# DECUS
## PROGRAM LIBRARY

| | |
|---|---|
| DECUS NO. | 8-547 |
| TITLE | ADVANCED AVERAGER PROGRAM (ROTTERDAM VERSION) |
| AUTHOR | L.T.M.E. Hillegers and S. Miller |
| COMPANY | Medical Faculty<br>Department of Anatomy<br>Rotterdam, The Netherlands |
| DATE | April 21, 1972 |
| SOURCE LANGUAGE | |

LABORATORY MANUAL

LAB-8
ADVANCED AVERAGER PROGRAM
(ROTTERDAM VERSION)

Use and description of the program

L.T.M.E. Hillegers,
Department of Anatomy,
Medical Faculty,
Rotterdam.
The Netherlands.

February 1972.

CONTENTS

# PREFACE

The Lab-8 Advanced Averager (Rotterdam Version) program was developed as a collaborative project between the neurophysiological laboratory, Department of Anatomy, Medical Faculty Rotterdam and the Technical University of Delft.

In the neurophysiological laboratory a PDP-8/L computer is used to provide on-line transient averaging of electrical signals recorded in neurophysiological investigations of pathways in the spinal cord of the cat. These pathways are thought to play a role in the coordination of movements such as stepping and jumping.

The Rotterdam Version of the Advanced Averager program was developed
a) to avoid reloading of Sections III, IV and V of the existing Advanced Averaging program each time an average is taken,
b) to add additional special purpose sections to allow on-line computation and oscilloscope display of the time course of the effects of conditioning stimuli upon test responses. For abbreviation in the listings (reference 7) such curves are referred to as "Simon" curves.

Certain features of the Advanced Averager program such as trend and confidence limits and X-Y plotting routines have been deleted. The program is designed to run with 4K memory. (See Chapter III).

I am very grateful to the Medical Faculty Rotterdam for the opportunity to work in the Department of Anatomy, to Dr. Simon Miller for his encouragement and criticism throughout the project, to Hans van der Burg for help with the electronic systems, and to Miss Edith Klink for typing the manuscript.

Tom Hillegers,
February 1972.

CHAPTER I

SYSTEM REQUIREMENTS

INTRODUCTORY DESCRIPTION OF THE PROGRAM

## System Requirements

For the Lab-8 Advanced (Rotterdam Version) program the following equipment is necessary:

PDP-8/L computer, 4K memory, with teletype

AX08 Laboratory Peripheral with Option XR

Oscilloscope

## Introductory description of the program

The Lab-8 Advanced Averager (Rotterdam Version) is designed to run on PDP-8 configuration set out above. The program performs transient averages of analog signals, and then excecutes some arithmetical computations and displays the results graphically on an oscilloscope. In the averaging process the computer sums the time-locked analog signals over a predetermined period (epoch) and for a number of trials (n). The sum is then scaled to the magnitude of the original signal by dividing by n. This results in an averaged signal (average) with the "noise" reduced by a factor of $1/\sqrt{n}$ assuming the "noise" to be randomly distributed. To perform these computations each electrical analog signal is digitized into numerical values (data points) by an A/D converter (Laboratory Peripheral AX08).

Both the raw digitized signals as well as the averaged signals are displayed under program control on an oscilloscope screen. Once the averaging process has been completed, data points specified by the operator, or the sum of such data points from each average, may be typed out on the Teletype.

A sort code (bit pattern) is attached to each signal and identifies that signal. At the same time that a signal is offered to an analog input of the AX08, the accessory sort code sets the sense line register (contingency register), which is read out by the program, thus making it possible for several signals to use the same analog input. In this way a wide variety of experimental situations can be handled.

The program also contains a part specially developed for neurophysiological investigations, in which the effects of conditioning of reflexes can be displayed graphically on an oscilloscope. Such curves are termed "Simon" curves for purposes of abbreviation in the listings (reference 7).

The Lab-8 Advanced Averager (Rotterdam Version) program is built up from three sections, I, II, III, which are loaded sequentially into core.

Section I is a compiler program in which the user specifies for Sections II and III a wide range of parameters to suit the particular experiment:

the number of data points into which a signal is digitized,
the duration of the interval (epoch) in which the samples are taken,
the moment at which the epoch begins,
the frequency at which the epochs are sampled,
the number of signals, the analog inputs they use and the sort codes assigned to them.

At the end of this Section the teletype punches a control paper tape, specifying the parameters to be used by Sections II and III.

Section II, together with the timing parameters loaded by the control tape produced by Section I, enables the user to calibrate the computer clock to the timing of the experiment and to set the levels of the external trigger input to the computer.

Section III is loaded after Section II and contains three parts:

Part 1: Signal averaging.

Part 2: Each average is displayed separately; the values of the data points can be typed out and can be added to form a sum (integral).

Part 3: Sums from the averages taken in Part 2 are used to form the graphic display of the conditioning curves ("Simon" curves).

Switching between the Parts of Section III and all other operations are controlled by keys on the keyboard of the teletype.

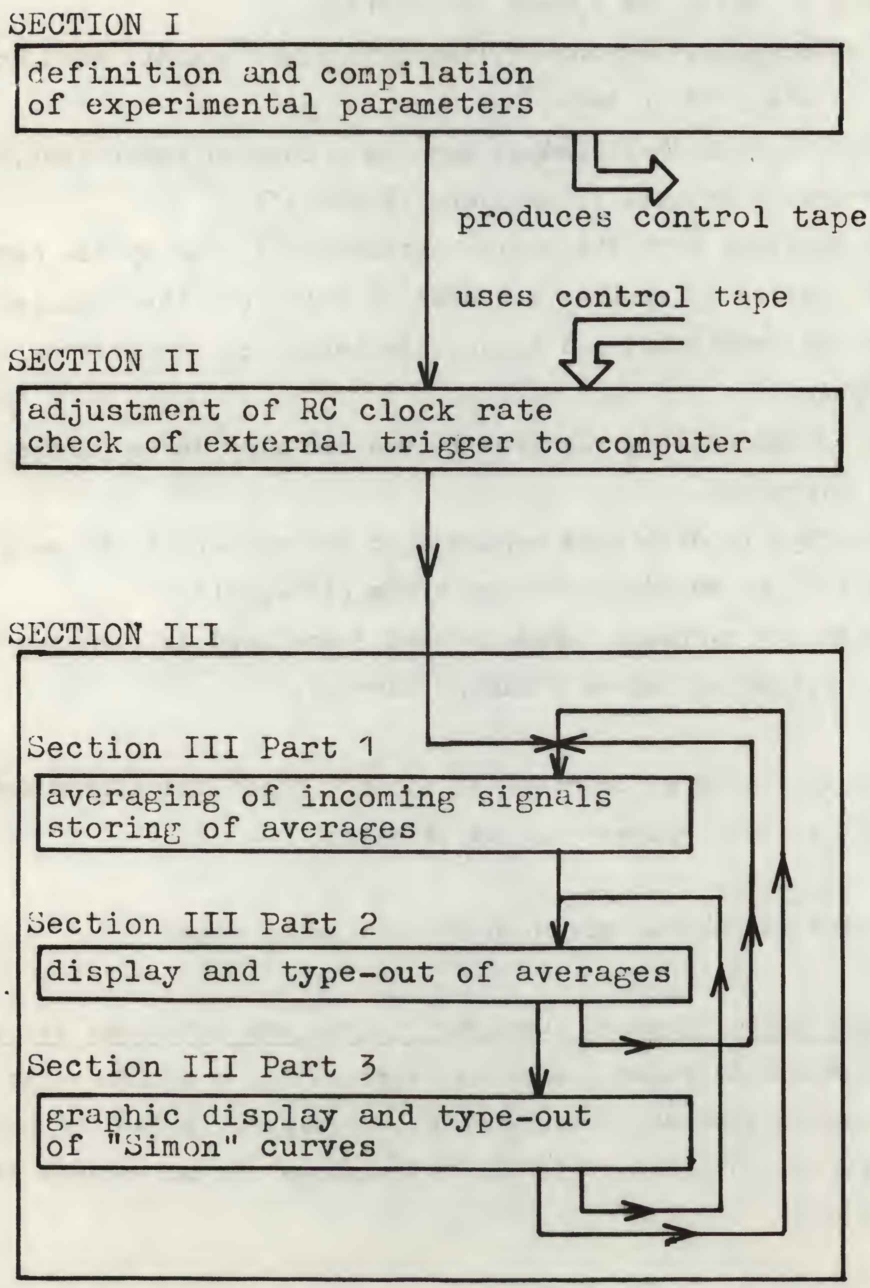
The global program structure is given on the following page.

## Differences between Lab-8 Advanced Averager Program and Rotterdam Version

The Lab-8 Advanced Averager (Rotterdam Version) is a modification of the program Lab-8 Advanced Averager (Reference 6), developed in 1970 by Digital Equipment Corporation. In Chapter III the differences and advantages of the programs are set out.

