17=Ø
Section 5 - SAVE SYS AAVG5.SV 64ØØ-7177;Ø6377=Ø
```

The Basic Averager can be loaded with the Absolute Loader in the order

*CONT.BN, BAD2.BN

where CONT.BN is the control tape and BAD2.BN is the Basic Averager. As with the paper tape system the control tape binary must be loaded then the binary of the Basic Averager. The binaries of the control tapes and the Basic Averager are stored on the user's OS/8 device. Any one of the eight control tapes can be used.

STARTING LOCATIONS:

If Sections 3, 4 and 5 of the Advanced Averager are to be used, start at 400, field Ø.

If they are not going to be used, the program can be started at 6505. It still can be started at 400 if the saved Sections 3, 4 and 5 of the Advanced Averager are present on the system device. A control tape and the Basic Averager can be saved after being loaded, if desired.

The save locations are:

BASIC Averager	Ø-577,	62ØØ-7577		
Control Tape 1	6134			
Control Tape 2	4175,	6147		
Control Tape 3	3233,	4221,	52Ø7,	6175
Control Tape 4	6134			
Control Tape 5	36Ø3,	6147		
Control Tape 6	2444,	3627,	5Ø12,	6175
Control Tape 7	6134			
Control Tape 8	3437,	6147		

Running the MS Basic Averager

Operation is the same as the paper tape system except the programs come off mass storage.

CTRL/P in output mode will chain in Section 3 of the Advanced Averager. From there on the Advanced Averager commands are active.

For example, to save Control Tape 2 the command would be:

SAVE SYS BASAV2.SV Ø-577,62ØØ-7577,4175,6147;4ØØ=Ø

Errors:

The BASIC Averager has only one error message:

USER ERROR 1

which indicates a chaining problem.

BOOK THREE

TIH, PST AND LATENCY

HISTOGRAM PROGRAMS

PREFACE

Book Three describes the operation of three programs: a) the Post Stimulus Time Histogram, b) the Latency Histogram, and c) the Time Interval Histogram program. Programs a) and b) are contained on one binary tape (DEC-8E-APLHA-A), program c) is contained on DEC-8E-ATINA-A-PB. In addition, the mass storage version of each is discussed: DEC-8E-ATIHA-A-PB for TIH and DEC-8E-APSTA-A-PB for PST.

The LAB8/E PST and Latency Histogram program records all signals crossing a set threshold after a given event. This program, which produces a histogram representing the sum of all activity following a particular stimulus, is most frequently used in experiments dealing with the response of a single nerve cell to a stimulus. The histogram is displayed on an oscilloscope as a frequency of event versus time plot.

The LAB8/E Time Interval Histogram program records the time intervals between events. These events are defined as a crossing of a set threshold. The program records the intervals by means of a histogram or frequency distribution graph; where the time interval is plotted against its frequency of occurrence. The program is generally used to record the spontaneous activity of single nerve cells or to establish the mean time interval between peaks of an analog signal like an electrocardiogram.

CONTENTS

TABLES

FIGURES

CHAPTER 1

TIH, PST AND LATENCY HISTOGRAM PROGRAMS

1.0 GENERAL DESCRIPTION

The LAB8/E Post Stimulus Time (PST) and Latency Histogram program records all signals crossing a set threshold after a given event. The program is used most frequently in experiments dealing with the response of a single nerve cell to a stimulus. After the primary event (S1 or a stimulus) the program measures the time of occurrence of all subsequent events (responses) above a certain analog threshold. The PST histogram represents the sum of all activity following a stimulus. It is not an analog summation (signal averaging) but an event summation which can be interpreted as a distribution of the probability that an event will occur at time T following a stimulus.

The histogram is displayed on the oscilloscope as a plot of frequency of event (y) versus time (x). The program allows the user to specify the number of firings following a stimulus to be recorded, the resolution of the time axis (bin width), the duration of the time axis, (number of bins), and the range of the time axis (the minimum time which must pass after the stimulus before analysis begins).

The Time Interval Histogram (TIH) program records the time between events occurring on one input (i.e., Schmitt trigger 2). The display is a frequency distribution showing the duration of the time interval versus its frequency of occurrence.

1.1 Hardware Requirements

4K PDP-8/E	
AD8-E	10 bit A/D converter
AM8-EA	16 channel multiplexer, 8 channels implemented
DK8-ES	Real-time clock
VC8-E	10 bit point plot display
Tektronix 602, 603, 604 or VR14 or RM503	Display oscilloscope

An OS/8 configuration is necessary to run OS/8 versions of histograms.

1.2 The Program

The PST and Latency Histogram program provides information about the activity which follows a stimulus or synchronizing event. The activity is defined by a signal which will fire a Schmitt trigger (the characteristics of a signal which cause a firing are described in Section 1.3). It is this event (the firing) and its time of occurrence which are detected by the program. In addition to recording the time of occurrence of events following a stimulus, the PST and Latency Histogram program can show a zeroth histogram. This is a graph showing how much activity followed each stimulus. The Y axis is the frequency of occurrence where the X axis is divided into units of epochs (1 epoch = 1 unit). This feature is useful in checking to see if the overall activity following a stimulus remains constant through time or decreases due to habituation or fatigue.

The TIH program also has a zeroth histogram. This display shows how much activity occurred during each time segment of the experiment. (The duration of the run is specified by defining a basic time unit and then specifying the number of units to be considered.)

Before data collection can start, the scheme for recording events and their time of arrival must be defined.

Latency: A post stimulus time histogram is a record of all activity following a stimulus for a set period of time. In certain studies, it is desirable to have a record of only the first few nerve potentials following a stimulus. This is called a latency histogram. The number of potentials recorded following the stimulus is called the order of the latency histogram. The program can form either a PST or latency histogram.

Time Scale: Since the LAB8/E is a digital device, time must be counted in increments rather than as a continuous function. The finer the increments, the more closely the real situation is approached. The majority of biological phenomena occur slowly enough that this departure from the real time world does not impose severe restrictions. Therefore, the first parameter we must assign is the time scale increment, or resolution. This is called the bin width. Since approximately 1ØØ μsec are required to acknowledge the tick of a clock, bin widths are defined in multiples of 1ØØ μsec.

Once the units of the time scale are defined, the length and range of scale must be assigned. This is done by supplying the number of bins which should be recognized, i.e., (bin width) x (number of bins) yields time scale length and time at which the time scale should begin. The latter parameter is defined in answer to the question "Minimum time?" (where minimum time is in terms of bins). As an example: A histogram is defined as having bin widths of 5.Ø milliseconds. It consists of 1ØØ bins and has a minimum time of Ø bins. The time scale would start at time zero (time zero is defined by the occurrence of the stimulus in the case of the PST and Latency Histogram program; in the TIH program, time is relative to each event), and range in increments of 5 milliseconds to time = 5ØØ milliseconds (5.Ø milliseconds x 1ØØ). If a minimum time had been specified, the starting and ending times of the range would be incremented by that amount (e.g., minimum time = 2 bins; therefore, 2 x bin width = 1Ø milliseconds total delay of the start of the range; thus the range would start at 1Ø milliseconds and end at 51Ø milliseconds.

Minimum time is used usually for one of two reasons. Either it is used to avoid artifacts which the stimulus might cause or it is used to move the time scale to the time of interest (i.e., if precise measurements had to be obtained at a specified time after the stimulus and no other time was of importance in the study).

Once the time scale is defined, the only other parameter that must be specified before data taking can start is the duration of the experiment. In the case of PST and Latency this is defined by specifying the number of stimuli to be recorded. The TIH has its duration defined by specifying a basic time unit; and the number of times this time unit should occur from the start of data collection. This timing feature is useful in studies where the nerve cell is prone to fatigue. After the preset time has elapsed, the program will notify the user and he can either output his data or return and take more data.

1.3 Connections

A lead which contains a synchronization pulse (a pulse occurring at the same time as the stimulus) should be connected to Schmitt trigger number 1. A second lead which contains the response activity or the spontaneous activity should be connected to the Schmitt trigger number 2. The Schmitt triggers are threshold detectors and can accept pulse or analog inputs. In either case, the circuit will fire when the trigger threshold has been exceeded after having been Ø.3 volts above or below the trigger threshold. The trigger threshold is set with the knob corresponding to the phone jack input connector. This threshold varies ± 5 volts. The input to the Schmitt trigger must be kept to within ± 5Ø volts. A technical description of the Schmitt trigger is contained in Appendix C. During the trigger initialization section, a lead containing an analog signal may be connected to analog input 1 or 2 but this is not necessary during the remainder of the program.

CHAPTER 2

PROGRAM EXECUTION

2.1 Program Loading

The Binary Loader (BIN) is used to load either of the histogram programs. If BIN is not present, refer to Appendix A for loading instructions. (For loading the program from the Disk Monitor system, follow the instructions given in the Monitor handbook. The SAVE command should be SAVE PST! Ø-4177, 742Ø-7577; 2ØØ and SAVE TIH! Ø-4177, 743Ø-7577; 2ØØ).

To load the histogram program, use the following procedure:

a. Place the tape in the tape reader.
b. Set the Switch Register to 7777.
c. Press the ADDR LOAD key.
d. If using a high-speed reader, set the Switch Register to 3777.
e. Turn the Teletype control knob to LINE.
f. Turn the scope to ON.
g. If using a low-speed reader, set the reader switch to START.
h. If using a high-speed reader, press reader ON switch.
i. Press the CLEAR and CONT keys on the computer.

See Chapter 3 for OS/8 version of the programs.

2.2 Running the Program

After the tape has loaded in, verify that the accumulator lights are all off. If they are not all off, there has been a reader error and the program should be reloaded. After loading is complete and correct, set 2ØØ in the Switch Register, press ADDR LOAD, then press CLEAR and CONT. At this point,the message

TRIGGER

is printed.

Trigger Initialization

At this point, adjust the Schmitt trigger thresholds (described in Section 1.3) by moving the clock knobs corresponding to the clock input connections so that the triggers are fired on data and not noise spikes. The trigger thresholds should be adjusted individually to avoid confusion, so during the initialization only one lead should be connected to the Schmitt trigger inputs at a time.

When the trigger fires, the scope will show a line across its face. The presence or absence of this line indicates whether or not the Schmitt trigger is firing. If desired, a lead may be attached in parallel (with the line going to the Schmitt trigger input) to the analog input number 1 or 2. Now, whenever the trigger fires, two traces will appear on the oscilloscope. One is a sweep of the analog input and the other is a baseline to represent the voltage at which the trigger fired. This feature is useful if a slowly varying analog signal is used to fire the trigger. By viewing the input, you can see where on the waveform the trigger is firing. The sampling rate of the analog channel is set by analog knob 3 on the front of the laboratory peripheral panel: counterclockwise to sample faster; clockwise to sample slower.

Once the trigger thresholds are set satisfactorily, all leads except S1 and S2 should be removed to avoid confusion. Type the RETURN key to advance the program.

NOTE

> After the program has left the trigger section, it is impossible to return to this section under program control. The program must be restarted by the switches at location 7424 for PST and 7432 for TIH. If data has been collected, this section may have been overwritten by data. If this is the case, the program must be reloaded to recalibrate the Schmitt triggers.

Parameter Setup

The user must now define the PST or Latency Histogram by answering a series of questions. The questions will be asked and the user must respond in the format described below. The user must type the RETURN key after all answers. Leading zeros are not required. If the question is improperly answered a ? is printed and the question is repeated. To back up one question to redefine a parameter, type the LINE FEED key.

If a mistake in answering is made, type the RUBOUT key to repeat the current question. To redefine all of the parameters, type CTRL/A (this is read as "control A"). To type it, hold down the CTRL key and type A. CTRL/C returns control to the Monitor, if one exists, at any time during the program. If no Monitor exists, refrain from issuing this command.

<u>Question 1</u> (not asked in TIH)

PST?

First of all, the program wants to know if it is to form a Latency Histogram or a PST Histogram (for description, see Section 1.2).

Answer Y (for yes), N (for no) followed by the RETURN key (RETURN is signified by ↵).

If Y is the answer, question 2 will be skipped and question 3 will be asked. If N is typed, the following question will be asked.

<u>Question 2</u> (not asked in TIH)

ORDER=

For a Latency Histogram, the program must know the number of theshold crossings following the stimulus to record. If a frequency distribution of the first pulse following a stimulus is desired, 1 should be typed. For a distribution of the first two pulses, 2 should be typed, etc.

Answer with a number between Ø and 4Ø95. This signifies the number of pulses to record following the stimulus. If all of the pulses following are to be recorded, Question 1 should have been answered Y.

<u>Question 3</u> (Question 1 for TIH)

BINW=

The question is asking for the bin width. The bin width is the basic time unit for the time scale used in building the histogram.

Answer with a number between Ø.1 and 4Ø9.5. The number is interpreted as milliseconds and must be in the form x.x ↵.

<u>Question 4</u> (Question 2 for TIH)

BINS=

The number of bins and bin width determine the range of the time scale.

Answer with a number between Ø and 189Ø. If more bins are requested, there will not be enough core for data storage; the message CORE? will be typed then the question will be asked again.

Question 5 (not asked for TIH)

EPOCHS= This question is to define the number of times the stimulus will be presented (the number of epochs).

Answer with a number between Ø and (189Ø - # bins). Again, it is possible for this answer to request more core than is available. The zeroth histogram is a graph showing how many firings were recorded consequent to each stimulus. Therefore, a core location is needed to record the count for each epoch. If core is exceeded, the message CORE? is typed and the question is asked again. Simply reduce the number of epochs or the number of bins until the answer is accepted.

Question 6 (Question 3 for TIH)

MINTIM= This question asks the starting time of the time range under consideration.

Answer with a number between Ø and 4Ø94. The units are bin widths.

Question 4 (for TIH only. Not asked for PST)

TIME\UNIT= This question asks what the basic time unit is for timing the experimental trail. It is unrelated to the histogram time units.

Answer with a number between Ø.1 and 4Ø9.5. The number is interpreted as milliseconds and must be in the form x.x).

Question 5 (for TIH only. Not asked for PST)

#UNITS= This question requests the number of basic time units defined in the previous question which is to be used for the experimental trial.

Answer with a number between Ø and (189Ø - # bins). Again, it is possible for this answer to request more core than is available. The zeroth histogram is a graph showing how many firings were recorded consequent to each stimulus. Therefore, a core location is needed to record the count for each epoch. If core is exceeded, the message CORE? is typed and the question is asked again. Simply reduce the number of epochs or the number of bins until the answer is accepted.

After the last question is answered, the program waits for CTRL/S to start data collection. (Type CTRL/A to repeat all questions; CTRL/C to return to the Monitor.) For an example of parameter setup, see Appendix K.

Data Taking

After the questions have been answered, type CTRL/S to start data taking. The LAB8 will now start displaying the first histogram. This is the PST or Latency Histogram which records the responses and their times of arrival following the stimulus; or the TIH displaying the occurrence of various intervals.

The X axis of the display is the time range. At the start of data taking, the time range consists of all the bins which were requested. These bins are shown with zero content at first so that their heights are coincident with the base line. As responses are registered, the bins will show themselves. If a minimum time is specified, the first bin on the left-hand edge of the X axis is the underflow bin. That is, the bin records all responses that occur at some time before the specified minimum time has elapsed. If no minimum time is requested, there will be no underflow bin and the first division of the X axis will be the first bin of the time range. The rightmost bin in the TIH is for all inter-pulse intervals greater than the largest bin. It is called the overflow bin.

NOTE

> The counts that fall into the underflow or overflow bins are not recorded in the zeroth histogram. The zeroth histogram records only those counts that occur during the specified time range.

The Y axis is also shown as a straight line. It differs from the X axis in that it contains 25 hatch marks spaced 40 raster points apart. These marks are provided so that bin content can be estimated visually during data taking. These hatch marks remain the same distance apart regardless of the Y scale factor.

There is one other display which can be viewed. This is called the zeroth histogram. To see this display, type <. This histogram is a graphical representation of the number of pulses which are received after each stimulus or during each time unit. The X axis here has an increment for each epoch (rather than for each bin). The first epoch is shown at the left-hand end of the X axis. This histogram is useful for determining the gross activity of the neuron through time.

The display will remain active during data collection. The collection algorithm for PST and Latency is the following. The program waits for a stimulus, S1 (Schmitt trigger 1). Until a stimulus is recognized, no responses are acknowledged. When S1 is received, the program accepts only responses (S2's) until the end of the time range is reached (i.e., total time = minimum time + (# of bins x bin width)). During this time, no S1's are acknowledged, so the user should make sure that the stimulus repetition rate is slower than the length of the analysis time period. Figure 3-1 describes the timing relationships and the signals that are acknowledged.

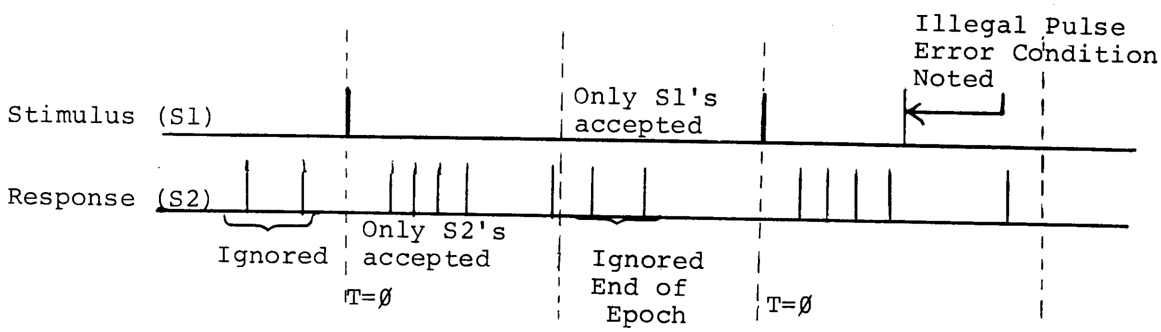


Figure 3-1 Timing Relationships

From the diagram it is clear that only those response pulses occurring during the analysis time are counted. An error condition is noted if an S1 (stimulus) is received during the analysis period. The time counter does not reset to zero so the S1 is effectively ignored. The occurrence of the untimely S1 is noted and is reported at a later time.

The collection algorithm for the TIH is the following: the time between events seen on Schmitt trigger 2 is measured and then recorded in the appropriate bin. If the time interval measured were 2.5 milliseconds and the histogram had no minimum time constraints with a bin width of 1 millisecond, the third bin of the histogram would receive a count. The first bin is bin Ø; so bin 2 would be incremented by one. If a minimum time is specified, the underflow bin, or the minimum time bin, is bin Ø. The overflow bin is always the last bin.

If at any time during data collection the contents of a bin in the histogram exceeds 4Ø96, the overflow is noted and the contents are set to 4Ø96. Every time a bin is filled over 4Ø96, the bin is set to its maximum and the overflow is noted. The errors are reported when data collection is finished.

Commands During Data Collection

During data collection there are several commands which can be executed. Most of these commands operate on the display. Table 3-1 is a list of possible "Group I Commands." Note that, although the display is changed, the data remains unaltered.

After data collection is complete, the program transfers to the output mode (OPM). This transfer takes place after the user-specified number of epochs is complete or after CTRL/Q or CTRL/P is typed during data collection. The computer outputs a carriage RETURN/LINE FEED, the message OPM, another carriage RETURN/LINE FEED, two four-digit numbers separated by commas, and then resumes the display.

ERROR CONDITION NOTIFICATION:

The two four-digit numbers that are typed represent the number of times two types of errors occurred during data collection. Error condition one (the first number) arises if an S1 or stimulus pulse is received during an epoch (refer to Figure 3-1). These S1 pulses have no effect on the data collection, but are noted as errors and reported here. Error condition two (the second number) arises if the contents of any bin exceed 4096_{10}. Each time the overflow occurs, it is noted, and the bin content is reset to 4096. Only error condition two is reported in the TIH.

At this time, all of the Group I commands are still active as well as the set of Group II commands described in Table 3-2.

Before using the program, the user should thoroughly familiarize himself with the Group I and Group II commands summarized in Appendix N.

Table 3-1

Data Collection Commands

COMMAND	ACTION
>	Display the next highest histogram. In the present histogram programs there are only two possible histograms to display. Therefore this command will only move the display from the zeroth to the first histogram.
<	Basically the same as > but moves the display back one histogram. Note that the display will not be forced to display nonexistent histograms by repeating these commands.
CTRL/Z	Stop data collection. Zero out all buffers and error counters. Wait for CTRL/S to restart collection.
CTRL/A	Go back to parameter setup and Alter parameters.
CTRL/Q	Quit data taking at the next end of epoch and transfer to output mode (data taking then may be resumed, saving all data by CTRL/R).
CTRL/P	Panic Stop. Immediately transfer to output mode. Do not wait for end of epoch or anything else.
CTRL/C	Immediately return to Monitor (this command should not be used if system Monitor does not exist).
C	Each time this command is issued, the display alternates between displaying Cursors or no Cursors. The cursors are two vertical lines which are controlled by analog knobs 0 and 1. Their relative positions are unimportant.
U	Scale the dimension Y Up by a factor of 2. This does not affect the actual bin values, only the display.
D	Scale Y Down by a factor of 2. (U and D may be given any number of times.)
E	Expand the area of the histogram enclosed by the cursors so that the area takes up the entire X axis. (E may be done any number of times.)

Table 3-1

Data Collection Commands (Cont.)

COMMAND	ACTION
B	Return to the Beginning expansion factor (i.e., no expansion). In other words, cancel all E commands.
S	Before display, Sum adjacent bins. This command adds adjacent bins and displays the resulting histogram. This effectively doubles the bin width. It also can be considered a type of smoothing routine. At the start of the program, the sum factor is set to one. Each time the S command is executed, the sum factor is increased by one. When sum factor = 1, each bin is displayed separately; when Sf=2, two bins are added together and displayed as one. When Sf=3, three bins are added together and displayed as one, etc.
O	Set sum factor to 1 or return to Original sum factor.
Z	Zero all display changes or set all displays to unmodified state.
A	Each time this command is issued, the display alternates between displaying an Axis or no Axis. This command is included for aesthetics only. Its sole purpose is to yield a clearer display.
V	Each time this command is given, the display switches between Viewing the histogram in a bar graph display or a point plot display.
CTRL/U	This is an undefined User command which can be implemented by following the instructions in Appendix L. Unless the program is modified, it will have no effect.

Table 3-2

OPM COMMANDS

<table>
<tr><th>Command</th><th>Action</th></tr>
<tr><td>T</td><td>
Type out the data enclosed by the cursors. Before the T command is issued, the cursors should be placed so that they surround an area of interest. When T is typed, the scope goes blank and the Teletype types the contents of the bins between the cursors. If the cursor line goes through any part of a bin, that bin is included in the typeout. To terminate the typeout before the right cursor is reached, type CTRL/Q. Until the user is familiar with the typeout portion of the program, the following procedure is suggested. Bring up a bar graph display. Position the cursors <u>on</u> the bins that delimit the area of interest. Type T. A sample of the output is the following:
1, ØØØ1, ØØØØØ1Ø, ØØØØ229, .Ø436, ØØ53, ØØ56,
ØØ53, ØØØ6, .6ØØØ, .Ø262,
ØØ55, ØØØ2, .2ØØØ, .ØØ87,
ØØ56, ØØØ2, .2ØØØ, .ØØ87,
Starting with the leftmost number on the first line, the interpretation is:
<table>
<tr><td>1</td><td>the histogram being typed out is the first-order histogram (not zeroth).</td></tr>
<tr><td>ØØØ1</td><td>the sum factor</td></tr>
<tr><td>ØØØØØ1Ø</td><td>the total number of counts between the cursors</td></tr>
<tr><td>ØØØØ229</td><td>the total number of counts in the entire histogram</td></tr>
<tr><td>.Ø436</td><td>the ratio of the number of counts between the cursors to the number of counts in the entire histogram is .Ø436 to 1 or 1Ø/229.</td></tr>
<tr><td>ØØ53</td><td>bin number 53 is the leftmost bin being typed</td></tr>
<tr><td>ØØ56</td><td>bin number 56 is the rightmost bin</td></tr>
<tr><td>ØØ53</td><td>on the second line, bin number 53 contains six (ØØØ6) counts</td></tr>
<tr><td>.6ØØØ</td><td>the ratio of the number of counts in this bin to the number of counts between the cursors is .6 to 1 or 6/10.</td></tr>
<tr><td>.0262</td><td>the rate of the number of counts in this bin to number of counts in the entire histogram is .0262 to one or 6/229.</td></tr>
</table>
Each bin is reported in this manner. If a bin has zero contents, its typeout is skipped. In this example, bin 54 was omitted because it contained nothing. Commas are provided to make any output tape compatible with BASIC/RT (refer to Book Seven).
Note that, if a minimum time is specified, bin Ø represents a count of all S2's occurring before the minimum time is elapsed. It is the Minimum Time Bin (or underflow bin). If no minimum time is specified, bin Ø is the first bin of the histogram. The total number of bins displayed is always equal to the user-specified number.
</td></tr>
</table>

Table 3-2

OPM Commands (Cont.)

Command	Action
	Another fact to be considered is that when the histogram has been subjected to the Sum command, the bin numbers refer to the bin numbers of the original, unaltered histogram. The bin number typed out refers to the left-most bin that is summed with its neighbors on the right. Example: Bin 1Ø has 4 counts Bin 11 has 3 counts Bin 12 has 8 counts Bin 13 has 5 counts If S is typed once and the T command is then issued, the typeout will be the following: BIN 1Ø; 7 COUNTS BIN 12; 13 COUNTS NOTE When S has been typed, a skipped bin number does not necessarily mean a bin of zero contents. Similarly, when the E command is executed, followed by T, the bin numbers refer to the original unaltered histogram.
K	Calibrate an X-Y analog plotter. When the K command is given, the outputs of the oscilloscope go to the maximum positive voltage for the X and Y coordinates. When any key on the keyboard, except CTRL/Q, is typed, the outputs go to the maximum negative voltage. After the outputs have gone to the maximum negative voltage, typing any other key causes the outputs to go back to positive voltage. This process may be repeated as many times as needed while the bias and gain controls for the X-Y plotter are adjusted (see Appendix H for details of attaching an analog recorder). When the plotter is calibrated, type CTRL/Q to resume the display.
P	Plot the histogram being displayed on the analog X-Y recorder. The histogram is plotted in the exact scale as it is displayed, in bar graph form, with an axis and no cursors. After the initial command, the program waits for the user to turn on his plotter and lower the pen. Type any key on the keyboard and the axis and histogram will be plotted. The speed of the pen can be controlled with analog knob 3. Adjustments of the speed can be made while the plotting is taking place. Turn the knob counterclockwise to accelerate the plot and clockwise to slow the plot.

Table 3-2

OPM Commands (Cont.)

Command	Action
	When the axis and histogram have been plotted, the program pauses again. Raise the pen and turn off the plotter. If the plotter is on while the display is active, the pen will shake furiously, trying to keep pace with the changes in analog voltage. When the plotter is turned off, type any key and the display will be resumed.
CTRL/B	Causes a binary dump of the histogram being displayed. NOTE The histogram is returned to normal size and scale before the binary dump takes place. After the initial command of B, the program prints HIGH?. Answer Y ⤶ (for yes) if the dump is to proceed on a high-speed punch or N ⤶ if the Teletype punch is to be used. After this initial question, the program prints ID. Enter a maximum of five characters followed by the RETURN key for an identification code. CAUTION The following characters may not be used in the ID code: RUBOUT, CTRL/C, or CTRL/A. A LINE FEED at this point repeats the question HIGH? Typing the RUBOUT key causes the ID question to be repeated. After the ID code has been entered, 64 leader/trailer punches are punched. If the low-speed punch is to be used, it should be turned on during the leader/trailer. Following the leader/trailer code, the ID code is punched in ASCII and followed by another 64 leader/trailer punches. After the identifying information has been output, a core image of the display list (the pertinent information relating to the histograms for display purposes), a core image of histogram (which was being displayed when the CTRL/B command was executed), and a checksum are punched. A core image dump allows the papertape to be read back into the computer in the same form as it was originally dumped. The binary loader can be used to reload the data. This feature allows the user to take data, dump it and then return at any later time to process it or review it. The user may write his own processing program or read the data back into the histogram program. For details of binary format, see Appendix M. To review the data with the program, read in the program as described in Section 2.1 (if it is in core, reloading is not necessary). After the program

Table 3-2

OPM Commands (Cont.)

Commands	Action
	is in core, load the data with the binary loader using the same procedure as if the main program were being loaded. After the data and program are in core, start the program at location 1375 (place 1375 in the front switches. Press ADDR LOAD and CLEAR and CONT.) Now operate the program is if the output mode had just been entered. Note Since the CTRL/B command dumps only the histogram being displayed, only one histogram will be displayed properly after data readin. If this program is run from disk or DECtape, the histogram can be dumped by the following procedure. 1. Type CTRL/C - Monitor returns. 2. Save locations 4ØØØ-7577 as a binary file. 3. Restart the program at location 1375. 4. Continue as if a binary dump had taken place. To review this data, load the program then the data. Start at location 1375. After the binary dump has taken place, the program returns to the active display and all of the commands are operative.
CTRL/I	This is a user Group II command which may be Implemented in the same manner as CTRL/U. See Appendix L. Until a patch is made to implement it, issuing this command does nothing.
CTRL/Z	Zero all buffers and error counters.
CTRL/R	Retain the data in the buffers so that data collection may be resumed without loss of data. The effect of this command on the first order histogram is to allow it to accumulate another preset number of epochs. The effect of this command on the zeroth histogram is to set the epoch pointer back to zero. That is, the first bin will be incremented by the number of pulses received during the next epoch. The error counters are not reset.
CTRL/S	Follows either CTRL/R or CTRL/Z to start data collection again.
CTRL/A	Return to redefine parameters.

CHAPTER 3

MASS STORAGE TIH, PST AND LATENCY HISTOGRAMS

3.0 GENERAL DESCRIPTION

The mass storage version of the histogram programs is the same as the paper tape program except that it runs under OS/8 and can read and write the histogram to and from the system device. The tape numbers for the binary program are:

DEC-8E-ATIHA-A-PB	for TIH
DEC-8E-APSTA-A-PB	for PST & Latency

3.1 Using MS TIH, MS PST and Latency Programs

Loading the Programs

The programs can be loaded via the Absolute Loader and started at 2ØØ of field Ø.

If the programs are to be saved with the SAVE command the core locations to be saved for both programs are Ø-7577 of field Ø and 74ØØ-7577 of field 1. For example, to SAVE PST, load the binary via ABS Loader, terminated by ALTMODE and save the core image via the SAVE command, SAVE SYS PSTD1.SV Ø-7577,174ØØ-17577;2ØØ.

Starting Locations

The normal start for both programs is 2ØØ of field Ø. If data is not to be collected but a file is to be read in and reviewed, the program can be started at 1375.

Commands

CTRL/W is used to read or write the histogram data to mass storage. When CTRL/W is typed, the OS/8 command decoder is called to request an input or output file. An asterisk is printed to request the file (use OS/8 command decoder format). CTRL/W is active only during the output mode of the program.

Output or input can be done only to or from the system device. Answer the asterisk as follows.

*FILE.X

to read FILE.X or

*FILE.X<

to output to file FILE.X on the system device. An input file must have been written by the same program that is calling it.

<u>Errors</u>

An error during data transfer causes the message

USER ERROR 1

and control returns to OS/8.

3.2 Data Output Format on Mass Storage

When a file is written out by a histogram program, page Ø, the zeroth histogram and the histogram are written to the system device. The format of the file is:

Block 1	Word 1	-	is a 2
	⋮	-	Location 1 of page Ø rest of page Ø
	256	-	up to location 256
Block 2	Word 1	-	Loc. 4ØØØ, location -1 of start of Øth histogram
	2	-	Loc. 4ØØ1, end location of Øth histogram
	3	-	Loc. 4ØØ2, location -1 of start of histogram
	4	-	Loc. 4ØØ3, end location of histogram
	5	-	Loc. 4ØØ4, start of Øth histogram
	6		End of block = histogram data as in core.
Block 3			continuation of data as in core, blocks continue to end of data

BOOK FOUR

AUTO- AND CROSS-CORRELATION

PREFACE

The Auto- and Cross-Correlation program for papertape, DEC-LB-U41B-PB, is designed to correlate data (five bit samples (4 bits + sign)) at sampling rates ranging from .1 to 2Ø4.7 milliseconds, on line, with the user controlling all parameters from the Teletype. It displays and scales data while computing and provides output that can be used with FOCAL, BASIC, or user's programs. An OS/8, version, DEC-8E-AACRA-A-PB, runs under OS/8 and transfers its data to and from the system device.

A listing is available through the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754.

CONTENTS

BOOK FOUR

TABLES

FIGURES

CHAPTER 1

AUTO- AND CROSS-CORRELATION PROGRAM

1.0 GENERAL DESCRIPTION

A variety of techniques are available for extracting signals from their noisy environments. These include filtering (frequency domain), signal averaging (amplitude domain), and correlation (time domain). Each technique has its own advantages and disadvantages and hence its own area of application.

Correlation, as it applies to waveforms, can be used to detect periodic signals buried in noise or provide a measure of similarity between two waveforms. Auto-correlation measures the similarity of a signal to a time delayed version of itself, while cross-correlation measures the degree of similarity of one source or input to a second source. No synchronizing events, such as the trigger required in signal averaging, need be available for the application of correlation techniques.

Typical applications for correlation include:

1. Determination of the transmission paths and propogation velocities of mechanical vibrations, acoustic and seismic waves.
2. Indication of epilepsy through comparison of electro-encephalograms from the two halves of the brain.
3. Measurement of impulse-response characteristics of servo control systems in the presence of noise.
4. Analysis of the tremor frequencies of patients afflicted with Parkinson's disease.
5. Detection of very weak and distant radio sources in space.
6. Speech research in telecommunications, education for the deaf, and non-computer interfaces.

1.1 Theory[1]

Mathematically, the auto-correlation function is defined as:

$$R_{xx}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{\phi}^{T} x(t)\,x(t-\tau)\,dt \qquad (1)$$

[1]The reference for the statistical equations is: Bendat, Julius S. and Piersol, Allan G., Measurements and Analysis of Random Data, John Wiley & Sons, Inc., N. Y. 1967.

where x(t), the time function of interest, is multiplied by a time delayed version of itself, x(t-τ), and the product of the two values averaged over the observation time T. The resulting average approaches an exact correlation as T approaches infinity.

Cross-correlation is very similar to auto-correlation, the difference being that the input is multiplied by a time delayed version of a second signal, rather than a time delayed version of itself. The cross-correlation function of a waveform is defined as:

$$R_{xy}(\tau) = \lim_{T \to \infty} \frac{1}{T} \int_{\phi}^{T} x(t)y(t-\tau)dt \qquad (2)$$

To apply digital computing techniques to the above equations, it is convenient to approximate the average by sampling the signal at regular intervals, for instance t, and then summing a finite number, k, of the sample products. Rewritten in discrete form, equation (1) becomes:

$$R_{xx}(\tau) \approx \frac{1}{k+1} \sum_{i=\phi}^{k} x(t_i)X(t_i-\tau) \qquad (3)$$

If the maximum time delay, τ_{max}, is divided into N intervals of $\Delta\tau$ and the time between samples Δt is made equal to the time delay increment $\Delta\tau$ equation (3) becomes:

$$R_{xx}(n\Delta\tau) \approx \frac{1}{k+1} \sum_{i=\phi}^{k} x(t_i)x(t_i-n\Delta\tau) \qquad \text{where: } n=\emptyset,1,2,3,\ldots N \qquad (4)$$

which is the equation used in the LAB8 program. Similarly, equation (2) can be approximated by:

$$R_{xy}(n\Delta\tau) \approx \frac{1}{k+1} \sum_{i=\phi}^{k} x(t_i)y(t_i-n\Delta\tau) \qquad (5)$$

1.2 Properties and Examples of Correlation

The auto-correlation function has the following properties:

1. Auto-correlation is an even function; i.e., $R_{xx}(\tau) = R_{xx}(-\tau)$, therefore, the correlogram need only be computed and displayed for positive values of τ. $\tau = \emptyset$ is displayed at the left of the screen.

2. The auto-correlation function has a maximum value for $\tau = \emptyset$. Sometimes it is convenient to set this value corresponding to $\tau = \emptyset$ to 1 and express all other data as a fraction. The CTRL/T command of the program does this automatically.

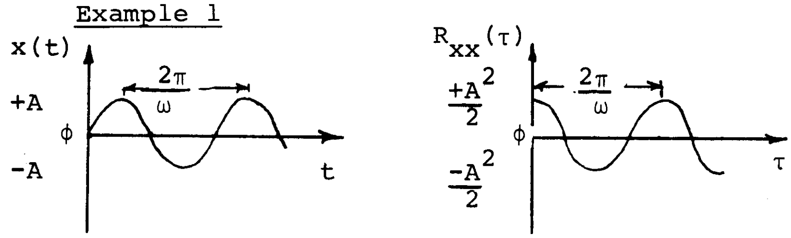
3. The value of the auto-correlated function for $\tau = \emptyset$ is the mean square voltage of the time function, substituting $\tau = \emptyset$ in equation (1).

$$R_{xx}(\phi) = \lim_{T \to \infty} \frac{1}{T} \int_{\phi}^{T} x^2(t)\,dt = \overline{x^2(t)}$$

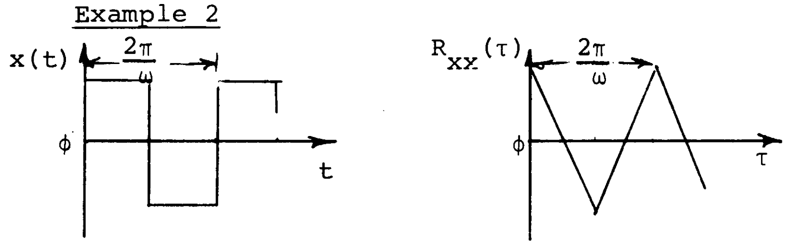
   The T command in the program automatically normalizes the data and types it out as mr^2.

4. If the time function contains periodic components, the auto-correlation function will contain components having the same periods.

Example 1

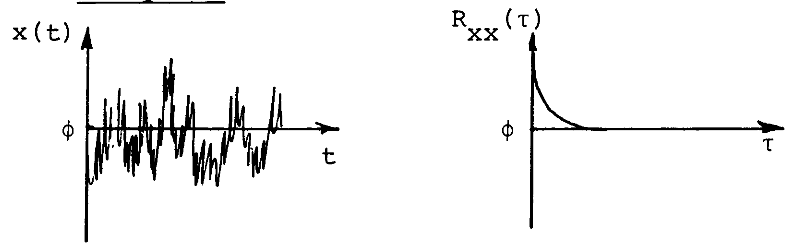


Example 2



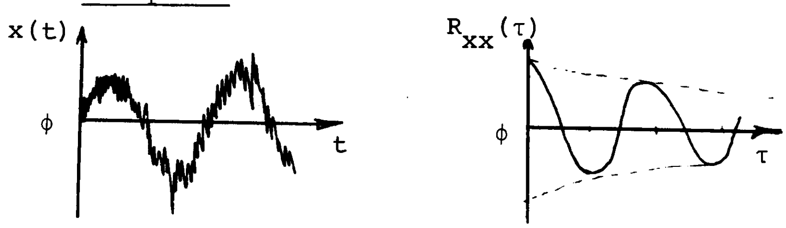
5. If the function contains only random components (noise) and no periodic components, the auto-correlation function will approach zero as τ increases. The rate at which it approaches zero is a measure of the randomness of the time function and, therefore, the bandwidth of the noise.

Example 3



6. If the time function is composed of two or more components, the resultant auto-correlation function will be the sum of the auto-correlation functions of each individual component.

Example 4



The cross-correlation function has the following properties:

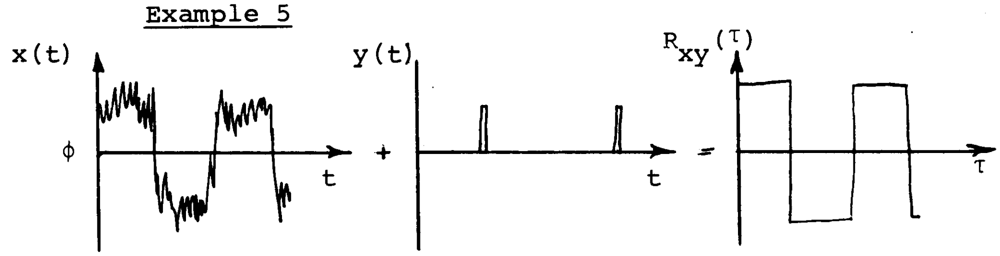
1. The cross-correlation function is not necessarily an even function; therefore, in general, $R_{xy}(\tau) \neq R_{xy}(-\tau)$. Consequently, a time shift of one input function in one direction with respect to the other does not produce the same result as an equal shift of the same function in the opposite direction.

2. $R_{xy}(\tau)=R_{xy}(-\tau)$. A shift in one input function x(t) in one direction must yield the same result as an equal shift of the other input function y(t) in the opposite direction.

3. $R_{xy}(\tau)$ does not necessarily have a maximum at $\tau=\emptyset$. The maximum value of $R_{xy}(\tau)$ will occur for that value of time shift τ for which the two input functions are most alike.

The following example illustrates the application of the cross-correlation function:

Example 5



1.3 Principle of Operation

A simplified block diagram of the LAB8/E correlation program is shown in Figure 4-1.

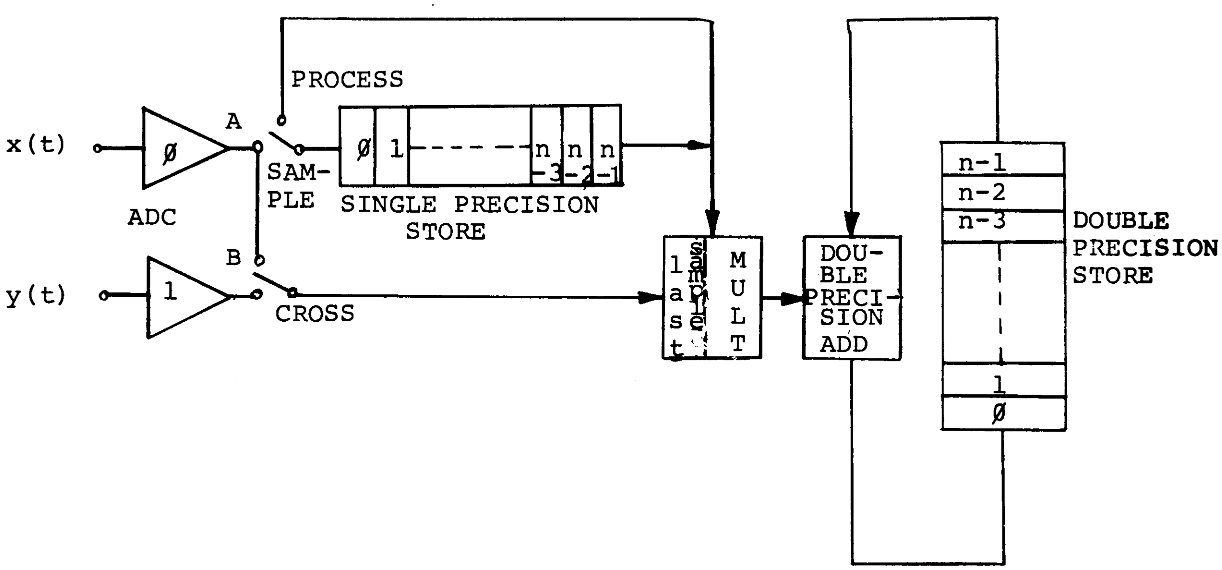


Figure 4-1 Simplified Block Diagram of the LAB8/E Correlation Program

When the program is started, the user is requested to reply to three questions: rate, number of points, and auto? Once these questions have been answered, the program does the following automatically:

1. Sets the clock to take analog samples at the specified rate. Sampling rate = $\Delta\tau$.

2. Sets up a single precision buffer to hold the delayed samples.

3. Sets up a double precision buffer to hold the correlogram data.

4. Sets the software auto/cross switch to the proper position.

The data collecting part of the program can now begin.

The memory updating process can be visualized as continually storing data in a closed loop. After each revolution of the loop, the oldest data point in the shift register is replaced by a new sample.

As each sample is converted into an equivalent digital value by the analog-to-digital converter, it is stored in the single precision store. This buffer acts in a manner similar to a shift register. That is, for each new sample entered, the oldest sample is lost.

As the data is being shifted in the single precision store, calculations are being performed to update the correlogram. Each data point is multiplied by the most recent sample ($\tau=\emptyset$) and the product is added to the corresponding cell in the double precision store. In cross-correlation, the multiplier is a sample from another channel. Table 4-1 shows the calculations formed for cross-correlation.

Table 4-1 Calculations for Cross-Correlation

Samples	Shift Register Number			
t_i	$\emptyset$	1	2	...n
$t_\emptyset$	$x(t_\emptyset)y(t_\emptyset)$	$x(t_\emptyset)y(t_\emptyset-\Delta\tau)$	$x(t_\emptyset)y(t_\emptyset-2\Delta\tau)$...	$x(t_\emptyset)y(t_\emptyset-n\Delta\tau)$
t_1	$x(t_1)y(t_1)$	$x(t_1)y(t_1-\Delta\tau)$	$x(t_1)y(t_1-2\Delta\tau)$...	$x(t_1)y(t_1-n\Delta\tau)$
t_2	$x(t_2)y(t_2)$	$x(t_2)y(t_2-\Delta\tau)$	$x(t_2)y(t_2-2\Delta\tau)$...	$x(t_2)y(t_2-n\Delta\tau)$
⋮	⋮	⋮	⋮	⋮
t_i	$x(t_i)y(t_i)$	$x(t_i)y(t_i-\Delta\tau)$	$x(t_i)y(t_i-2\Delta\tau)$...	$x(t_i)y(t_i-n\Delta\tau)$
	$\sum_{i=\emptyset}^{k} x(t_i)y(t_i)$	$\sum_{i=\emptyset}^{k} x(t_i)y(t_i-\Delta\tau)$	$\sum_{i=\emptyset}^{k} x(t_i)y(t_i-2\Delta\tau)$..	$\sum_{i=\emptyset}^{k} x(t_i)y(t_i-n\Delta\tau)$

1.4 Hardware Requirements

PDP-8/E	
AD8-ES	1Ø bit A/D converter
DK8-EP	Real-time clock
VC8-E	1Ø bit point plot display
VR14, Tektronix 602, 603, 604 or RM503	Display oscilloscope
AM8-EA	8-channel multiplexer must be implemented to perform cross-correlation and to obtain a printed or plotted output.

1.5 Program Loading

The binary tape of the Auto- and Cross-Correlation program, DEC-LB-U41B-PB, is loaded using the binary loader. First, place the tape in the high- or low-speed reader and advance to the leader/trailer portion. Load 7777 in the Switch Register (SR) and depress ADDR LOAD. If loading with the high-speed reader, set bit Ø to Ø. Depress CLEAR then CONT keys.

After the tape has been loaded, load Ø2ØØ in the SR, depress ADDR LOAD, push CLEAR, then CONT keys.

The binary tape, DEC-LB-U41B-PB, can be loaded with the disk monitor system by following the disk monitor's instructions for loading tapes, i.e., SAVE AUTO! Ø-3177,62ØØ-7577;2ØØ

1.6 Using the Auto- and Cross-Correlation Package

System Initialization

Load the binary tape as described in Section 1-5.

Once the program has been started, it prints:

RATE ?

Type the sampling rate (τ) in milliseconds. The number typed must be in one of three formats:

1) xxx.x

2) xx.x

3) x.x

"x" signifies a decimal digit. The value of the number typed must be in the range of 1 to 2Ø4.7. If the number exceeds this range, the message

ØØØ.Ø<RAT<2Ø4.8

is printed. In practice, the rate specified should be at least four times the maximum frequency being sampled.

If something other than a digit is typed, it is ignored and not echoed.

The sampling rate is the time between the taking of samples from the analog to digital converter. Once the rate has been entered, the message

NO. OF POINTS?

is printed. This parameter defines the number of points that will make up the correlogram. Input is a decimal number of no more than three digits. Its range is 4<number<513. If the number entered is outside this range, the message

4<# OF POINTS<513

is printed.

In all cases, when typing a number to define the rate and number of points, input can be restarted by typing a RUBOUT, which echoes a backslash (\ ). Input can be prematurely terminated when typing the number of points by typing the RETURN key.

Once the number of points and the sampling rate have been defined, the computer prints:

AUTO?

If the signal is to be auto-correlated, respond with a Y; if it is to be cross-correlated, type an N. Any other characters are ignored and not echoed.

After the reply is entered, the program prints either

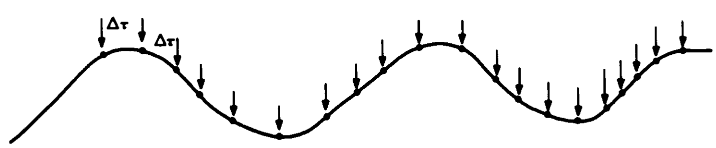
BATCH MODE

or

NORMAL MODE

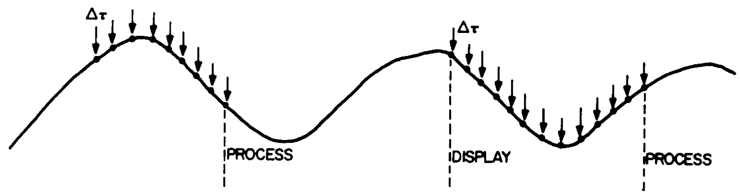
A small digression is necessary to explain the difference between BATCH MODE and NORMAL MODE. It was found that to process one point took at least 2ØØ μsecs. To process N points took at least (2ØØN) μsecs. Thus the user, when defining the rate and number of points , had to make sure he was allocating at least 2ØØ μsecs per point. For example, suppose the rate is defined as 2Ø.Ø millisecs., and the number of points as 1ØØ. 2Ø.Ø millisecs. = 2Ø,ØØØ microsecs., $\frac{20,000}{100}$ = 2ØØ μsecs per point. But what if the rate were 2Ø.Ø msecs., but with 2ØØ points; this would mean 1ØØ μsecs. per point; which is not enough time to process one point. When there are 2ØØ or more microseconds per point, the correlator operates in NORMAL MODE, that is, it does one entire multiplication sweep, reads in a new value from the ADC, does another sweep, etc. This provides a steady sampling rate. However, if the time allocated to the processing of one point is less than 2ØØ μsecs., then BATCH MODE is entered. In this mode, the entire buffer is filled with samples taken at the defined rate. Once N+1 samples have been taken, the buffer is multiplied by the last sample and displayed, after which new data is taken. The following graphs illustrate the two modes:

NORMAL MODE



The arrows represent the places on the graph where samples are taken. Δτ is the time between samples, or the rate. During this time, the buffer is processed.

BATCH MODE



Because of its nature, BATCH MODE takes longer to yield a clean correlogram. Therefore, the user should operate in NORMAL MODE whenever possible.

After the computer notifies the user of the mode, it displays the empty buffer and waits for CTRL/S to start data collection.

NOTE

> During the time of parameter setup, typing CTRL/A restarts the parameter setup routine, or CTRL/C jumps to 76ØØ and returns to the Monitor, if it exists.

Input Connections

If doing auto-correlation, the lead must be connected to channel Ø. If cross-correlation, the two leads must be connected to channels Ø and 1. The shift register is filled with samples from channel Ø. In either case, the input should be externally adjusted to the range of ±1.Ø volt. For best results, the input should be as close as possible to ±1.Ø volt.

Commands

There are two major groups of commands:

Group 1: These commands are active during data collection and output.

Group 2: These commands are active during data output and are inactive during data collection. They are ignored if typed while data is being correlated.

The commands are described in Table 4-2.

1.7 Miscellaneous

Program response to keyboard commands and the integrity of the display will differ between collection modes during data collection (refer to System Initialization for description of modes).

Because of the nature of data collection, BATCH MODE allots less time to keyboard monitoring and display. This has no effect on the quality of the data, and a fully active display and responsive keyboard will return when data collection has ended.

Table 4-2

Commands

Command	Action
CTRL/S	Starts data taking. It clears the buffer and all counters before starting.
CTRL/C	Stops data collection, turns off the clock and jumps to 76ØØ in field Ø. If running with the disk monitor system, this command returns control to the monitor.
CTRL/A	Immediately transfers control of the parameter setup routines and clears the buffers.
CTRL/Z	Clears the buffer and the sweep counter. The program then waits for CTRL/S to start data collection. CTRL/Z must be preceded by a CTRL/Q if data taking is in progress. In batch mode, CTRL/Z cannot be typed until the display appears on the scope.
CTRL/R	Restarts data taking, but does not clear the buffer or the sweep counter. It is generally used for restarting data taking after a CTRL/Q command.
CTRL/Q	Stops data collection after the current sweep and enables all Group 2 commands.
CTRL/U	This is an unimplemented user command. Details of its implementation may be found in Appendix O.
U	Expands the display of the correlogram by a factor of 2. The scale factor is initially set to 2^4.
D	Contracts the display of the correlogram by a factor of 2. Display of input is not affected.
V	Determines whether the input or the correlogram is displayed. The input is initially displayed. Typing V will switch the display from input to correlogram, or vice versa. When switching modes, the display may deteriorate due to interrupts in the middle of the display loop.
X	Effects only the display of the correlogram. The base line is shifted by the value of knob 2. The base line will be biased by the same amount each time X is typed. The correlogram can thus be moved up and down on the scope.
Z	Restores the correlogram to its unbiased X axis. It is used to counteract all previous X commands.
	Group 2
C	Alternately turns on and off the display of cursors. The two cursors are initially off. Their locations on the X-axis are determined by knobs Ø and 1.

Table 4-2

Commands (Cont.)

Command	Action
	NOTE The inputs to analog inputs Ø and 1 must be disconnected before the cursors are used.
N	Prints the number of sweeps and the current scale factor in the following format: xxxxxxx (number of sweeps) 2 ±xxxxxxx (scale factor) where x is a decimal digit. The number of sweeps is defined as the number of times the first position of the shift register is multiplied by the results of an A/D sample.
K	Calibrates an X-Y analog plotter. When the K command is given, the outputs of the oscilloscope go to the maximum output voltage for the X and Y scope coordinates. When any non-command key on the keyboard is typed, the outputs each go to the minimum voltage. When any other key is typed, the outputs go back to the maximum. This process may be repeated as many times as needed while the bias and gain controls for the X-Y plotter are adjusted. (Refer to Appendix H for details of attaching an analog recorder.) When the plotter is calibrated, type CTRL/Q to resume the display.
P	Plots the displayed waveform on an analog plotter. After P is typed, the first point is displayed, and the plot routine waits for a character to be typed before proceeding with the plot. The plotter should now be turned on. Once a character has been typed, the data is plotted. The speed of the plot is determined by knob 3 on the laboratory peripheral. Turn it counterclockwise to go faster, clockwise to go slower. This should be adjusted until the plotting speed is appropriate for the plotter being used. At the end of the plot, the routine stops, waiting for a character to be typed before proceeding. This allows the user time to turn off the plotter. After the character is typed, plot mode is terminated, and the display returns to normal.
CTRL/I	is an unimplemented user command. For details of implementation, see Appendix O.
CTRL/B	causes a binary dump of the correlogram. (only used in paper tape system, DEC-LB-U41B-PB NOTE The correlogram is returned to normal size and scale before the binary dump takes place.

Table 4-2

Commands (Cont.)

<table>
<tr><th>Command</th><th>Action</th></tr>
<tr><td></td><td>After the initial command of CTRL/B, the computer prints:

OUT?

Answer H if the dump is to proceed on a high-speed punch or L if the Teletype punch is to be used. (Any non-command character is ignored and not echoed.) After this initial question, the computer prints:

NAME:

Enter a maximum of six characters followed by the RETURN key for an identification code.

CAUTION

The following characters can not be used in the identification code:

RUBOUT CTRL/C CTRL/A

Typing RUBOUT causes input to be zeroed and then restarted.

After the identification code has been entered, leader/trailer code is punched. If the low-speed punch is to be used, it should be turned on during the leader/trailer output. Following the leader/trailer code, the identification code is punched in ASCII and followed by more leader/trailer code.

After the identifying information has been output, a core image of the correlogram and a checksum are punched. A core image dump allows the paper tape to be read back into the computer in the same form as it was originally dumped. The binary loader can be used to reload the data. This feature allows the user to take data, dump it, and then return at any later time to process or review it. The user may write his own processing program or he can read the data back into the auto- and cross-correlation program. For details of binary format, see Appendix P. To review the data with the auto- and cross-correlation program, load the program as described in Section 1.5 (if the program is in core, give the command CTRL/Z; reloading is not necessary).

After the program is in core, load the data with the binary loader. After the data and programs are in core, start the program at location 7560 (place 7560 in the front switches, press ADDR LOAD, CLEAR and CONT keys). Now operate the program as if the output mode had just been entered.</td></tr>
</table>

Table 4-2

Commands (Cont.)

Commands	Action
	NOTE Since only the correlogram is dumped and not the temporary buffer, reloading of the data may cause a bad display of the temporary buffer. This will not effect the proper operation of the program. If this program is run from disk or DECtape, the correlogram can be dumped by the following procedure: 1. Type CTRL/C. Monitor returns. 2. Save page zero and locations 32ØØ-6177 as binary files. 3. Reload the program and start at 2ØØ. To review this data, load the auto- and cross-correlation program, then the data. Start at location 756Ø. After the Binary dump has taken place, the program returns to the active display and all of the commands are operative.
T	Prints the data between the cursors. If the cursors overlap one another, the typeout routine is immediately exited. An example of the output follows: ØØØØØØ1, ØØØØØØ4, ØØØØ8Ø4, ØØØØØØ4, ØØØØ2ØØ Ø175872 ØØ57856 ØØ35Ø72 -ØØ15872 ØØØØØØ1 signifies the leftmost cursor is on the first point of the correlogram ØØØØØØ4 the position of the rightmost cursor (i.e., the fourth point of the correlogram) ØØØØ8Ø4 is the sweep count which indicates the number of times the first position of the shift register was multiplied by an ADC sample ØØØØØØ4 is the number of points whose values are being printed ØØØØ2ØØ 2ØØ x 1ØØ μsec is the amount of time represented by the distance between the cursors Ø175872 is the normalized value for the first position of the correlogram

Table 4-2

Commands (Cont.)

<table>
<tr><th>Command</th><th>Action</th></tr>
<tr><td></td><td>ØØ57856 is the normalized value for the second position of the correlogram.

Before the data is typed, it is normalized by one of the following standard equations:

For "normal mode" normalization:

Normalized value of the i^{th} point =

$$\frac{(\text{raw value of the } i^{th} \text{ point})}{(\text{number of sweeps}) - (i)}$$

For "batch mode" normalization:

Normalized value of the i^{th} point =

$$\frac{(\text{raw value of the } i^{th} \text{ point})}{(\text{number of sweeps})}$$

The units for the data points are (millivolts).

NOTE

If the LAB8/E is equipped with a single analog input, a typeout can only be obtained by placing a +5 volt signal on the input and typing T. The contents of the entire buffer are dumped.

The typeout continues until all the data between the cursors has been output. Typeout can be stopped prematurely by using the CTRL/Q command.</td></tr>
<tr><td>CTRL/T</td><td>is similar to the T command in that it also types data. The first line of output is identical to the first line printed by the T command. However, the data points are printed as correlation coefficients, ranging from +1 to -1, rather than as normalized values.

NOTE

This command should only be used while doing auto-correlation. The results are meaningless for cross-correlation.

An example of the output follows:

ØØØØØØ1, ØØØØØØ4, ØØØØ8Ø4, ØØØØØØ4, ØØØØ2ØØ

+Ø.99999E+ØØ
+Ø.32827E+ØØ
+Ø.19863E-Ø1
-Ø.92564E-Ø1</td></tr>
</table>

Table 4-2

Commands (Cont.)

Commands	Action
	The correlation coefficients are output in floating point form. The five digit decimal number to the right of the decimal point should be multiplied by 1Ø raised to the power of the number following the E (E is for exponent). The equation by which the correlation coefficient is reached is the following: Correlation coefficient of the i^{th} point = $\frac{\text{(raw value of the } i^{th} \text{ point)}}{\text{(raw value of the first point)}}$ In the above example printout, the correleoation coefficients can be interpreted as follows: 1st point / 1st point = +Ø.99999E+ØØ 2nd point / 1st point = +Ø.32827E+ØØ 3rd point / 1st point = +Ø.19863E+ØØ NOTE The correlation coefficients obtained from this routine when cross-correlation is performed bear no resemblance to the correct correlation coefficients. The routine is valid only for obtaining correlation coefficients from an auto-correlogram.

CHAPTER 2

CORRELATION MS

2.0 GENERAL DESCRIPTION

The correlation program that runs under OS/8 writes the correlogram to the system device and has the ability to read it back at a later time. The binary of the program is

DEC-8E-AACRA-PB

2.1 USE OF CORRELATION MS FOR OS/8

Auto and cross-correlation MS is the same as the paper tape system with the addition of writing data to and reading from the system device. The data written by the correlation program can be read by it at a later time.

Loading

The program's binary can be loaded via the Absolute Loader and started at location 2ØØ. To save the core image after the binary is loaded, use the SAVE command

SAVE SYS CORD3.SV Ø-3177,62ØØ-7577,174ØØ-17577;2ØØ

MS Command

CTRL/W is the command to read or write data. The OS/8 command decoder is loaded and requests a file. In response to the * an output file can be given to write the correlogram,

*FILE.XX<

or an input file can be given to read a previously written file.

*FILE.XX

NOTE

If a file other than one written by the correlation program is given on input, the program will crash.

After the file is given, the program reads or writes the file and returns to display mode with the correlogram on the display.

MS I/O Errors

An error to the command decoder causes an OS/8 error message. An error reading or writing the data, closing the file or filling the system device causes an error message to be printed. After an error OS/8 is running, restart the correlation program at 2556, field Ø.

Running with no Data Collection

If the program is to be run by reading a file and not collecting data, the following procedure can be followed. Start at location 200 and give dummy answers to the questions asked. Type CTRL/S followed by CTRL/Q. The program is now in output mode. Type CTRL/W to read in the file.

2.2 Data Format

The data written out is page Ø, the temporary buffer at 32ØØ followed by the correlogram.

Block 1	-	Page Ø, 1. Word 1 = 3, words 2 to 256 are locations 1 on.
Block 2	-	temporary buffer starting at 32ØØ in single precision followed by the correlogram in double precision.
Block 3, etc.	-	is written as needed to get all current data.

The temporary buffer contains the A/D readings, and the correlogram contains the summed product as described in Table 4-1.

The temporary buffer is N+1 words long and the correlogram is 2N words long where N is the answer to the setup question "No. of Points". The data in the correlogram is a double precision number, referred to as the raw data under the T command.

When a T command is given, the i^{th} value printed is

$$\#\ \text{printed}\ (i) = \frac{\text{Raw Data}(i) \times 4\emptyset 96}{\#\ \text{of Sweeps} - i}$$

The data in core which is written to the system device is "RAW DATA (i)".

BOOK FIVE

DAQUAN

CONTENTS

BOOK FIVE

CONTENTS

CHAPTER 1

USING DAQUAN

1.0 GENERAL DESCRIPTION

DAQUAN is used to acquire data by boxcar, multi-sweep time averaging from one instrument at a time and to display the results. A wide variety of subsequent processing techniques such as smoothing, differentiation, curve simulation, etc., are then used to reduce the data.

Input	=	one channel
Data rate	=	30 seconds/point to 8.2KHz (using Schmitt triggers)
Maximum Number of Points	=	2Ø47

DAQUAN is intended for the following applications:

- spectra comparison, stripping, and simulation
- Gaussian and/or Lorentzian fitting
- deconvolution of fused peaks (by Gaussian, Lorentzian or mixed technique)
- integration
- differentiation
- multiplication
- scaling
- plotting (standard X-Y recorder)
- peak detection and summary reporting of positions and percent areas.

Using DAQUAN's data handling commands, the researcher can interpret the data as required by his experiment in seconds. A sloping baseline can be aligned, a spectrum can be scaled, integrations can be performed, and two spectra can be compared simultaneously, each operation requiring only a single command.

An X-Y recorder can be interfaced parallel to a scope channel of the computer for accurate plotting of the results of any operation.

The applications of DAQUAN can include limited gas chromatography, slow mass spec, autoanalyzers, atomic absorption, infra-red, and U.V. spectrometers.

The output from DAQUAN is fully compatible with BASIC/RT so that the latter can be used for more sophisticated processing of the data acquired and/or processed by DAQUAN. It is possible to successively read 2Ø47 point blocks from paper tape, process the data with output to the high-speed punch and, when complete, read the output tape into the BASIC/RT program and perform a wide variety of subsequent analyses.

1.1 Hardware Requirements

The minimum configuration for using DAQUAN is:

> Advanced LAB8/E - 15 system
>
> An X-Y Analog Recorder is recommended for hard copy, using a relay driver controlled by the channel select bit of the display register (see Appendix Q).

1.2 Initial Starting Procedure

Both the RIM and Binary Loaders must be loaded before DAQUAN can be loaded. Refer to Appendix A for Loading Procedures for RIM and BIN.

To load DAQUAN use the following procedure:

1. Place the DAQUAN tape (DEC-8E-ADQNA-A-PB) in the proper reader (with leader/trailer under read head), either Teletype or photoelectric. NOTE: The binaries of DAQNFP and DAQUAN.8E are combined on the paper tape DEC-8E-ADQNA-A-PB.
2. Set the Switch Register (SR) to Ø and press EXTD ADDR LOAD. Set the SR to 7777 (the starting address of the BIN Loader). Press the ADDR LOAD switch and turn on the reader.
3. If using the high-speed reader, set SR bit Ø to Ø. Press CLEAR then CONT keys.
4. When the reader stops, DAQUAN is in core memory.

If the accumulator (AC) does not contain zero when the reader halts or if the tape does not stop on trailer code, a checksum discrepancy exists indicating that the DAQUAN tape has been read <u>incorrectly</u>. Load the tape again by repeating the procedure described above.

The OS/8 save locations for DAQUAN (paper tape) are 0-7577.

To start the DAQUAN program initially, set the SR to Ø2ØØ and press ADDR LOAD. Clear the Switch Register, press EXTD ADDR LOAD switch, next press CLEAR, then CONT keys.

DAQUAN indicates it has been successfully loaded into the computer by printing the message

DAQUAN LIVES!
TITLE:

Commentary of any length can be typed after TITLE. The information is not stored with the data and is used only to supply a titled paper copy record of data operations performed. Type CTRL/A to exit from the title phase. A command may now be issued (refer to Chapter 2). Refer to Appendix T for the core map of DAQUAN.

1.3 Inputting Data

Most calculations are performed with a Floating Point Package (refer to Chapter 8 of Introduction to Programming for a description of Floating Point), but final array results are single precision (12 bit) integers, normally scaled to the range Ø-1ØØØ$_{1Ø}$. When a numerical value is requested, most conventional formats are acceptable. Thus, for example, the decimal value 1Ø may be entered as 1Ø.Ø, 1Ø, 1E1, or .1E+Ø2, etc., as described in the above manual. After a numeric value has been typed in response to a command, any character except Ø to 9, E, or . terminates input for that entry. A space is recommended as the terminator. Typing the RETURN key does not automatically generate a line feed.) In response to questions, only Y or N is an acceptable answer. Any other response generates a question mark on the Teletype and is ignored. No terminator is required after a Y or N response.

In all cases, typing RUBOUT before a terminator deletes all input up to the preceding terminator and allows the correct value to be entered. A RUBOUT during numeric input echoes as an exclamation mark on the Teletype.

If, during scope display, a command unacceptable to DAQUAN is typed, a ? is printed on the Teletype and the program returns to the same scope display.

Because most DAQUAN commands replace the previous contents on the scope, it is recommended that any raw or interpreted data that may possibly be used at a later time be saved on tape (see PAPER TAPE command, Section 2.3) to prevent accidental elimination from the display channel.

1.4 Stopping and Restarting DAQUAN

If an operation must be halted immediately, press the console HALT switch. This should not be used haphazardly; if arrays were being modified, the data may be lost. Routines requiring input parameters or initial dialogue can be halted during that stage and before the input is complete. To restart DAQUAN after an emergency stop, set HALT switch to run position (switch in up position), set Switch Register to Ø2Ø1. Push ADDR LOAD switch, clear Switch Register and push EXTD ADDR LOAD switch. Next, push the CLEAR switch then the CONT switch. A new DAQUAN command can be issued when the display is restarted.

To Reassemble:

The sources for the paper tape DAQUAN system are:

DAQUAN.8E
DAQNFP

on the LAB8/E paper tape source DECtape, DEC-8E-ASPTA-A-UA4.

DAQNFP is the floating point package used by DAQUAN. To reassemble, each file should be assembled separately (with PAL8) to create the binary paper tapes. The floating point binary must be loaded (via the binary loader) before the DAQUAN binary.

CHAPTER 2

DAQUAN COMMANDS

2.0 INTRODUCTION

After starting DAQUAN (refer to Section 1.2), a scope buffer should be designated to accommodate the incoming data which is loaded into the computer's memory area directly from an analog input channel, the Teletype keyboard, and the high-speed or Teletype reader. Any of DAQUAN's commands can then be used to interpret the data. For example, the baseline can be linearized and the resulting spectrum integrated.

Each DAQUAN command is described in this chapter. The values requested, acceptable ranges, formulas used and available options for each are included in the discussion. After each command is completed, the correct spectrum is displayed (except the MODIFY command). Another command may be issued at that time.

Scope Buffer

DAQUAN provides three buffers for viewing data:

buffer 1 - assigned by user
buffer 2 - assigned by user
buffer 3 - buffers 1 and 2 simultaneously

Data supplied from the associated instrument or from paper tape is assigned to buffer 1 by typing the number 1 on the Teletype; similarly, data is assigned to buffer 2 by typing the number 2 on the Teletype. The choice of buffer 1 or 2 determines which data will be used. If a number between 4 and 9 is typed, buffer 1 is assumed. Typing Ø produces a response of question mark on the Teletype.

Each buffer can accommodate up to 2Ø47 data points. By typing 1 or 2 a buffer can be assigned to accommodate data when a new spectrum is being collected or when another DAQUAN command has been completed. The present contents of that buffer are displayed on the scope. When new data is assigned to a buffer which already contains data, it replaces the old data.

Buffer 3 permits the user to view the data in buffers 1 and 2 together. When buffer 3 is being used, analog knobs Ø and 1 control the X and Y offset for buffer 2 to allow comparison of data sets. It can be viewed at any time between commands, but should be followed by a 1 or 2 before calling a command. If left at buffer 3, the next command assumes buffer 2.

Most of the remaining DAQUAN commands are called by typing at least the first two letters of the command name and a colon after the termination of the preceding command. Refer to Appendix R for a summary of the commands. Commands such as AV, PA, CA and CO require specification of a channel, i.e., 1AV or 2CA.

2.1 AVERAGE

The AVERAGE command (AV:) implements time averaged analog input from the associated instrument. The parameters of the scan are specified on the Teletype in response to AVERAGE's questions and then that operation is performed. The data collected is displayed from the buffers specified before the AVERAGE command was initiated. Because of the need to store the intermediate sum in double precision format, if more than 1024 points are requested, storage will overlap the alternate display buffer and destroy any data kept there.

To collect data, proceed as follows:

1. Prepare the analog instrument for the experiment and connect the analog signal line to any available channel and the sync or trigger to Schmitt input 4.

2. Specify a scope buffer number (1 or 2) to accommodate the data.

3. Type AV: indicating an averaging operation is to be performed.

4. The computer prints:

    CHANNEL =

   requesting the analog channel number (decimal) to which the instrument is connected. Type in the analog channel number to be used by this experiment and press the RETURN key.

5. The computer responds with the message

    NO. POINTS =

Type in the number of points to be collected in the scan where Ø<POINTS<2Ø48 and then press the SPACE bar.

6. The next question is

   SEC/SCAN =

   Type in a value for the length in seconds of each scan where SEC/SCAN= seconds/point x points/scan. Remember that a minimum of Ø.123 millisecond/point and max of 30 sec/pt. is required. Press the SPACE bar. If the resulting rate is too high [(SEC/SCAN/NO. POINTS)<Ø.123 msec.], the routine restarts, asking for

   NO. POINTS =

7. When the rate is acceptable, the next question is:

   DELAY (SEC) =

   asking for the delay from receipt of the trigger pulse to the start of data acquisition for each scan. Values from Ø to 2Ø seconds are legal; resolution is 1Ø milliseconds. Press the SPACE bar after typing the value.

8. The message

   NO. SCANS =

   is printed last and the reply must be in the range 1 to 2Ø47. Type in the value and press the SPACE bar. Input will begin with the arrival of the first trigger pulse.

9. Activate the instrument to begin the experiment. The AVERAGE command will now collect the data according to the specified parameters.

The AVERAGER implements two display routines of its own, totally separate from the main scope routine of DAQUAN. This was necessary because of internal timing and intermediate data structure peculiar to the AVERAGER.

During data acquisition, the value for each point in the current scan is displayed as soon as it is available. Note that the X axis is not scaled to the number of points; thus, if more than 1Ø24 points were requested, the trace will wrap-around such that the 1Ø25th point will be displayed at the first location, the 1Ø26th at the second location, etc.

Between the end of each sweep and the trigger for the next, the current summed data is displayed. By typing C for check, subsequent triggers can be ignored. The Y amplitude can be magnified at this time by typing M; or decreased by typing D. Type P to proceed or S to stop the run. When S is typed the run is terminated, the number of scans completed to that time is indicated by the message

NO. SCANS =

the summed data is divided by the number of scans taken, and the average is displayed on the scope.

2.2 TIME

The constant corresponding to the computer's memory cycle time used to determine timing accuracy for averaging during data acquisition can be reset, by using the TI command. The AVERAGE routine can be calibrated as follows:

1. Disconnect all inputs to Schmitt triggers. A dummy run is made to measure the actual time required for a two minute sweep vs. the time based on an assumed memory cycle time of 1.2 microseconds. This is done by manually triggering the scan by flipping the Schmitt trigger polarity switch and measuring via a stopwatch the elapsed time until the static scope display appears after the sweep.

2. Call the AVERAGER by typing AV: then use the values CHANNEL Ø, 1ØØØ POINTS, 12Ø SEC/SCAN, Ø for DELAY, and 2 for NO. SCANS.

3. Time the program from when the Schmitt trigger slope switch is struck to when the static display begins, approximately 12Ø seconds later. Call this value T. The memory cycle time, MCT, is then determined by

$$MCT = T \times \frac{1.2 \times 10^{-6}}{120} = T \times 10^{-8} \text{ sec.}$$

4. Enter the desired value by typing TI: and supplying the value of MCT after the message

   MCT(SEC)=

2.3 PAPER TAPE

Data can be input to the computer from papertape via the high- or low-speed reader. The data[1] may have been punched directly by the analog instrument or may have been prepared by the DAQUAN command OUTPUT. The procedure for inputting data from the tape is as follows:

1. Type PA: to call the routine.

2. Answer the message

   NO. POINTS=

   with a value less than 2Ø48, not including skipped points.

3. The next message is

   PTS TO SKIP=

[1]DAQUAN accepts data in USA-ASCII format with and without spaces and/or non-numeric characters before a numeric value (each number must be properly terminated, as described in Section 1.3).

Type in the number of points to be skipped from the start of the tape before data is retained (less than 2Ø48) and press the SPACE bar.

4. Two parameters are requested by

   YRANGE & MIN:

   The range on the Y axis should be at least as large as the difference between the largest and smallest values on the tape. Type the Y-range, a comma, and the minimum value, which should be no larger than the smallest Y value in the data. Press the SPACE bar.

5. The last question is

   TTY I/O?

   Prepare the reader before replying because the tape will be read in immediately after the reply. Answer Y if the data is coming from the low-speed reader. A reply of N will start the high-speed reader.

2.4 CALCULATE

Spectra can also be calculated from the Teletype keyboard. DAQUAN requests information on the peaks, parameters, number of points, and baseline. (Refer to Appendix S for definition of the equations used.) The spectrum resulting from this calculation is displayed. Interaction from a peak that is more than eight times the half-width away from its position (or crest) is ignored. This gives a considerable saving in time for narrow width, multi-peak spectra. Note that all peak widths must be less than 1/4 x (X RNG) (step 4). For ease of input, the peak width for only the first peak is entered (step 5). See Appendix S if it is desired to change these conditions. The CALCULATE procedure is as follows:

1. Call the routine by typing CA:

2. Respond to the message

   NO. POINTS=

   with a number less than 2Ø48 and press the SPACE bar. Note that both buffers 1 and 2 are affected by this value; thus, if the CALCULATE operation is using buffer 1, only that number of points will be available for display of the data in buffer 2.

3. After the message

   NO. PEAKS=

   type the desired number of peaks, in the range Ø<PEAKS≤3Ø.

4. More data is requested by

   X RNG, X1, INT MPLR, BASE, LOR FR. (Ø-1):

   The first three parameters allow the user to define any coordinate grid in which to contain the calculated data. These parameters scale the data to a 1ØØØ x 1ØØØ point grid for compact integer storage. This does, however, impose a minimum on resolution, especially for peak position and width, defined as X RNG/1ØØØ. Type in the parameters: X range, initial X location, multiplier which is the Y axis scaler, the location of the base line on the scaled Y axis, and the Lorentzian fraction where Ø is a 100% Gaussian fit and 1 is a complete Lorentzian fit. The fraction must be in the range Ø.Ø to 1.Ø. A value of 1 for INT MPLR implies a range of Ø to 1ØØØ on the Y axis, a value of 1Ø implies a range of Ø to 1ØØ, etc. Base and peak heights should then fall within this scaled range. Peak positions should be in the range X1 to X1+X RNG. A negative X RNG is legal; e.g., for X1=5ØØ and X RNG= -5ØØ, the X axis runs from 5ØØ on the left to Ø on the right.

5. The dimensions of each peak, requested next by the message,

   HGT,H-WIDTH,POS:

   are height, half width at half height, and position on the scaled X axis. Type in each value followed by a comma or space and then press RETURN after the parameters for each peak. Thus, a two peak spectrum can include the data:

   678,5,35
   789,5Ø

   Enter a width for the first peak only. That value is assumed for all other peaks. A carriage return - line feed is issued when the parameters for each are complete.

6. After all the necessary data has been typed in, the computer indicates it is busy performing the calculation by the message

   COOL IT.

   on the Teletype. When calculated, the spectrum described by steps 2 through 5 is displayed on the scope.

After a display buffer has been selected and data has been brought into the computer's memory, any of the following commands can be issued. Any number of them can be requested and in any order. After the data has been interpreted by any of the available commands, it can be stored on tape for later use.

NOTE

> Because of parameter storage requirements, use of the CALCULATE routine will remove MODIFY (Section 2.22) from DAQUAN.

2.5 ALTER

The parameters specified by the most recent CALCULATE command can be modified by an ALTER command (AL:). The command is implemented as follows:

1. Type AL: to call the ALTER routine.

2. The message

   PK,PA,VL:

   is printed on the Teletype, requesting the peak index number. PK = peak number where the first peak typed in during CALCULATE is 1, the second is 2, etc. PA = parameter index where height is 1, half-width is 2 and position is 3. VL = new value for that parameter. Type the peak number, a comma or space, the parameter index, a comma or space, and the new value. Press the SPACE bar. All parameters except X RNG and X1 may be altered. The equal widths option has no effect at this time; any width may be altered. Consider the following sequence:

   PK,PA,VL: 12,3,674

   This series will position the twelfth peak at 674 on the scaled X axis.

Other spectral parameters can be modified by typing one of the sequences listed below and then the new value in response to the above message.

Sequence	Parameter to be Modified
Ø,1:	intensity multiplier
Ø,2:	baseline
Ø,3:	Lorentzian fraction
n,Ø:	print parameters for peak n
Ø,Ø:	print all parameters

The last two sequences exit to the display without recalculating the spectrum.

3. Type Ø,-1 and a SPACE to signal the end of alterations.

   The computer indicates it is recalculating the spectrum by printing the message

   COOL IT.

on the Teletype. When it is completed, the new spectrum is displayed.

2.6 COPY

The contents of the currently displayed buffer can be copied into the other buffer, leaving the two with identical data, by typing CO:. This is particularly useful if the manipulated data is to be compared with the original data.

2.7 XINVERT

The left to right X axis relationship of the displayed array can be inverted by typing XI:.

2.8 YINVERT

Each Y value of the displayed buffer is subtracted from 1ØØØ and the differences stored in that buffer, thus effectively inverting the Y axis by the command YI:.

2.9 SCALE

If, when the data is displayed on the scope, the maximum and minimum points "wrap around" because they are out of the scope's range or if the range is smaller than desired, SCALE (SC:) can be used to bring the Y data into the range $\emptyset \leq Y \leq 1\emptyset\emptyset\emptyset$. The scaling parameters used, original minimum value, original maximum value, and multiplier, are printed on the Teletype after the computation, as, for example:

