Automated EEG Spectrum Analysis System

by

Marinela C. Laguna

Dissertation submited to the Faculty of the Graduate School
of the New Jersey Institute of Technology
in partial fulfillment
of the requirements for the degree of
Master of Science in Biomedical Engineering
1985

APPROVAL SHEET

Title of Thesis: Automated EEG spectrum analysis system

Thesis and Abstract Approved:

______________________ ____________

Dr. Stanley Reisman Date

Associate Professor

Electrical Engineering

______________________ ____________

Date

______________________ ____________

Date

______________________ ____________

Date

VITA

Name: Marinela C. Laguna

Permanent Address:

Degree and date to be conferred: M.S.B.E., 1985

Date of birth:

Place of birth:

Secondary education: Liceu "Gh. Lazar", Sibiu, Romania, 1976

Collegiate Institutions attended	Date	Degree	Date of Degree
I.P.T.V., Timisoara, Romania	9/76-7/81	M.S.E.E.	July, 1981
N.J.I.T., Newark	1/83-5/85	M.S.B.E.	May, 1985

Major: Electrical Engineering

ABSTRACT

TITLE: Automated EEG Spectrum Analysis System

Marinela C. Laguna, Master of Science in Biomedical Engineering, 1985

Thesis directed by: Dr. Stanley S. Reisman, Associate Professor, Department of Electrical Engineering

A computer system has been designed to acquire EEG data from monkeys and to perform the spectral and statistical analysis. The system is capable of processing two channels simultaneously, acquire 64 Kbytes of EEG data for each channel on floppy disk, do the spectrums and averages of the spectrums, and calculate the histogram amplitudes of 13 frequency bands in a range of 0-71.1 Hz.

The software gives the user a lot of choices with respect to data acquisition and processing. The relative spectral power differences between "no-drug" and "drug" experiments give information about the drug effects on the central nervous system.

The present system is easier to control with respect to previous similar systems because it is capable of performing the experiments in an automated manner without any human intervention. Another quality of the system is the simplicity and small space needed.

Being based on software, the system is very flexible and easy to extend. One of the features of the new system is new programs for the statistical comparisons can be done automatically. Another

feature is related to the extension of the system to process 16 channels simultaneously.

Experiments have been performed to validate the system using the previous system results. Results are presented to show the quality of the system.

Investigations have been done into other ways of analyzing the spectral data in order to improve the system.

The results of this system can be successfully used in classifying new drugs.

ACKNOWLEDGEMENTS

I would like to thank Dr. Reisman for his advisement and support of this thesis; Joseph Sia for his supervision and aid in the thesis; Terry Smart for his laboratory aid and help in interpretation of the experimental results; Dr. J. Sepinwall for his encouragement and support; and Domenica Iannicelli for typing this thesis.

TABLE OF CONTENTS

List of Figures

Page

Page

Page

Chapter 1

INTRODUCTION

1.1 Electroencephalography (EEG) signals used in diagnosis and pharmacology

EEG signals are obtained by placing electrodes on different parts of the scalp. They reflect the brain activity, but the brain represents the coordinatory center of all the activities of the body, so people realize that observing EEG patterns can be very important in clinical situations. Years before, visual interpreted EEG was used in the diagnosis of different diseases. However, because of the low signal levels and as a result of years of experience, the research people decided to quantify the EEG signals (Quantified Electroencephalography, QEEG) in order to obtain more accurate conclusions with respect to diagnosis. Computerized EEG analysis was the next step in this research field. With the computer, the analysis can be done in time or in frequency giving the possibility to visualize the power of each type of wave, Δ, θ, α_1, α_2, β_1, β_2, in different clinical cases.

One of the recent studies [2], uses computerized spectral EEG analysis to compare normal and dyslexic children behavior. This study and its experimental results are very important not only for improving the methods of diagnosis in such a disease, but also for the way that EEG spectrum results have to be considered and correlated during an experiment.

Three minutes of passive eyes-closed and eyes-open EEG were recorded before and after 4-5 hours of behavioral tasks in 10-12 year-old boys of normal intelligence and neurological status. Half were severely reading disabled, half were reading normally. Bilateral, central, parietal, and mid-temporal EEG referenced both to vertex and to linked ears were recorded. The EEG was digitized at 256 points/sec. and FFT was done for each second epoch (after eliminating the artifacts) with 69 points/sec. Averages of FFT power spectra of artifact-free 1 sec. epochs for 2-2.5 minutes were computed. Based on these values, several coefficients were calculated for both normal and dyslexic children and big differences were found (Fig. 1.1a, 1.1b).