LAB-8 ADVANCED AVERAGER ROTTERDAM VERSION

GLOBAL PROGRAM STRUCTURE



## CHAPTER II

### THE USE OF THE PROGRAM

### PROGRAM STRUCTURE

### FLOW CHARTS

(It is assumed that the user has a working knowledge of the Lab-8 System. Further references are given on page 46 ).

### 1. Program Loading

Section I and II of the Lab-8 Advanced Averager (Rotterdam Version), and the control tape produced by Section I are loaded into core by the Binary Loader (Start address 7777) using the High or Low Speed Reader. Zero accumulator indicates correct loading of each tape. The tape must be reloaded if the accumulator is not zero. All the sections contain a Paper Tape Autostart and Overlay Part (DEC program LBSYS) such that the Binary Loader jumps to the first instruction of the section of program just loaded.

### 2. Section I

#### Definition of experimental parameters

2.1 After correct loading (see II.1. above) press CONTinue to proceed. On the oscilloscope the following text will appear:

```
LAB-8 IS READY AND
TOM WISHES YOU
SUCCESS

HIT RETURN TO PROCEED
```

#### Note on use of keys:

CR (Carriage Return), LF (Line Feed) and RUBOUT

The CR key of the teletype will be used throughout the program to go to the next step, or to "execute" an instruction previously typed in.

The LF key is used to jump back to a previous question or step - for example, to put right a parameter incorrectly specified.

The RUBOUT key erases characters typed in and displayed on the oscilloscope.

2.2. Pressing CR after the introductory display will cause the following to appear on the oscilloscope:

SYNC ON INPUT: $_

The blank space can be filled in with 0, 1, 2 or 3 — the number of the BNC connector on the front panel of the AX08 to which the external trigger is connected.

The next three questions concern the timing of the sampling process (See Fig. II.1). The program asks:

a) the number of data points into which each analog signal should be digitized;

b) the length of the sampling period (epoch) in seconds or milliseconds;

c) the delay (amount of time either before or after the external trigger pulse, with which the sampling period begins).

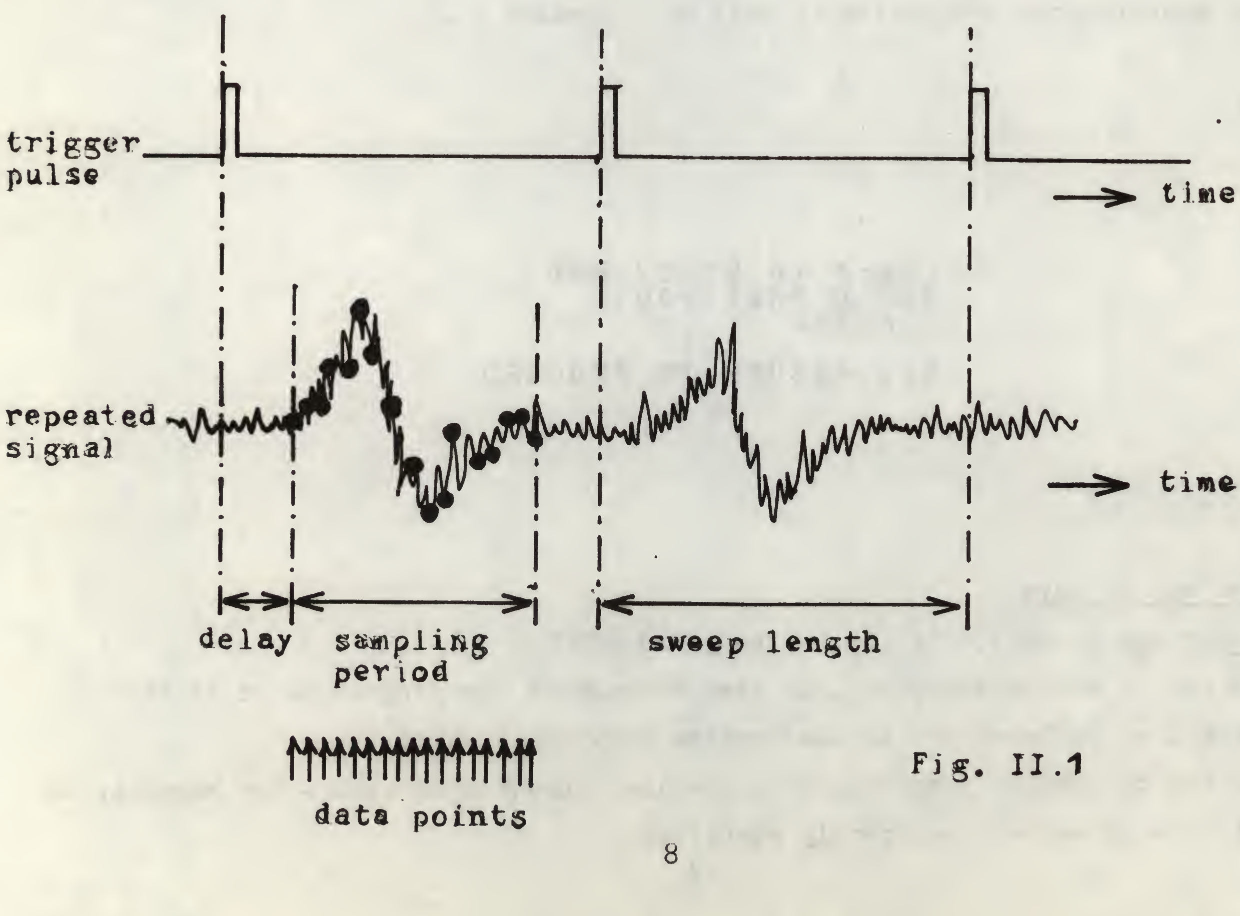



Fig. II.1

2.3 The following display now appears:

```
---- DATA POINTS
IN ONE AVERAGE
```

It requests the number of data points. The amount of core available limits the number of data points. For one signal the maximum number of data points is 517. The number per signal (or average) is reduced by the number of analog inputs and averages selected. The limit may be calculated by the formula given in 2.11.

2.4 After entering the number of data points, followed by CR, the following will be displayed:

```
LENGTH: ----- -SEC
```

This asks the user to define the length of the sampling period. Note that there is a space between the first five and the sixth blanks. This indicates that the first five blanks are for 1 to 5 figures (or 1 to 4 figures plus the decimal point).

Type CR. The sixth space is for either an M, for milliseconds, or SPACE or CR, for seconds. Numbers may be entered in any position. Leading or trailing blanks are not required. Leading zeroes are also not required. The sampling rate (the sampling period divided by the number of datapoints) can be between 175 $\mu$sec/point and 2 sec/point.

2.5 The next parameter to be specified is the time at which the sampling period begins with reference to the trigger pulse:

DELAY: _____ _SEC

Note that this display is similar to the preceeding. However, a minus sign (hyphen on the keyboard) is allowed to indicate the sampling period starting before the trigger pulse. If this negative delay is used, the trigger pulse must arrive no later than the time equivalent to the last data point. In other words, a negative delay cannot be longer than the length of the sampling period.

Otherwise the following error message will appear:

? SWEEP ENDS EARLY

indicating that the sweep would end before the trigger pulse arrived.
Type CR to respecify the delay.
Type LF to change both the length and the delay.

A negative delay may be used in an experimental situation where the trigger pulse is derived from an asynchronous event and where the analog waveforms preceding this event require analysis. The sampling epoch thus begins before the trigger pulse.
A positive delay may be used if the analog waveforms are to be expected some time after the trigger pulse. In this case the sampling epoch begins after the trigger pulse.

Note:
In any display requiring a numerical answer, typing CR alone will be interpreted as a zero.

2.6 After defining these three timing parameters the following sweep summary will appear (the actual figures are only examples):

```
BEGIN -9.800MSEC
RATE   350  USEC
END    24.85MSEC

_: CHANGE? [Y OR N]
```

Changes can be made by typing Y (=YES). If any change is made, <u>all</u> <u>three</u> parameters defining the sampling period must be respecified. If no change is made type N (=NO) or simply CR.

<u>Note</u>:

In all conversational mode YES or NO questions SPACE, CR, or any character other than Y will be interpreted as NO.

In the preceding example the first point is made at 9.8 msec before the trigger pulse, the sampling rate is 350μsec per point, and the last point occurs at 24.85 msec after the trigger pulse.

<u>Note</u>:

a) All measurements (such as DELAY, SORT AT, etc.) in the Lab-8 program refer to the leading edge of the trigger pulse.
b) Both the sampling period and the delay should be integer multiples of the sampling rate selected. If the user selects fractional multiples, the program will calculate and display the closest approximation. This is the sweep summary display.
c) There is an uncertainty of 1 sampling interval in the sampling period start time. This is expressed in the sweep summary by not rounding the start and end times.