```
MIN=  -54
MAX=  75
MPY=  7.752
```

where

$$MPY = \frac{1\emptyset\emptyset\emptyset}{(MAX-MIN)} \quad \text{and} \quad Y_i = (Y_i - MIN) \times MPY$$

NOTE

If the range between minimum and maximum exceeds 2047, the Multiply (not the Scale) Command must be used to adjust the data.

2.10 MULTIPLY

Data can be scaled to an arbitrary range other than the Ø to 1ØØØ range assumed by the SCALE routine. The MULTIPLY routine is used mainly in spectrum stripping when a standard and/or background spectrum is to be subtracted from a raw data set after appropriate scaling.
The command is used as follows:

1. Type MU: to call the MULTIPLY routine.

2. The parameter

   MIN=

   is then requested. This value is to be subtracted from each data point before MPY is applied.

3. When

   MPY =

   is printed, type in the desired multiplier. Then the following computation is performed:

   $$Y_i = (Y_i - MIN) \cdot MPY$$

   The final data should be in the range -2Ø47 to $2Ø47_{1Ø}$. Values outside of this range are truncated to these limits. Final values less than -5 or greater than 1Ø19 will "wrap around" when displayed.

2.11 SMOOTH

Collected data can be smoothed by an 11 point digital filter routine via the SMOOTH command (SM:). The new spectrum is displayed on the scope. The first and last five points are not altered. See Appendix S for functional equation.

2.12 CURSORS

Typing CU: displays two movable bright dots on the scope referred to as cursors. They are used to implement the INTEGRATE and STRIP commands by delimiting the data points to be modified. The two cursors are controlled by four analog channel knobs as follows:

CURSOR	Analog Channel knob	Direction of movement
left	Ø	horizontal
	1	vertical
right	2	horizontal
	3	vertical

When the two dots appear on the display initially, knobs Ø and 1 must always position one of them to the left of the other.

A grid of $1ØØØ_{1Ø}$ x $1ØØØ_{1Ø}$ on the scope is assumed. The actual range extends slightly beyond, but the data points should be within that range.

Typing CU: a second time removes the cursors from the scope.

Note that the cursors can be used to inspect peak amplitudes, valleys, widths, etc., by positioning them appropriately and noting the values, using the LIST command.

2.13 LIST

To get a Teletype listing of the decimal values of the cursors, type

LI:

The values are printed in the following order: X, Y left; X, Y right.

2.14 INTEGRATE

The INTEGRATE command (IN:) provides two options:

1. Integrate the data between the cursors by using them to set a pivot or baseline, print out the area, and leave the data array unchanged.
2. Integrate all data using the cursors (extrapolated) to define a pivot line and store a scaled running integral in the data array, then print out a scale down multiplier.

The data is defined as lying on a 1ØØØ x 1ØØØ grid, independent of the total number of points in the array. The area at a given point, i, is expressed as

$$A_i = \sum_{i=s}^{j-1} (X_{i+1}-X_i) \; [(Y_{i+1}-P_{i+1}) + (Y_i-P_i)] \; /2$$

where $X_\emptyset = \emptyset$, $X_n = 1\emptyset\emptyset\emptyset$ and s is the starting point which is equal to the X value of the left cursor for option 1 or is equal to Ø for option 2. Similarly, $j \equiv$ x value of the right cursor and P_j $(=P_i)$ is the value of the pivot line at that point.

An integration is performed as follows.

1. Type CU: and position the cursors as required for the desired option (refer to Section 2.12).
2. Type IN: to call the routine.
3. The message

   SCAN INT'L?

   is printed. Type Y if a running integral (option 2) is desired; type N is a partial integral and area printout (option 1) are desired.

If, when the routine is first called, the message

BAD X POINTS!

is printed, the left-right relationship of the cursors has been inverted. The display is restarted to allow correct positioning. When corrected, type IN: to call the routine again.

2.15 STRIP

Three options are available for altering portions of the displayed spectrum.

1. Replace the data between the cursors with the best straight line.
2. Subtract a sloping baseline from all data points.
3. Subtract a straight line interpolated between the cursors from the data between the cursors.

The STRIP command is implemented as follows:

1. Type CU: and position the cursors as explained below.

2. Type ST: to call the STRIP program.[1]

3. The message

   STRIP PEAK?

   is printed on the Teletype and the two dots are displayed on the scope. If some of the data points are to be replaced by the best straight line, position the left dot at the first data point to be replaced, and the right dot at the last data point to be replaced. Then type Y and the delimited portion of the spectrum is replaced by a straight line. The STRIP program is exited. Continue with another DAQUAN operation. If one of the other STRIP operations is desired, type a response of N.

4. A reply of N generates the message

   FULL BASE?

   If a (sloping) baseline is to be subtracted from all the data points, the two dots should have been positioned on the scope with the desired slope and amplitude. The dots need not be on the data curve nor on the X axis extrema because they are extrapolated. Type Y and the new display will appear. If a straight line is to be subtracted from some data points, i.e., partial baseline restoration is desired, type N.

If, when the STRIP routine is called, the message

BAD X POINTS!

is printed, the left and right dots have been inverted. The original display with the cursors appears. Correct the dots by adjusting the knobs; and type ST: to call the routine again.

2.16 DIFFERENTIATE

The derivative curve of a spectrum is computed by the DIFFERENTIATE command (DE:). Derivatives to a depth of at least six can be calculated for some spectra with minimal distortion of the data. These are produced, not by adjacent point differences, but by the following procedure.

[1] If STRIP is called without calling CURSORS first, stop and restart the program (refer to Section 1.4), call the CURSORS and then STRIP routines.

$$Y_i = Y_{i+2} - Y_{i-2}$$

with

$$Y_i \equiv Y_2 \equiv Y_3 \text{ and } Y_{n-2} \equiv Y_{n-1} \equiv Y_n$$

If multi-depth derivatives are desired, scaling and smoothing are suggested before each level to reduce quantization error or "stairsteps" that result from integer arithmetic. It may also be necessary to strip out the first and last five points if they interfere with scaling (they are unaffected by smoothing).

2.17 PLOT

Any displayed spectrum can be plotted on an X-Y recorder interfaced to a scope channel for a hard copy of that spectrum using the PLOT command (PL:). The size and rate of the plot are controlled by the user; the pen is controlled by a relay driver triggered by the channel select bit of the display enable register. See Appendix Q for details. To generate a plot of the presently displayed spectrum, proceed as follows:

1. Type PL: to call the plotting routine.

2. The first message printed on the Teletype is

   LINE PLOT?

   Type Y if a line plot is desired or type N if a point plot is preferred.

3. The plotting routine is now waiting for the user to calibrate an area on the X-Y recorder. A small dot appears on the scope and will make the same movements as the plotter pen. The speed for drawing the axes, as well as for the actual plotting, is controlled by knob 3 of the analog channel controls and can be adjusted at any time while using the PLOT command. Turning knob 3 clockwise increases the rate of plotting; turning it counterclockwise decreases the rate. The axes are calibrated by typing the following letters to perform the indicated operations.

Letter	Operation
X	locates maximum X coordinate
Y	locates maximum Y coordinate
O	returns pen to X-Y origin

   The pen on the X-Y recorder should be set initially to the origin. Type X and then manually reposition the pen to the desired maximum X coordinate. After the $X_{max}Y_{\emptyset}$ position has been determined, type Y. Set the pen to the maximum Y coordinate ($X_{max}Y_{max}$) similarly. The axes have now been determined for the plot. Type O to move the pen automatically back to the origin ($X_{\emptyset}Y_{\emptyset}$).

4. The axes and quadrant markers of the graph can now be marked off on the plot if desired. Type M to mark off the frame and quadrants. Knob 3 controls the pen speed.

5. When the axes have been marked, the spectrum is ready to be plotted. Type G to initiate the plot. As before, analog channel knob 3 is used to adjust the plotting speed.

   Any characters other than X, Y, O, M or G during this sequence produce a question mark on the Teletype and are ignored.

6. When the spectrum has been plotted, the following message is printed.

   PLOTTER OFF?

   The plotter should be turned off or set to stand-by and then Y typed. (N is also a legal reply.) Typing any other character returns a ?. If the plotter is not turned off, the pen will start to move wildly in its effort to follow the scope analog signals when the display starts.

2.18 SUBTRACT

Using the SUBTRACT command (SU:), data in the displayed buffer will be subtracted from data in the other buffer. The result will appear in the displayed buffer replacing the previous data.

2.19 ADD

The two buffers can be averaged and the result seen in the displayed buffer by issuing an ADD command (AD:). As with SUBTRACT, the data in the displayed buffer will be lost when the ADD command is executed.

2.20 SWAP

The data currently in buffer 1 can be placed in buffer 2 and the data in buffer 2 placed in buffer 1 with the SWAP command (SW:). This facility is especially useful with ADD and SUBTRACT.

2.21 SQUEEZE

The SQUEEZE command (SQ:) compresses the data array by a factor of 2 by averaging adjacent data points. It is recommended that this command be used immediately after data input because the number of points is halved for each issuance of this command. If the current buffer is 1 or 2, only that buffer is affected, but if both buffers are being displayed (3 was typed during display), both are halved. The purpose of this command is to reduce the number of points in order to diminish flicker and allow faster smoothing, scaling, etc.

2.22 MODIFY

The DAQUAN program itself can be modified by a routine that is similar to ODT-8 (DEC-8-COCO-D) which makes changes in core locations. (See Note on CALCULATE routine (Section 2.4). MODIFY is not available if CALCULATE was used.) After calling (with MO:) the MODIFY routine, any of the following can be performed in a logical sequence. Each must begin at the left margin.

open location	Type the location (octal) and press the SPACE bar. The contents of that location are printed followed by a colon.
change contents	After the present contents of a location have been printed, type the new contents and the letter C to enter the correction and press the SPACE bar.
inspect next location	After pressing the SPACE bar, type N. The next location and its contents are printed followed by a colon.
reinspect same	After pressing the SPACE bar, the contents of the last location opened can be printed again by typing S.
transfer control to	Type R and the location and press the SPACE bar. Execution of the program will continue from that location.

Pressing RUBOUT at any time will terminate the operation and restart the routine.

2.23 OUTPUT

Collected data can be listed and/or output on paper tape by using the OUTPUT command (OU:). After calling the routine, the question

TTY I/O?

is asked. A response of Y implies the Teletype punch, and a response of N implies the high-speed punch. Be sure the device is prepared before responding to the question. The format is 1Ø columns of four digit integers.

2.24 RESTART

The DAQUAN program can be restarted from the beginning at any time. The message

DAQUAN LIVES! or (under OS/8) DAQUAN OS-8!

is printed. This command (RE:) is equivalent to restarting at Ø2ØØ as described in Section 1.2.

2.25 PEAK Report

The PEAK processor will request X axis scaling information (to allow setting up X as time, wavelength, mass units, furlongs, etc.) and a detection sensitivity parameter. Then a pass is made through the stored data to detect the peaks and integrate between the minima found. Finally, a summary report is printed containing the minima, maxima, their scaled X value, the scaled raw area between minima, the intensities normalized to the largest peak of 1ØØ, and the % area. The data array is left unaltered and can be subsequently processed as desired.

Before calling the PEAK processor routine, the user should call the CURSOR (Section 2.12) and STRIP (Section 2.15) routines to remove any sloping or non-linear baseline deviation and to remove spurious points. SMOOTHing is also advantageous. It is not necessary to set the baseline down to zero, only that there be a constant offset. This is corrected for when areas and intensities are created.

Type

PE:

DAQUAN prints

X1,XMPY,GATE:

requesting the X value of the first data point for X1, the X axis multiplier or delta X for XMPY, and the detection GATE. The detection algorithm requires a GATE's worth of consistently increasing data points followed by a GATE's worth of consistently decreasing values. This does not imply consecutive changes, just that there be no major deviations from the trend. Thus, the GATE is usually set to about 1/3 to 1/5 the number of points over the narrowest peak in the data with values of 3-6 being typical. Small values give added sensitivity whereas large values allow small or narrow peaks to remain undetected.

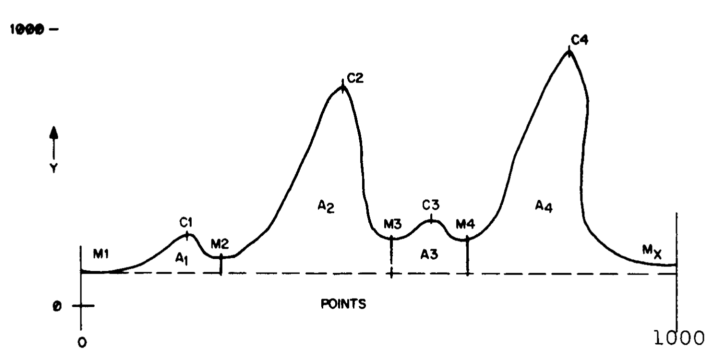
Type RETURN after specifying these three values. The computer prints:

```
MIN,X,MAX,X,A,I,%A:
TTY I/O?
```

The top line labels the subsequent columns where MIN, the first column, is the leading minimum value for the peak, the second column is its scaled X value. Similarly, columns 3 and 4 are the maximum and X value, then column 5 is the scaled raw area, column 6 the intensity normalized to the largest peak with a value of 1ØØ, and finally, column 7, the percent area. Areas and intensities are computed based on the lowest or minimum point in the data array which is thus defined as baseline.

The TTY I/O question requests the output device to be used. A response of Y specifies Teletype punch; N specifies high-speed punch. Typing any other character causes a ? to be printed. Be sure the device specified is prepared before responding to the question.

For example, this display



when processed by PEAK would report the four peaks with leading minima, maxima, and area as M_1, C_1, A_1, M_2, C_2, A_2, etc. with areas computed above the horizontal baseline through M_1, the lowest point, using perpendicular line drop at the valleys.

I.e.,

$$A_j = \sum_{Pt.\ M_j}^{Pt.\ M_{j+1}} (Y_i - M_1) \cdot XMPY$$

where M_x is the lowest point between C_4 and the last data point. The normalized intensities would be computed as

$$I_j = \frac{C_j - M_1}{C_4 - M_1} \cdot 100$$

CHAPTER 3

DAQUAN EXAMPLE

The printout on the left side of the page is an actual DAQUAN sequence. The commentary on the right was added to indicate the operation performed. Underlined information on the left is that typed by the user.

DAQUAN LIVES!	PROGRAM IS READY
TITLE:DEMO1	NAME THIS SEQUENCE "DEMO1"
OK, HIT ME!	READY FOR A COMMAND. BUFFER 1 IS ASSUMED.
1CA:	CALCULATE SPECTRUM.
NO. POINTS=1ØØØ	1ØØØ POINTS
NO. PEAKS=5	5 PEAKS
XRNG, X1, INT MPLR, BASE, LOR FR (Ø-1): 1ØØØ,Ø,1,Ø,Ø	PARAMETERS FOR SPECTRUM
HGT, H-WIDTH,POS:	
1ØØ,5,1ØØ	SPECIFY WIDTH FIRST PEAK ONLY
2ØØ,2ØØ	SUBSEQUENT PEAKS ASSUME SAME WIDTH
3ØØ,3ØØ	
4ØØ,4ØØ	
5ØØ,5ØØ	
COOL IT	SPECTRUM BEING COMPUTED
CO:	COPY DATA ONTO BUFFER 2
1AL:	ALTER SPECTRUM ON BUFFER 1
PK,PA,VL:1,1,5ØØ	ALTER HEIGHT OF FIRST PEAK TO 5ØØ
PK,PA,VL:2,2,1Ø	CHANGE HALF WIDTH OF PEAK 2 TO 1Ø
PK,PA,VL:3,3,8ØØ	CENTER THIRD PEAK AT LOCATION 8ØØ
PK,PA,VL:Ø,-1	ALL THE CHANGES REQUIRED
COOL IT	SPECTRUM BEING COMPUTED
1SQ:	HALVE DATA POINTS ON BUFFER 1
1CU:	DISPLAY CURSORS (POSITION CURSORS)
1IN:	INTEGRATE BUFFER 1
SCAN INT'L? N	PARTIAL INTEGRAL REQUESTED
AREA= Ø.11Ø3Ø9E+Ø6	AREA UNDER CURVE BASED ON 1ØØØ X 1ØØØ GRID
SM:	SMOOTH DATA

CO:	COPY DATA INTO BUFFER 2
CU:	DISPLAY CURSORS (POSITION CURSORS)
ST:	STRIP OUT A
STRIP PEAK? N	
FULL BASE? Y	(SLOPING) BASELINE
MU:	SCALE DATA
MIN=-52	WITH THIS MINIMUM AND THIS
MPY+.5	MULTIPLIER (AXIS IS HALVED)
21AD:	ADD DATA IN DISPLAY BUFFER 2
OU:	OUTPUT THIS DATA
TTY I/O? N	ON HIGH-SPEED PUNCH
2AV:	ACQUIRE DATA IN DISPLAY BUFFER 2
CHANNEL=1	INPUT FROM BUFFER 1
N. POINTS=1ØØØ	1ØØØ POINTS,
SEC/SCAN=1Ø	1Ø SECONDS/SCAN
DELAY(SEC)=1Ø	1Ø SECOND DELAY,
NO. SCANS=3	3 SCANS MAXIMUM
AV:	CALL AVERAGER TO TIME MEMORY CYCLE
CHANNEL=Ø	USE THESE PARAMETERS
NO. POINTS=1ØØØ	
SEC/SCAN=12Ø	
DELAY(SEC)=Ø	
NO. SCANS=1Ø	TIME THE PROGRAM FROM WHEN THE SCHMITT POLARITY SWITCH IS STRUCK TO WHEN STATIC DISPLAY BEGINS.
1TI:	CHANGE TIME OF THIS PROGRAM
MCT(SEC)=1.22E-6	SET CONSTANT TO 1.22 MICROSECONDS
12SW:	VIEW AND THEN SWAP BUFFERS
SU:	SUBTRACT BUFFER 2 FROM BUFFER 1
1PL:	PLOT SPECTRUM ON X-Y RECORDER
LINE PLOT? Y	LINE PLOT
OXYOM	CALIBRATE AXES THEN MARK BOX AND QUADRANTS
G	PLOT
PLOTTER OFF? Y	DONE PLOTTING
RE:	RESTART DAQUAN
DAQUAN LIVES!	PROGRAM IS READY
TITLE:DEMO2	NAME THIS SEQUENCE "DEMO2"
OK, HIT ME!	READY FOR A COMMAND
PA:	READ IN FROM PAPERTAPE:
NO. POINTS= 6ØØ	6ØØ POINTS
PTS TO SKIP=Ø	START WITH FIRST POINT
YRANGE & MIN: 4ØØ,Ø	4ØØ IS RANGE OF DATA WITH MIN VALUE OF Ø
TTY I/O? N	VIA HIGH SPEED READER
XI:	INVERT LEFT-RIGHT RELATIONSHIP ON X-AXIS
YI:	INVERT Y-AXIS

```
PE:                                    WANT PEAKS PROCESSED WITH FIRST
XI,XMPY,GATE:1 1 3                     POINT HAVING AN X VALUE OF 1,
MIN,X,MAX,X,A,I,%A:                    DELTA X OF 1, AND A SENSITIVITY OF 3

TTY I/O?Y                              TO BE PRINTED ON TELETYPE

  448,   1.ØØ,   5Ø6,   3.ØØ,  Ø.6222ØØE+Ø4,   52.22,  15.Ø7Ø8,
   13,  22.ØØ,    81,  28.ØØ,  Ø.358ØØØE+Ø3,    7.2Ø,   Ø.8671,
   44,  32.ØØ,   957,  53,ØØ,  Ø.13459ØE+Ø5,  1ØØ.ØØ,  32.6ØØ2,
  935,  56.ØØ,   943,  67.ØØ,  Ø.12584ØE+Ø5,   98.52,  3Ø.48Ø8,
  16Ø,  8Ø.ØØ,   799,  97.ØØ,  Ø.8662ØØE+Ø4,   83.26,  2Ø.981Ø,

DE:                                       CALCULATE FIRST DERIVATIVE OF
                                          DISPLAYED DATA
```

CHAPTER 4

DAQUAN MS FOR OS/8

4.0 GENERAL DESCRIPTION

DAQUAN MS is DAQUAN DEC-8E-ADQNA-A-PB changed to run under OS/8. DAQUAN MS can be loaded by OS/8 and returns to OS/8 via a CTRL/C. In addition, data can be read and written via mass storage (system device). To accomplish this the CALCULATE command has been taken out. Hence, the commands CALCULATE and ALTER are not present. A new command MA: (mass storage) has been added.

The binary for MS DAQUAN is DEC-8E-ADAQA-A-PB, or on DECtape, DEC-8E-ALMSA-A-UB, the binary file is DAQD5.BN.

4.1 Loading DAQUAN MS

DAQUAN MS can run with the EAE or non-EAE floating-point package. The two sources that can be used are

DEC-8E-NEAEA	-	EAE floating-point package
DEC-8E-NFPPA	-	non-EAE floating-point package

The floating-point binaries provided with DAQUAN are:

Paper-tape:	DEC-8E-ADFEA-A-PB	-	EAE Floating-Point Package
	DEC-8E-ADFPA-A-PB	-	non-EAE Floating-Point Package
	DEC-8E-ALMSA-A-UB:		
DEC-tape	FPPNE.BN	-	non-EAE floating point
	FPPEAE.BN	-	EAE floating point

Reassembling the Floating-Point Sources:

The EAE package is ready to be assembled. However, the non-EAE sources must be edited before assembly. Edit the origin statement