2.7 If the user finds the sweep summary satisfactory, he types CR and the following message appears on the screen:

SORT AT ----- -SEC

This refers to the contingency feature and specifies the sort time with reference to the trigger pulse. Each digitized sweep is held in temporary storage and is assigned to specific permanent storage locations defined by the sort code (See note 2.9 below). The sort time is the instant at which the computer will look at the logical status of the eight digital contingency inputs. At sort time the computer will store the pattern of 0's and 1's indicating the states of the eight contingency inputs. When the averages are updated with the current sweep, this pattern is compared with the sort code specified for each analog input. If the input sort pattern is identical to the corresponding sort code for that input, the current sweep in temporary storage will be added to the permanent storage of the average corresponding to this sort code. If not, the sweep will be discarded and not stored.

<u>Note</u>:

a) In order for the state of a contingency input to be read as 0 at the sort time, this input must have been at -3 Volt during the entire interval between the trigger pulse and the sort time. For the state to be read as 1, the voltage at the input might have changed only for an instant from -3 to 0 Volt (and then back again) at any time between the trigger pulse and the sort time.
b) Since the computer will not look for the next trigger pulse until after the sort time, this feature can be used to lengthen artificially the sweep for purposes of disregarding trigger pulses occurring too soon after the end of the previous sampling period.

2.8 After typing CR the following scope display will appear:

AVERAGE * 1
ON
ANALOG INPUT --

In this display ANALOG INPUT — asks which analog input will be used for average number 1. If the first input is used 0 can be typed, followed by CR. Analog inputs can be associated with specific averages in any order. For example:

```
AVG  1  ANALOG INPUT 3
AVG  2  ANALOG INPUT 1
AVG  3  ANALOG INPUT 2
```

Note:

Since octal notation is used to specify the inputs, there are no inputs 8, 9, 18 and 19. In the basic configuration, only inputs 0 - 3, or 34 - 37 can be specified. (Inputs 34 - 37 are the four potentiometers on the front panel of the AX08). Other inputs can be added with further AX08 options: option XM adds inputs 4-7; the first option XC adds inputs 10-13, etc. Multiplexer channels 30 - 33 are never implemented.

2.9 Typing CR results in the following display:

**SORT CODE: _ _ _**

Note:

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

SORT AT _ _ _ _ _ SEC and SORT CODE: _ _ _ are the two displays which specify the parameters of the contingency condition. The contents of the temporary storage buffers arenot automatically emptied into the series of memory locations allocated to the accumulation of a certain average. This is done only if certain conditions, or contingencies, are met. The sort code is what specifies the condition for the addition to memory of recently digitized data. For example, if a sort code of 001 were specified for average number 1, data acquired during each sweep would only be added to those memory locations allocated to average number 1, if, at sort time, the status of the inputs in the contingency register were as follows:

| C0 | C1 | C2 | C3 | C4 | C5 | C6 | C7 | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | (binary notation) |
| 0 | | | 0 | | | 1 | | (octal notation) |

<u>Note</u>:

The contingency feature can be used to generate a wide series of different average waveforms from the same analog input, following changes in experimental conditions. These changes in turn can result in different settings of the logical levels of the contingency register. This feature could be used, for example to edit abberant responses. The contingency register could be set so that a response containing an artefact would be discarded or saved in another location of memory.

If the contingency register is not to be used for a certain average, 0, or CR, should be typed for the sort code.

2.10 After typing CR the following will be displayed:

```
AVERAGE * 2
ON
ANALOG INPUT __
```

This can be answered in the same manner as in the two previous steps.

2.11 Additional averages may be set up as required until the following display appears (after the appropriate answer for the sort code for that average has been given):

```
? INSUFFICIENT MEMORY
```

This means that the capacity of the memory to set up buffers and to store data has been exceeded. The formula below can be used to compute how many words (W) will be used for this purpose:

$$W = (2P + 10).A + (P + 71).I + 6, \quad \text{(decimal notation)}$$

where P = number of data points per average, A = number of averages, I = number of analog inputs.

The total number of words available apart from the program is 1640, so that

$$W \leqq \frac{1640}{10}$$

Type CR to respecify this average.
Type LF to respecify all averages.
To change the number of data points per average type CTRL/Z, which will cause a return to display 2.2.

2.12 In certain cases, where a sampling rate as close as possible to the maximum 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 nc longer possible to maintain a relatively fast sampling rate.
Type CR to respecify the last average.
Type LF to respecify all averages.
Type CTRL/Z to respecify the timing parameters of the averages.

2.13 When sufficient averages have been set up, typing LF discontinues this part of Section I and brings up the following display:

```
LEAST SWEEP INTERVAL:
----- _S [> 579.1 MS]
```

This indicates that to do the calculations which the program makes between sweeps, the trigger pulses can be no closer together than 579.1 msec (in this example). External trigger pulses can thus be at regular intervals - greater than 579.1 msec, or they can be at irregular intervals, the shortest of which is 579.1 msec. If the SO trigger <u>output</u> is used, the program will supply pulses at exactly the interval specified in answer to the above question. Trigger pulses occurring at shorter intervals from the last acknowledged trigger pulse will be ignored.

<u>Note</u>:

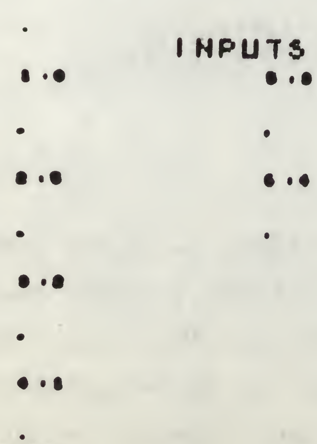
a) The time left between the least sweep interval and the user specified minimum interval (typed in above) will be used by the program to stabilize the display of inputs. The shortest interval desired or expected under experimental conditions and not the shortest interval allowed by the program should normally be selected.
b) Since the cycle time of a PDP-8/L is variable by a factor of ± 20%, trouble may develop if the user specifies a least sweep interval which is very close to the shortest interval allowed. This is especially so in a the 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 shortest sweep interval.

2.14 Typing CR brings up the following Preset Sweep Counter Display:

AVERAGE ____ SWEEPS

The largest number which can be typed in answer to this display is 4095. Answering 0 or CR is equivalent to requesting 4096 sweeps. In the signal averaging section (Section III Part 1) typing R will allow a further preset number of sweep counts without destruction of the data already accumulated in permanent storage.

2.15 The following Inputs Display shows the positions the actual inputs will occupy on the oscilloscope screen during the View Input mode of the on-line operation in Section III Part 1. For example:



The numbers 1-6 indicate that in this example there will be six different oscilloscope waveforms in the View Input mode. The second numbers indicate the analog input numbers associated with each waveform. If the positions, lengths and sizes of all these waveforms are satisfactory, CR may be typed.

2.16 If it is desired to change one of these parameters, typing LF will bring up the following display:

**CHANGE --**

The number to be typed in here corresponds to the number of the first average to be changed to a different position. As soon as the number and CR are typed, display 2.15 will reappear, but with the waveforms to be changed in a different position. Its position, height and width are now controlled by the four potentiometers on the AX08 front panel, as follows:

KNOB 0 : Horizontal position
KNOB 1 : Vertical position
KNOB 2 : Horizontal gain ) in 4 discrete increments
KNOB 3 : Vertical gain )

KNOB 0 is connected to analog input 34, KNOB 1 to input 35, etc.. When these four have been adjusted so that the position, height and width of the selected waveforms are acceptable, LF can again be typed to bring up display. 2.16. This allows the number of the next waveforms to be changed, to be typed. The process can be repeated until all of the waveforms are satisfactory.

2.17 When all waveforms are satisfactory, typing CR will bring up the following Averages Display. For example:

**AVERAGES**

| | |
|---|---|
| . | |
| 1 | 5 |
| . | . |
| 2 | 6 |
| . | . |
| 3 | 7 |
| . | . |
| 4 | |
| . | |

If no more than one contingency condition has been associated to the signals fed to one analog input, 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, AVG 3, etc. If LF is now typed, display 2.16 will again appear. The number now typed should correspond to one of the numbers shown in the Averages Display 2.17. After the number is typed in followed by CR, the Averages Display will reappear with the average to be changed shifted to a different position. KNOBs 0-3 are again activated and their functions are the same as in step 2.16. When the average to be changed has been satisfactorily adjusted, LF can be typed to change another average.

2.18 If it is not desired to change any more averages typing CR will bring up the following display:

```
PUNCH CONTROL TAPE? _
<Y: YES>
```

Punching a control tape allows the user to:

a) establish a precisely repeatable procedure for averaging,

b) vary certain parameters during the averaging section (Section III Part 1) by reading in different control tapes by the Binary Loader and restarting Section III at address 7603. The only restriction on the changes of parameters is that the sampling rate must remain constant (i.e. the number of data points per second or millisecond must be unchanged). If the time base is changed, Section II must be repeated to reflect the time change.