```
*FLPT-26ØØ
```

to read

```
*FLPT-2ØØØ
```

Delete the origin statement *FLPT-25ØØ.

Reassemble the new source with PALIII or PAL8. The source of the floating-point package is distributed on three paper tapes (DEC-Ø8-NFPPA-A-PA1, -PA2, and -PA3). Only tapes 2 and 3 are used and should be assembled with PAL III or equivalent in the order they are numbered (tape 2 first, then 3). Tape 2 ends with a PAUSE statement while tape 3 ends with a dollar sign. Tape 1 is the extended functions which are not used.

If the package is assembled with PAL 8, the user must define several floating-point and several 8/E instructions which are used by the interpreter, but not contained in the PAL 8 symbol table. The user should create a paper tape which looks like:

```
FIXMRI  FADD=1ØØØ
FIXMRI  FSUB=2ØØØ
FIXMRI  FMPY=3ØØØ
FIXMRI  FDIV=4ØØØ
FIXMRI  FGET=5ØØØ
FIXMRI  FPUT=6ØØØ
FEXT=ØØØØ
FNOR=7ØØØ
SWP=7521
CAM=7621
MQA=75Ø1
MQL=7421
SGT=6ØØ6
PAUSE
```

and use it as the first tape of the assembly (before tape 2) of the package.

To load DAQUAN, use the Absolute Loader as follows. In response to the asterisk, type

```
*FPP.BN,DAQD5.BN
```

where FPP.BN is the binary of the floating-point package being used and DAQD5.BN is the binary of DAQUAN.

The OS/8 save locations for the floating-point package and DAQUAN MS are 0-7577.

<u>Starting Address</u>

Start DAQUAN at location 2ØØ of field Ø, restart at 2Ø3 and start at 2Ø2 to reenter the display routine. Do not restart at 2ØØ.

4.2 Using DAQUAN MS

Under OS/8 DAQUAN MS prints DAQUAN OS-8! when started. All commands are the same as paper tape DAQUAN except that CALCULATE and ALTER are not present. To write or read from mass storage (system device), the MA: command is given. The OS/8 command decoder is called to get the input or output file. In response to the asterisk, type the standard OS/8 reply

    *FILE.xx

for an input file, and

    *FILE.xx<

for an output file. If an output file is given, the current channel being shown is written to the system device. The format of the data on disk is a core image, the first word of the data starting in the first word of the mass storage file.

If the file is an input file, two more questions are asked:

    PTS. TO SKIP=

and

    NO. POINTS=

Respond with a number less than 2Ø48 followed by the RETURN key for each question. If more than 2Ø47 points need to be skipped, the file can be reconfigured by the CONVERT program.

4.3 Data Format

Data stored on mass storage from DAQUAN is in single precision, the first word on mass storage is the first word of the channel buffer that is written. Data read by DAQUAN is single precision data. The values should be between Ø and 1ØØØ. If not, the MULTIPLY command should be used on the data to scale it down.

4.4 Errors

Any errors in the command string to the command decoder cause an OS/8 error message. An error with I/O writing or reading, or closing of the file causes an error message to be printed. Check to see if there is room on the system device and try again. Restart DAQUAN at 2Ø2.

CHAPTER 5

DAFFT OVERLAY FOR DAQUAN MS

DAFFT runs with DAQUAN MS and the EAE floating point package DEC-8E-NEAEA. DAFFT (Data Acquisition and FFT) contains all functional aspects of DAQUAN MS except that:

1. the MODIFY function is removed.
2. the SQUEEZE function is removed.
3. the number of points is limited to 1Ø24.

It includes these added functions:

1. Performs complex FFT from single-precision data to give complex single precision results.
2. Performs complex inverse FFT similarly.
3. Computes scaled power spectrum.
4. Allows temporary storage of 1Ø24 values in a third buffer area.
5. Has the option of 3-point or 11-point digital filtering in time domain and Hanning filtering in frequency domain.

DAFFT allows signal averaging in the time domain (up to 1Ø24 double-precision points), FFT into frequency domain giving up to 1Ø24 real and 1Ø24 complex coefficients (half of which are in positive and half in negative frequency), computation of the power versus frequency spectrum as the scaled sum of the squares of those coefficients, input and/or output of any of these arrays using the Teletype or high-speed reader/punch, and allows retention of an extra 1Ø24 point array in core for recall of data from a previous step.

The software package consists of the DAQUAN MS, EAE floating point package and DAFFT OVERLAY binaries. The DAFFT OVERLAY contains a copy of DECUS 8-144, J. W. Rothman's FFTS-C, and the modifications and additions to DAQUAN MS. Also incorporated is the patch for FFTS-C to correct a roundoff error.

5.1 Loading

To load DAFFT, the Absolute Loader is used. In response to the asterisk, type

```
*FPPEAE.BN, DAQD5.BN, DAFFT3.BN
```

where FPPEAE.BN is the binary of the EAE version of the floating-point package DEC-8E-NEAEA. DAQD5.BN is the binary of DAQUAN MS followed by the binary of DAFFT. The OS/8 save locations for DAFFT are: 0-7577,10000-11575. The start location is 2ØØ of field Ø, the same as DAQUAN.MS. If no mass storage is available, the programs can be loaded by the binary loader and started at 2ØØ. The program then runs as long as CTRL/C or the MA: command is not given. The binary of DAFFT is DEC-8E-ADAFA-PB.

Refer to Appendix T for core maps of DAFFT.

5.2 Operation

Since MODIFY and SQUEEZE are not available, requesting them causes type-out of a question mark. Similarly, do not at any time request more than 1Ø24 points. Also, any array upon which FFT or IFFT is to be done must be some power of two in length, i.e., 4, 8, 16...256, 512, or 1Ø24 points. If the array is not a power of two, the program halts and must be reloaded with OS/8. Also, any array to be transformed must reside in buffer 1.

Relationship to Acquisition

The FFT of 2^N real or complex input points yields 2^N complex coefficients, half of which are positive frequency and half of which are negative frequency. Most applications use only positive frequency, therefore, 2^{N-1} points result from the transform. For a desired final result of Ø-f max bandwidth, the input should be band limited at twice fmax and sample rate should equal twice fmax to prevent aliasing and to allow accurate frequency recovery.

In order to remove DC offset from the data, before computing the FFT the mean value of the array is subtracted from each datum. This "zero mean" correction forces the integral of the array to zero.

5.3 Command Description

Table 5-1 summarizes the new commands.

Table 5-1

DAFFT Commands

Command	Action
FT:	Calculate FFT of complex time domain data. Real coefficients are in buffer 1 and imaginary coefficients are in buffer 2 before calling.
ZT:	Calculate FFT of real valued time domain data. Real coefficients are in buffer 1. Buffer 2 is set to zero before transform.
IT:	Calculate inversed FFT of complex frequency domain data. Real coefficients in buffer 1 and imaginaries in buffer 2.
PO:	Compute power versus frequency spectrum as sum of squares of real and imaginary coefficients. Buffer 1 contains real coefficients and buffer 2 contains imaginary coefficients before calling.
SH:	Shift to or from buffer 1 and storage buffer.
SM:	Perform Hanning, 3-point, or 11-point filtering.
OF:	Move the display's x axis from the bottom to the middle or from the middle to the bottom of the scope. This enables plus and minus y values to be displayed with zero at mid-scopt.

FT and ZT create arrays including both positive and negative frequencies. Zero frequency is set at left margin with increasing values to the most positive frequency in the center of the scope. The most negative frequency is adjacent to that and remainder of display is negative frequency going to zero again on the right margin. The power display is analogous.

If an FT or ZT command is used to obtain an FFT of the data and then an IT command is used to return to the original data, the result of the IT operation must be multiplied (not scaled) so that the data displayed represents the original data.

After data is entered (by AVERAGE or PAPER TAPE commands) ensure that the real coefficients are in buffer 1 and imaginary coefficients (if any) are in buffer 2. The real and imaginary coefficients are output or input as separate arrays.

Shift Command

Typing

SH:

causes

SAVE?

to be printed. Answer Y if the data in buffer 1 is to be placed in a storage buffer. Answer N if the data in the storage buffer is to be loaded into buffer 1. Type the RETURN key after either response, to continue execution.

SMOOTH Command

A choice of two time domain filters or a frequency domain filter is available. When

SM:

is typed, the message

H,3,11(Ø,1,2):

is printed, requesting a value of Ø to execute Hanning smoothing of the real and imaginary Fourier coefficients stored in buffers 1 and 2, respectively. The expression used is:

$$Y_i = -1/4 \cdot Yi-1 + 1/2 \cdot Y_i - 1/4 \cdot Y_{i+1}$$

It is designed to remove fractional frequencies and to enhance resolution in the frequency domain.

A response of 1 causes the data in the currently-displayed buffer to be smoothed with a three-point filter defined as follows:

$$Y_i = 1/4 \cdot Y_{i-1} + 1/2 \cdot Y_i + 1/4 \cdot Y_{i+1}$$

Responding with 2 executes the 11-point filter as described in Chapter 2 (Section 2.11) and Appendix S.

Complete the desired response with the RETURN key.

FFT Commands

Typing FT:, IT:, or ZT: causes the action described in Table 5-1. When the transform is completed, DAFFT prints:

FACTOR= xx

where xx is a power of two by which the stored values are to be multiplied to get the actual Fourier coefficients. All stored values were divided by that power of two to prevent overflow of the single-precision format. The coefficients are inferred to be fractions between +1 and -1 but for all intents and purposes can be considered as integers between +2Ø48 and -2Ø48. Then the actual Fourier coefficient is found as

$$FC_i = (Y_i/2048) \cdot 2^{FACTOR}$$

Since DAFFT, for purposes of display, assumes a number range of Ø to 1ØØØ, display of the transforms will show scope wraparound. This is allowed in order to retain maximum resolution of the data. If desired the arrays can be scaled down using the MULTIPLY command (Section 2.10) with a MIN of 2ØØØ (or 1ØØØ) and MPY of Ø.25 (or Ø.5). This should be done for both the real array in buffer 1 and the imaginaries in buffer 2. Use of the SCALE command is discouraged since that routine assumes a data range of +1Ø24 to -1Ø24 and the FFT or IFFT gives a range of +2Ø48 to -2Ø48. However, if the power spectrum is to be computed, it is recommended that no scaling be done. This then allows the full resolution to be used.

POWER Command

To compute the power spectrum from the stored real and imaginary coefficients, type

PO:

DAFFT then requests

FACTOR=

Enter a value from 6-12 followed by the RETURN key. This is to allow scaling to the desired range. The larger the FACTOR the smaller the result. The power coefficient is computed as a 24-bit value then divided by 2^{FACTOR}. Use of a large FACTOR allows keeping all power coefficients on scale whereas a smaller FACTOR allows presentation of the weak power components. Only the low order 12 bits are retained.

The expression for computing power is

$$P_i = (Y_{1i}^2 + Y_{2i}^2) \cdot 2^{-FACTOR}$$

where Y_{1i} is the real coefficient (in buffer 1) and Y_{2i} is the imaginary coefficient (in buffer 2). The power array replaces the data in buffer 1.

5.4 DAFFT Example

```
DAQUAN LIVES!
TITLE: DAFFT DEMO (↑A)

OK, HIT ME!

AV:                       ACQUIRE 10 SCANS OF 1024 POINTS AT 1024HZ FROM
CHANNEL= 2                CHANNEL 2 WITH NO PRESCAN DELAY.

NO. POINTS= 1Ø24
SEC/SCAN=1
DELAY(SEC)=Ø
NO. SCANS= 1Ø

SH:
SAVE?Y                    SAVE TIME AVERAGED DATA IN STORAGE BUFFER.
21ZT:                     ZERO OUT COMPLEX COEFFICIENT ARRAY AND COMPUTE FFT.
FACTOR= 4                 DATA WAS AUTOMATICALLY DIVIDED BY 16 (2^4).
21PO:                     COMPUTE POWER SPECTRUM WITH FACTOR OF 2^-9.
FACTOR= 9
1SH:                      SHIFT STORAGE BUFFER BACK TO BUFFER 1, I.E.,
SAVE?N                    RESTORE TIME AVERAGED DATA.
1ZT:
FACTOR= 4                 COMPUTE FFT AS ABOVE
SM:
H,3,11(Ø,1,2)=Ø           APPLY HANNING FILTER
PO:
FACTOR= 9                 COMPUTE POWER SPECTRUM AS ABOVE.
```

CHAPTER 6

PAFFT OVERLAY FOR DAQUAN MS

PAFFT (Power Average by FFT) has all the features of DAFFT (Chapter 5) but instead of time domain averaging, frequency domain averaging of the power spectrum is done. This is accomplished by acquiring a synced sweep of data in time, resettling to a zero mean, doing the FFT, applying Hanning filtering, computing power, adding to the double precision power average, and returning to allow the next sweep. A 1Ø24 average requires about four seconds dead time per sweep (set by FFT time). PAFFT is also limited to 1Ø24 points and does not have the MODIFY or SQUEEZE functions.

6.1 Loading

The Absolute Loader is used to load PAFFT. In response to the asterisk, type

```
*FPPEAE.BN, DAQD5.BN,PAFFT2.BN
```

where FPPEAE.BN is the binary of the EAE version of the floating-point package DEC-8E-NEAEA, DAQD5.BN is the binary of DAQUAN MS followed by the binary of PAFFT. The OS/8 save locations for PAFFT are: 0-7577, 10000-11575. The start location is 2ØØ of field Ø, the same as DAQUAN MS. If no mass storage is available, the programs can be loaded by the binary loader and started at 2ØØ. The program then runs as long as CTRL/C or the MA: command is not given. The binary of PAFFT is DEC-8E-APAFA-PB.

Refer to Appendix T for core maps of PAFFT.

6.2 Operation

The AVERAGE routine as described in Chapter 2 (Section 2.1) was modified by inserting the FFT and power average computations after each sweep. Also, when the run is complete the number of points for display is halved. This will present only the positive frequency power spectrum.

Also, when AVERAGE is requested and after NO. POINTS= (power of 2) is specified, PAFFT requests

FACTOR=

The response is a value of 6-12 followed by RETURN. This FACTOR is analogous to that described in Chapter 4 (Power Command).

PAFFT requires using buffer 1 for the sweep acquisition, both buffers 1 and 2 for the Fourier coefficients, and the storage buffer to retain the double-precision power average. Therefore, no other arrays, preceding data, etc., can be contained in core when a power average is being computed.

When the average is terminated the double precision power average is divided by the number of sweeps taken, truncated to the limits of ±2Ø47, and stored in buffer 1.

6.3 Command Description

The PAFFT commands are the same as the DAFFT commands described in detail in paragraph 5.3. Table 5-2 summarizes the new commands.

Table 5-2

PAFFT Commands

Command	Action
FT:	Calculate FFT of complex time domain data. Real coefficients are in buffer 1 and imaginary coefficients are in buffer 2 before calling.
ZT:	Calculate FFT of real valued time domain data. Real coefficients are in buffer 1. Buffer 2 is set to zero before transform.
IT:	Calculate inversed FFT of complex frequency domain data. Real coefficients in buffer 1 and imaginaries in buffer 2.
PO:	Compute power versus frequency spectrum as sum of squares of real and imaginary coefficients. Buffer 1 contains real real coefficients and buffer 2 contains imaginary coefficients before calling.
SH:	Shift to or from buffer 1 and storage buffer.
SM:	Perform Hanning, 3-point, or 11-point filtering.
OF:	Move the displays x axis from the bottom to the middle, or from the middle to the bottom of the scope. This enables plus and minus y values to be displayed with zero at mid-scope.

6.4 PAFFT Example

DAQUAN LIVES! TITLE: PAFFT DEMO (↑A)	
OK, HIT ME!	
AV: CHANNEL=Ø	ACQUIRE 1024 POINT POWER AVERAGE WITH SCALE FACTOR OF 8 AND 10 SCANS MAXIMUM. FINAL ARRAY IS 512 POINTS WITH 0 - 170 HZ BANDWIDTH (1024/3 ÷ 2)
NO. POINTS= 1Ø24 FACTOR= 8 SEC/SCAN= 3 DELAY(SEC)= Ø NO. SCANS= 1Ø	
NO. SCANS= 4	AVERAGE STOPPED AT FOUR SCANS
SC:	SCALE DATA TO SIZE FOR PLOTTING
MIN= Ø MAX= 2Ø47 MPY= Ø.489	
SM: H,3,11(Ø,1,2):1	APPLY 3 POINT FILTER
PL: LINE PLOT?Y G PLOTTER OFF?Y	MAKE A LINE PLOT OF DATA
AV: CHANNEL= 2	ACQUIRE POWER AVERAGE WITH SCALE FACTOR OF 9 TO GIVE 512 POINTS WITH BANDWIDTH OF 0 - 512 HZ
NO. POINTS= 1Ø24 FACTOR= 9 SEC/SCAN= 1 DELAY(SEC)= Ø NO. SCANS= 1Ø	

B O O K   S I X

C O N V E R T   P R O G R A M

CONTENTS

BOOK SIX

TABLES

CHAPTER 1

LAB8/E DATA CONVERSION PROGRAM

The conversion program is used to change data from one format to another. It reads a file from mass storage and writes the new file on mass storage. The program accepts single, double, or floating-point data, FORTRAN IV and FORTRAN II data, and data from the LAB8/E programs. The output of the data file can be single, double, or floating point, FORTRAN IV or FORTRAN II format. The binary tape of this program is DEC-8E-ACVTA-A-PB. The program name on DECtape is COND14.BN(DEC-8E-ALMSA-A-UB) and COND14(DEC-8E-AMASSA-A-UA1).

1.1 Loading and Starting

The CONVERT program can be loaded by the Absolute Loader and starts at location 2ØØØ of field 1. To run the program, type .R ABSLDR to OS/8; in response to the asterisk, answer

```
*CON/G=12ØØØ$
```

where CON is the binary of the CONVERT program and $ represents an ALT MODE. The OS/8 save locations are: 12ØØØ-17177.

1.2 Commands

Once the CONVERT program is loaded and started, it asks two questions: the FILE question and the CONVERT question.

FILE Question

The OS/8 command decoder is used to request the files. In response to the asterisk, type

```
*OUTFIL<INFIL↵
```

where OUTFIL is the new file being created and INFIL is the old file. If a file already exists with the name of the output file, it will be deleted when the new file is closed.

For example, the command

```
*FILE1<FILE1↵
```

takes FILE1 and converts it to the new format with the name FILE1. When the new FILE1 is closed, the old file is deleted. The standard OS/8 rubout format is active and CTRL/C takes the program back to OS/8. If any errors are detected, the command decoder outputs an OS/8 error message and reasks the question. The files are specified in OS/8 format and can be read from and written to any OS/8 device that handles binary data.

CONVERT Question

After the files are entered, the program prints an asterisk. Respond with the CONVERT type in the format

```
*XX<YY↵
```

or

$*XX<YY/Z_1/Z_2...$

where XX is the output type and YY is the input type. Z_1, Z_2, etc., are conversion switches and are discussed later. After the command is given the program changes the file and closes the new file. When it returns to process another file an asterisk is printed on the Teletype for file input.

A RUBOUT deletes the last character typed and types a backslash. Successive RUBOUTs delete the next preceding characters.

Switches

The five switches, H, T, D, S, and R, which can be specified to the CONVERT question are described in Table 6-1.

Table 6-1

Switches

Switch	Explanation
/H	Used only with Histogram input (PST or TIH output). If present, each value of the histogram is divided by 2. This is used because the output of the histogram program is a positive 12-bit number, which would be processed as negative if bit Ø were set.
/T	Allows an ID TITLE to be put on the output file from CONVERT. The TITLE is written in OS/8 packed ASCII in block 1 of the file. Block 2 will then start the data. The ID TITLE is terminated with a CTRL/Z, so that PIP can list the TITLE on the printer or Teletype and return to OS/8. When a /T is given, the question ID= is printed. Respond with the title. A RUBOUT deletes the last character. Successive RUBOUTs delete preceding characters. A RETURN, LINE FEED must be typed to start a new line. Up to 255 characters can be input. CTRL/Z ends the title input. /T can not be used with the S switch.
/D	Used to delete block 1 from the input file. Block 1 should be an ID title block but does not have to be one. /D allows the title to be changed on a file. /D cannot be used with the S switch and can not be used with AV, CO or HI input.
/S	Used to skip the first block of the input file. This allows a titled file to be converted without changing the title. /S cannot be used with the /D or /T switches nor with CO, AV or HI input.
/R	Used to delete any number of numbers from the front of the file and to write a desired number of numbers in the output file. When /R is given, two questions are asked: SKIP #= and READ #= In response, the number of numbers to be skipped and the number of numbers to be read are entered and terminated with the RETURN key. The new file then consists of the number of numbers read. /R can be used only on SI or DO input.

Error Conditions

The four types of errors which may occur are:

Message	Meaning
USER ERROR 3	An incorrect command was given.
USER ERROR 2	A mass storage read or write error occurred.

Message	Meaning
USER ERROR 1	The output device is full.
USER ERROR Ø	Error occurred during a create, close or in the handler.

1.3 File Types

Input

F4	FORTRAN IV Floating-Point format
F2	FORTRAN II Floating-Point format
DO	Double precision
SI	Single precision
DA	DAQUAN data
FS	FORTRAN II single precision data
HI	Histogram (TIH, PST) mass storage output
AV	Advanced and Basic Averager mass storage output
CO	Correlation mass storage output

Output

F4	FORTRAN IV Floating-Point format
F2	FORTRAN II Floating-Point format
DO	Double precision
SI	Single precision
DA	DAQUAN data
FS	FORTRAN II single precision data

Note that FORTRAN IV is RTPS FORTRAN and FORTRAN II is OS/8 8K FORTRAN.

Convert Chart

	Output					
Input	F4	F2	DO	SI	DA	FS
F4	-	Y	Y	Y	Y	-
F2	Y	-	Y	Y	Y	-
DO	Y	Y	Y	Y	Y	Y
SI	Y	Y	Y	Y	Y	Y
DA	Y	Y	Y	Y	Y	Y
FS	Y	-	Y	Y	Y	-
HI	Y	Y	Y	Y	Y	Y
AV	Y	Y	Y	Y	Y	Y
CO	Y	Y	Y	Y	Y	Y

where Y = yes and blank means no.

1.4 Input Data Formats

AV, HI, and CO are the output of the averagers, histograms and correlation programs, as written to mass storage by the mass storage WRITE command. The format of the data is described in the write-up of those programs.

F4 input is the output of FORTRAN IV or OS/8 BASIC. The data is 3-word floating-point numbers, 85 data points to a 256-word mass storage block. The last word of the block is not used. The floating-point data has the format, word 1 is a 2's complement exponent. Words 2 and 3 are a 24-bit 2's complement mantissa with the decimal point between bits Ø and 1 of word 2 when the number is normalized.

F2 input is the output of OS/8 FORTRAN II floating-point numbers. The numbers are 3-word floating-point numbers written under an A6 format and then packed in standard OS/8 ASCII packing, three ASCII characters per two words. Hence, a 3-word floating-point number occupies four words of the mass storage block. A maximum of 63 data points can be stored per block.

The data must be written in A6 format, and an example of a WRITE statement is:

```
     WRITE (4,1Ø) X
 1Ø  FORMAT (12ØA6)
```

where X is an array of 120 points. Each X value would be written in ASCII and packed,

WORD 1	1	2
WORD 2	3	4
WORD 3	5	6

→

3H	1
3L	2
6H	4
6L	5

where 1 is the 6-bits, Ø-5, of word 1 and is stored as 8-bit ASCII in word 1 of output. The six bits of 3 are changed to 8-bit ASCII, then the top four bits go in word 1 (3H) and the low four bits in word 2 (3L). At the end of each write there is an 8-bit packed ASCII 215,212. The next write starts with the next 8-bit ASCII character.

An F2 floating-point number is 3 words, 27 bits of mantissa, 8 bits of exponent, and a sign bit as follows

<table>
<tr><td>A</td><td>B</td><td>C</td></tr>
<tr><td colspan="3">D</td></tr>
</table>

A is the sign bit, B the exponent, C the high-order 3 bits and D the low 24 bits of mantissa. The value is expressed as sign magnitude times an exponent. The exponent is stored as exponent -2ØØ octal, and Ø is expressed as all zeros.

DO, Double precision, input is assumed to be 24 bit 2's complement numbers, with the high order first followed by the low order. There are 128 double precision numbers to a block.

SI, single precision, input is 12 bit 2's complement numbers, 256 numbers to a block.

DA is single precision numbers from DAQUAN.

FS is single precision numbers from OS/8 FORTRAN II. The numbers must have been written in an A2 format by FORTRAN. Like F2 output, the 6-bit ASCII is packed as 8-bit ASCII by OS/8, the record ended by an 8-bit packed ASCII 215,212. The next record follows immediately. An exmaple of a write by FORTRAN II is

```
      WRITE (4,1Ø)IX
  1Ø  FORMAT (12ØA2)
```

where IX is an array of 120 points.

NOTE

> The output of CONVERT in F2 and FS formats can not be later converted by the CONVERT program. Each block output by CONVERT has two dummy characters at the end and as a result is not the same as the output format of F2 and FS data.

1.5 Output Data Formats

F4 output is the same as input. Data is written 85 floating-point numbers per block until the input file is exhausted.

F2 output is written in OS/8 packed 8-bit ASCII so that it is readable under A6 format by OS/8 FORTRAN II. Each block of output contains up to 63 floating-point numbers. The last four words of the block are used to fill out the block and end the record. Hence, the longest record is 63 values. The format of the output block is

Words 1 - 252	63 floating-point values in A6 format
Words 253 - 256	Ø, Ø, Ø, Ø, 215, 212 in packed 8-bit ASCII

If the last block is not full the above four words follow the last floating-point number. To read 200 data points of converted data, the following FORTRAN II statements can be used,

```
    DO 1 I=1,3
    READ (4,1Ø) (X(J+(I-1)*63),J=1,63)IX,IY
1Ø  FORMAT(63A6,2A2)
1   CONTINUE

    READ (4,11) (X(I),I=19Ø,2ØØ)

11  FORMAT(11A6)
```

Note that IX and IY are dummy variables to read the two single-precision numbers at the end of the data before the end of record mark.

DO, double precision, output is the same as for DO input.

SI, single precision, output is the same as for SI input.

DA, DAQUAN, output is single-precision numbers.

FS, single precision, output for FORTRAN II is in OS/8 packed 8-bit ASCII in A2 format. There are 189 numbers per block. The last four words of the block are the same as for F2 output. If the last block of output is not full, the last number is followed by the four words to mark the end of records as in F2 output. To read 200 words of single-precision output, the following FORTRAN II statements can be used.

```
    READ (4,1Ø)(IX(I),I=1,189),IY,IZ
1Ø  FORMAT (191A2)
    READ (4,11)(IX(I),I=19Ø,2ØØ)
11  FORMAT (11A2)
```

NOTE

IY and IZ are dummy variables to read the two single precision dummy values at the end of the block.

1.6 Histogram CONVERT Output

The output of the histogram program is page Ø and two histograms. The histogram values are single precision 12-bit positive numbers. However, if the convert output is to be used by the DAQUAN program or is single precision data, the histogram values are treated as a 2's complement number. The H switch on the convert type causes the histogram data to be divided by two before conversion to ensure all positive results.

NOTE

Only the numbers in the histogram are divided, not any other output of the histogram program.

Output of Histogram Input

Parameters	1st Number - number of points in zeroth histogram
	2nd Number - zeroth histogram bin width
	3rd Number - number of points in histogram
	4th Number - bin width of histogram
	5th Number - min. time
	6th Number - is the ORDER for PST (unimportant value for TIH)
Values	7th Number - is the first value of the zeroth histogram
	8th Number - is the second value of the zeroth histogram followed directly by the histogram

The values accepted by the histogram program during setup go up to 4Ø95, hence if a value was given that is greater than 2Ø47 and the histogram data is converted to single precision, the input values will be treated as 2's complements and converted. If the output is double precision or floating-point, the setup value will always be positive. The output of the setup value is:

	INPUT TO HIST.	OUTPUT OF CONVERSION
ORDER	Ø to 4Ø95	Ø to 4Ø95
BIN WIDTH	.1 to 4Ø9.5	1 to 4Ø95
BINS	Ø to 189Ø	Ø to 189Ø
EPOCHS	Ø to 189Ø	Ø to 189Ø
MIN TIME	Ø to 4Ø94	Ø to 4Ø94
TIME/UNIT	.1 to 4Ø9.5	1 to 4Ø95
UNITS	Ø to 189Ø	Ø to 189Ø

The histogram numbers range from Ø to 4Ø95 as output of the histogram program. If the numbers are not scaled by /H, they will be treated as 2's complement numbers if converted to single precision but will always be positive Ø to 4Ø95 if converted to double precision or floating point.

1.7 Basic and Advanced Average CONVERT Output

The Averager program writes out the job lists followed by the Average data. The job lists are seven words long and describe the average. The average data point can have three different formats: average alone; average and confidence limits; or average, confidence limits and trend. The average and confidence limit are in double precision and the trend is single precision. The high-order part of the average and confidence limit is the integer part and the low order the fraction. When the data is typed out by the Averager program in Section 4, with a scale factor of Ø, the high order plus one bit of low order is typed for the average and the confidence limit.

If the output of the CONVERT program for the Averager is single precision or DAQUAN, the high order part of the average and confidence limit is output. The trend is single precision to begin with. For double precision or floating-point output, both the high and low parts are converted by the CONVERT program without outputting the decimal point. The CONVERT program treats this number as a 24-bit integer while converting it; the user must divide the result by 4096 to find the average integer.

The output file format is as follows:

1st Number = 1 if low resolution average
Ø if high resolution average

2nd Number = Channel number Ø and up

3rd Number = Data type =1 if Average only
=2 if Confidence limits
=3 if Average, Confidence Limit and Trend

4th Number = Sort code

5th Number = Number of points in the average

6th Number = Number of sweeps

7th Number = Data points
etc.

Following the data of an average will be the first number of the next average or -1 if no more averages.

A data point is as follows for Type 3 data:

Average value
Confidence Limit
Trend

and for Type 2 data:

Average Value
Confidence Limit

For Type 1 data it is:

Average

If the output is going to DAQUAN the data is rearranged so that all the average values are together followed by the confidence limits and then the trends, if they exist, as follows:

1 - 6th	numbers	=	average data header
7th	number	=	number of sweeps
8 - Xth	numbers	=	average data
$X+1-Y^{th}$	numbers	=	confidence limits, if they exist
$Y+1-Z^{th}$	numbers	=	trends, if they exist
$Z+1^{th}$	number	=	first number of next average, or -1 if no more averages.

1.8 Correlation CONVERT Output

The output of the Correlation programs is page Ø, the temporary buffer and the correlogram. The temporary buffer starts at the first word of block 2 and is N+1 words long, where N is the number of points.

The output of the conversion program is as follows:

1st Number	=	zero if Auto, -1 if Cross-correlation
2nd Number	=	RATE input to program, 1 to 2Ø47 represents 1 to 2Ø4.7
3rd Number	=	number of points
4th Number	=	sweep count
5th, 6th	=	dummy numbers
7th Number etc.	=	correlogram

The sweep count is in double precision and the correlogram data is in double precision; hence, if the output is in single precision, the low order part of the correlogram and sweep count is obtained. However, if the output is in single precision format, the sweep count low-order is in word 4 and the sweep count high order in word 5. If N is the number of points collected for the correlation, the N numbers starting at number 7 are the correlogram. The correlogram numbers which are converted are the double precision core values of the correlation program. The value typed by the correlation program under the /T command is

$$\text{VALUE(i)} = \frac{\text{RAW DATA of } i^{th} \text{ point}}{\text{(number of sweeps)} - i} \times 4\emptyset96$$

The data which is output by the CTRL/W command in the correlation program and then converted by the convert program is the "RAW DATA of i^{th} point".

B O O K S E V E N

B A S I C / R T

CONTENTS

TABLES

CHAPTER 1

USING BASIC/RT

INTRODUCTION

BASIC/RT provides the researcher with a powerful and complete software package for total programmable control of all LAB8/E peripheral devices: analog channels, Schmitt triggers, crystal clock and display scope. In addition, commands to specify parameters such as pulse rate and response time permit optimum experimental flexibility. Buffer allocation commands for the display and analog channels facilitate computer efficiency. Using the timing commands, a delay before sampling, a pause until a user response is typed on the Teletype, or a pause until an interrupt occurs, can easily be included in a program. Another feature, the user's command, permits customized system software.

All of the features provided by 8K BASIC are also implemented in BASIC/RT, permitting total programmable I/O control and user coded functions. The entire BASIC/RT command and function set is included in Appendix U of this manual. For additional information on BASIC fundamentals, refer to the EduSystem-10 System User's Guide; for details on 8K BASIC refer to the 8K BASIC User's Manual, DEC-Ø8-SKXA-D. BASIC/RT does not run under the OS/8 system.

Loading Procedure

Use the following procedure to load BASIC/RT:

1. Load the RIM Loader into field 1. Either the high- or low-speed reader can be used. Set the data and instruction fields to 1 (use the EXTD ADDR LOAD key) before toggling in the first instruction. Note that after pressing DEP or EXAM, the LAB8/E always indicates the Program Counter location, not the Memory Address location, in the upper row of lights. To see what content has just been loaded, set the multiple register knob to MD (Memory Buffer).

2. Load the BIN Loader into instruction field 1. (Use the EXTD ADDR LOAD key to load in this field setting.)

Steps 1 and 2 can be replaced by loading the HELP Bootstrap.

3. Read in the BASIC/RT paper tape. Use data field 1 and instruction field 1. When loading via the BIN Loader, press START CLEAR then START CONT keys to initiate reading of the tape. When program is loaded, it prints the message READY.

1.0 SCOPE COMMANDS

The display scope on the LAB8/E can be programmed to plot points on its screen. The scope commands provide complete control for graph location and size, display time and number of points displayed.

1.1 PLOT

When plotting on the scope, which is rectangular, BASIC/RT uses these dimensions for its perimeter:

$\emptyset \leq X < 1.1$
$\emptyset \leq Y < 1.\emptyset$

Thus, any plot must be within the above limits, which can easily be accomplished by inserting a scaling factor.

The PLOT command causes the appropriate point to be displayed on the scope. It is issued in the format:

```
PLOT x,y
```

where x and y are any expressions which equal the actual X and Y coordinates of the point to be plotted. An acceptable sequence for plotting a straight line across the middle of the scope is:

```
 5  B=.5
10  FOR A=0 TO 1.1 STEP .01
20  PLOT A,B
30  NEXT A
```

Remember that every X,Y set must be within the specified ranges. If it is not, that data set is simply not displayed.

BASIC/RT displays points on the scope when it is not doing any internal calculations. Data is displayed, for example, while waiting for input or output. If a calculation is required during a plotting sequence, the data is not displayed until all the calculations are completed. Thus, in the following example, plotting a decaying sine wave, the function is not displayed until all the points have been calculated.

```
200  FOR S=0 TO 1.10 STEP .006
210  PLOT S, SIN(S*35)*EXP(-S*2.5)/3+.5
220  NEXT S
```

1.2 DELAY

BASIC/RT provides a command that refreshes the scope after each calculation so that the progress of the graph can be seen. This command, DELAY, causes BASIC to display all x,y points calculated up to this statement. Thus, the decay of the sine wave above can be viewed after each point is calculated by adding to the above example the command

```
215  DELAY
```

The DELAY command provides the additional time for BASIC to display the point before continuing to the next statement.

1.3 CLEAR

BASIC/RT also permits erasing the scope under program control. By inserting the statement:

```
CLEAR
```

all points currently displayed are removed from the screen. Thus, if more than one plot is required by a user program and it is not necessary for them to overlap, a CLEAR command between calculations erases the scope for the second plot.

In the next example, two compounded interest sums, $400 at 7% and $450 at 6.25% per annum compounded yearly for 30 years, are plotted. If the intercept point is to be noted, then line 12Ø can be omitted and at completion the two curves will be displayed together.

```
1ØØ  I=.Ø7
1Ø2  P=4ØØ
1Ø4  T=3Ø
11Ø  GOSUB 5ØØ
12Ø  CLEAR
13Ø  I=.Ø625
134  P=45Ø
14Ø  GOSUB 5ØØ
15Ø  STOP
5ØØ  FOR N=1 TO T
51Ø  X=N/35
52Ø  Y=(P*((I+1)↑N)/4ØØØ)
53Ø  PLOT X,Y
54Ø  DELAY
55Ø  NEXT N
56Ø  RETURN
```

Lines 51Ø and 52Ø include scaling factors for the scope.

1.4 Display Buffer

The technique for plotting points employed by BASIC/RT includes creating a buffer in the user's area of core to store all the calculated points before they are displayed. (This buffer area is considered as a dimensioned variable; thus, executing the commands SCRATCH, RUN, and END removes the buffer from core. The CLEAR command does not delete the buffer, it merely erases its contents.) When a PLOT command is encountered, BASIC/RT checks to see if a display buffer has already been assigned and, if not, then space sufficient for about 500 points is allocated. If this amount of room is not available in core, the error message TOO BIG is printed.

1.5 USE

The area created by the PLOT command is approximately equal to a 333-dimensional array. In order to conserve space, if less than 500 points are to be plotted, or to plot more than 500 points, a buffer dimensioning command is provided so that core can be allocated optimally. This command, USE x, is implemented as an array, as follows (x is always a variable):

```
20  DIM P(200)
30  USE P
```

Line 30 says : use P as a storage buffer for a future PLOT command; do not create an additional array at that time. Line 20 creates enough room for about 300 data points[1]. If a user-assigned or BASIC/RT generated buffer is not large enough and overflows during execution the error message TOO BIG is printed.

Only one USE statement is effective for any plot sequence and, if it is to be used, it must be issued before the PLOT command. The variable associated with the USE command is active until one of the buffer removing statements (RUN, SCRATCH, END) is encountered, but it can be used for another purpose when not currently required for displaying.

In the example in Section 1.3, the maximum number of points to be plotted is 30, so a considerable amount of core can be conserved by specifying a smaller buffer. The following code can be added to optimize the program.

```
10  DIM M(40)
20  USE M
```

[1]To determine the minimum array size for point displays, consider that each point requires two values (X and Y) and that every three values require one word of memory. To display P points an array dimensioned for W words is necessary:

$$W=.67(P)$$

CHAPTER 2

CLOCK COMMANDS

2.0 GENERAL DESCRIPTION

The hardware clock provided in the LAB8/E, like all the other devices, can be handled under program control to maximize its performance, by specifying parameters such as pulse rate and initial pulse time.

2.1 SET RATE

To set the clock to interrupt at a specific rate, use the command

```
SET RATE mode,time
```

where mode[1] is the desired clock speed (Ø-7) and time is the number of clock "ticks" between interrupts, up to 4Ø96 counts. The appropriate mode value is derived from the next chart[2].

mode	rate
Ø	Stop
1	external input
2	1Ø$^{-2}$ seconds
3	1Ø$^{-3}$ seconds
4	1Ø$^{-4}$ seconds
5	1Ø$^{-5}$ seconds
6	1Ø$^{-6}$ seconds
7	Stop

Thus, for the clock to interrupt at 1 second intervals, an acceptable command is

```
SET RATE 3,1ØØØ
```

which causes the clock to wait 1000 one msec. ticks. If the specified clock rate is too fast, so that the interrupt cannot be serviced in time, the error message RATE ERROR AT (line number) is printed and the clock stopped. The line number printed is that of the statement currently being executed.

---

[1]If an illegal mode is requested, BASIC tries to use the value: -3 will be taken as 3, .Ø3 as Ø, etc.

[2]Refer to DK8-EP Clock Description in the PDP-8/e Small Computer Handbook.

Note that in the 100-200 μsec elapsed total time range BASIC/RT is servicing the interrupt correctly, but is not executing any BASIC/RT commands because of the high rate; in this case, the processing has been suspended. Type CTRL/C, which will cause the RATE ERROR message to be printed, to restart.

2.2 SET CLOCK

BASIC/RT provides another command for setting the clock rate for the Schmitt triggers. Its format is the same as that for the SET RATE command, namely

SET CLOCK mode,time

except that mode is a 12-bit decimal number which will be used to load the clock enable register[1], thus permitting the user to enable any function he chooses. Refer to Table 1 to determine the appropriate value. The time parameter is specified in the same manner as with the SET RATE command[2].

Any time either of the SET statements is encountered, the time counter is zeroed. Then, any time a clock interrupt occurs, this counter is incremented. Up to approximately 16,000,000 counts can occur before this counter resets itself to zero.

2.3 TIM

At any time in the program, the current count (number of elapsed interrupts) can be determined via the function TIM(n).

This function can be used in conjunction with any of the BASIC commands so that the value can be printed or the next action to be performed by the program can be dependent on the count. The format of the function call is TIM(n) where n is any argument (the argument is not checked by BASIC/RT). In the following program the elapsed time for the plot determines the next action; print the count and halt or, for 50 data points, print the sines and the terminating count and then stop.

In the first run, the time elapsed. By changing line 1Ø, the sine table was generated.

---

[1]BASIC/RT always forces the interrupt request bit on.

[2]The 2's complement of the value of time is performed so that it is in a form suitable for the clock.

```
10   A=10                          RUN
20   SET RATE 2,20                 10
30   FOR M=0 TO 1 STEP .01
40   PLOT M,M↑2                    READY.
50   DELAY
60   NEXT M
70   IF A>TIM(0) THEN 200          10 A=12
80   PRINT TIM(23)                 RUN
90   STOP                           0
200  FOR Z=0 TO 1 STEP .02          .01999867
210  PRINT SIN(Z)                   .03998933
220  NEXT Z                            .
230  PRINT TIM(C)                      .
240  END                               .
```

2.4 WAITC

Another application of the clock is to halt program execution until a clock interrupt occurs. The WAITC command performs this function, thereby permitting BASIC/RT to display on the scope while waiting for the interrupt to signal continuation of program execution.

2.5 CLS and CLC

There are two other functions that are used with the Schmitt triggers. The first, CLS(n), obtains the status bits of the clock after the previous interrupt (those obtained from the PDP-8 CLSA (Clock Status to AC) instruction). This value can be tested, for example, to determine how the interrupt occurred. The other, CLC(n), performs a CLCA (Counter to buffer to AC) and returns the value as a floating point number. Refer to PDP-8/e Small Computer Handbook.

Table 7-1

Clock Enable Register Functions

AC Bit	Function
∅	Enables clock overflow to cause an interrupt.
1 & 2	Mode ∅∅ Counter runs at selected rate. Overflow occurs every 4∅96 counts. Flag remains set. ∅1 Counter runs at selected rate. Overflow causes Clock Buffer to be transferred to the Clock Counter, which continues to run. Overflow remains set until cleared with IOT 6135. 1∅ Counter runs at selected rate. When an enabled event occurs, the Clock Counter is transferred to the Clock Buffer, and the Counter continues. 11 Counter runs at selected rate. When an enabled input occurs on channel 3, the Clock Counter is transferred to the Clock Buffer, and the Clock Counter continues to run from zero.
3, 4 & 5	Count Rate 111 Stop 11∅ 1 MHz 1∅1 1∅∅ KHz 1∅∅ 1∅ KHz ∅11 1 KHz ∅1∅ 1∅∅ KHz ∅∅1 External input ∅∅∅ Stop
6	Overflow starts ADC. (When the Clock Counter overflows, the analog-to-digital converter, type AD8-EA, is started.)
7	When set to a 1, the Crystal Clock is inhibited from generating clock pulses that increment the counter.
8	Events in Channels 1, 2, or 3 cause an interrupt request.
9, 1∅ & 11	Enable Events 1, 2, and 3 9 - Event 3 1∅ - Event 2 11 - Event 1

CHAPTER 3

A/D COMMANDS

3.1 ADC

Any A/D channel can be sampled at any time by using the single function ADC(n), where n is the channel number, Ø to 15, to be sampled in a direct or indirect statement. The ADC function performs an immediate conversion; the clock, however, can be incorporated so that sampling occurs at an established clock rate. In the next program, BASIC/RT waits for a clock tick and then prints the value of the clock using the TIM function, and of A/D channels 3 and 4 for fifty samples.

```
3ØØ  SET RATE 2,6Ø
31Ø  FOR P=1 TO 5Ø
32Ø  WAITC
321  A1=ADC(3)
322  A2=ADC(4)
323  T1=TIM(Ø)
33Ø  PRINT T1,A1,A2
34Ø  NEXT P
35Ø  STOP
```

3.2 REAL TIME

The ADC function is restricted to non-time critical work because the finite amount of time elapsed between clock ticks may not be sufficient to perform the tasks requested between ticks (for example, printing three values above). Also, more than one channel cannot be sampled in the same time quanta (for example, sampling channels 3 and 4 above).

For time critical operations, the REAL TIME command should be used because it provides a buffer to hold the sampled value prior to processing. Its format is:

```
REAL TIME v,c1,n1,n2
```

where v is a subscripted variable to be used as the data buffer. The variable is assigned in a manner analogous to the USE command for the scope, namely, as a dimensioned array. Because only one value is to be taken per sample, three samples are stored per buffer word. Thus, a dimension of 100 can store 300 data items. The array cannot be dimensioned larger than approximately 750, or a maximum of approximately 2,200 points. The parameter C1 is the first channel to be sampled, nl is the number of consecutive channels to be sampled, and n2 is the number of clock ticks for which to sample. To prepare to sample channels 1 and 2 once every millisecond for 150 milliseconds, suitable code is:

```
SET RATE 4,1Ø
DIM G(1ØØ)
REAL TIME G, 1,2,15Ø
```

Operation of the REAL TIME command is independent of the BASIC/RT statement processing speed; as long as there is sufficient buffer space, the REAL TIME statement will work.

3.3 ACCEPT and REJECT

The REAL TIME statement only creates the specified data buffer. To actually initiate sampling, the statement ACCEPT is required. Then sampling will start at the next clock tick. There must be an active REAL TIME statement or the ACCEPT is ignored. A REAL TIME command becomes inactive when the clock count equals zero. To suspend sampling use the command REJECT. This command is also useful for executing subsequent REAL TIME statements.

In the next example, the ACCEPT and REJECT statements are used to be sure sampling occurs at the specified rate for only the designated number of counts. Statement 500 stops the clock, 530 prepares it so that the first sample is taken after processing statement 540. The REJECT at 560 assures that after 100 counts no extra samples are taken at the rate of 10 mseconds.

```
5ØØ  SET RATE 7,Ø
51Ø  DIM G(1ØØ)
52Ø  REAL TIME G,1,2,15Ø
53Ø  ACCEPT
54Ø  SET RATE 2,1Ø
55Ø  IF TIM(4)<1ØØ THEN 55Ø
56Ø  REJECT
57Ø  SET RATE 3,1Ø
58Ø  ACCEPT
59Ø  END
```

3.4 Buffer Capacity

In the last REAL TIME example, the buffer, G, is subscripted to accommodate 3*1ØØ=3ØØ items. The program, however, could have been required to sample a greater total number of points. If the 520 was REAL TIME G, 1,6,150 then a total of 6*150=900 items must be accommodated. To handle this variation without having to allocate large amounts of core as buffer space, BASIC/RT uses a ring buffer so that data can be removed from the buffer before it is overwritten by a new, incoming item. To meet this condition, the sampling can be:

1. At a slow rate for a long time (consistent time sampling) or

2. At a fast rate for a short time.

If a fast rate is required for a longer period, a large buffer can be created, provided this space is not required by BASIC/RT for variables, etc. If the buffer becomes completely full such that data is being overwritten, the message A-D FULL is printed. The maximum throughput rate is about 4 KHz or 250 μs.; the steady state long term rate is about 50 points/second.

3.5 ADB

To retrieve data collected by REAL TIME and ACCEPT sequences that are placed in a buffer, the ADB(n) function is required. Data is withdrawn from the buffer in the same order in which it was entered. Thus, if four A/D channels, 1-4, are being sampled, the order of the data in the buffer is

$$1_i 2_i 3_i 4_i \quad 1_{i+1} 2_{i+1} 3_{i+1} 4_{i+1} \quad 1_{i+2} \cdots\cdot$$

The argument in the ADB function is ignored. The items will be removed consecutively. If there is no REAL TIME or ACCEPT statement or if there is no data remaining in the buffer (because the number of clock ticks, n2, in the REAL TIME statement had expired), the error message NO A-D and a line number are printed.

This example is an expansion of the one in Section 3.3 to incorporate the ADB function. Lines 590 through 610 have been added so that the sampled values can be printed.

```
5ØØ  SET RATE 7,Ø
51Ø  DIM G(1ØØ)
52Ø  REAL TIME G,1,2,15Ø
53Ø  ACCEPT
54Ø  SET RATE 2,1Ø
55Ø  IF TIM(4)<1ØØ THEN 55Ø
56Ø  REJECT
57Ø  SET RATE 3,1Ø
58Ø  ACCEPT
59Ø  FOR A=1 TO 15Ø
6ØØ  PRINT ADB(1),ADB(2)
61Ø  NEXT A
63Ø  END
```

The user must keep track of the items in the buffer (number requested and number removed) and their ordering by channel number. In the above example there will be 75 lines printed, each with two data values in the format:

${}^{1}{}_{i}$	${}^{2}{}_{i}$
${}^{1}{}_{i+1}$	${}^{2}{}_{i+1}$
${}^{1}{}_{i+2}$	

If line 140 was PRINT ADB(Ø),ADB(6),ADB(2), the information printed would be in the order:

${}^{1}{}_{i}$	${}^{2}{}_{i}$	${}^{1}{}_{i+1}$
${}^{2}{}_{i+1}$	${}^{1}{}_{i+2}$	${}^{2}{}_{i+2}$
${}^{1}{}_{i+3}$		

CHAPTER 4

TEST AND PAUSE COMMANDS

4.1 TST

For automated environments requiring a response typed on the Teletype to determine the subsequent action, a testing function, TST(n), is available which checks the Teletype to see if a character has been typed. If it has, a non-zero is returned; if it has not, a zero is returned. The TST function is particularly handy in conjunction with the TIM function for specifying a maximum response time. The TST function must be reset by doing a GETC function for the TST statement to be used more than once in a program.

4.2 WAIT

If a pause time is required by a program, the WAIT command can be included in the program. BASIC/RT processing will be halted until <u>any</u> interrupt occurs. Note that a clock interrupt is sufficient to reactivate BASIC/RT.

CHAPTER 5

User Commands and Functions

5.1 User Command (UCOM)

The USER COMMAND is a feature in BASIC/RT which allows the user to insert his own code and to effectively communicate with BASIC/RT. This command differs from the UUF function in that it is more sophisticated and hence more complicated in its operation. To briefly explain the differences: UUF is a function, such as SIN and COS, while UCOM is a command. A function takes an expression in parentheses as its argument and returns a value to be used later. For example, Y=INT(4.5) takes 4.5 and returns 4.0. A command is a statement which may take arguments (such as READ or INPUT), or may not have any (such as CLEAR or END), or they may be optional (as with PRINT). In a function such as UUF, BASIC/RT has already evaluated the expression and left the results in the FAC before calling the function so the function has no control over what is in it. With a command, it is up to the user code to scan and determine what is wanted. To successfully use this command, a knowledge of PDP-8 code and the floating point package is essential. It is suggested that the coding for commands such as CLEAR, PLOT, USE, (and any others) be studied in order to understand the nature of the commands. Some brief examples are illustrated next to show how to insert some simple commands.

In this example, a UCOM is created such that whenever it is issued, it executes a 6123 (some random IOT) with the AC all 7's. The first thing to do is allocate space. There is room in upper and lower core. In this case, let us overlay the EXP function with the UCOM. Put a pointer in the execute table to the location of the command. The table entry will be about location 306 (refer to a listing for the exact location). Now, whenever a UCOM is given, control will transfer to where this location points. The code will then look like:

```
*EXP
CLA CMA                 /Set the AC to all 7's
6123                    /Execute the IOT
CLA                     /Clear the AC
JMP I (DEVCOM           /and continue with BASIC
```

The routine DEVCOM checks to see that the next item on the line is a CR (carriage RETURN). If it is not, then an error message will be given. This is to prevent illegal user code from getting through: UCOMING HOME FOR DINNER? The previous sentence would call the UCOM processor, but since the next item is not a carriage RETURN, an error message would be given.

There are two more routines that can be used: DEVCON and SKIPIT. DEVCON does the same thing as DEVCOM except that it checks the present item for CR rather than the next item. SKIPIT skips past the rest of the line until it finds a CR and never gives an error message.

To modify the previous example to take two expressions and multiply them together and then send out the low 12 order bits to the device via a 6123, the command would look like:

```
UCOM A,B
```

where A and B are any expressions. The coding is:

```
*EXP
JMS       GETWD       /skip past the UCOM element.
JMS I     (MEVAL      /evaluate next element of line.
FENTER
FSTA      TEMP        /save away
FEXIT
JMS I     (MEVAL      /evaluate next element of line.
FENTER                /enter the FPP package
FMU       TEMP        /multiply A*B.
FEXIT                 /exit now.
JMS I     (FIX        /fix the AC now.
CLA                   /clear the AC.
TAD       AC3         /pick up low word.
6123                  /send to the device.
CLA
JMP I     (DEVCON     /and exit.
```

Notice that the exit is via DEVCON rather than DEVCOM. This is because MEVAL has evaluated the expression until it found a delimiter. This _must_ be a comma in the first place and a CR in the second place. If it is not, then DEVCON gives an error message. As stated previously, the user should look at PLOT and USE and perhaps a few other commands to fully appreciate the UCOM command.

It is possible to put the UCOM code in field 1 with only some minor changes. Since some commands are dispatched again in field 1, place in 306 a pointer to 7743. (7743 is the upper core dispatcher.) Then, in field 1 at 744, place the address of the UCOM command code. On UCOM, BASIC/RT will now go to this location in field 1. Cross-field communication is done through a variety of functions, all of which are mentioned in the internal documentation. The novice user is advised against putting instructions in field 1 until he has successfully run functions and commands from field Ø. Note that functions can also reside in field 1. The user should examine ACCEPT, TIM and possibly REAL TIME to get a better understanding of cross-field communications.

It is possible to allocate more space in field 1 for functions/commands by changing the value of LIMIT. LIMIT is the first location of upper core used by BASIC/RT. By changing PLIMIT at 2562 to, say, 2000, about 1000_8 locations are gained for user functions and commands. Note that this space is lost from the user program and variable area, but very often this is worth the sacrifice.

5.2 User Coded Function

As in 8K BASIC, the user can include his own function in BASIC/RT, calling it as UUF(n), to perform a special task. The UUF function is coded in the same manner as described in the 8K BASIC User's Manual, except that the suggested locations for the code are 0064 to 0177 in field 1. Other smaller areas are available in field 0 and many more can be created by deleting some of the mathematical functions.

CHAPTER 6

ERROR MESSAGES

The computer checks all commands before executing them. If for some reason it cannot execute the command, it reports the fact by printing one of the messages in Table 7-2 and the number of the line in which the error was found.

Table 7-2

BASIC/RT Error Messages

Message	Explanation
SYNTAX ERROR	The command does not correspond to the language syntax. Common examples of syntax errors are misspelled commands, unmatched parentheses, and other typographical errors.
FUNCTION ERROR	A function was used which was deleted when the system was loaded and hence is not available. A DEF statement will produce this error if the DEF capacity has been deleted.
TOO-BIG ERROR	The combination of program and variables exceeds the capacity of the computer. Reducing one or the other may help. If the program has undergone extensive revision, try punching it out, typing SCRATCH, and reloading.
SUBSCRIPT ERROR	A subscript has been used which is outside the bounds defined in the DIM statement.
LINENO ERROR	A GOTO, GOSUB, or IF references a nonexistent line.
FOR ERROR	FOR loops are nested too deeply.
NEXT ERROR	FOR and NEXT statements are not properly paired.
GOSUB ERROR	Subroutines are nested too deeply.
RETURN ERROR	GOSUB and RETURN statements are not properly paired.
DATA ERROR	There are no more items in the data line.
ARGUMENT ERROR	A function has been given an illegal argument, for example, SQR(-1).
RATE ERROR	Clock rate was too fast to service interrupts.
A-D FULL	No more room in the data buffer for remaining items.
NO A-D	No input coming from specified analog channel.

BOOK EIGHT

NMR SIMULATOR

CONTENTS

TABLES

CHAPTER 1

USING NMR SIMULATOR

1.0 INTRODUCTION

NMR SIMULATOR[1] will calculate, display and plot theoretical NMR spectra of any system of spin-1/2 nuclei containing up to six spins. Its inputs, in addition to chemical shifts and coupling constants, are sweep offset, sweep width and spectrometer frequency. It will, if desired, punch the calculated transitions on paper tape for later analysis. While the principal use of this program will undoubtedly be in proton NMR, it is not limited to proton range values: shifts, coupling constants, width and offset can be anywhere from 10^{-616} to 10^{616} Hz.

The principal value of NMR SIMULATOR is in the analysis of those NMR spectra which are so closely coupled that the usual first order approximations no longer hold, e.g., when the chemical shifts are no longer large compared to the coupling constants. This program provides a method of calculating and plotting theoretical NMR spectra directly in the laboratory. In addition to the program's value as a theoretical or analytical tool, it has great utility as an educational aid.

The NMR SIMULATOR program does not run under OS/8.

1.1 Storage Requirements

NMR SIMULATOR occupies locations 0-7577 in field 0. It stores data in locations 20-7245 of field 1.

1.2 Hardware Requirements

The minimum configuration is a LAB8/E with 8K of core, ASR33 Teletype, and VC8E display control. In order to use the punched tape and transition listing options, a high-speed reader/punch is also required.

---

[1]Based on "NMR SIM", D. F. Juers, R. S. Boettcher, V. J. Hull and H. E. Zimmerman, DECUS 8-194. Additional Reference to: G. Fraenkel and J. W. Cooper, Tetrahedron Letters 1968, 1825.

1.3 USE OF NMR SIMULATOR

1. Load the binary tape of the NMR simulator as follows:

   - o Place the tape in the reader.
   - o Set the Switch Register (SR) to 7777.
   - o Press ADDR LOAD.
   - o If the high-speed reader is used, set bit Ø to Ø.
   - o Press CLEAR then CONT.

2. Start the program as follows:

   - o Set the SR to Ø2ØØ (Binary ØØØ Ø1Ø ØØØ ØØØ)
   - o Press LOAD ADDR
   - o Press CLEAR and CONT.

3. The Teletype will type:

   NMR-SIMULATOR
   NUMBER OF SPINS=

   Enter any value from 2 through 6. Do NOT follow with a carriage RETURN. If you answer with Ø, the program will shift directly to a display of the contents of the display array. This is most useful if the program was inadvertently restarted after a spectrum has been calculated.

4. The next question is

   PUNCHED OUTPUT?

   If you wish a binary tape containing all of the transitions and their intensities, answer Y. Answering with any other character produces no tape. Since 5- and 6-spin cases take a significant amount of time to calculate, some time is saved by using this option <u>if</u> you will be changing the offset and sweep width. For smaller systems, it is usually just as fast to allow the computer to recalculate the spectrum from scratch.

5. The program then types

   SPECTROMETER FREQ. (MHZ) =

   Answer with the frequency of your spectrometer, i.e., 60., 90., 100., etc., followed by the RETURN key. This frequency is used only if you enter shifts and coupling constants on the delta or tau scales. If you plan to enter these values on the frequency scale, this parameter can be answered by pressing the RETURN key

6. The next two questions are

SWEEP OFFSET =

and

SWEEP WIDTH =

Set these to the closest approximation possible of the experimental spectrum you wish to simulate. A <u>positive</u> value for the offset produces an <u>upfield</u> shift of the spectrum. These two values must be in Hz (cps). Follow these two values with a RETURN.

7. The program next asks for the shifts

CHEMICAL SHIFTS
H(1) =
...etc.

These values can be entered according to one of three possible scales.

a) The Frequency scale -

This is defined so that TMS is Ø Hz and the frequency increases positively in the <u>downfield</u> direction. Just type the decimal number followed by the RETURN key.

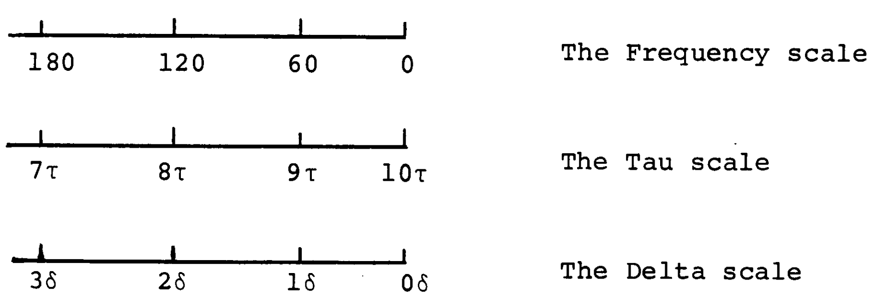
b) The Tau scale -

This is defined so that TMS is 10 tau, and so that tau values decrease in the downfield direction. To use this scale, terminate each shift value with a T instead of a RETURN.

c) The Delta scale -

This is defined so that TMS is Ø delta, and so that delta values increase in the downfield direction. To use this scale, type D after each shift value instead of RETURN.

These three scales are illustrated below for a 60 MHz spectrometer:



8. The final parameters to be entered are the coupling constants:

```
COUPLING CONSTANTS
 J(1,2) =
 J(1,3) =
 ...etc.
```

Answer these values with decimal numbers in the frequency scale (or, if you wish, in the tau or delta scales). The subscripts following each J refer to the numbers of the nuclei as determined by their order of entry. Thus, J(1,2) is the coupling constant between H(1) and H(2).

If you make a mistake while entering any decimal number, typing RUBOUT will delete the entire number. You can then re-enter the number starting from its first digit. The program does not print any character for RUBOUT.

All decimal numbers may also be entered according to the exponential or E format. This is described fully in Introduction to Programming[1] and in Programming Languages[2].

9. For 4-, 5- and 6-spin cases, the computer will type COOKING... to assure you that the calculation is proceeding normally. Be sure that the high speed punch is turned on if the PUNCHED OUTPUT? option was specified.

10. Approximate execution times for various size cases are given below:

Number of Spins	Execution Time in Seconds
2	<1
3	2-3
4	8-10
5	30-90
6	300-600

11. When the calculation is complete, the Teletype will type