2.19 If a control tape is not desired, type N or CR and a summary of the averaging parameters will be typed. For example:

```
    [illegible] SUMMARY
[illegible]   -1.000 [illegible]SEC
RATE     350 [illegible]SEC
[illegible]      34.95 [illegible]SEC

     AVERAGES
AVERAGE      ANALOG INPUT     [illegible]
   1              1
   2              1
   3              2
   4              2

SYNC IS INPUT: S1
31    SWEEPS AT [illegible]   MS
STOP AT 500   MS
(VC2,1,231 -3375)
!
```

After this sweep summary an ! and a [ will be typed, requesting the loading of Section II for clock calibration and trigger adjustment. Turn on the reader and Section II will begin loading.

2.20 If Y for YES is typed, followed by CR, the following display will appear:

```
TITLE:
---------------------
```

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

| | | | |
|---|---|---|---|
| ! | exclamation mark | % | percentage sign |
| | space | < | left angle bracket |
| * | multiplication sign | > | right angle bracket |
| = | equal sign | # | number sign |
| / | slash | ( | left parenthesis |
| ↑ | up arrow | ) | right parenthesis |
| ← | left arrow | | |

After twenty-one or fewer characters have been typed, CR will indicate the end of the title. At this point the oscilloscope display will disappear and ↑ will be printed on the teletype.

2.21 To punch a control tape:

a) push button marked ON on the box at upper left of teletype, if using the Low Speed Punch, making sure that the leftmost switch (bit 0) of the Switch Register is up (a binary 1), or

b) with High Speed Punch, push button marked POWER on the punch and make sure that the leftmost switch (bit 0) of the Switch Register is down (a binary 0). Now type CTRL/P. This will cause the punch to begin punching out the control tape. A typical example of this listing follows:

```
?!!?))!??!!??????????????!!???))!3???????    ?))!

DEMONSTRATION-EXAMPLE

    SWEEP STEPS [illegible]
BEGIN -0.94 MSEC
RATE   350   MSEC
END    94.88 MSEC

     AVERAGES
AVERAGE      VALUE TABLE      BIT CODE
    1                1              3
    [illegible]      1              [illegible]
    3                0              3
    4                0              0

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
```

The first part of the tape will be the twenty-one or fewer character title in hand readable letters. As many control tapes as required can be generated with one loading of Section I. The program will return to step 2.2 to permit the user to specify a new set of parameters, if Y was answered to PUNCH CONTROL TAPE? The last part of the control tape (separated by a length of tape with continuous holes on the right side) contains the actual parameters required by the computer in concentrated form. To save time, only this last part need be read in immediately after Section II.

<u>Note</u>:

The user may return to question 2.2 at any time during Section I (except between questions 2.15 and 2.16) by keyboard control CTRL/Z.

## 3. SECTION II

Reading a control tape

Calibration of RC clock

Trigger adjustment

### 3.1 Reading a control tape

The information in the third part of a control tape determines the parameters of the averaging process. It is in binary format and is read in by the Binary Loader* after Section II has been loaded. For convenience the Sections II and III of Lab-8 Advanced Averager Rotterdam Version are on a separate paper tape.

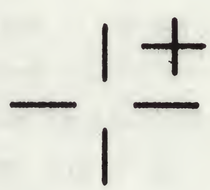
* START ADDR. 7777

### 3.2 RC clock calibration

After correct loading of Section II and the control tape (accumulator = zero), press CONTinue on the console. The following will be typed:

TIMING ....

and the following display appears:



If no cross appears, rotate timing range switch on AX08 clockwise a few positions. the purpose of this display is to calibrate and set the RC clock by adjusting the Coarse and then the Fine RC clock controls until the two sets of cross hairs are lined up:

The RC clock is then oscillating at the correct rate to supply timing information consistent with the parameters selected for this particular experimental run.

3.3 <u>Trigger Adjustment</u>

As soon as CR is typed, the following message will be printed:

TRIGGER ....

The operator should now consult the sweep summary type out, which corresponds to the parameters now in core. These include the statement SYNC ON INPUT S_ , followed by 0, 1, 2 or 3. If S0 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. The operator should ensure that a trigger pulse is connected to the BNC socket on the front panel of the AX08 corresponding to the trigger input selected in his program. The input pulse or analog signal must be in the range - 0.5 to - 2.5 Volt at the moment it is desired that the Schmitt trigger circuit fires. To test for correct firing an analog signal of $\pm$ 1 Volt is applied to Analog Input 0: if the trigger is firing this signal will be displayed on the oscilloscope every time a trigger pulse is received. The start of the sweep, as viewed on the oscilloscope, occurs without delay from the onset of the trigger pulse.

(It should be noted that in Section III the movement of the display from left to right across the screen is NOT related in time to the arrival of the trigger pulse or to the actual beginning of the sweep; it is merely the beginning of another <u>display cycle</u>.)

The knob next to the BNC trigger socket being used, should now be adjusted to ensure reliable, regular firing of the Schmitt trigger every time a trigger pulse is received. If displays appear on the screen periodically regardless of the knob setting, the trigger pulse goes more negative than -2.5 Volt each time. If it is important to synchronize the sweep with a particular point on the waveform arriving at the trigger socket, the amplitude of this point should first be measured and then the Schmitt trigger level adjusted to this value. Once the trigger has been satisfactorily adjusted, type CR to go to the on-line averaging section (Section III). This will be read in automatically because of the Overlay Part at the end of this Section.

Note:

If it is desired to reset the timing before proceeding to Section III, CTRL/Z should be typed.

## 4. SECTION III

### On-line signal averaging
### Display and type out of averages
### Conditioning ("Simon") curves

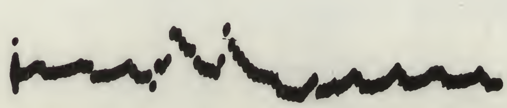
Section III is divided into three parts:

1. In Part 1 the actual averaging process is done. The incoming signals are digitized each sweep, sorted according to their sort codes, and - when these match - added to previous sweeps and stored in their own particular locations of core. Both the raw inputs and the averages can be displayed on the oscilloscope.
2. In Part 2 the averages obtained in Part 1 can be displayed separately with an identifying number. The values of the data points from which it is built up can be typed out. Also, on command from the keyboard, an integral of the absolute value of an average can be made. This integral is typed out and stored for the calculation of a conditioning curve ("Simon" curve).
3. Part 3 is the special purpose part of the program for the computation and graphic display of conditioning ("Simon") curve. ("Simon" curve is an abbreviation used in the listings of the program (reference 7)).

## 4.1 SECTION III Part 1

### On-line signal averaging

The CR typed after the trigger level adjustment will cause an ! and a [ to be typed. If the Low Speed Reader is being used, turn the reader to START. After the tape has been read in, turn to STOP. Verify that the accumulator is 0; if not, reread Section III. Press CONTinue on the console to proceed with Section III. The program will type < and start up in the View Input mode. For example:



This means that whatever is displayed on the oscilloscope initially is the actual analog input to the system and not an accumulating average. The operator can then adjust the gains of his amplifiers to produce outputs as close to ± 1 Volt as possible without clipping the highest amplitude peaks. This correction should be carried out before averaging is begun.

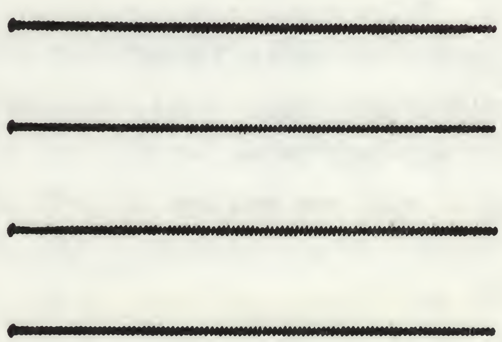
The traces which are displayed when the on-line section of the program starts up have the same heights and widths and are in the same positions as specified finally in the Inputs Display part of Section I (2.15, 2.16). If many inputs are being used, or if their positions on the oscilloscope are unusual, it may be advisable to photograph the Inputs Display during Section I to use as a reference for on-line operation.

Note:

If a relatively fast sampling rate has been specified, a large number of inputs selected or relatively little time allowed between trigger pulses, the program has so many operations to perform that it does not have time to display more than every fourth of fifth data point. Display is considered a low priority operation: A/D conversion, multiplexing and storing away the results of A/D conversion are considered more important. For this reason, the quality of the display may be somewhat degraded under certain circumstances during on-line operation.

V

Typing V switches the view mode to display the contents of pairs of memory locations in which the averages will be accumulated. By entering this Part the averages are automatically zeroed, as are the buffers and locations reserved for the calculations and displays of Part 3 of the program. All traces in this display should therefore appear as straight horizontal lines. For example:



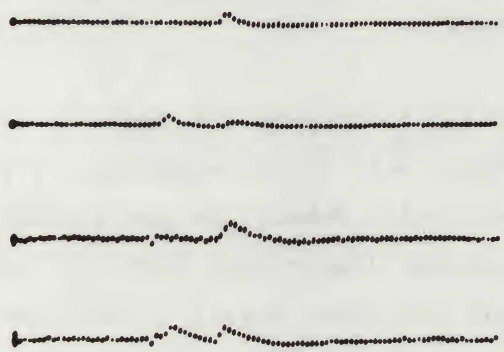
Typing V again will switch the view mode back from View Average to View Input

CR

Typing CR starts the averaging process. Waveforms should now appear to grow in the place of the straight lines of the previous display. Typing V changes the view mode alternately to View Input and back to View Average. If a very fast sampling rate was specified only a flash of the display cycle may be seen.

LF

Typing LF will interrupt the averaging process and change the view mode to View Input. This enables the operator to verify that an average is being accumulated. V should now be typed to change the view mode back to View Average. To stabilize the display on the oscilloscope the upper (coarse) timing control should be turned a few steps clockwise. For example:



To resume averaging, first reset the coarse timing control to its original position. (If this is ignored further sweeps will be added to the averages already taken, at a different sampling rate.)
Type CR. The averaging process will now restart without altering the data so far accumulated.

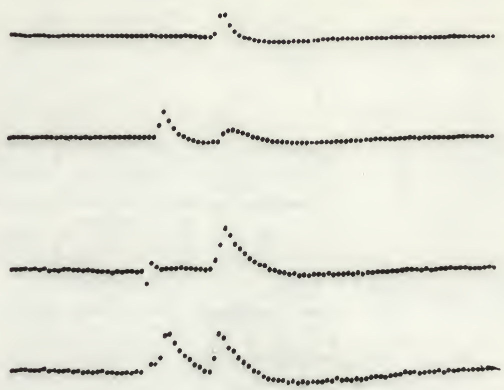
C

Typing C contracts the vertical display by a factor of 2.

X

Typing X expands the vertical display by a factor of 2.

The character 〉 is printed when the preset number of sweeps has been reached. The averaging process stops. The view mode remains in View Average. The display shows the unscaled sum of the data accumulated from the sweeps. For example:



Z

Typing Z zeroes the locations storing the accumulated averages and sets the program to the start of Part 1 again.

Typing CR starts the averaging process again.

R

Typing R resets the sweep counter to zero without destroying the data already accumulated and allows, after CR has been typed, a further preset number of sweeps to be taken. The data of this second group are added to the averages already taken.

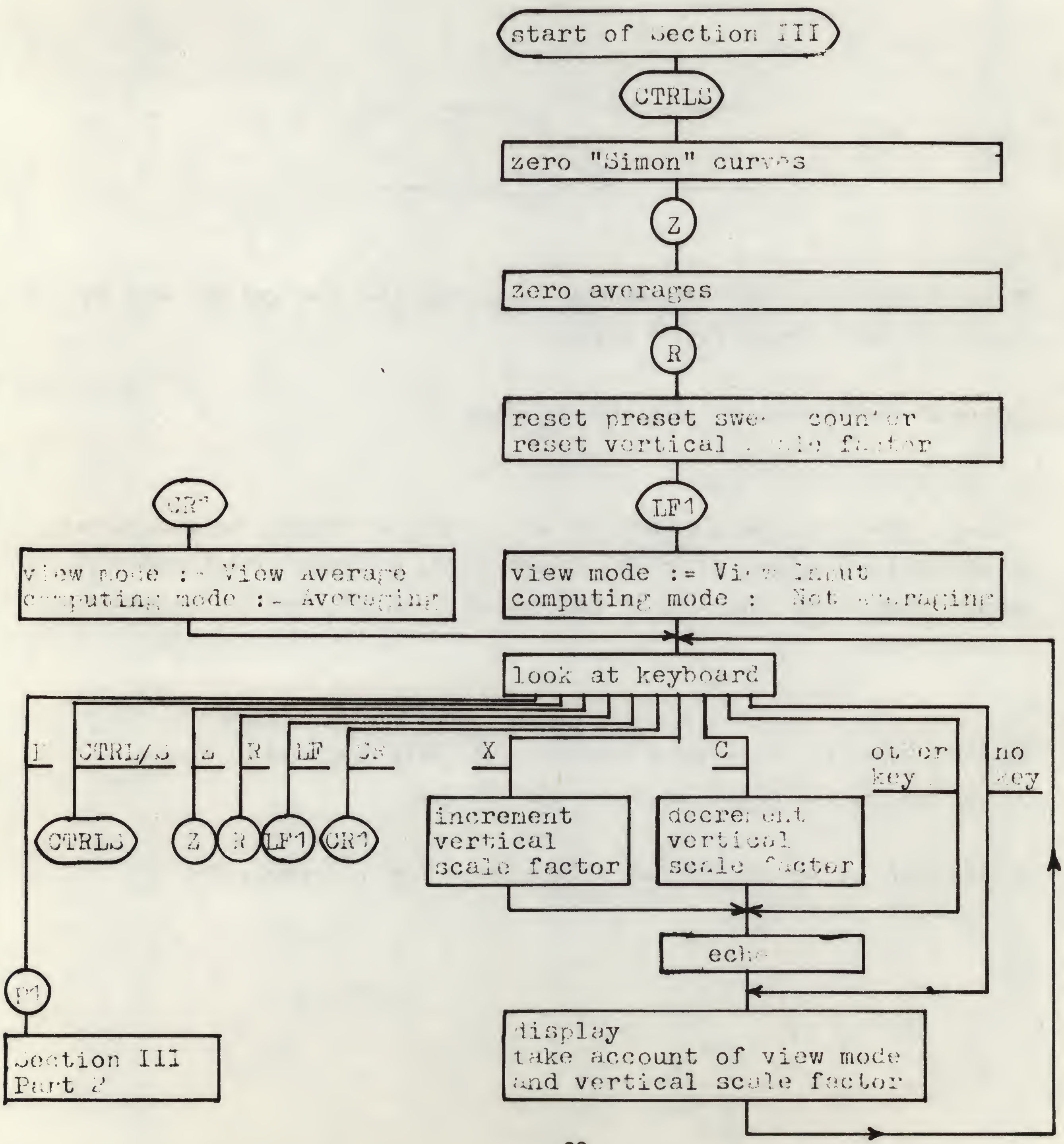
P

When the averaging process has been satisfactorily completed, typing P allows the program to proceed to Section III Part 2.

A flow chart of Section III Part 1 is given on the following page.

FLOW CHART OF SECTION III PART 1

(the working of the preset sweep counter has been omitted)



## 4.2 SECTION III Part 2

### Display and type out of averages

Typing P in Section III Part 1 enters this part of the program. The oscilloscope now displays average number 1. For example:


The average number is in the upper right corner.
Note the two brightened points. These cursors can be moved along the waveforms: the left one by turning analog input knob 0 and the right one by turning analog input knob 1.

CR

Typing CR brings up average number 2. For example:

The other averages can be successively displayed by typing CR. After the last average the program reverts to displaying average 1.

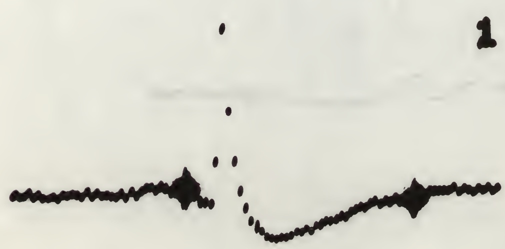
LF

Average number 1 can also be selected at any time by typing LF.

X, C

X and C expand or contract the displays in the vertical axis. For example:





T

Typing T instructs the teletype to type out the values of the data points between the left and right brightened points (cursors) on the display. For example:

analog input number
average number
sort code
number of sweeps
point position of left cursor
point position of right cursor

```
1  1  003  34    37   84
   52    48    46    39    36    35    74   269   280   166
   95    57    31    17     5     0  -  9  - 11  - 14  - 15
-  13 -  12 -   9 -   6 -   5 -   3     0     5     9    11
   17    14    17    13    18    23    24    27    29    31
   31    33    34    34    36    37    38    38
```

The values are in millivolts and have, of course, the same range as the analog inputs, i.e. from -1024 to +1020 mV. The program begins the type out of the data points with the left cursor point and continues until the right cursor or the last point. Thus when the right cursor is on the left of the left cursor <u>all</u> the points to the right of the left cursor are typed; when the left cursor is not visible, no data point is typed.

I

Typing I outputs on the teletype the integral (sum) of the absolute values of the data points between the left and the right cursor. For example:

analog input number
average number
sort code
number of sweeps
point position of left cursor
point position of right cursor

```
1  1  003  34    37   84
   1883
```

The sum is also stored for the calculations of points for the conditioning ("Simon") curves of Section III Part 3.