```
DONE!
```

and ring its chime. The spectrum will then be displayed on the scope of the LAB8/E.

---

[1] Digital Equipment Corporation, Introduction to Programming, 2d Ed.; (Maynard, Mass., Sept. 1970), p. 1-35.

[2] Digital Equipment Corporation, Programming Languages, 1st Ed.; (Maynard, Mass., May, 1970), p. 16-11.

CHAPTER 2

DISPLAY MODE COMMANDS

While the spectrum is being displayed, the commands in Table 8-1 can be issued by pressing the indicated key on the Teletype.

Table 8-1

Display Mode Commands

Command	Meaning
R	Restart the program from the beginning.
J	Computer asks COUPLING CONSTANTS, etc. Enter new coupling constants as described in Section 2.0.
S	Computer asks CHEMICAL SHIFTS, etc. Enter new chemical shifts as described in Section 2.0.
O	Computer asks SWEEP OFFSET =, SWEEP WIDTH =. Enter new offset and width as described in Section 2.0. After asking for the offset and width values, the computer asks TAPE?. If you have punched no tape, answer N and wait for the program to recalculate the spectrum. If a tape was punched, answer Y. The computer responds with LOAD AND CONT. and halts. Load the punched tape into the high-speed reader and press CONT on the console.
T	Computer types the transitions and intensities. It is necessary that a tape be punched in advance to use this option. The computer will type MINIMUM INTENSITY = For small cases, or for accurate fitting, answer Ø and RETURN to get all of the transitions. To list only the major peaks, the minimum intensity may be increased. The computer responds with LOAD AND CONT. At this point, load the punched tape into the high-speed reader and press CONT on the console. The computer will respond with ENERGY INTENSITY, and the listing will be typed out on the Teletype. The display will return when the listing is complete.

Table 8-1

Display Mode Commands (Cont.)

Command	Meaning
P	Plot out the displayed spectrum. By connecting the voltages driving the scope display to an X-Y plotter, one can obtain a hard copy of the theoretical spectrum. The computer types L.L. and displays a dot in the lower left hand corner. Adjust the plotter bias and sensitivity to place the pen in this corner. Then type one of the following: LINE FEED — The computer types U.R. and displays a dot in the upper right hand corner RETURN — The display returns to the lower left. The plot begins at a rate specified by Knob Ø. If the plot rate is too fast, interrupt it by typing CTRL/O . When the plot is complete, or when it is interrupted by CTRL/O, the computer types TURN OFF PLOTTER. The program waits until RETURN is typed before returning to the display. This is done so that the rapidly varying voltages of the display will not cause the plotter pen to gyrate wildly and scribble all over the freshly plotted spectrum.
L	Lorentzian line shape calculation. The computer types: LINE WIDTH = Answer with any decimal number followed by RETURN. The program begins to calculate the curve. The length of time required for this calculation varies from about 2 to 30 seconds, depending on the number of lines and the line width. When the calculation is complete, the computer types: DONE! and displays the curves on the scope. At this time the following commands are available: R - restart the program from the beginning J - change the coupling constants S - change the chemical shifts U - shift the display up D - shift the display down CTRL/O return to the stick figure display P - plot out the Lorentzian curves CTRL/C return to the Monitor (DECtape or Disk systems only) Thus, one can vary the line width by returning to the stick display via CTRL/O and re-entering the Lorentzian routine by typing L.

CHAPTER 3

METHOD OF CALCULATION

The simplest method of hand calculation of an NMR spectrum requires that all possible spin functions be written and sorted according to the number of +1/2 and -1/2 spins in each. Using the usual notation[1,2] spin -1/2 is represented by β and spin +1/2 by α; the order in which the spins are written represents the nuclei to which they belong. Thus, for a three spin system, the spin functions can be written as follows:

$\alpha\alpha\alpha$	$\alpha\alpha\beta$	$\alpha\beta\beta$	$\beta\beta\beta$
	$\alpha\beta\alpha$	$\beta\alpha\beta$	
	$\beta\alpha\alpha$	$\beta\beta\alpha$	

Since quantum mechanics requires the nuclei to be indistinguishable, the difference between $\alpha\alpha\beta$ and $\alpha\beta\alpha$ and $\beta\alpha\alpha$ cannot be determined. Thus, these spin functions must be mixed[2,3] and an equal number of new, mixed ones created. This is done by the computer in a manner somewhat different from hand calculation.

For each grouping of spin functions, an H-matrix is constructed as follows:

The diagonal elements, H_{uu}, are calculated from

$$H_{uu} = \sum_{i=1}^{n} (S_i \nu_i + \sum_{j=1}^{n} T_{ij} J_{ij}) \qquad (1)$$

where S_i = -1/2 for a β spin
+1/2 for an α spin

ν_i = the frequency of the absorption in Hz

T_{ij} = +1/4 if nuclei i and j have the same spin
-1/4 if they have opposite spins

J_{ij} = the coupling constant between i and j in Hz

The off-diagonal elements are calculated from the following:

$$H_{uv} = 1/2\ U_{ij} J_{ij} \qquad (2)$$

---

[1] J. A. Pople, W. G. Schneider and H. J. Bernstein, High Resolution Nuclear Magnetic Resonance, (McGraw-Hill, New York, 1959), pp. 103ff.

[2] J. D. Roberts, An Introduction to the Analysis of Spin-Spin on High Resolution Nuclear Magnetic Resonance, (W. A. Benjamin, New York, 1961), pp. 37ff.

[3] J. D. Roberts, Notes on Molecular Orbital Calculations, (W. A. Benjamin, New York, 1962).

where

J_{ij} is as above

U_{ij} = 1 if the basic functions differ only in the interchange of i and j
= Ø if the condition is not satisfied

This matrix is then diagonalized, and the diagonal elements (or trace) of the matrix become the energies (eigenvalues) of the "mixed" spin functions. The mathematical method of matrix diagonalization[1] produces a second matrix, the U matrix, which contains the final spin eigenvectors. These are used to calculate the probability of each transition between the calculated energy levels.

Transitions occur only between spin functions, the sum of whose spins, F_z, differs by 1. Thus,

$$F_z = \sum_{i=1}^{n} S_i \tag{3}$$

and ΔF_z must =1 for a transition to occur.

The probability (intensity) of this transition is given by

$$I = \left(\sum_u \sum_v C_u C_v' \quad A\right)^2 \tag{4}$$

where

C_u = the u^{th} element of one spin function's eigenvectors.

C_v' = the v^{th} element of the other set of eigenvectors.

A = 1 if C_u and C_v' differ by one spin
= Ø in all other cases.

These matrices are formed and diagonalized, a group at a time, and the transitions calculated. A reduced copy of these transitions is stored in a 500 word display array, with all transitions within the specified range placed in one of these 500 bins.

The Lorentzian line shape calculation is performed by setting T_2 = 1/(line width), and applying the equation of Pople:[2]

$$g(\nu) = \frac{\text{line height}}{1 + (2\pi\ T_2(\nu-\nu_0))^2}$$

[1]P. R. Bevington, Data Reduction and Error Analysis for the Physical Sciences, (McGraw-Hill, New York, 1969).

[2]Pople, et al, pp. 103ff.

CHAPTER 4

Program Description

1. The program types its name and asks for the number of spins.

2. Enter the number of spins. The program tests for a value between 2 and 6 and stores it in N.

3. Set switch if punched output is desired.

4. Enter spectrometer frequency. The program stores it for use in calculating Hz from delta or tau values.

5. The program sets up basic product functions (BPF's).

   a. It calculates NARRY array which contains relative addresses of the BPF's of different F_z value.

   b. It calculates NUSE array. This array actually contains the BPF's. They are stored in groups, each group having F_z value one greater than the preceding group.

6. Enter sweep offset and sweep width.

7. Enter chemical shifts.

8. Enter coupling constants.

9. SPEC array is initialized

   a. This array eventually contains the displayed spectrum

   b. It is initialized to Ø except for 25 calibration points which are set to $4ØØØ_8$.

10. First H and U matrices are calculated.

    a. First H matrix is always a 1 X 1 which does not require diagonalization.

    b. First U matrix is always a 1 X 1 with its only element equal to 1.

11. The program sets FLAG which is the negative of the number of times the loop from 12 to 18 must be executed.

12. The program sets N2, calculates EN and UOLD arrays.

    a. It copies the trace of the H matrix into the EN array.
    b. It copies the U matrix into the UOLD array.
    c. It sets N2 equal to the size of the H matrix just copied.

13. The size of the next H matrix is calculated and put into N1.

14. The program determines which transitions will be allowed between the BPF's represented by UOLD and those that will be represented by U. This is stored in the array called TABLE.

BOOK NINE

NMR SIGNAL AVERAGER

CONTENTS

TABLES

CHAPTER 1

NMR AVERAGER

1.0 GENERAL DESCRIPTION

NMR AVERAGER is a signal averaging program designed for use with any NMR spectrometer configuration in which the application of a sawtooth voltage from -5 to +5 volts will cause a linear sweep of the spectrometer frequency (or field). Since the computer actually does the sweeping, no trigger signal is required. This leads to a highly reproducible relationship between memory channel and spectrometer frequency, and an extremely accurate method of calibration. Once the average is accumulated, NMR AVERAGER can be used for display manipulation, plotting, calibration, digital and analog integration, differentiation, curve smoothing and decimal and binary output.

1.1 Hardware

The minimum computer configuration includes a LAB8/E, Teletype, and an X-Y analog plotter. An NMR spectrometer with an internal lock and autoshim is required. It must have some modification such that the application of a -5 to +5 sweep voltage will cause a frequency sweep in an adjustable range. One method of doing this is by substituting a digital frequency synthesizer or voltage controlled oscillator for the sideband modulation oscillator.

Optional Equipment: a digital frequency synthesizer, frequency counter and high-speed punch.

1.2 Storage

NMR AVERAGER occupies locations 0-3377 and uses locations 3600-7577 to store 1024 double precision data points. After data normalization (↑E)[1] locations 3600-5577 are used as a display buffer. The data are stored in single precision from 5600-7577.

1.3 Loading

NMR AVERAGER is loaded using the standard Binary Loader. Its starting address is 0200. (Refer to Chapter 4.)

NMR Averager can not be run under OS/8.

[1] ↑E is the command used to enter Normalized Display Mode and is typed by holding down the CTRL key and typing E.

1.4 Execution Time

The execution time is entirely I/O bound. The sweep time is variable in one second increments from 1 to 4095 seconds. The actual timing is calculated by an integer approximation of the correct number of crystal clock pulses per sample point. Consequently, while the time may be slightly slower or faster than that specified, it will be reproducible. Plotting time is variable by adjusting Knob 3 on the LAB8/E. Digital output is via the Teletype (10 characters per second) or the high-speed punch (50 characters per second). All display commands are executed essentially instantaneously.

1.5 Abbreviations and Conventions Used in this Book

CR/LF is used to mean a carriage return, line feed combination output by computer.

CTRL/H or ↑H is used to mean the non-printing character produced by holding down the CTRL key and simultaneously typing the alphabetic character H.

ADC and DAC are used to mean analog to digital and digital to analog converters.

CHAPTER 2

INTRODUCTION TO SIGNAL AVERAGING

The signal averaging method assumes that a very weak, but coherent, signal is buried under random noise. Analog to digital converters (ADC's) are used to convert the varying analog voltage output of the spectrometer into digital information. After the passage of a specified time interval, the signal is again examined and digitized. These signals, separated in time by the total sweep time divided by the number of memory channels used, are stored in successive locations in the computer's memory during the first sweep. During succeeding sweeps, the signal is added to the information already contained in the memory locations. The noise, if random in nature, averages to zero while the signal to noise ratio grows in magnitude[1]. According to theory, signal enhancement is of the order $\sqrt{N}$, where N is the number of sweeps taken.

Obviously, then, the greater the number of sweeps, the greater the signal enhancement. If there were no other constraints it would seem apparent that signal averaging should be carried out at the highest possible sweep rates. Unfortunately, in NMR spectroscopy, this is not possible because of the necessity of satisfying conditions of slow passage[2]. Thus, since there is a minimum desirable sweep time, a multiple sampling technique was developed in the NMR AVERAGER to increase the statistical significance of each sweep. In this method, each data point is examined several times (up to 20) before the computer goes on to the next channel. These twenty A-D conversions are summed and placed in memory. This permits the performance of twenty sweeps in the time of one, without sacrificing resolution to rapid passage conditions.

The next problem is: how can one legitimately sample several times unless he is sure that the exact same frequency is being examined? In most signal averaging systems, the computer and spectrometer sweep simultaneously but independently once they start. This means that the spectrometer continues to sweep while the

[1]R. R. Ernst, "Sensitivity Enhancement in Magnetic Resonance", Advance in Magnetic Resonance, 2, J. S. Waugh, Ed., (Academic Press, New York, 1966).

[2]J. A. Pople, W. G. Schneider and H. J. Bernstein, High Resolution Nuclear Magnetic Resonance, (McGraw-Hill, New York, 1959) pps. 37, 38.

computer "stands still". The answer is, of course, that multiple sampling techniques (here known as boxcar integration) cannot be performed accurately under these conditions. This problem has been overcome, however, in NMR AVERAGER, since the sweeping is controlled by the computer, and it can hold its output voltage at the exact same level as long as necessary to obtain all 20 samples, before incrementing it to the next frequency level and channel. Thus the multiple sampling technique is a form of "digital filtering" rather than of boxcar integration.

Once the signal data has been accumulated, there are still additional problems which must be solved. It is obviously easy to plot out the spectrum using a pair of DAC's to drive an X,Y plotter, but once the data has been plotted, how can it be calibrated? In the ideal case, there are two or more reference lines in the spectrum which make it easy to obtain approximate chemical shifts for each spectral peak. But in the less-than-ideal, that is to say the usual case, there may be no known lines at all, although the frequency values of the extreme limits of the plot may be known. The obvious solution, which some chemists have always used, is linear interpolation between two known points. However, when these known points are the extreme ends of the spectrum, this interpolation can be disastrous. Neither the spectrometer sweep nor the DAC computer sweep is exactly linear, and in the case of simultaneous, but independent sweeping, the two sweep voltages may even be bowed in different directions. To see just how nonlinear some of these sweep voltages are, plot two of them against each other on an X,Y plotter, and lay a ruler alongside.

However, since NMR AVERAGER actually does the sweeping itself, there is always a direct correspondence between memory channel and sweep voltage. This fact is then used in the calibration routine, in which the sweep voltage can be advanced manually until the plotter pen hovers over the peak of interest, and the corresponding frequency can be read from a frequency counter.

To make such a calibration routine completely accurate, it must be recognized that the computer spends some of its time putting out the DAC voltage and some examining and converting the input signal. During the time spent in calculation and conversion, there is no voltage coming from the DAC. This means that the average voltage "seen" at the output of the DAC depends on the sweep time. Since the examining and conversion time is constant and the voltage output time varies with the sweep time, it is necessary to take this "wasted" time into account in the calibration routine. Thus, in NMR AVERAGER, the amount of time spent in putting out the calibration voltage is proportional to the sweep time used to acquire the data. This ensures a highly accurate calibration.

CHAPTER 3

CONNECTING THE COMPUTER AND THE SPECTROMETER

There are undoubtedly numerous methods of interfacing various spectrometers with the LAB8/E. The most successful methods have used a digital frequency synthesizer or voltage controlled oscillator as either the modulation frequency source or as the radio frequency source.

In the case of Varian HA-model instruments the technique is extremely simple. The x-axis scope output of the LAB8/e is connected to both the scope input and the sweep input of the synthesizer, using a BNC or UHF T-connector, and appropriate adaptors. The frequency output of the synthesizer is connected to MAN OSC OUT at the rear of the lock box, and the MANUAL OSCILLATOR tuning network card is removed. The synthesizer has now replaced the manual oscillator as a modulation source. The spectrometer is then set up as for FIELD SWEEP, so that the irradiating channel is the just modified manual oscillator channel. The output voltage which sweeps the synthesizer will then cause a frequency sweep of the sample.

An alternative method, for synthesizers whose frequency is high enough for the nuclei being studied, is to use the synthesizer as an actual radio frequency source. The actual connections made will depend on the characteristics of the probes and detectors used.

For the JEOL C60-HL and PS-100, the input voltage necessary to sweep the spectrometer is 0 to 190 volts on older models, and 0 to +25 volts on the newer models. Interfacing thus requires an inexpensive differential amplifier. The output to JEOL recorders is 10 millivolts full scale and will thus also need amplification to the ±1 volt range required by the LAB8/E.

For the Perkin Elmer-Hitachi R20B, the input voltage is -15 to +15 volts. The output voltage should require no amplification. Digital Equipment Corporation will be glad to furnish to its users all available information on spectrometer interfacing. Simply contact the Chemistry Marketing Manager.

Regardless of the frequency input method, the output remains the same. Virtually all spectrometers have an analog output jack marked "external recorder". This is connected to analog input Ø on the LAB8/E front panel. If, for some reason, an input other than Ø is used, it is necessary to deposit the number of this input channel in location 126 after NMR AVERAGER has been loaded. The input jacks are numbered Ø-3 from top to bottom.

Still another approach is to modify the spectrometer such that the sawtooth output is used to cause the direct sweeping of the spectrometer's sweep coils. The actual modifications depend to a large degree on the spectrometer used.

3.1 Homogeneity

NMR AVERAGER does not control the homogeneity of the spectrometer. Consequently, it is necessary that both field and homogeneity drift be minimized by the spectrometer itself. This requires that the spectrometer have an internal lock and some sort of autoshim circuitry. For spectrometers without automatic shim control, it is recommended that signal averaging not be carried out over a period greater than two hours, unless experiments show that drift is negligible over a longer period.

3.2 Filtering

The response time, or amount of "filtering" should be adjusted to a moderate to large value for the compound being examined. Ernst[1] has reported that it is preferable to perform filtering before signal averaging rather than afterwards, to avoid down-conversion of high frequency noise. If, after the spectrum has been accumulated, it is found that the filtering was insufficient, the curve can be smoothed by use of the F command.

---

[1]Ernst, 2, 1-135.

CHAPTER 4

DETAILED DESCRIPTION OF THE USE OF THE AVERAGING PROGRAM, NMR AVERAGER

1. Load the binary tape of the averaging program as follows:

   a. Turn on the power to the computer and Teletype and switch the Teletype to LINE.

   b. Place the binary tape in the tape reader while the reader is set to FREE.

   c. Turn the reader to START.

   d. Set switches 6-8 to field in which BIN is loaded. Set switches 9-11 to field in which program is to be loaded. Press EXTD ADDR LOAD.

   e. Set the Switch Register to 7777. This is the position with all 12 switches having their tips pointing up.

   f. Depress ADDR LOAD. All the lights in the Program Counter should come on.

   g. Depress CLEAR then CONT. (To use the high-speed reader, change bit Ø from 1 to Ø before pressing CLEAR. Bit Ø is the leftmost bit of the Switch Register.) The tape should begin reading in. If it does not, the Binary Loader program is not resident and must be reloaded. Refer to DEC instructions (PDP8/E Small Computer Handbook 1971) on reloading the Binary Loader.

   h. When the tape finishes reading in, it should stop at the last set of holes and the computer should halt with all 12 lights in the accumulator out. (The link may be either on or off.) If the accumulator is not equal to Ø, there is a tape-reading error. To rectify this, reload the tape, starting at instruction d. If this error persists, reload the Binary Loader program.

2. Start up the averaging program as follows.

   a. Set the Switch Register to 0200 (Binary 000 010 000 000).

   b. Depress ADDR LOAD and then CLEAR and CONT. The Teletype should type:

      NMR-AVERAGER
      SWEEPS=

   c. Enter any value between 1 and 4095 for the number of sweeps. If a mistake is made, type RUBOUT. The computer types ? followed by a carriage return and line feed (CR/LF). Enter a new value. Typing any noninteger character, or entering a number larger than 4095 will also cause this error response. You can tell the computer to accumulate more than 4095 scans by answering Y to REPEAT? (see 2.g).

If the response to this question is Ø or the RETURN key, the program immediately shifts to the display section. If normalization has been performed since memory was last erased, the display is the Post-Normalization Display. If no normalization has been performed, the display is the Pre-Normalization one.

This is most useful as an error recovery, when the program has been inadvertently restarted.

When a correct value has been entered, type the RETURN key.

d. After responding with a CR/LF, the computer types:

SWEEP TIME =

Answer with any value less than 4096. Correct any error as above by typing RUBOUT, and re-entering the value. Then type RETURN. If the value is more than 4095 the computer responds with a ? and reasks the question.

e. If the value of the sweep number is acceptable, the Teletype will follow the CR/LF with

SAMPLES =

This parameter refers to the number of times each point is sampled during a single sweep. Respond with any number between 1 and 20. The purpose of this parameter is to increase the statistical significance of each sweep, by examining each point several times. Follow the typed value with RETURN.

f. The program types out

PAUSE TIME =

This is the number of seconds between sweeps. During this time a voltage is generated corresponding to the left end of the X-axis. This allows time for the oscillator and/or the recorder pen carriage to return to the left end before the next sweep. Answer with any integer between Ø and 4Ø and type RETURN.

g. The Teletype does a CR/LF and types

REPEAT?

If Y is typed, the computer sweeps indefinitely, typing "END" after every N sweeps. If any other character is typed, the sweeping is not repeated.

h. The Teletype does a CR/LF and types

TYPE RETURN

If RETURN is typed, the memory is erased and averaging is begun.

If LINE FEED is typed, the memory is not erased, and additional signals are added to whatever data has already been accumulated. This feature allows change of the sweep parameters without disturbing the accumulated data, by using the R command.

i. Typing either RETURN or LINE FEED causes a CR/LF on the Teletype, and displays a dot at the left hand edge of the scope, representing the rate and position of the sweep. This linear scope sweep is the actual voltage which is also sweeping the spectrometer. If the averaging is to take place over a long period, turn down the scope intensity to avoid burning a dot on the scope, and turn off the Teletype.

Refer to Appendix V for a summary of use of the NMR Averager.

ON-LINE AVERAGING

During the actual averaging, no other operations can be performed by the computer, but, at the end of any sweep, the computer can be stopped and the accumulated data displayed (refer to Chapter 6).

Stop Commands	Meaning
S	Causes the sweeping to be interrupted at the end of the current sweep. After viewing the display type G to continue or R to restart the program.
↑H	(CTRL/H) Causes an immediate halt. This command is only for use as an error recovery, if sweeping has been inadvertently started. It should <u>not</u> be used for an interrupt if the data acquired so far is to be saved. The reason for this warning is obvious; the computer adds each new data point into memory immediately after receiving it, and, if it is interrupted in the middle of a sweep, part of memory will contain one more sweep than the rest.

CHAPTER 6

THE DISPLAY MODE

When all the sweeps are completed, the computer prints

END

and shifts to a display of the accumulated data. This section of the program is called the Display Mode. If sweeping is interrupted either by the S or CTRL/H commands, the computer shifts to Display Mode without printing anything.

If a large number of sweeps have been taken, the computer displays a group of disconnected dots. These can be reconnected, however, by contracting the display several times. Display Mode commands are described in Tables 9-1 and 9-2.

Table 9-1

PRE-NORMALIZATION MODE COMMANDS

Command	Meaning
C	Contract the display by a factor of 2 each time C is typed.
X	eXpand the display by a factor of 2 each time X is typed.
G	Go - continue averaging without erasing the memory. If the specified number of sweeps have not been completed, sweeping continues until this number is reached. If all sweeps have been completed, an additional N sweeps are performed. If the answer to REPEAT? was Y, sweeping continues indefinitely; if the Teletype is turned on, END is typed after every N sweeps.
CTRL/Z	Zero (erase) the memory and the display buffer. When the memory is erased, the number of sweeps is also reset to zero, so new spectra can be accumulated using the same parameters.
R	Restart the program at the beginning. This is equivalent to pressing HALT, ADDR=0200, and CLEAR then CONT.
N	Print the Number of sweeps completed. Thus, to find out how many sweeps have been completed, type S to Stop the sweeping, and then type N. Continue the sweeping by typing G.
P	Plot routine. When the computer enters the Plot routine, it prints LOWER LEFT and displays a dot at the lower left-hand corner of the scope. At this time,the outputs to the scope from the LAB8/E are (-5,-5) volts. Refer to the LAB8/E User's Manual for instructions on connecting the LAB8/E to the plotter. The plotter should be adjusted so that this voltage places the pen in the lower left-hand corner of the plotter paper. Now type the LINE FEED key. The computer responds with UPPER RIGHT and displays a dot in the upper right-hand corner of the scope. The LAB8/E outputs are now (+5,+5) volts. After making tentative gain and zeroing adjustments to the plotter, type RETURN, which will again bring the pen to the lower left corner. By typing the LINE FEED and Carriage RETURN keys alternately, the plotter can be adjusted to plot a spectrum of any desired size and shape.

Table 9-1

PRE-NORMALIZATION MODE COMMANDS (Cont.)

Command	Meaning
	When the plotter adjustments are complete, retype P to begin the plot. The rate of plotting is controlled by knob 2 on the LAB8/E. Adjust this knob until an appropriate rate is found. If the plotting is too fast or too slow during the first trial, adjust knob 2 to vary the plot rate and type P to restart the plot. When an appropriate rate is found, lower the pen to make a hard copy of the spectrum. When the plot is complete, or when it is interrupted by typing CTRL/O, the message TURN OFF PLOTTER will be printed on the Teletype. The program waits for the user to type the RETURN key before returning to the display. This is so that the rapidly changing voltages of the display will not cause the plotter pen to gyrate wildly.
K	Calibrate. The calibration routine can be entered either from the plot subroutine or from the Display Mode. When K is typed, a single dot whose coordinates are controlled by knobs 2 and 3 is displayed on the scope. This subroutine provides an extremely accurate method of calibration of the plotted spectrum. To calibrate a plotted spectrum, simply use the knobs to adjust the plotter pen so it coincides with the peak of interest on the spectrum. Since the x-axis voltage is connected to the frequency synthesizer, it is only necessary to connect a frequency counter to the synthesizer to find the frequency of the plotted line. After using this calibration routine several times, it will become apparent that the speed of response of the dot on the scope to the control knobs varies with the sweep time. This is entirely normal, since the average output voltage varies somewhat with the sweep time. Thus, in order to make the calibration as accurate as possible, the computer calculates how long to spend displaying each point before checking the knobs to see if they have been changed. This rate is based directly on the sweep time currently set in the computer. It is therefore necessary to calibrate a spectrum with the sweep time set to the same value that it had during the actual averaging process. Exit from the calibration routine can be accomplished by typing CTLR/O on the Teletype. The computer then returns to either the pre-normalization or post normalization display.

Table 9-1

PRE-NORMALIZATION MODE COMMANDS (Cont.)

Command	Meaning
	In order to calibrate a spectrum approximately from the display alone, use the digital I integration routine to adjust cursor 2 to be on top of the line of interest. Then type I again, wait for the numerical integral to be typed, and type K. The x-coordinate of the dot has now been adjusted to exactly the same point on the scope as the cursor occupied. Then read the frequency from the frequency counter.
B	Binary punch. This command tells the computer to punch out the contents of its memory in binary form, which can be read by the Binary Loader. It includes a load address of 3600 and a checksum at the end. When B is typed, the computer asks H.S. PUNCH? Turn on the punch to be used, and answer Y for output on the high-speed punch. Input other than Y causes output on the low-speed punch. Type CTRL/O to abort the punching. The computer responds with CR/LF and returns to the Display Mode.
T	Type out the contents of memory as 1024 double-precision decimal words. As above, the computer first asks H.S. PUNCH? If Y is answered, the output will be in ASCII code on the high-speed punch. This saves time, since the tape can be listed later off-line, while the computer is doing something else. Be sure to turn on the punch before answering Y. As in the binary punch routine, the printing/punching can be interrupted at any time by typing CTRL/O and the RETURN key.
CTRL/C	Shifts from the display mode to location 7600 octal. This command is used with the Disk System Monitor or the OS/8 Monitor, as a method of direct return to the Monitor. If the system does not have one of these Monitors (available with DECtape or Disk systems) typing CTRL/C will probably wipe out the Binary Loader and NMR AVERAGER as well.
CTRL/E	Enters the normalized Display Mode. The double-precision numbers stored in memory are converted to single precision numbers based on the number of contractions which have been performed on the display before normalization. Thus, the display must be on scale and contracted so that there is no "wraparound" before normalization is performed. This command is irreversible: the double precision pre-normalization mode cannot be reentered once this command has been issued. The additional 1024 points of memory gained by this normalization are used as a display buffer in which data can be modified without destroying the original spectrum. Thus, curve smoothing, integration and differentiation are performed non-destructively.

Table 9-2

POST-NORMALIZATION MODE COMMANDS

Command	Meaning
CTRL/D	Differentiates the spectrum as displayed. The digital approximation of the first derivative is calculated and displayed.
U	Shifts the display up by 5 percent.
D	Shifts the display down by 5 percent.
F	Filters (or smooths) the data by the use of a three-point smoothing routine. This routine operates <u>only</u> on the display buffer, leaving the original data intact. This smoothing can be repeated as many times as desired, and its result plotted by typing P or integrated by typing I or A. To return to the unsmoothed data, type CTRL/O and the RETURN key. Care should be taken that the entire display is on scale and not "wrapped around" before smoothing, since the computer will attempt to connect lines off scale at the top of the scope with lines off scale at the bottom of the scope. The commands C, X and S also return to the unsmoothed display, once contracted, expanded or in unmodified form. Thus, if the display is off scale when F is typed, just retype C and F. The filtered spectrum can be shifted up or down without destroying it. The filter routine uses the simple equation[1]: $y'(x) = 1/4(y(x-1)+2y(x)+y(x+1))$ to replace points 2-1023 of the 1024 points. Thus, while points 1 and 1024 are included in each calculation, they are never replaced. Instead, points 1 and 1024 are set equal to points 2 and 1023 after each smooth.
A	The computer calculates the analog integral so it will fit on scale, but the display may need vertical shifting. This can be accomplished, as usual, by the U and D commands. Once the integral has been calculated, and is on scale, it can be filtered, plotted or again integrated by typing A. To return to the unintegrated display, type CTRL/O and the RETURN key.

[1]P. R. Bevington, <u>Data Reduction and Error Analysis for the Physical Sciences</u>, (McGraw-Hill, New York, 1969), pp. 255-259.

Table 9-2

POST-NORMALIZATION MODE COMMANDS (Cont.)

Command	Meaning
I	When I is typed, two new lines will appear on the scope. These are two vertical cursors and are controlled by knobs 2 and 3. Adjust the two cursors to enclose the area of interest and type I again. This will produce the printout of a number on the Teletype. This number is proportional to the area surrounded by the cursors. In order for this digital integral to be meaningful, it is, of course, necessary to integrate two or more areas and compare them. A rough calibration can be performed on a displayed spectrum, without plotting it, by typing I, adjusting cursor 2 to the point of interest, typing I again, ignoring the number printed and typing K. Since the x-position of the calibration dot is also controlled by knob 2, the position of the dot will be the same as that of the cursor. The frequency can then be recorded from the frequency counter. The Display Mode is re-entered by typing CTRL/O and the RETURN key.
CTRL/O	Means "whatever you are doing - stop!" and has the effect of halting several operations performed by commands from the Display Mode. If the computer is punching or typing the contents of memory in decimal or binary form, the computer does a CR/LF and returns to Display Mode. If a smoothed spectrum, an analog integral, a first derivative, or an analog integral of a smoothed spectrum is being displayed, the computer does a CR/LF and returns to the unsmoothed, unintegrated display. If the computer is plotting, it ceases plotting and prints TURN OFF PLOTTER. Typing the RETURN key shifts to the display of whatever data was being plotted. Smoothed or integrated data is not destroyed unless CTRL/O is typed again. If CTRL/O is typed when normal, unmodified data are being displayed, the command will have no effect.
Other Commands	The following commands from the pre-normalization mode are also active: C — Contract X — eXpand CTRL/Z — Zero memory K — calibrate P — plot the spectrum currently displayed CTRL/C — return to the OS/8 Monitor N — Number of sweeps completed R — Restart G — Go - start averaging - used only after CTRL/Z

CHAPTER 7

ERROR RECOVERIES

This Chapter lists methods of recovering from errors made while running the NMR AVERAGER.

7.1 AVERAGING SECTION ERRORS

Error		Recovery
Typed the wrong	{ number of sweeps samples sweep time	Type RUBOUT. Teletype types ?, and does a CR/LF. Enter a new value.
REPEAT?	answered with Y	Restart at 200, or if already sweeping, interrupt with S or CTRL/H. Change by restarting with R command.
In answer to TYPE RETURN:		
Typed LINE FEED so memory was not zeroed		Interrupt with CTRL/H then zero with CTRL/Z and type G.
Typed carriage RETURN instead of LINE FEED so memory was accidentally zeroed		None.
Sweeping started inadvertently:		
If you want the result		Type S to get to Display
If you do not want the result		Type CTRL/H, CTRL/Z and, when ready, G.

7.2 Display Section Errors

Error	Recovery
Typed too many C commands	Type an X for each extra C.
Typed too many X commands	Type a C for each extra X.
G was typed inadvertently	Type S if you want data in memory, CTRL/H if you do not. Type CTRL/Z to erase, and G when ready.
R was typed inadvertently	Answer Ø to SWEEPS= to return to the display.
P was accidentally typed	Type CTRL/O and the RETURN key to return to display.
K was accidentally typed	Type CTRL/O and the RETURN key to return to display.
Plot rate is too fast (or too slow)	Type CTRL/O, the RETURN key, and then P to start over or type P to restart the plot immediately.
Spectrum has been filtered too much	Type CTRL/O, the RETURN key, then E to start filtering again.
I was accidentally typed	Type I again to return to display.
B or T was accidentally typed	Type CTRL/O and the RETURN key to return to display.
CTRL/Z or CTRL/E was accidentally typed	If binary tape of data has been punched, reload it with the Binary Loader. If not, recovery is not possible.

CHAPTER 8

MAINTENANCE OF NMR AVERAGER FOR THE LAB8/E

8.1 Introduction

NMR AVERAGER is a signal averaging program specifically designed for cases where the x-axis DAC voltage can be used to sweep the laboratory instrument, in this case an NMR spectrometer. It acquires 1024 data points in double precision, and will plot and display them without further action. Other routines are accessible by normalizing the data to single precision, and using the space gained as a display buffer.

It differs from the usual signal averager in that it can sample each of the 1024 data points up to 20 times before proceeding. Given that the computer is sweeping the spectrometer, the result is that each of the 20 samples are taken at the same point in the spectrum rather than the sequential points obtained when the spectrometer continues to sweep independently. Under these conditions this corresponds not to boxcar averaging, but to a form of digital filtering which is referred to here as multiple sampling. To better understand the following paragraphs, obtain a listing by running source tapes DEC-LB-U31A-UA, DEC-UB-U32A-UA, and DEC-LB-U33A-UA available from DEC's Software Distribution Center.

8.2 Storage

NMR AVERAGER occupies locations Ø-3377. Locations 36ØØ-7577 are used to store the double precision data before normalization. The high and low order halves of each word are stored separately as two lists, with the low order words starting at 3600 and the high order words starting at 5600. After normalization, locations 3600-5577 are the display buffer, and 5600-7577 contain the normalized data.

8.3 Startup

The program is loaded as usual with the Binary Loader and starts at 0200. Each message is unpacked using the character unpacking routine. The program asks for the number of sweeps and, if zero, jumps directly to the pre-normalization display. It asks for the sweep time in seconds and rejects zero. It asks for the number of samples per point, accepting only values between 1 and 20. The pause time is acquired and any value between 0 and 40 seconds is accepted.

8.4 Calculation of the Pause Time

If PAUSET is non-zero, it is multiplied by 100_{10} and its negative saved in PAUSET. This value is loaded into the clock buffer register and the clock is set to tick at 10^{-2} seconds.

$$\text{number of ticks} = \text{PT sec} \times 100 \text{ ticks/sec}$$

8.5 Calculation of the Sweep Time

The minimum time between clock ticks equals the shortest sweep time divided by the number of points times the number of samples per point:

$$t_{min} = \frac{1 \text{ sec}}{(1024 \text{ points})(20 \text{ samples/point})} \cong 5 \times 10^{-5} \text{ seconds/sample}$$

Thus, the base clock rate is 5×10^{-5} seconds per sample. The actual number of ticks per sample for a given clock rate is calculated as follows:

number of ticks/sample =

$$= \frac{(T \text{ sec/sweep})}{(5 \times 10^{-5} \text{ sec/tick})\ (1024 \text{ points/sweep})\ (S \text{ samples/point})}$$

$$\cong \frac{T\ (2 \times 10^{4})}{S\ (10^{3})} \text{ ticks/sample} = \frac{20T}{S} \text{ ticks/sample}$$

This double precision number is then divided by 1000_{10}. The result is the number of 5×10^{-2} second ticks per sample, and its negative is stored as DELAY2. The remainder is the number of 5×10^{-5} ticks per sample and its negative is stored in DELAY1. During actual sampling, both counters are incremented to 0 between each sample.

8.6 Sweeping and Sampling

The sweep and sample counters are set, the clock started at 100 KHz, and the clock buffer set to -5. The x-axis register is loaded with 7000 and displayed until the clock overflows. The clock overflow is counted by DL1. When DL1 reaches zero, DL2 is tested. If it is non-zero, the counter TCNT, initially set to -1000_{10}, is incremented. When both DL1 and DL2 are zero, an A/D conversion is initiated.

The A/D conversion is taken from the channel specified by CHANEL. CHANEL is set to zero in the released version of the program, but it can be changed by hand and a new tape dumped using a binary punch routine.

Following the A/D conversion, the value is tested for sign and added into the double precision storage in the usual manner. The sample per point counter AGN is incremented and, if zero, the pointers and x-axis display are advanced using the subroutine PADV. If AGN is not zero, another sample is taken at the same output voltage.

8.7 Interrupt

The sweep routine does not use the interrupt facility, as it would take more instructions to detect a clock interrupt, and the sweep rate could not quite make it at the one second rate. After each set of samples are taken on a given point, the keyboard flag is tested, and if it is set, a character is read. If it is S, the register STOP is set to one. If the struck character is CTRL/H, the sweeping is interrupted immediately and the pre-normalization display begun. Any other character is ignored and sweeping is resumed.

8.8 Pre-Normalization Display

During this display the low order words are loaded one-by-one into the y-axis register and displayed. When the display is contracted, this causes an arithmetic right shift of each double precision word CCNT times, where typing C increments CCNT once. The low order result of this rotation is loaded into the y-axis register. The x-axis is incremented by one after each display by the routine PADV.

8.9 Contraction and Expansion

The register CCNT is incremented by one each time C is typed and decremented by one each time X is typed. If CCNT becomes negative, the rotation is to the left instead of to the right. Rotations are performed by the routine pointed to by CONTRC.

8.10 Binary Punch

The binary punch routine is called by B, and it produces a tape of locations 3600-7577 with a load address of 3600 and a checksum at the end. The double precision storage is not reshuffled to the standard high-low order. This tape can, of course, be reloaded using the standard Binary Loader.

Entering the routine asks the question, H.S. PUNCH?, which if answered by Y sets the register HPUN to 1. The actual punching is done by BPUNCH which tests the register and uses the high speed punch if HPUN = 1. It also allows interrupts from the keyboard. If CTRL/O is typed, the punching halts and the pre-normalization display returns.

8.11 Decimal Typeout

The contents pointed to by HIWD and LOWD are loaded into HISAV and LOSAV and typed out using the signed decimal point routine Digital-8-25-U.Sym. The numbers are typed eight to a line, separated by commas. As in the binary punch routine, either punch can be used.

8.12 Normalization

When CTRL/E is typed, the program counts the number of contractions to date, and subtracts 2. This corresponds to the difference between the 1Ø-bit display and the 12-bit memory word. Each double precision word is then rotated this many times right. If CCNT-2 is negative, no left rotations are performed. The array starting at 3600 becomes the display buffer and the normalized data is stored at 5600-7577. Commands operating on the data, such as smoothing, integration, differentiation, and vertical shifting, place their results in the display buffer, leaving the original data untouched. The flag EDONE is set to 1, so that the program can test whether normalization has been performed.

8.13 Plotting

The clock is set to tick at the rate of 1ØØ Hz. Knob 2 is read and biased to a positive number by the addition of $1ØØ1_8$. This value is converted to its negative and loaded into the clock buffer register.

The next point in the display buffer (if normalized) or the next double precision word, appropriately contracted (if not normalized) is obtained, loaded into the y register and the display intensified until the clock overflows. Then the knob is read again, the next data point is loaded into the buffer, and the process continues until all 1Ø24 data points have been displayed.

8.14 N Command

The location LOSAV is loaded with the sum of NSWP and SWEEP. Location HISAV is cleared. NSWP is minus the number of sweeps remaining and

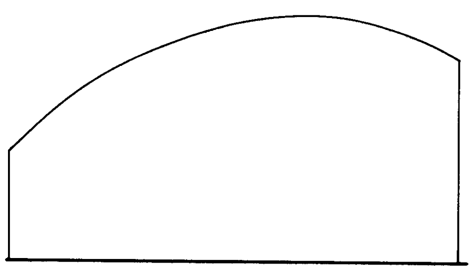
SWEEP is the total number of sweeps asked for. Their sum is the number completed. This value is printed by SDPRNT as the number of sweeps.

8.15 Integration

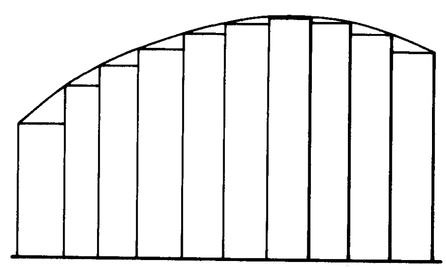
The same subroutines are used by both A and I to calculate the actual integral. Different driver routines are used, however. The integration itself will be discussed first.

Integration under a curve can be simulated by the rectangular rule, which states that the area under the curve is approximately equal to the sum of the areas of very small rectangles drawn under the curve, or

$$\int_{0}^{1024} B(x)\,dx \qquad\qquad \sum_{x=1}^{1024} x\, B(x)$$



Integral = Area under the curve



Integral ≅ sum of the areas of the rectangles

The integration routines assume that the display on the scope is properly contracted, and that it has been shifted so that it does not "wrap around" from top to bottom.

8.16 Analog Integration

Analog integration in this case refers to the simulation of the RC integral of a varying voltage, using digital means. It proceeds as follows:

a) The number of times the display has been shifted vertically from zero is found in UCNT and the buffer rebiased by this amount.

b) The sum of all buffer words is calculated and stored in INTHI-INTLO.

c) INTHI is loaded into the AC and it is rotated left until the first non-zero bit is found. If no non-zero bit is found, the same test is performed on INTLO. This number of rotations is subtracted from 15_{10}. The difference is the number of times the sum of the buffer words must be rotated so that the final sum will be on scale for a 10-bit DAC display. The negative of this value is stored in RSET.

d) The sum of the buffer is again calculated a word at a time. After each word is added, the sum is copied into HITEMP-LOTEMP and this pair of words rotated right -RSET times. This right-shifted sum is contained entirely in LOTEMP and each time is loaded back into the buffer location just examined.

   Thus, when this operation is complete, the display buffer contains the running sum of all data points in the original display - so divided that the total sum is never greater than the maximum value of the display.

   This sum of data points is equivalent to the rectangular rule integration with the x-distance assumed equal to one.

8.17 Digital Integration

The buffer is displayed once and then Knobs 2 and 3 are read. A cursor is displayed at the x-value represented by each knob and 61 points above to 61 points below the y-value of the buffer at that point.

When I is typed again, these cursor values are converted to buffer addresses

Buffer Address = Knob value + 777 + LOSTRT

and the points between these values summed and typed out using SDPRNT.

8.18 Filter

The equation $y'(x) = 1/4\ (y(x-1) + 2y(x) + y(x+1))$ is applied to points 2 through 1023. This is an extremely simple routine since 1/4 and 2 can be represented by bit rotation alone. The result of this calculation is placed into the buffer, and the double precision memory is untouched. Since neither x_1 nor x_{1024} are operated on (since there is no x_0 or x_{1025} to use in the calculation), there is some tailing of the baseline after an extensive number of filterings.

8.19 Calibration

The calibration routine basically controls the x and y coordinates of a point by reading Knobs 2 and 3. The added features are those designed to make the voltage as close as possible to that produced during the actual sweeping. This is accomplished by re-reading the two knobs only as often as DL1 reaches Ø, and all multiple samples have been "taken".

APPENDIX A

LOADING PROCEDURES[1]

A.1 READ IN MODE (RIM) LOADER

The RIM Loader is a program used to load the Binary Loader. The RIM Loader must be toggled into memory using the switches on the computer console.

Load the RIM Loader as follows:

a. Set ØØxx in the Switch Register (xx=field in which RIM is to be loaded) and press EXTD ADDR LOAD

Follow the procedure below for each location listed.

a. Set the Switch Register (SR) to the location value.
b. Press ADDR LOAD switch on the computer console.
c. Set the SR to the appropriate reader instruction value.
d. Press DEPosit switch. Repeat steps c. and d. until loading is completed.

Location	ASR33 Reader	High-Speed Reader
7756	6032	6014
7757	6031	6011
7760	5357	5357
7761	6036	6016
7762	7106	7106
7763	7006	7006
7764	7510	7510
7765	5357	5374
7766	7006	7006
7767	6031	6011
7770	5367	5367
7771	6034	6016
7772	7420	7420
7773	3776	3776
7774	3376	3376
7775	5356	5357
7776	0000	0000

[1] Refer to Introduction to Programming, Volume 1, for detailed information.

A.2 BINARY (BIN) LOADER

The BIN Loader is a program used to load the Advanced Averager program into memory. The BIN Loader tape is loaded after the RIM Loader. To load the BIN Loader, follow the procedure below:

a. Put the BIN Loader tape in the reader (always put leader/trailer code over reader head, never blank tape). Set ØØxy in SR (x = field in which RIM was loaded, y = field in which BIN is to be loaded) and press EXTD ADDR LOAD key.

b. Set SR to 7756_8 (the starting address of the RIM Loader).

c. Press ADDR LOAD on the computer console.

d. If the low-speed reader (ASR33) is used, turn the Teletype control knob to LINE and set the reader switch to START. If using the high-speed reader, set the high-speed reader switch to ON.

e. Depress the CLEAR and CONT switches on the computer console. Tape should begin reading in. If not, check the RIM Loader and repeat steps a. to d.

f. After the program is read, depress the HALT switch on the computer console. BIN is now loaded.

APPENDIX B

RUNNING THE LAB8/E PAPERTAPE ADVANCED AVERAGER FROM DISK OR DECTAPE (4K DISK MONITOR SYSTEM)

To run the LAB8/E Advanced Averaging program from disk or DECtape, use only tapes DEC-8E-AAPTA-A-PB and DEC-8E-AAAPA-A-PB. These tapes contain the necessary changes to enable the programs to run from disk or DECtape.

DEC-8E-AAPTA-A-PB is analogous to DEC-8E-AAP1A-A-PB in that it contains all four sections of the Advanced Averager (separated by leader/trailer). This tape should be used for operation of the program without control tapes. DEC-8E-AAAPA-A-PB is an expanded version of Section II. This should be saved separately and used only in conjunction with control tapes.

B.1 SAVING THE PROGRAM ON THE MASS STORAGE DEVICE

DEC-8E-AAPTH-A-PB

Save each section of the program as system programs. Consult the monitor document for your system for the specific procedure.

The locations to be saved in each section and the starting address for each section are:

	Locations to be Saved	Starting Address
Section I	Ø-277; 6ØØ-7577	6ØØ
Section II	64ØØ-7577	653Ø
Section III	64ØØ-7577	7ØØØ
Section IV	Ø-177; 64ØØ-7577	64ØØ
DEC-8E-AAAPA-A-PB		
Section II	Ø-177; 626Ø-7577	653Ø

NOTE

Save DEC-8E-AAAPA-A-PB as a binary file whose starting address is specified at load time.

B.2 RUNNING THE ADVANCED AVERAGER FROM DISK OR DECTAPE

To run the Advanced Averager from Disk or DECtape, call Section I into core. If the program was saved as a system file, it will start itself. Run Section I as described in this manual. At the end of

Section I, control will return to monitor. Call in Section II and proceed as described. Do this for all sections.

If it is desired to run the program using a control tape, call the system loader program and load Section II first; then load the control tape from the reader. Specify a starting address of 653Ø. Proceed as described in the manual.

APPENDIX C

SCHMITT TRIGGER

The Schmitt triggers of the LAB8/E are part of the DK8-ES real time clock. They can interact with the clock in many different ways (see description of the DK8-ES in the LAB8/E User's Handbook, DEC-LB-HRZA-D for details).

The Schmitt trigger circuit produces standard, predictable levels as a result of external activity at their inputs. The trigger will set (or fire) if it sees the input go through a .3 volt excursion and then cross a preset threshold. The direction of crossing is set with a switch. The threshold is variable between ±5 volts and is set with a knob. The .3 volt excursion, threshold crossing, and return crossing may take a minimum of 2 microseconds. In other words, a 1ØØ nanosecond pulse through the threshold and back will not fire the trigger. Re-cross is not necessary to fire the trigger.

Detailed specifications for the Schmitt trigger are:

Nominal input voltage range	+ or - 5V.
Input type	Differential
Input resistance	5ØK
Minimum duration input pulse	2µs
Maximum permissible input voltage	+ or - 5ØV
Hysteresis	Ø.3 V
Common Mode rejection	35 db
Propagation	6ØØ ns
Input threshold	Variable between + and - 5V
Source	+, -, or line frequency zero crossing.

APPENDIX D
EQUATIONS FOR STATISTICAL CALCULATIONS[1]

The 95 percent confidence limits are the following:

$$C_i = 2S_i / N^{1/2}$$

where N is the number of sweeps and S_j (standard deviation) is

$$S_i = \left\{ 1/(N-1) \left[\sum_{i=1}^{N} (A_{ij})^2 - (\sum A_{ij})^2 (1/N) \right] \right\}^{1/2}$$

where A_{ij} is the value of the j^{th} point of the i^{th} sweep.

The trend function is:

$$T_i = \frac{\sum_{i-1}^{N} (A_{ij} - A_{(i-1)j})^2}{(N-1) S_i^2}$$

[1]The reference for these equations is E.L. Crow, F.A. Davis, M.W. Maxfield, Statistics Manual Dover Publications Inc. N.Y., N.Y.

APPENDIX E
DATA POINT STRUCTURE

SECTION II

	Representation	Word Order [1]
Sum (of jth point for N sweeps)	$\alpha_j = \sum_{i=1}^{N} A_{ij}$	Low High
Sum of Squares (of jth point for N sweeps)	$\beta_j = \sum_{i=1}^{N} (A_{ij})^2$	Low Mid High
Value of jth point in (i-1)th sweep	$A_{(i-1)j}$ where $A_{0j} = 0$	Low
Sum of Squares of Successive Differences (of jth point for N sweeps)	$\gamma_j = \sum_{i=1}^{N} (A_{ij}-A_{(i-1)j})^2$	Low Mid High

[1] Values are stored in either single, double, or triple precision as indicated by the Word Order.

Sections III and IV

	Representation	Word Order[1]
AVERAGE (of ith point for N sweeps)	α_i/N N = Number of sweeps	High Low
CONFIDENCE LIMITS (of ith point for N sweeps)	$C_i = 2S_i/N^{1/2}$ where, $S_i = \left\{ 1/(N-1) \left[\beta_i - (\alpha_i^2/N)\right] \right\}^{1/2}$ (S_i is standard deviation)	High Low
TREND (of ith point for N sweeps)	$T_i = K\gamma_i/(N-1)S_i^2$ where K = 1000 (output scale factor)	Low

The values of AVERAGE, CONFIDENCE LIMITS, and TREND are stored in contiguous locations for each data point. Locations are allotted only for those functions requested in Section I (compiler). If CONFIDENCE LIMITS and TREND were not requested, a_j would be immediately followed by a_{j+1}; that is, all values pertaining to one point are kept together in the format shown. These would be followed by the values for the next data point. There are no unused core locations between data points.

[1]Values are stored in either single or double precision as indicated by the Word Order.

APPENDIX F
HOW TO ACCESS DATA

Core Map

For a generalized core map of the LAB8/E Averager, see Appendix G (Memory Organization). The structure pictured is for the Advanced Averager. A and B sweeps refer to high and low resolution capability of the Advanced Averager.

The locations of the areas can be found in certain locations on page zero. Below is a table stating the name of the area and where its location can be found. Example: The starting location -1 of the job lists is found in location 23. C(23) + 1 should be read, "the contents of location 23 plus 1."

Area	Basic Averager	Advanced Averager
Job List	C(23) + 1	C(23) + 1
MPLXR List	---	C(56) + 1
Chan Display List	C(24) + 1	C(24) + 1
ADC Buffers A Sweep	C(47) + 1	C(47) + 1
B sweep	---	C(54) + 1
Calculation (storage) Buffers	the location of these is obtained from job list (explained below)	

The LAB8/E Averaging Programs utilize what is called a "job list". The job list contains seven words which define the average. Each average has a job list particular to itself. If three averages are taken, three separate job lists are created by the compiler. They would be stored sequentially, starting at address 230_8. At the end of the last job list, there is a word containing zero to denote the last job list. This list is used whenever the data is referenced by any part of the program.

230_8	Word 1:	Bit 0:	0 for high resolution sweep; 1 for low
		Bits 1-5:	analog channel # pertaining to this average
		Bit 6:	always 1 to prevent possiblity of zero word
		Bits 7-11:	position of this average in sampling list
	Word 2:	Bits 0-3:	job type (Avg. only = 0400); Conf. lim.= 1000; (TRN = 1400)
		Bits 4-11:	sort code
	Word 3:		Minus the number of data points in the data block containing this average (all the numbers pertaining to an average are stored in a data

Word 3 cont.	block). Each average has its own data block (does not include points that are in block extension).
Word 4:	CDF (change data field) instruction for the data block.
Word 5:	address-1 of this block
Word 6:	defines the scaling of the display of this average
Word 7	position on a display scope where x = 0 and y = 0 for the averager

Of interest here is Word 5 (location 234_8). This gives the location -1 of the data block which pertains to this average and its associated numbers. Contents of Word 5 + 1 is the address of this data block. The first location contains the number of sweeps taken. The following locations contain the sum(in double precision), the confidence limits (in triple precision), and the trend (in triple precision + 1 location) for each point. A data block during Section III would look something like the following:

```
c (word 5) = 1263
1263\
   4\           start of block (# of sweeps)
   5\ lo }      order of sum of all point 1's
   6\ hi }
   7\ lo  }
1270\ mid }     order of the sum of the squares for
   1\ hi  }     all point 1's
   2\           value of last point:  necessary in trend calculation
   3\ lo  }
   4\ mid }     order of sums necessary for trend calculation
   5\ hi  }
   6\ lo }      order of sum of all point 2's
   7\ hi }
etc.
```

In Sections III and IV the arrangement of the data changes. From the raw statistics gathered during the averaging section the final statistics are calculated. Sections III and IV contain the data in the following format:

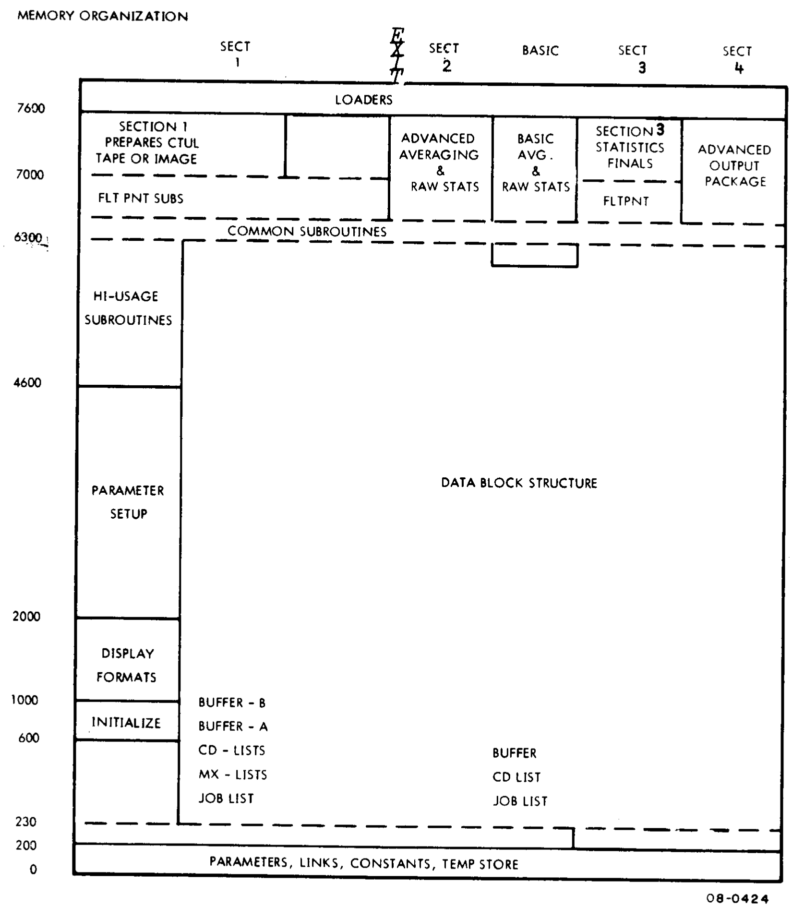
Starting address of the data block is the same

```
1263\
   4\           # of sweeps
   5\ hi }      order of sum of point 1
   6\ lo }
   7\ hi }      order of confidence limits for point 1
1270\ lo }
   1\           1000 x trend for point 1 x 2^12
   2\ hi }      order of sum of point 2
   3\ lo }
```

Note the reversal of the order.

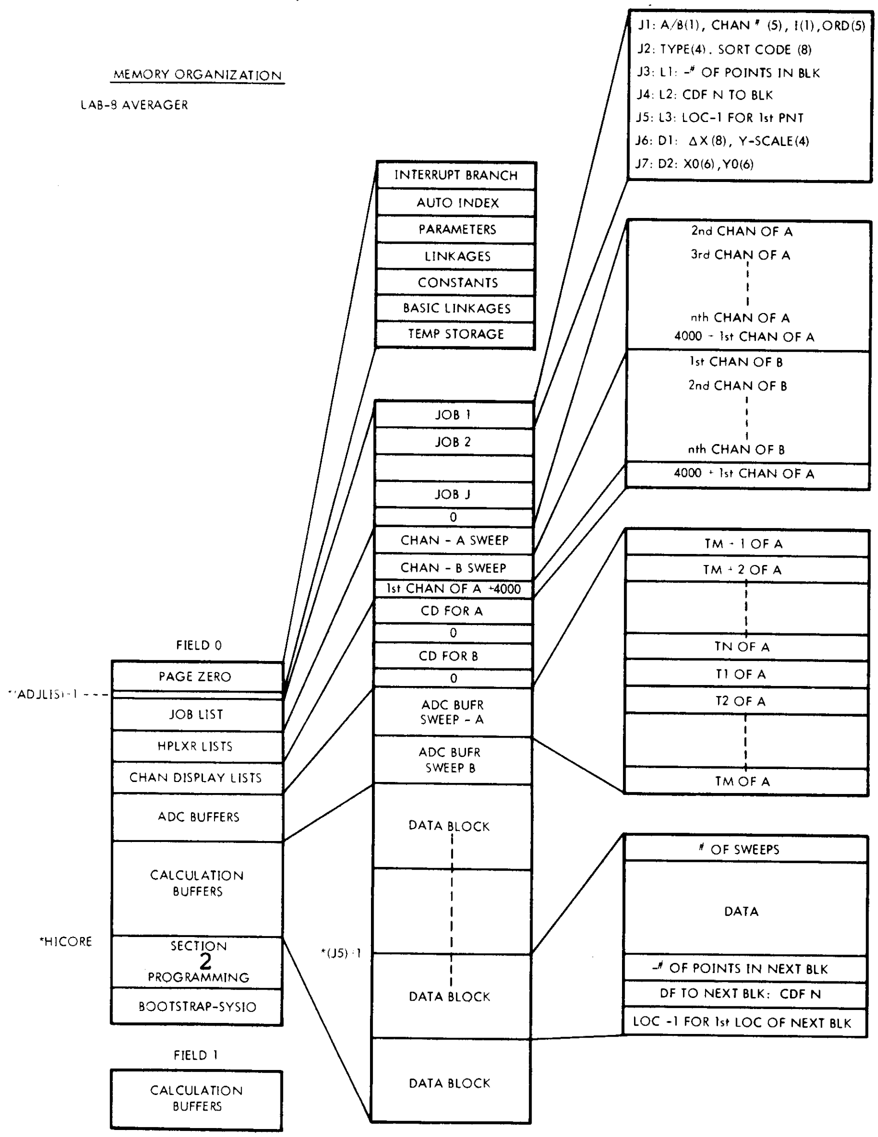
MEMORY ORGANIZATION

Memory Organization



MEMORY ORGANIZATION

LAB-8 AVERAGER



APPENDIX H

ATTACHMENT OF AN ANALOG X-Y PLOTTER

Plotting of a displayed waveform may be easily accomplished by the use of an analog plotter. Since the display is itself an analog device, the information carrying signal is available at the X and Y input connections of the scope. The plotter should have an effective analog range of ±5 Volts.

To attach the plotter to the scope, simply run two wires from the plotter to the scope inputs for each of the X and Y connections and their respective grounds. The wires need not be shielded. Now, whatever voltage is available to the scope is also available to the plotter. To plot a waveform, one slow display cycle should be gone through.

<u>CAUTION</u>

While the display is running at its normal speed, the plotter should be disconnected or turned off because the pen will shake furiously attempting to follow the display.

APPENDIX I

DATA BLOCK STRUCTURE

To provide for multiple field machines, the LAB8/E Averager saves data in linked blocks. In general, LAB8/E will end one block and link to another whenever it comes to an end of field. Linking is done in Section I.

An example is shown for N sweeps and M data points broken into two data blocks. In the example, the block bridges Field Ø to Field 1.

-# of points in 1st block: -L
CDF to 1st block: CDF Ø
Location -1 of 1st word Block -1

Words J3-J5 of Job List Entry for this Job.

Block 1 in Data Field Ø

BLK1--

# of sweeps: N 1st data point 2nd data point 3rd data point
. . . .
Lth data point
-# of points in block extension: L-M
CDF to block extension: CDF 10
Location -1 of 1st word of block extension: BLK EXT-1

BLK1 EXTENSION

L+1 data point
. . . .
Mth data point
-# in next block: Ø

BLOCK 1 EXTENSION

in Data Field 1

Each data point consists of information pertaining to the mean and, if requested, standard deviation and trend for the signal at that point in time. The number of locations used for each of these functions for each data point and the contents of each location depends on which section of the LAB8/E Averager is in core.

NOTE

> If the data block does not have an extension, there is a zero word for termination. This terminator must appear at the logical end of every data block or data block extension.

INDEX

APPENDIX J

THE GENERATION AND MODIFICATION OF CONTROL TAPES FOR THE LAB8/E BASIC AVERAGER

When the LAB8/E Basic Averager is used, it must be preceded or followed by control tapes. The purpose of these tapes is to define the specific parameters involved in the average to be taken. The Basic Averager Program is general in that it can be adapted to a variety of situations. The control tapes accomplish specific adaptation.

There are eight different control tapes available from the DEC Program Library. These allow the maximum number of points to be taken, in all cases, as the number of channels and types of statistics are varied. In some cases, the user may want to decrease the number of points taken to realize a shorter period of stimulus repetition or to allow himself core for his own programs or he may want to sample some number of channels not covered by the existing control tapes. If the machine contains more than 4K of memory, the user may wish to fully utilize the core available to him.

All of the situations above are easily reconciled once the user understands what a control tape is and how it can be generated or modified. (Although it is possible to generate a control tape using no other tools than the Symbolic Editor and PAL III Assembler, it is easier to modify existing control tapes so that they will conform to the new situations.)

Before attempting modification, a knowledge of the Symbolic Editor and PAL III Symbolic Assembler should be obtained.

Briefly, what is done is the following: Using the Editor program, the necessary changes are made in the ASCII control tape that is most similar to the desired product. An ASCII tape containing the changes is punched, and then it and DEC-LB-UØØA-PA (a symbol reference tape) are assembled (i.e., translated into binary code) using the PAL III Assembler. PAL III then punches out the modified control tape which can be used with the Basic Averager.

EDITING

Paper tapes marked DEC-LB-UØ1A-PA through DEC-LB-UØ8A-PA are the ASCII versions of the control tapes that come with the LAB-8 system.

These are available through the Software Distribution Center (DEC - Maynard, Mass.).

NOTE

> Programs that are written in the ASCII language are readily understandable to us, the users, since they are written using alphanumeric characters and can be heavily commented. Because of the straightforward nature of ASCII, it is used exclusively in writing, debugging, or modifying programs. When the program is in its corrected form, it must be translated into the language understood by the computer (i.e., binary). Once a binary version is created it is used to instruct the computer.

To edit the ASCII control tapes, load the Editor Program and have it read the desired ASCII control tape. For clarity of discussion, a listing should be created at this time (type L and the RETURN key).

The first item printed will be:

```
/ (n) CHAN BASIC (AVGR) CONTROL TAPE (UØ-A)
                 (SD)
                 (TREND)
```

The slash denotes a comment following.

The next item printed will be DECIMAL; this will tell the Assembler that all numbers following should be considered decimal numbers (i.e., numbers of the base 10) until the command, OCTAL, is given.

NCHAN = (1, 2 or 4)	The number should be changed if a different number of channels are to be sampled.
POINTS =	Change if a different number of points are desired.

As with all systems, there is a limit to the number of points that can be taken, because there is a finite amount of core available for storage. In the Basic Averager there are 3027 words available for data storage. The LAB8/E averaging programs are set up in such a manner that the incoming data is first stored in a temporary buffer (one core location per point). The values are then added to the sum, which is saved in double precision (two core locations per point). If confidence limits are requested, the value is squared and then added to the sum of the squares, which is saved in triple precision (three core locations per point). The trend function requires four additional locations per point.

In summary, for one point the following core is required:

	Core locations
Temp Buffer	1
Sum Buffer	2
Conf Lim Buffer	3
Trend Buffer	4

To take an average with confidence limits and trend would require 10 core locations per point. If there were 3000 locations available, a maximum of 300 points could be taken using a 4K machine. If two channels were involved and both required data for confidence limits and trend, 150 points/channel would be maximum.

JLIST=

The LAB8/E Averager programs use what is known as "job list processing". Each average taken has a job list associated with it that defines the parameters of that particular average (detailed description follows). Job lists are seven words long. If there is more than one, they follow each other sequentially in memory. Following the last list there must be a word containing zeros. The symbol "JLIST" designates the total length of the string of job lists. This is equal to 7*nchan + 1 and must be specified as the result of a summation or one decimal number.

SUMLEN=

This statement defines the core required to store the sum associated with 1 channel. This is always equal to 2 times the number of points or POINTS + POINTS.

SDLEN=

This statement defines the core required to store the data for calculating the Standard deviation or confidence limits for one channel. This should remain unchanged.

TRNLEN=

This defines the core required for the trend data. Do not change this.

BLKLEN=

This defines the amount of core necessary to store all the data for one channel.

Stored data is kept in blocks, each average having its own block. The length of the block is determined by the number of data points and the job type (associated statistics, if any).

Job Type	BLKLEN
AVG	2*#Points + 1
AVG & Confidence Limits	5*#Points + 1
AVG, CLIM and TREND	9*#Points + 1

Again, the length must be specified as the result of a summation or one number (it cannot be specified as the result of a multiplication). One location is needed at the end for an end-of-the-block word.

```
    OCTAL
```

The following numbers are to be considered in octal.

The next line of the listing will be the following:

```
    *MEMTOT
        0
```

The asterisk signifies that MEMTOT is a location that will have its value specified later by the symbol reference tape. This enables us to tell the assembler that we want to place a number in a particular location in memory. The zero following is that number. This indicates the number of additional 4K stacks of memory on this machine (i.e., an 8K machine would have *MEMTOT).
1

```
    HICORE-LOCORE-4
```

Defines the amount of core available for data

```
    *ADJLIS
        LOCORE-1
```

Specifies the address -1 of the start of the job list. It is used by the program to locate the job list.

```
    *ADCHNL
        CHLIST=LOCORE + 3LIST -1
        CHLIST
```

Specifies the address -1 of the start of the channel display list.

```
    *SMASK
        S1
```

This location contains the sync mask. This is used to determine if the proper Schmitt trigger has fired. The Basic Averager is set up to respond to Schmitt trigger one.

```
    *FAST
        ISZ MCHAN or NOP
```

This instruction is used in the A/D sampling routine. If only one channel is sampled NOP should be placed there. More than one channel requires ISZ MCHAN.

```
    *KMCHAN
        -NCHAN
```

Sets up a location to contain minus the number of channels that was specified above.