Q

Typing Q causes the program to quit or interrupt the type out of data points or integrals.

CTRL/S, Z, P

The function of the keyboard controls CTRL/S, Z and P is given in the flow chart of Section III Part 2.

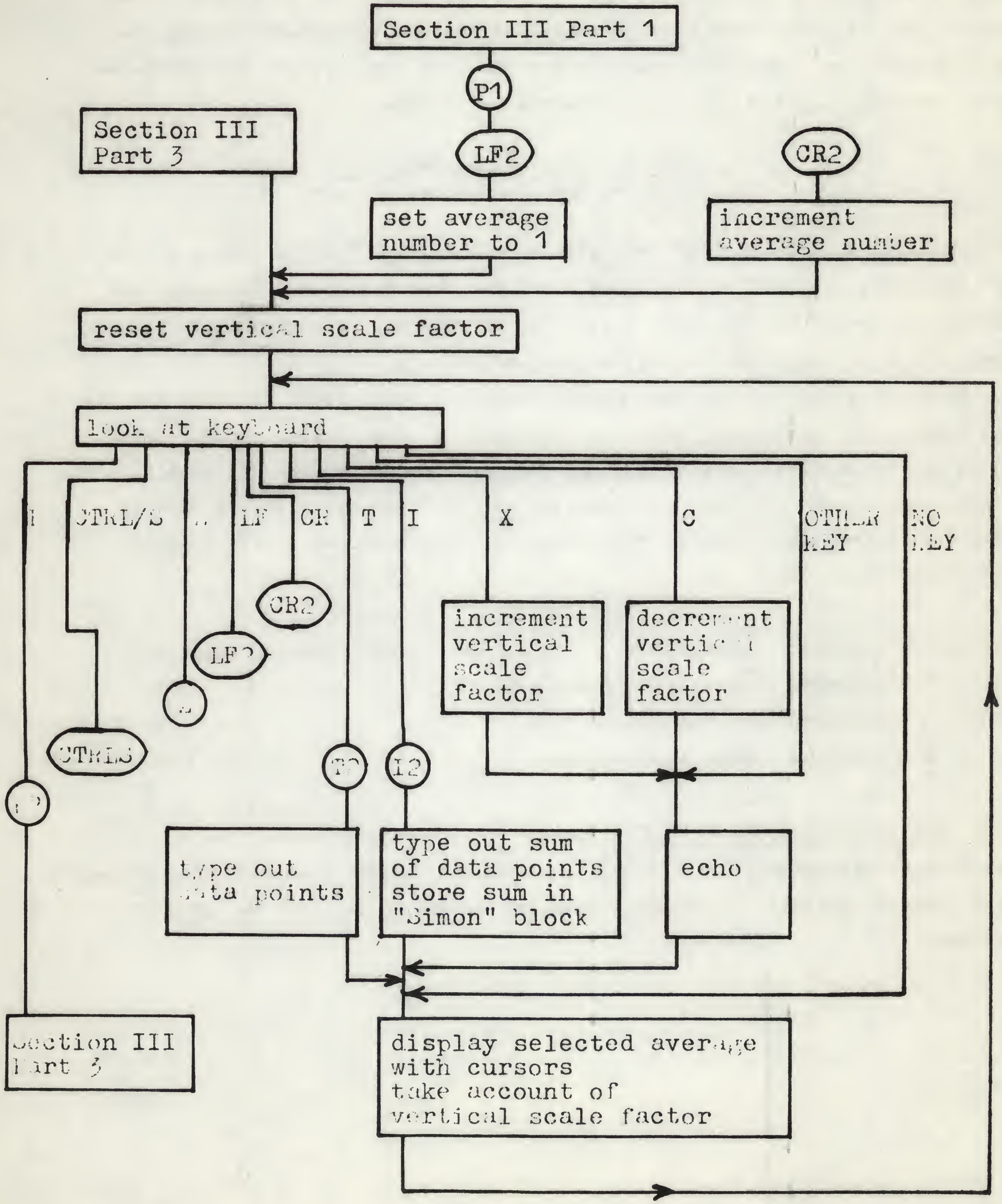
The flow chart of Section III Part 2 is given on the following page.

## 4.3 SECTION III Part 3<br>Conditioning ("Simon") curves.

### 4.3.1 Explanation of Conditioning Curves

This last part of the program can be considered separately from the other parts. It has been developed for use in neurophysiological research and performs some specific calculations on pairs of averages. It should however be emphasized that the Lab-8 Advanced Averager Rotterdam Version program has so been designed that the averaging process and the type out of the averages can be performed independently of Part 3. In experiments in the laboratory in Rotterdam the stimuli are so arranged that alternate Test Stimuli are preceded by Conditioning Stimuli. This results in alternate Test and Conditioned Responses which are fed to one of the analog inputs of the AX08. The Test and Conditioned Responses are matched in phase by their appropriate sort codes so that one region of core accumulates Test, and another, Conditioned Responses during averaging. In neurophysiological research the effects of conditioning stimuli upon reflex responses may be investigated by recording the response to a test stimulus in the absence of conditioning (Test Response) and then the response to the test stimulus during stimulation of another structure (Conditioned Response). The magnitude of the effect of this conditioning can be estimated by comparing Test and Conditioned Responses. The time course of the conditioning effect can be investigated by estimating its magnitude at different time intervals between Conditioning and Test Stimuli.

FLOW CHART OF SECTION III PART 2



In Section III Part 2 of the program the magnitude of each averaged response is estimated by obtaining the sum (integral) of the absolute values of the data points between the cursors set for each average. In the simplest case where there are two averages (one of the Test Responses and the other of the Conditioned Responses of one analog input) the magnitude of the conditioning effect is estimated by dividing the sum (integral) derived from the averaged Conditioned Responses by the sum from the averaged Test Responses, and by multiplying this by 100 to obtain a percentage:

$$\frac{\text{Sum averaged Conditioned Responses}}{\text{Sum averaged Test Responses}} \times 100$$

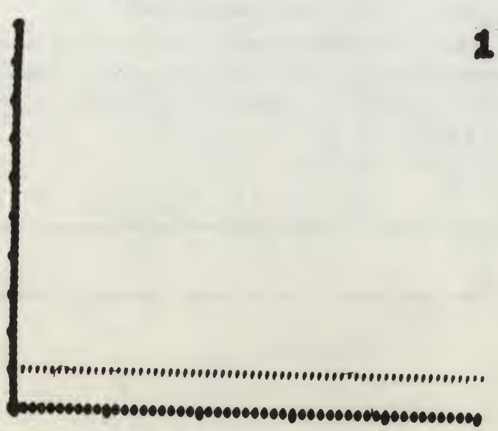
On this basis a value of 100% indicates no change, $<$ 100% a depression of the Test Response, and $>$ 100% a facilitation of the Test Response. The value of this percentage is stored in Part 3 of the program and is used in the construction of the conditioning curves (See below).

In the protocol of the experiment both the Conditioned Response and the Test Response of one analog source are fed to the <u>same</u> analog input. This avoids any differences between the two responses caused by unequal amplification or A/D conversion. The Test and Conditioned Responses are distinguished by their sort codes: the computer recognizes the Test Response by its <u>smaller</u> sort code. For example:

| | | | |
|---|---|---|---|
| average 1 = | averaged test response 1 | sort code 2 | analog input 1 |
| 2 = | averaged conditioned response 1 | 3 | 1 |
| 3 = | averaged test response 2 | 2 | 2 |
| 4 = | averaged conditioned response 2 | 3 | 2 |

4.3.2 <u>Use of Section III Part 3</u>

Part 3 is entered from Part 2 by typing P. On the screen a conditioning ("Simon") curve appears. At first no values appear on the graph and only the axes are displayed:



Which conditioning ("Simon") curve appears depends on the analog input number of the response that was last displayed in Part 2 before Part 3 was entered. The number of the analog input is displayed in the upper right corner. The X-axis is divided in steps of 10 msec, with every 100 msec double dotted. The Y-axis is divided in steps of 20%, with every 100% point double dotted and with the line Y = 100% drawn at lower intensity.

A

Before a point in the conditioning ("Simon") curve can be entered, the appropriate sums of the Test and Conditioned Responses must have been typed out during Section II Part 2 of the program. The percentage value of the quotient of these sums (see above) forms the Y-coordinate of the "Simon" point (admissable range: 0 - 4095%). To input the X-coordinate of this point, $T_{ct}$, the interval in msec between the onset of conditioning and test stimuli, must be typed (admissable range: 0 - 4095 msec, or pro rata, if other values are assigned to the scale markings of the X-axis).