```
    *ASI
      -1
```

This is needed for the compatability of sections 3 and 4.

*KADC

300

This is the mode setting for the A/D converter. 300 is used for multiple input, 200 for single input.

*KMODE

5640

This is the mode setting for the clock. This will enable the clock to tick at one microsecond a tick, to reset itself, and to enable the A/D converter to start on its overflow.

*SAMA

-POINTS

This is minus the number of samples to be taken

*NCHA

NCHAN

This is the number of channels being sampled.

*ADBUFA

This tells the program where to store the A/D samples. This is the temporary buffer and it proceeds the channel display list.

*KBLA

This is equal to minus the total number of samples taken during a sweep of all of the channels.

The next area that may be edited is the job list itself. Until this point, all variables that have been defined have been common to all the averages in this experiment. Now, each average must have its particular parameters defined. The first list will always start at location LOCORE (0230_8). After the end of the last job list, a zero word is inserted to indicate the end of the lists.

Word 1:	Bit Ø	always = Ø
	Bits 1-5:	specify the analog input number pertaining to this average.
	Bit 6:	always =1
	Bits 7-11:	indicate the position of this average in the sampling list (usually the same as bits 1-5)

Word 2:	Average Type Code	
	AVG =	for average only
	SD =	for average and confidence limits
	TRN =	for average, confidence limits, and trend (using these abbreviations will lead to the correct translation)

Word 3:	-POINTS	This will assemble to be the negative of the number of data points in the block of this average.
Word 4:	CDF n ;	change data field instruction, CDF_n(=62n1), where n is the number of the data field (i.e., the block of 4K; n = 0 for 1st 4K; n = 10 for 2nd 4K) where the first data point for this average is to be stored.
Word 5:	ADCALn,	where n is the number of the average. First average is always zero. This is the address -1 of the beginning of the data block belonging to this average.
Word 6:	Display Word 1	(see below)
Word 7:	Display Word 2	(see below)

Channel Display List

The channel display list contains two words for each average. These are the same as words 6 and 7 of the job list. In assembling, the channel display list comes directly after the end-of-list word in the job list.

Display Word 1

The leftmost eight bits of display word 1 are the x-increment used between points in the display cycle. The first five of these eight bits are the integer part of the delta-x; the last three are the fractional part.

Since the display scope has a maximum resolution of 1024 points in each direction, it is useful to be able to use a fractional increment for full screen utilization. For instance, if the average contained 682 points, an increment of 1.5_{10} (00001.100_2, using 8 bits) would just fill the scope with the display. An increment of 1.5_{10} will display a data point every 1.5 x locations.

A simple formula can be used to determine the DELTAX value to use, given: a.) POINTS, and b.) PC, the percent or fraction of the screen that you wish to cover with the display:

$$DELTAX = \frac{1024 * PC}{POINTS}$$

This number should be translated into a binary integer and fraction and placed in the left most eight bits of the first display word.

The last four bits of display word 1 are the Y scale factor (YS) for the particular average. As each data point is displayed, its value is divided by 2^{YS}. Thus, for full height display, YS would be zero; for half height it would be one, etc.

Display Word 2

The first six bits of this word are X origin for the average display; the last six bits are the Y origin. Both of these numbers are multiplied by sixteen before being used as starting points for the display. This makes it possible to start the display anywhere on the screen. The first bit of the origin is interpreted as a sign bit so that it is possible to start below zero. For instance, a Y origin of -200_8 would imply a 50_8 as the low-order end of display word 2.

Generating the Display Words

Here is an example of how to set up the display words for a 500 point, two channel, average-only control tape. The averages are assumed to be displayed one above the other, the full width of the screen.

DELTAX: Since there are 500 points in each sweep, our formula gives DELTAX $=1024/500 = 2.00_{10} = 00010.000_2$, to 8- bit accuracy.

YS: Each average should cover half the screen, so each data point value must have been divided by 2^1. Therefore, YS should be 1. We now have display word 1 as

$000100000001_2 = 0401_8$
DELTAX YS

XZ: Since both traces start at the left-hand side of the scope, XZ will be -777_8 for both averages.

YZ: The first average must cover the top half of the scope ($0 - 777_8$) and its zero point should then be 400_8. This means that the second half of display word 2 will be 20_8 for this average. Similarly, the second average must have the Y origin at -377_8 and the second half of the word will be $-20_8 = 60_8$. The channel display list will now appear as:

```
/CHANNEL DISPLAY LIST
*CHLIST + 1

0401    /THESE WORDS ARE FOR
4020    /FIRST AVERAGE.
0401    /THESE ARE
4060    /FOR SECOND.
0000    /THIS IS END OF LIST.
```

The last symbol to be defined is the end-of-block symbol. This is a zero word at the end of the data block. Since there must be a zero following each data block, this is accomplished by the assembler. The address is ADCAL + BLKLEN + 1.

Possible Changes

The easiest change to make is the number of points taken in an average. The only changes necessary are POINTS and the DELTAX value. The assembler (PAL III) effects the consequent changes.

To enable the Basic Averager to use more than 4K of core is more difficult. The difficulty arises when data must be stored and retrieved in upper core. This is easily accomplished if the data blocks are in the lower 4K or upper 4K of core (i.e., a block is in one or the other but not both). Word 4 of the job list defines the data field of the first point in the data block. $ADCAL_n$ (the address -1 of that data block) directs LAB8/e within that data field. So, if a data block is in upper 4K, only the CDF instruction need be changed. The user must keep in mind that when addressing upper core (i.e., locations >7777) all addresses are relative to location 0000 of that block of 4K.

NOTE

If OS/8 is being used, page 7600 of all fields can not be used.

To define an address in upper core, the user must type in the command FIELD n, where n is the data field. All addresses specified thereafter will be located in that data field (the first 4K is field 0).

When a data block lies in both upper and lower core, the LAB8/E switches between data fields by a process called "linking." Linking is accomplished by a set of three linkage words at the end of the block in the lower field. These words inform LAB8/E of the number of data points (not core locations or values) that are needed to complete this average, the data field in which these points lie, and the location -1 of the first word in the next data block. These linkage words must be supplied by the user.

The upper limit of the data storage area in the lower 4K is 6277_8. The Basic Averager program is located above this. So, at some point before this limit, the linkage words must be inserted. (After assembly, the user should check to see that these words were placed in locations lower than 6277_8).

Note that word 3 of the job list pertaining to this average should show only the number of points found in the first part of the data block (i.e., it must not include the point count contained in the part of the block that is in upper core). The CDF instruction and ADCAL word of the job list should refer to the address and field of the first data point contained in the part of the block which is in lower core. The linkage words should contain the information on the points that are in upper core.

Example: Assume an extra 4K of memory

1 channel Avg, Conf. Lim, and Trend 600 points start with tape DEC-LB-UØ7A-PA

Changes:

1.)	Points = 600	
2.)	*MEMTOT$_1$	
3.)	In the job list;	Word 3 should be minus the number of points contained in lower core. In this case, 200 points will be stored in lower core. DECIMAL -200_{10} OCTAL
4.)	Word 6	DELTAX must be changed to correct the display for increased number of points . . . # of points = 600 for full-screen display. DELTAX = 1.705_{10} = 1.551_8 (for decimal-to-octal conversion see <u>Introduction to Programming</u> F1-4), so word 6 becomes: Delta X 000 011 010 000 = 0320 i.e. 1.5 + y scale y-scale remains zero
5.)		The first channel display word must also be changed to 0320.
6.)	Linkage Words	These three words must be located directly after the last value pertaining to the last point in this block. This location is determined by starting with ADCAL0. This is the address -1 of the first word of the data block. The first word of the data block contains the number of sweeps, and the following words contain the data. Therefore, the last location is equal to: ADCAL0 + 1 + (No. of locations needed for data storage) + 1

In this case, 200 points are stored in this block. Each point needs two words for the sum, three words for the confidence limits, and four words for the Trend; a total of nine words per point. Therefore, the address of the word following the data is:

$$ADCAL0 + 1 + 1800_{10} + 1$$

Linkage Word		
1	-400_{10}	/-No. of points in next block
2	6211_8	/CDF for next field
3	99_{10}	/location -1 of first word

Note

Linkage word 3 was chosen arbitarily.

Define these in the following manner:

```
DECIMAL
*ADCAL0 + 1 + 1800 + 1
-400
OCTAL
6211
DECIMAL
99
```

Now the end-of-block word must be inserted. It lies in Field 1, 400_{10} points or 3600_{10} words from the start of the buffer. There is no sweep count included in this block, so the location of the word following the last data word is:

(starting address -1; specified in linkage word) + 1 +(number of words used for storage)

In this case, the location is 3700_{10}.

Define this by the following:

```
FIELD 1     /Change Fields
DECIMAL
*3700
0
$
```

The dollar sign signifies the end of the characters to be assembled.

Once the user has familiarized himself with the procedure and logic of editing and assembling control tapes, he may go on to further, more elaborate modifications.

When all modifications are completed, a tape should be punched by typing P, RETURN, turning on the punch (high speed or low speed), and pressing CONTinue. After the tape is punched, type F and RETURN, then T and RETURN.

The output tape may now be assembled by any of the DEC assembly programs. The output of the assembler may be used as a control tape.

APPENDIX K

EXAMPLE OF PARAMETER SETUP

PST AND LATENCY

In the following example, the underlined portion of the dialog is generated by the program. The reply is not underlined.

1. Program started at location Ø2ØØ.

TRIGGER ↵	Check the Schmitt trigger thresholds. Carriage RETURN tells the program this has been done.
PST? Y↵	PST or Latency? PST chosen.
BINW= 5.Ø ↵	Bin width is set to 5.Ø milliseconds.
BINS= 1ØØ↵	There are 1ØØ bins in the histogram.
EPOCHS= 4Ø ↵	There will be 4Ø stimulus-response sequences accepted.
MINTIM= Ø↵	No minimum time constraints.

This dialog will produce a PST histogram. It will measure arrival times of responses from time zero (mintime = Ø) up to 5ØØ milliseconds (binwidth x # of bins) with a resolution of 5 milliseconds following the stimulus. Forty stimulus-response sequences will be accepted.

2. Program started at location Ø2ØØ.

TRIGGER ↵	Check the Schmitt trigger thresholds. Carriage RETURN tells the program this has been done.
PST? N↵	PST or Latency? Latency chosen.
ORDER= 5 ↵	The first five pulses following the stimulus should be recorded.
BINW= .5 ↵	First answer was not in correct format. Question is repeated.
↵	
?BINW= Ø.5↵	Bin width is .5 milliseconds or 5ØØ µsec.
BINS= 19ØØ↵	19ØØ bins exceed the maximum allowable. User is notified of this and the question is repeated.
CORE?↵	
BINS= 18ØØ↵	18ØØ bins are requested and accepted by the program.

EPOCHS= 1ØØ↵ Again the core limits are exceeded and the question is repeated.

CORE?↵

EPOCHS= 7Ø↵ 7Ø is accepted.

MINTIME= 1Ø↵ Minimum time is 5 milliseconds (1Ø x .5).

Here a latency histogram is requested. It will be a composite fifth order latency histogram (the histogram will be the frequency distribution of the first five pulses arriving following the stimulus). It will measure the time of arrival starting at 5 milliseconds after the stimulus (mintime = 1Ø x bin width) until 955 milliseconds after the stimulus (bin width x # bins + mintime) with a resolution of 5ØØ μsec. 7Ø stimulus-response sequences will be accepted before automatic transfer to output mode.

TIME INTERVAL HISTOGRAM

1. Program is started at location Ø2ØØ

TRIGGER ↵ Check the Schmitt trigger thresholds. Carriage RETURN tells the program this has been done.

BINW= 5.Ø ↵ Bin Width is set to 5.Ø milliseconds.

BINS=1ØØ ↵ There are 1ØØ bins in the histogram.

MINTIM= Ø↵ No minimum time constraints.

TIME\UNIT=2ØØ.Ø ↵ The basic time unit for timing the program is 2ØØ.Ø milliseconds.

#UNITS= 2ØØ ↵ There are 2ØØ of the basic units so the experimental run will last 4Ø seconds (2ØØ ms x 2ØØ).

APPENDIX L

IMPLEMENTATION OF USER COMMANDS FOR PST

A listing should be consulted before either of the user commands, CTRL/U or CTRL/I, is utilized.

The user commands are designed to be patched in and will do anything that the user can program. CTRL/U is a Group I command and will be executed during data acquisition. To utilize CTRL/U, change location 1356 from 124Ø to the location of the user routine. At the end of the user patch there should be a CLL CLA JMP I 121 (5521). This will return control to the main program.

CTRL/I is a Group II command to be executed only after data collection is finished. To implement CTRL/I, change location 1357 from 124Ø to the location of the user routine. Again, at the end of the routine there should be a CLA CLL JMP I 121.

NOTE

> The user patch for CTRL/U must not turn the interrupt ON or OFF. It also must not disturb the clock in any way.

To find free areas of core, consult a listing.

The MS version of TIH and PST used CTRL/I for CTRL/W. If a user command is needed in Group II the histogram size has to be made smaller and the command put in the command list.

APPENDIX M

HISTOGRAM BINARY OUTPUT FORMAT

When the CTRL/B command is given, a binary dump of the histogram being displayed will proceed on either the high or low speed punch. The dump will start with 64 leader-trailer punches (these are channel 8 punches). For a description of paper tape punching format, see Introduction to Programming, Vol. 1, 4-14.

After the first leader-trailer punches, the ID code will be punched in ASCII format. This will be followed by more leader-trailer. The core image is preceded by an address. This is signified by a 7-channel punch with the most significant 6 bits of the address on the same line followed by the least significant 6 bits on the next line. A core image follows the address.

The first core image is the display list. This gives the starting location -1 of zeroth histogram followed by the ending address, followed by the starting address -1 of the first order histogram, followed by the ending address of the first order histogram.

After the display list, the address of the histogram being dumped is punched followed by a core image of the histogram. After all information has been punched, a checksum is punched.

A checksum is a means of checking the validity of read-in. It is a 12-bit word which represents the sum of all of the punches in the tape that are not leader-trailer or ASCII information. It is formed by adding every line of punched data (7 channels) to a 12-bit word. Overflows are ignored, and the resulting 12-bit sum is punched at the end of the data. This is followed by more leader-trailer code.

When the binary loader reads a binary formatted tape, it keeps a running 12-bit sum of all punches between the end of the leader code and the checksum.

NOTE

The checksum punches are not included in the checksum. If the result is non-zero, an error on read-in has occurred and the tape should be read-in again.

APPENDIX N

SUMMARY OF COMMANDS FOR PST, TIH AND LATENCY HISTOGRAM PROGRAMS

Group I: Commands for Data Collection and Output

Command	Action
>	Increase order of display.
<	Decrease order of display.
CTRL/Z	Stop, zero buffers wait restart.
CTRL/A	Stop, return to alter parameters.
CTRL/Q	Quit taking data after next epoch.
CTRL/C	Return to Monitor.
CTRL/P	Panic stop.
U	Scale display up by power of 2.
D	Scale display down by power of 2.
E	Expand area between cursors.
B	Cancel all E commands, and show beginning display.
S	Sum n bins before display.
O	Negate all S commands - return original display.
Z	Return to unaltered display.
A	Complement axis - no axis switch.
C	Complement cursor - no cursor switch.
V	Complement view bar graph or point plot display.
CTRL/U	User Group I command.

Group II: Commands Cannot be Used during Data Collection

Command	Action
P	Plot the histogram being displayed.
K	Calibrate the plotter.
T	Type the data between the cursors.
CTRL/B	Do a binary core image dump.
CTRL/R	Retain the data, ready for restart.
CTRL/I	User Group II command.

APPENDIX O

IMPLEMENTATION OF USER COMMANDS FOR AUTO- AND CROSS-CORRELATION [1]

Two user commands are available without reassembly. These are CTRL/U and CTRL/O. CTRL/U is a group 1 command, that is, it can interrupt data taking, and CTRL/O is a group 2 command which indicates that it operates only after data taking has ceased.

CTRL/U, because it operates while data taking, should be a short, fast utility routine, because calling it could affect the timing of the correlogram. In normal mode, the clock is interrupt enabled, that is, whenever it ticks an interrupt results. Thus, in normal mode, CTRL/U could be interrupted in the midst of carrying out its task. This is why it is recommended that CTRL/U be fast. However, if CTRL/U is long, precautions must be taken not to affect the data and pointers used by the routines which actually perform the algorithm. This applies for Batch and normal modes.

These limitations need not apply to CTRL/O, because it can be called only after data taking has ceased. CTRL/O must not destroy vital flags or pointers, but has available temporary registers normally used by the algorithm routines.

How to PATCH in a User Program

The location which specifies which location to jump to in case of a CTRL/O is COMGO+1. The user should put the starting address of his CTRL/O routine in this location. NOTE: The routine is not called with a JMS, but a jump is made to it. Exit is made by one of the two following methods:

1) JMP I CHECK1 - This will return to the location following the call to the command checker. This should be used only by a routine that does not modify locations used by the correlator.

2) JMP I CONT3 - This restarts the display loop, causing all display variables to be reinitialized. A link to CONT3 is on page Ø.

The location which specifies the Starting Address of the CTRL/U command is COMGO+11_8. All other specifications are as above.

[1]It is assumed that the user has a listing (available from the DEC Software Distribution Center, Maynard).

The user, by looking at the core maps, can see which core is free for his use.

To get the octal values of the symbols used, consult a listing.[1]

Useful Subroutines

There are several useful subroutines which are available for user implementation. Many of these are in the form of pseudo-ops. A description follows:

Subroutine 1: Calling Sequence:

```
SAVE
Address
Return
```

Function: This routine puts the contents of the Floating Accumulator, locations EXP, HORD and LORD in the address pointed to by the word after the call.

Subroutine 2: Calling Sequence:

```
NORM
Return
```

Function: NORMalizes FAC (Floating AC).

Subroutine 3: Calling Sequence:

```
DIVF
Address
Return
```

Function: Divides the FAC by the Floating Point Word whose exponent is pointed to by CALL+1.

Subroutine 4: Calling Sequence:

```
SORTJ
LIST1-1
LIST2-LIST1
Possible Return
```

Function: This routine compares the contents of location CHAR with that of the entries in LIST1. If a match is found then a jump is made to the contents of the corresponding location in LIST2. LIST1 is terminated by a negative number. If a match is not found, then SORTJ returns to CALL+3.

Example:

```
SORTJ/SEE IF CHAR = "+";"-"; OR "!"
LIST1-1
LIST2-LIST1
JMP CONT/DOESN'T CONTINUE
LIST1,"+/PUT ASCII VALUES
"-
"!
-1/END LIST
LIST2,ADD/JMP TO ADD IF A"+"
SUB/JMP TO SUB IF A"-"
IOR/JMP TO IOR IF A"!"
```

[1]Available from Digital Equipment Corporation, Software Distribution Center, Maynard, Massachusetts.

Subroutine 5, 6: Calling Sequence: SETDAT
Return

Calling Sequence: SETUP
Return

Functions: These routines set up data for display routine. This data must be set up before displaying.

Modifies: NUMER, DENOM

Subroutine 7: Calling Sequence: DISPLAY
Return

Function: Displays the buffer once.

Subroutine 8: Calling Sequence: MESAGE
MES-1
Return

Functions: Types a CR-LF and then the string of 8-bit ASCII characters pointed to by CALL+1. The list is ended by a Ø. Return to CALL+2.

Example:

```
MESAGE/TYPE "GOO"
MES1-1
JMP CONT/CONTINUE
MES1"G
"O
"O
".
Ø/END
```

Subroutine 9: Calling Sequence: PRINTC
Return

Function: Prints (AC) if AC not = Ø, print (CHAR) if AC = Ø.

Subroutine 1Ø: Calling Sequence: READC
Return

Function: Reads a CHAR from TTY. Checks to see if it is a command, and then puts the character in location CHAR.

Subroutine 11: Calling Sequence: CRLF
Return

Function: Types a CRLF combination

Subroutine 12: Calling Sequence: DNEG
High ADD
Return

Function: This routine performs a double precision negation of the double precision word whose high order word is pointed to by CALL+1. It is assumed that the low order word is located in core right after the high order.

Subroutine 13: Calling Sequence: DECIN
Return

Function: This routine inputs a decimal number from the Teletype. Two input modes are possible.

1) Input with a decimal point. In this case, input is in one of the following formats:

A) XXX.X
B) XX.X
C) X.X
D) .X

In all cases a period and a digit after it must be typed. This input mode is used when location X = Ø.

2) Input without a decimal point. This input is used when location X<Ø. Up to three decimal digits may be typed. Input may be prematurely terminated by a Carriage Return.

In both cases a RUBOUT restarts input. After input is done, the converted decimal number is left in location number.

In Case 1, that is input with a decimal point, the value returned is that of the number typed in if it did not have a decimal point. For example, if 2Ø.Ø is typed, the value returned is $31Ø_8$ or $2ØØ_{1Ø}$.

Subroutine 14: Calling Sequence: CLEAR
Return

Function: Clears the buffers.

Subroutine 15: Calling Sequence: CURSOR
Return

Function: This reads in the values of knobs Ø and 1, and displays cursors at the X point that corresponds to their values.

Subroutine 16: Calling Sequence: ADC
Return

Function: This routine samples the A/D channel specified by the AC upon entry. When it returns, the value read is in the AC.

Subroutine 17: Calling Sequence: DECOUT
Return

Function: This routine types out the decimal value of the number in the AC upon entry. The (AC) may be negative. No CR/LF is typed before or after output.

Subroutine 18: Calling Sequence: DOUT
Return

Function: Types the decimal value of the double precision number held in locations HIGH1, LOW1. This is signed output, no CR/LF is typed before or after output.

Subroutine 19: Calling Sequence: EFFECTIVE JMS SHIFTL
High Address
Return

Function: Shifts the double precision word pointed to by CALL+1 the (AC) times left. It then returns to CALL+2. The (AC) must be positive.

Subroutine 20: Calling Sequence: EFFECTIVE JMS PNCH
Return

Function: Punches the (AC) on high speed punch. Should be called with interrupt off.

Subroutine 21: Calling Sequence: EFFECTIVE JMS SHIFTR
High Address
Return

Function: Shifts the double precision word pointed to by CALL+1 (AC) places right, then returns to CALL+2. The AC>Ø.

Subroutine 22: Calling Sequence: EFFECTIVE JMS DPDIV
Return

Function: This routine divides the double precision word, which may be signed, held in locations HIGH1, LOW1, by the double precision word, which must be positive, held in locations HIGH, LOW. When it returns, the double precision quotient is in locations HIGH2, DIGBOX, and the remainder in locations HIGH1, LOW1. If there is division by zero, then an error typeout of "DIVISION BY Ø" results.

Subroutine 23: Calling Sequence: EFFECTIVE JMS ADD
High Address 1
High Address 2
Return

Function: This routine adds the double precision number pointed to by CALL+1 to the double precision number pointed to by CALL+2, putting the result in the word pointed to by CALL+2. Return is to CALL+3.

Subroutine 24: Calling Sequence: EFFECTIVE JMS DFLOAT
Return

Function: This routine floats the double precision number held in locations HURD, LORD. The exponent is left in location EXP. Thus, the result is left in the FAC.

Subroutine 25: Calling Sequence: EFFECTIVE JMS ACMINS
Return

Function: Negates the FAC.

APPENDIX P

CORRELOGRAM BINARY OUTPUT FORMAT

When the CTRL/B command is given, a binary dump of the correlogram will proceed on either the high-or low-speed punch. The dump will start with leader/trailer code (these are channel 8 punches). For a description of paper tape punching format, see <u>Introduction to Programming, Vol. 1</u>, 4-14.

After the first leader/trailer code, the identification code will be punched in ASCII format. This will be followed by more leader/trailer. The core image is preceded by an address. This is signified by a 7 channel punch with the most significant 6 bits of the address on the same line followed by the least significant 6 bits on the next line. A core image follows the address.

The first locations to be dumped are the parameters of the correlogram that are necessary for proper display and accurate typeout. In order of their dumping, these parameters are: CLKCNT (minus the clock rate); WRDCNT (minus the number of points in the correlogram); ENDTMP (end the address of temporary buffer -1); ENDPRM (end address of permanent buffer-1); SWEEPH and SWEEPL (the high and low order words of the sweep count); NUMBER (number of points); BTCFLG (this = Ø for normal mode, -1 for batch mode); and FLAG (this = Ø for auto-correlation, -1 for cross-correlation).

After the display and typeout parameters are punched, the address of the correlogram is punched, followed by a core image of the correlogram. After all information has been punched, a checksum is punched.

A checksum is a means of testing the validity of read-in. It is a 12-bit word formed by adding every line of punched data, which represents the sum of all of the punches in the tape that are not leader/trailer or ASCII information. Overflows are ignored and the resulting 12-bit sum is punched at the end of the data. This is followed by more leader/trailer code.

When the binary loader reads a binary formatted tape, it keeps a 12-bit sum of all punches between the end of the leader code and the checksum (note: the checksum punches are not included in the checksum). When the checksum is read, it is subtracted from the 12 bit sum accumulated during read-in. If the result is non-zero, an error on read-in has occurred and the tape should be read in again.

APPENDIX Q

CHART RECORDER INTERFACING INFORMATION TO LAB8/E VC8-E

The VC8-E (10 bit point plot display controller) consists of two modules (M869 and M885) which plug into the OMNIBUS. The M885 module includes the "male" portion of a "Berg Connector". The user can plug the "female" portion (DEC Part No. 1210090 and pin Part No. 1210089 - one pin per signal required) with the chart recorder cable assembly into the male portion. A sketch of the female Berg Connector with the pin locations is shown below:

BERG													
B	D	F	J	L	N	R	T	V	X	Z	BB	DD . . .	VV
A	C	E	H	K	M	P	S	U	X	Y	AA	CC . . .	UU

The pin assignments and their functions follow:

Pin	Function
D	Analog Y output (±5V into 1kohm minimum requirement)
F	Analog Ground
L	Analog X output (±5V into 1kohm minimum requirement)
N	Analog Ground
X	Z output (Not used for chart recorder)
V	Logic Ground
Z	Channel Select (TTL level into 1kohm minimum)
BB	Logic Ground

NOTES

1) If the chart recorder is differential, run the "X output" and the "analog ground" into two inputs and use the "logic ground" as a shield.

2) If the chart recorder is single-ended, tie the "analog ground" into the "recorder frame ground".

APPENDIX R

DAQUAN COMMAND SUMMARY

At least the first two characters of a command must be typed before colon is typed.

INPUT-OUTPUT COMMANDS

AVERAGE:	Accept time averaged analog data
PAPERTAPE:	Input data from paper tape or keyboard.
OUTPUT:	Print/punch paper tape
PLOT:	Plot data on X-Y analog recorder

PROCESSING COMMANDS

CALCULATE:	Calculate Lorentzian and/or Gaussian spectrum
ALTER:	Alter parameters input by previous CALCULATE command
COPY:	Copy CDC[1] into NDC[2]
XINVERT:	Invert X axis
YINVERT:	Invert Y axis
SCALE:	Scale to range of Ø-1ØØØ
MULTIPLY:	Scale to arbitrary range
SMOOTH:	Apply eleven point digital filter
CURSORS:	Set up two cursors on scope
LIST:	Print decimal values of cursors
INTEGRATE:	Integrate between cursors or running integration (preceded by CURSOR command)
STRIP:	Strip out data or baseline (preceded by CURSOR command)
DERIVATIVE:	Form differences (derivatives)
SUBTRACT:	Subtract CDC from NDC: results in CDC
ADD:	Add NDC to CDC; results in CDC
SWAP:	Swap CDC and NDC
SQUEEZE:	Average adjacent points of displayed channels
PEAK REPORT:	Find peaks and report on them
MASS STORAGE:	Read or write from mass storage (system device)

SPECIAL COMMANDS

MODIFY:	ODT-like core modifier
TIME:	Set machine cycle time constant to calibrate AVERAGER
RESTART:	Restart program

[1]CDC = currently displayed channel
[2]NDC = non-displayed channel

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (SPR) form (see the HOW TO OBTAIN SOFTWARE INFORMATION page).

Did you find errors in this manual? If so, specify by page.

____
____
____
____
____
____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

____
____
____
____
____
____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

____
____
____
____
____
____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name ______________________ Date ______________

Organization ______________________________

Street ______________________________

City ______________ State ______________ Zip Code or Country ______________

If you do not require a written reply, please check here. ☐

Please cut along this line.

Fold Here

Do Not Tear - Fold Here and Staple

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Communications
P. O. Box F
Maynard, Massachusetts 01754

APPENDIX S

CALCULATE ROUTINE

S.1 FUNCTIONAL EQUATIONS

Functional equations used by CALCULATE:

The equation for a sum of Lorentzian curves at point i is

$$Y_i = B+\sum_{j=1}^{P} H_j W_j^2/[(X_i-P_j)^2+W_j^2] = B+L_i$$

where

- P is number of peaks
- B is baseline
- H_j is intensity of j^{th} peak
- W_j is half width at half height of j^{th} peak
- P_j is position on X axis of j^{th} peak
- X_i is X coordinate with resulting net intensity of Y_i

The equation for a sum of Gaussian curves is

$$Y_i = B+\sum_{j=1}^{P} H_j e^{-\frac{1}{2}\left(\frac{X_i-P_j}{W_j}\right)^2} \equiv B+G_i$$

with similar parameter definitions except that here W_j is the sigma value. Combination of the two by means of the LORentzian FRaction is

$$Y_i = B + LORFR \cdot L_i + (1-LORFR) \cdot G_i$$

DIGITAL FILTER EQUATION:

The expression is effectively a low pass filter designed to retain the integrity of peak areas. Intensities may be diminished if there are fewer than 15 points between the peak's inflection points (approximately the width at half-height).

The equation takes the form of a nonlinear, weighted average of a data point and its ten nearest neighbors:

$$Y_i = \sum_{j=\emptyset}^{5} C_j \cdot (Y_{i+j}+Y_{i-j})$$

where

$C_Ø$ = Ø.1Ø38
C_1 = Ø.1956
C_2 = Ø.16Ø8
C_3 = Ø.Ø488
C_4 = Ø.Ø21
C_5 = -Ø.Ø3Ø

and

Y_i is the resultant, filtered value.

S.2 Modifying the CALCULATE ROUTINE

The CALCULATE routine makes these two stipulations:

1. interaction of a peak beyond the range of eight times its width is negligible
2. for initial entry, peak widths are all equal.

These can be removed by changing contents of location 2417 to 5242 and contents of 2212 to 7ØØ1, respectively. The first change will include all interaction and is desired for pure Lorentzian spectra. The second requires the user to enter the width for all peaks.

If desired, these changes should be made using MODIFY (Section 2.22 of Book 5) and before requesting CALCULATE.

APPENDIX T

CORE MAPS

T.1 Core Map of DAQUAN

Field Ø

Page	Contents
Ø	Constants, subroutine links, temp. storage
1 (200)	Main display loop, fix and float subroutines
2 (400)	MODIFY*, TTY subroutines
3 (600)	F.P. input, output subroutines, plot axis marker
4 (1000)	plotter calibration, answer subroutine, delay, A/D, DAC, relay subroutines
5 (1200)	Leader/trailer subr., X display increment setup, PAPER-TAPE input, X-Y analog PLOT setup, Y-INVERT
6 (1400)	OUTPUT, SCALE, increment x coordinate subroutine
7 (1600)	Analog plotter driver, message output subroutine
8 (2000)	Keyboard command decoder, command dispatch table
9 (2200)	SUBTRACT, COPY and mass storage
10 (2400)	Mass storage
11 (2600)	PAFFT routine, 7600 field 1 storage
12 (3000)	SWAP, SMOOTH, ADD, CURSOR routines, clockdriver
13 (3200)	cursor decoder, STRIP
14 (3400)	INTEGRATE
15 (3600)	More calculate routines, array initializer, DERIVATIVES, SQUEEZE
16 (4000)	AVERAGE setup, TIME calibration, LIST cursors
17 (4200)	Average acquisition routine, output setup for PEAK REPORT
18 (4400)	Intersweep average display, X-INVERT, MULTIPLY
19 (4600)	Output routine for PEAK REPORT
20 (5000)	PEAK REPORT peak detector
5131-5377	6-bit packed ASCII messages
5400-7577	Floating Point Package

*MODIFY overwritten by parameters when CALCULATE is used.

Field 1

Ø-3777	channel 1 data
4000-7777	channel 2 data, note 7600-7777 is saved in 2600-2777 of field Ø

Field 1 Utilization

A. AVERAGE requires 2N locations starting from base of the channel where N is the number of points. Hence, a 1024 point average into channel 1 just fits. More than 1024 points will overrun into the other channel.

B. PEAK REPORT requires N+9p locations where N is the number of points and p is the number of peaks detected. The peak parameters are stored immediately after the data array.

T.2 Core Map for DAFFT

Only the major departures from DAQUAN Core Map (T.1) are presented here.

Field Ø

Page	Contents
2	Field 1 dispatcher, POWER, SHIFT, SMOOTH, dispatcher overlay MODIFY
15 (3731-3756)	SQUEEZE overlayed by message text and FACTOR output from FFT's.

Field 1

Ø	Constants, storage for FFTS-C
1 (200)	FFT drivers, Hanning filter, zero mean routine
400-1577	FFTS-C
1600-3577	Channel 1 data
3600-5577	Channel 2 data
5600-7577	Storage buffer

T.3 Core Map for PAFFT

The departures from core maps of DAQUAN (T.1) and DAFFT (T.2) are shown below:

Field Ø

Page	Contents
None	

Field 1

Ø	Hanning filter, zero mean subroutine
1(200)	FFT drivers, power average subroutine

APPENDIX U

BASIC/RT COMMAND AND FUNCTION SUMMARY

U.1 Statements

ACCEPT	Start A/C sampling
CLEAR	Erase Scope
DATA	Provide initial data for a program
DEF	Define a function
DELAY	Display scope buffer
DIM	Define subscripted variables
END	End a program
FOR TO STEP	Set up a program loop
GOSUB	Go to a subroutine
GOTO	Change order of program execution
IF THEN	Conditionally change order of program execution
IF THEN nn	Conditionally execute the statement nn
INPUT	Get variable values from the terminal
LET	Assign a value to a variable (LET is optional)
LPT	Assign line printer as output device
NEXT	End a program loop
PLOT X,Y	Plot on the scope the point x,y
PRINT	Print out the indicated information
PTP	Assign high-speed paper tape punch as output device
PTR	Assign high-speed paper tape reader as input device
READ	Initialize variables to value from the data list
REAL TIME	Create buffer for A/D samples
REJECT	Stop A/D sampling
REM	Insert a program comment
RESTORE	Restore the data list
RETURN	Returns from a subroutine
SET CLOCK	Set clock enable register
SET RATE	Set clock interrupt rate
STOP	Stop program execution
TTY IN	Assign Teletype as input device
TTY OUT	Assign Teletype as output device
USE	Specify a display buffer
WAIT	Pause execution until next interrupt
WAITC	Delay program execution until next clock interrupt

APPENDIX V

SUMMARY OF THE USE OF NMR AVERAGER

1. Load with DEC Binary Loader, ADDR=0200, CLEAR, CONT.

2. Averaging -

NMR-AVERAGER	
SWEEPS =	Ø<n<4Ø96 (Ø shifts directly to display)
SWEEP TIME =	Ø<t<4Ø96
SAMPLES =	Ø<s≤2Ø
PAUSE TIME =	Ø≤p≤4Ø
REPEAT?	Y for continuous sweeping
	N or RETURN to do only N sweeps
TYPE RETURN	RETURN to erase and begin averaging
	LINE FEED to begin averaging without erasing memory
	CTRL/O to restart the program from the beginning
S	Halts the program at the end of the current sweep
CTRL/H	Halts the sweeping immediately (error recovery only)
END	

3. Pre-Normalization commands

C	Contract
X	eXpand
P	Plot
CTRL/Z	Zero (erase memory)
N	type Number of sweeps completed
K	calibrate
B	Binary punch of accumulated data
T	Type out decimal values of accumulated average
CTRL/C	return to OS/8 or disk monitor
G	Go - continue averaging
R	Restart program from beginning
CTRL/E	Enter the normalized section

4. Post-Normalization commands

C	Contract
X	eXpand
U	Up
D	Down
P	Plot
CTRL/Z	Zero memory
N	Number of sweeps completed
R	Restart
K	calibrate
F	Filter - curve smoothing
I	cursor Integration
A	Analog (RC) integration
CTRL/O	return unmodified display
G	Go - start averaging - effective only after CTRL/Z
CTRL/D	Differentiate display
CTRL/C	return to monitor

APPENDIX W

LAB/8E MASS STORAGE PROGRAMS
BINARY DECTAPE

All binaries used by the LAB/8E MS programs are contained on a DECtape DEC-8E-ALMSA-A-UB. The files are as follows:

AAVG1.BN	Section 1 of the Advanced Averager
AAVG2.BN	Section 2 of the Advanced Averager
AAVG3.BN	Section 3 of the Advanced Averager
AAVG4.BN	Section 4 of the Advanced Averager
AAVG5.BN	Section 5 of the Advanced Averager
BAC1.BN	Basic Averager control tape 1
BAC2.BN	Basic Averager control tape 2
BAC3.BN	Basic Averager control tape 3
BAC4.BN	Basic Averager control tape 4
BAC5.BN	Basic Averager control tape 5
BAC6.BN	Basic Averager control tape 6
BAC7.BN	Basic Averager control tape 7
BAC8.BN	Basic Averager control tape 8
BAD2.BN	Basic Averager
PSTD1.BN	PST & Latency Histogram
TIHD2.BN	TIH Histogram
CORD3.BN	Auto & Cross Correlation
COND14.BN	Convert Program
DAQD5.BN	DAQUAN Without a Floating Point Package
FPPEAE.BN	EAE Floating Point Package for DAQUAN
FPPNE.BN	non EAE F.P.P. for DAQUAN
DAFFT3.BN	DAFFT Overlay to DAQUAN
PAFFT2.BN	PAFFT Overlay to DAQUAN