Typing A causes the computer to ask for the condition-test interval, by printing

T.CT =

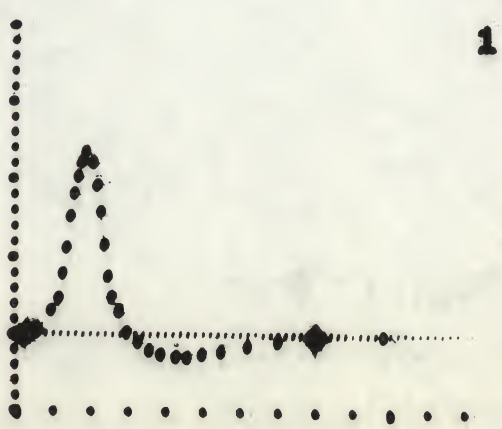
on the teletype. Type now the appropriate time interval in milliseconds (no fractional part is admitted) and input the number with CR. The "Simon" point has now been entered on the graph displayed on the oscilloscope.

CR, LF

These keyboard controls both cause the program to return to Section II Part 2 and to display an averaged Conditioned Response —— LF for the analog input of the momentary displayed "Simon" curve and CR for the next analog input.

Z

Typing Z brings the program back to Section III Part 1. Now new averages can be taken at a different condition-test intervals. Cycling successively through Section III Parts 1, 2 and 3, allows a whole conditioning ("Simon") curve to be composed. For example:



The maximum number of points for each "Simon" curve is 30. Any attempt to exceed this number results in the error message:

FULL

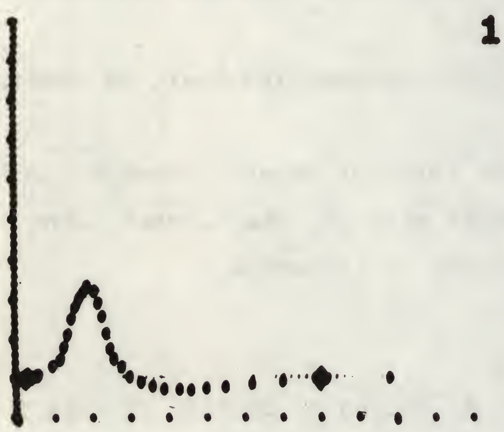
on the teletype.

X, C, W, N

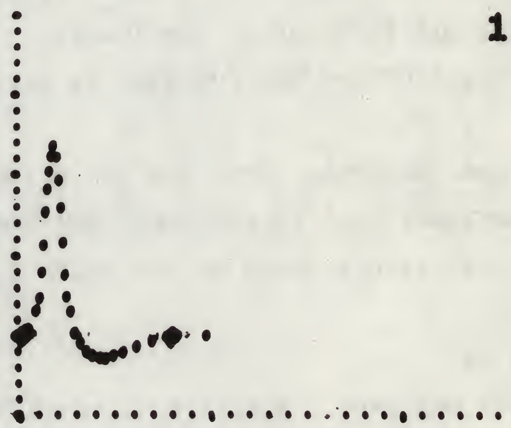
Keyboard controls X and C expand and contract the display vertically. Keyboard controls W and N widen or narrow the display horizontally.

Note: Contract and Narrow cannot be used beyond the limits for the scales of the axes specified on p. 34. If they are exceeded the program loops in the display subroutine. Should this happen stop the program and restart at Load Address $4615_8$.

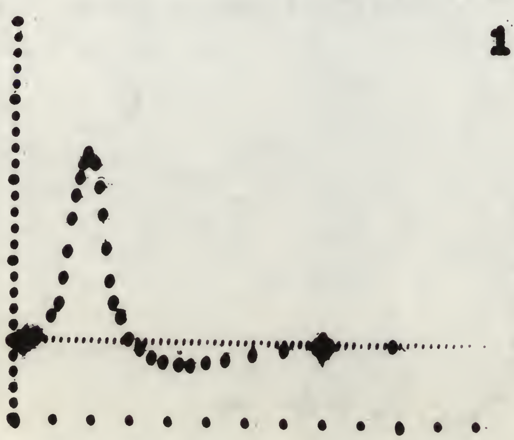
Examples:




control




"X" and "N"




"W"

CURSORS

Note that the left and right cursors can be moved along the "Simon" curve by analog input knobs 2 and 3 respectively.

T

Typing T causes a type out of the points on the graph between left and right cursors.

analog input number
conditioned response number
test response number
point position of left cursor
point position of right cursor

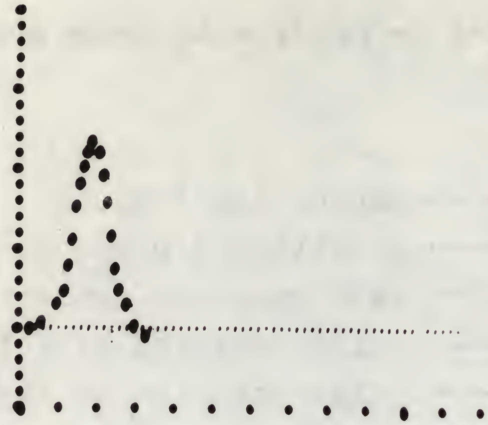
```
1   1   2      1   29
    3     100
    6     102
   10     132
   12     148
   13     180
   14     213
   15     253
   16     292
   17     304
   18     323
   19     336
   21     322
   22     293
   23     252
   24     217
   25     177
   26     154
   28     131
   30     104
   33      91
   36      80
   39      75
   43      72
   46      71
   50      75
   55      79
   62      86
   73      93
   80      93
```

Q

Typing Q causes the program to quit or interrupt the type out of the points.

E

Typing E erases the points between the left and right cursors.

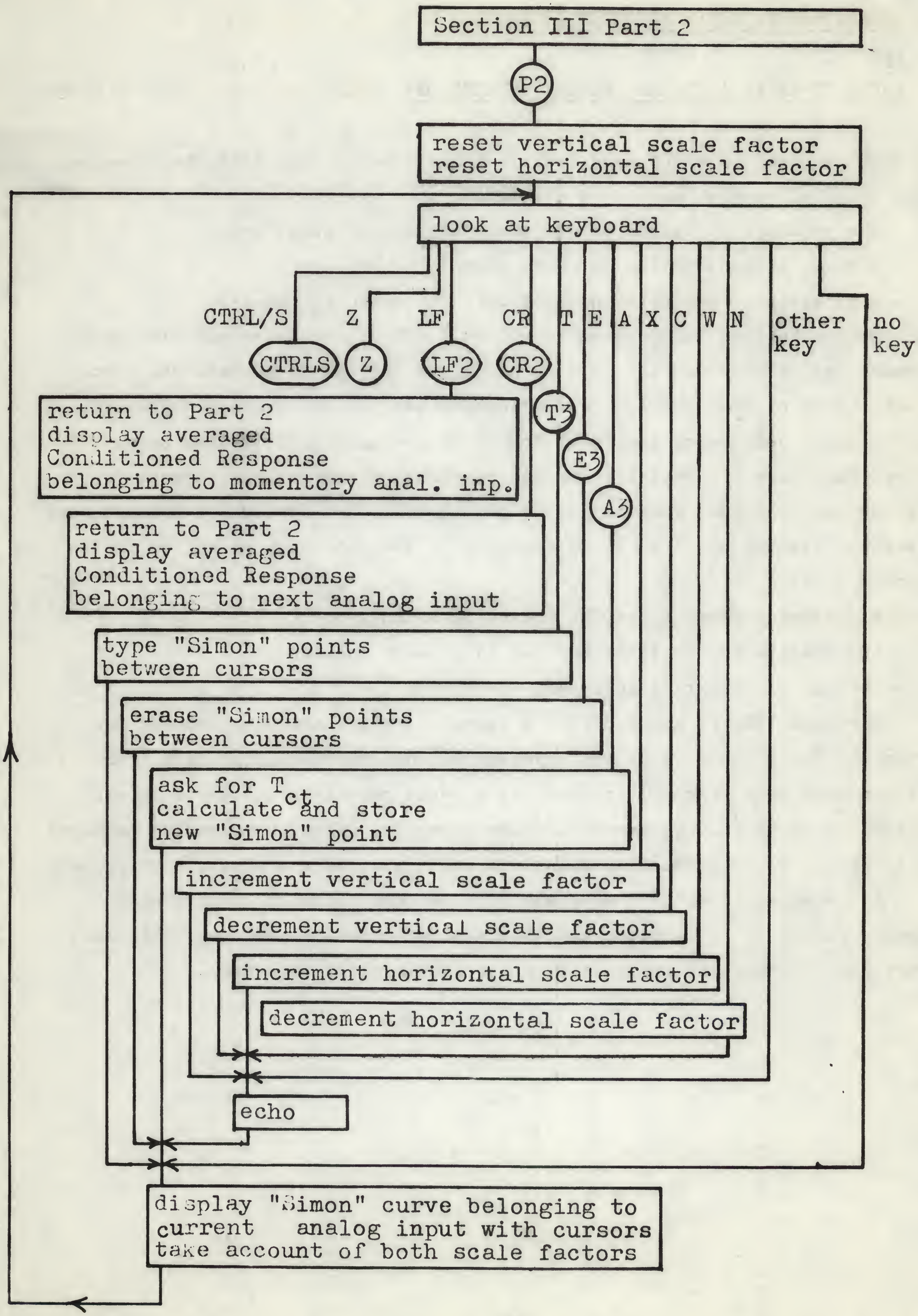


CTRL/S

Typing CTRL/S zeroes all the locations storing the "Simon" points and returns program control to Section III Part 1 for data to be obtained for new conditioning ("Simon") curves.

The flow chart for Section III Part 3 is given on the following page

FLOW CHART OF SECTION III PART 3



## CHAPTER III

### COMPARISON OF LAB-8 ADVANCED AVERAGER

### AND

### LAB-8 ADVANCED AVERAGER (ROTTERDAM VERSION)

Both programs have the same overall structure. Certain features, however, of the original version have been deleted:

- the possibility to run on different Lab-8 configurations,
- with or without option XR,(sort codes),
- with extended memory or without only the basic 4K memory.

The Rotterdam Version is designed to run on a PDP-8/L computer with the basic 4K memory and with option XR. It does not offer the use of an external clock or the definition of high and low resolution periods for the averaging process. It cannot compute confidence limits or trends of averages or plot data on an X-Y plotter. These are all restrictions and therefore disadvantages, except for users who have the configuration described above in the System Requirements, and who wish to use the Lab-8 Advanced Averager in the same general way as in the Rotterdam Version.

The Rotterdam Version has the following advantages:

- the compilation section (Section 1) is much shorter,
- Section II is little modified,
- Sections III, IV and V of the original version have been reduced to Section III Parts 1 and 2, so that cycling between the averaging part (Part 1) and the output part (Part 2) is possible without reloading the paper tapes. This feature saves a large amount of time otherwise spent in reloading sections of program and is of particular value where a number of averages are to be made,
- it is expanded with a more specific program allowing conditioning ("Simon") curves to be computed and displayed graphically on-line. This last feature can of course be replaced by other user-specific programs.

## CHAPTER IV

### ORGANIZATION OF THE DATA STORAGE

This chapter provides an introduction to the core map of the Advanced Averager Rotterdam Version program, and allows access to the averages and "Simon" curves. The information is essential for any user wishing to modify or extend the program, or to understand its full working. Careful notice should be taken of the listing of the Lab-8 Advanced Averager (reference 7) and of the listings in the Appendix below. These two can be regarded as the listing of the Lab-8 Advanced Averager Rotterdam Version.

Where the points of the averages and the points of the "Simon" curves will be stored by Section III, is defined during Section I. At the end of Section I this information is punched into the control tape. When the control tape is read in after loading of Section II, the framework of the data blocks is built up. A core map of Section III (on the following page) shows this organization:

## CORE MAP OF SECTION III

### in respect to

### THE ORGANIZATION OF THE DATA STORAGE

| address | | |
|---|---|---|
| 0000 | parameters, linkages, constants, temporary storage | |
| 0227 | | |
| 0230 =C(JOBLIS)+1 | job 1 | job list |
| | job 2 | |
| | ⋮ | |
| | last job | |
| | 0000 | |
| C(ADMPXA)+1 | multiplex register list | |
| C(ADCNL)+1 | channel display list | |
| | 0000 | |
| | 0000 | |
| C(ADBUFA)+1 | A/D conversion buffer | |
| C(ADBUFB)+1 | average block 1 | |
| | average block 2 | |
| | ⋮ | |
| | last average block | |
| | 0000 | |
| C(ADSIM) | "Simon" block 1 | |
| | "Simon" block 2 | |
| | ⋮ | |
| | last "Simon" block | |
| | 0001 | |
| | free | |
| 3400 | Section III program instructions | |
| 7600 | loaders | |

Each average defined in Section I gets its own job, a set of 7 words (J1 to J7) which describe that average:

job J1: bit 0 = always 0
bit 1-5 = analog input number of this average
bit 6 = always 1
bit 7-11 = position of this average in sampling list

J2: bit 0-3 = always 001
bit 4-11 = sort code of this average

J3: minus the number of data points in this average

J4: always 6201

J5: startaddress -1 of accessory average block

J6: horizontal scale factor for the display of this average in the View Average mode

J7: bit 0-5 = (position on the screen where x = 0)/10
bit 6-11 = (position on the screen where y = 0)/10

Beginning at address 0230 = C (ADJLIS) + 1 the jobs are sequentially stored in core. After the last job follows a 0 to denote the end of the job list.

The multiplex register list, startaddress C (ADMPXA) + 1, gives the program information about which and how many analog inputs are used.

In the channel display list, startaddress C (ADCNL) + 1, for each analog input there are two words (DW1 and DW2) that give information about the way of displaying the analog input:

DW1: horizontal scale factor for the display of this analog input in the View Input mode

DW2: bit 0-5 = (position on the screen where x = 0) /10
bit 6-11 = (position on the screen where y = 0) /10

The channel display list is closed with two 0's.

When the averaging process is working during Section III Part 1 the samples are temporally stored in single precision format in the A/D conversion buffer, startaddress C (ADBUFA) + 1. After the sampling period the samples are added to the accessory data points provided that the sort code matches with the status of the contingency register. The length of the A/D conversion buffer is ID as

I = number of analog inputs

D = number of data points per average.

With each average corresponds one job and one average block. The startaddress -1 of an average block is given in J5 of the corresponding job. The structure of an average block is as follows:

average block

| | |
|---|---|
| C(J5)+1 | sweep counter |
| | lower part of first point of this average |
| | upper part of first point of this average |
| | lower part of second point of this average |
| | upper part of second point of this average |
| | ⋮ |
| | lower part of last point of this average |
| | upper part of last point of this average |
| | 0000 |
| | 6201 |

In the average block the points are stored as nonnormalized sums in double precision format. Notice the unusual way of storage: first the lower part and then the upper part. To get their proper values they have to be divided by the sweep counter. The sweep counter is incremented by 1 every time a new set of samples is added to this average. After the last average block follows a 0 and an unused word. The length of an an average block is 2D + 3 as D is the number of data-points per average.

For each analog input one "Simon" block is set up:

"Simon" block

| |
|---|
| S1: minus number of points in this "Simon" curve |
| S2: analog input number of this "Simon" curve |
| S3: average number of Conditioned Response belonging to this "Simon" curve |
| S4: average number of Test Response belonging to this "Simon" curve |
| upper part of conditioned sum |
| lower part of conditioned sum |
| upper part of test sum |
| lower part of test sum |
| x-coordinate of first "Simon" point |
| y-coordinate of first "Simon" point |
| ⋮ |
| x-coordinate of thirtieth "Simon" point |
| y-coordinate of thirtieth "Simon" point |

The first "Simon" block starts at C (ADSIM). The last "Simon" block is followed by a 1. The length of a "Simon" block is $104_8$.

The total length of all these buffers, lists and blocks is:

$$W = (2P + 10)A + (P + 71)I + 6 \qquad \text{(decimal)}$$

where

$P$ = number of data points per average

$A$ = number of average

$I$ = number of analog inputs

The rest of the core up to 3400 is free. When few averages are computed this area could be used for other user programs

## APPENDICES



### References

1 Introduction to Programming, PDP-8 Handbook Series,
Digital Equipment Corporation, Maynard, 1970.

2 Programming Languages, PDP-8 Handbook Series,
Digital Equipment Corporation, Maynard, 1970.

3 PDP-8 Family Paper Tape System User's Guide,
Digital Equipment Corporation, Maynard, January 1970.
(no. DEC - 08 - NGCC - D)

4 PDP-8 AX08 Laboratory Peripheral,
Digital Equipment Corporation, Maynard, November 1969.
(no. DEC - 08 - H6BA - D)

5 Lab-8 Basic Averager User's Manual,
Digital Equipment Corporation, Maynard, June 1969.
(no. DEC - LB - T20A - D)

6 Lab-8 Advanced Averager User's Manual,
Digital Equipment Corporation, Maynard, February 1970.
(no. DEC - LB - T30B - D)

7 Listing of Lab-8 Advanced Averager (Ver C), includes Papertape Version and DSK/DTA Version,
Digital Equipment Corporation, Maynard, October 1969.
(no. DEC - LB - U17/18C - LA)

Restart addresses of Lab-8 Advanced Averager Version M

| | | |
|---|---|---|
| Section I | 7603 | |
| Section II | 7603 | |
| Section III Part 1 | 7603 | |
| Section III Part 1 | 6530 | (without destroying "Simon" curves) |
| Section III Part 2 | 4026 | |
| Section III Part 3 | 4615 | |