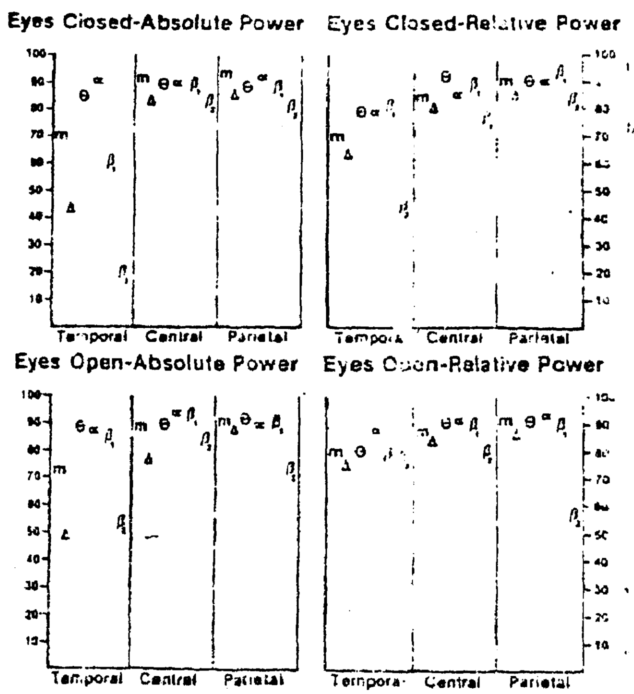


Fig. 1.1.a. Control Group Reliabilities

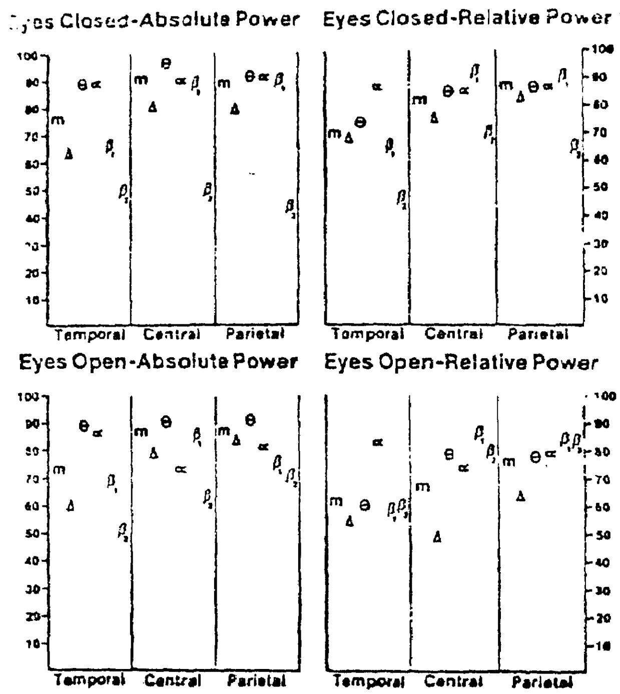


Fig. 1.1.b. Dyslexic Group Reliabilities

It was found also that absolute power is as reliable as relative power and is warranted whenever possible, since the interpretation of findings based solely upon relative power can be ambiguous (for example, a decrease in relative delta activity may result from decreased delta activity, increased activity in other bands, with delta activity unchanged, or some combination of the above). These findings support the utility of EEG power spectra as a reliable index of brain functions for studies of normal and learning disabled children.

Many research people were involved in correlating inter-individual variations in EEG activity with intellectual capacity [3]. Since 1933, studies are done in this area and during the years, many pros and cons have been presented and theoretically argued, relative to this subject. The reason for the discrepancies might consist in the heterogeneity of the sample with respect to the intellectual capacity (IQ), insufficient or inaccurate quantitative evaluation of EEG activity (alpha rhythm was predominantly used), heterogeneity with age (EEG changes with age and will change the correlations with IQ), inadequacy in measuring intelligence. However, recent studies [3], demonstrate that age-standardized EEG parameters can be correlated to IQ scores. They are based on some hypotheses:

a) small but consistent relationship between EEG at rest and IQ scores in normal children;
b) large correlations for mildly mentally retarded children;
c) the correlations depend on the maturation of brain function (more mature EEG parameters correspond to higher IQ scores) and are larger in the frequency bands where developmental change with respect to EEG parameters takes place;
d) the frequency distribution of the EEG activity is more important to the size of correlations than the topographic distribution.

Correlations between EEG and IQ scores were computed separately for a group of normal and a group of mildly mentally retarded children and the hypotheses were experimentally verified.

The most important conclusion of this study was that the earlier discrepant results are due to a visual rather than a computerized analysis of the EEG. Spectral parameters and their standardization for age and highly complex IQ tests were possible only by using the computer.

One of the other areas of diagnosis where EEG showed to be of a lot of importance is in finding the mild degree of dementia [4]. Using two groups of elderly patients, one having the disease and a control normal group, it was found that the theta and beta activity for those two groups is very different, no difference in alpha and delta activity, and decreasing in the average mean frequency for demented subjects, mean frequency for (5-20) Hz interval being defined as:

$$\text{mean freq. (5-20 Hz)} = \frac{\sum_{f=5}^{20\ \text{Hz}} (F(f) \times f)}{\sum_{f=5}^{20\ \text{Hz}} F(f)}, \quad F(f) = \text{the power spectrum.}$$

This type of relationships have to be standardized with the age in order to be used for diagnosis.

A big problem in medicine was controlling the brain function before and after the open-heart surgery. Research done with respect to this problem shows that pre- and post-operative quantitative EEG (QEEG) give the right information [5]. Fast Fourier Transform (FFT) was found to be the best method, in comparison with combined period and amplitude analysis. Correlations between post-operative cerebral complications and FFT mean frequency were found. The parieto-occipital regions appeared to reflect the operative strains more clearly than the fronto-central regions and generally, the correlations EEG - Brain function before and after open-heart surgery were found to depend on the type of the cardiac disease.

EEG evaluation is not important only in diagnosis, but also in pharmacology. The quantitative analysis of human electroencephalograms with a digital computer, known as "computer-analyzed EEG"

(CEEG), is one of the most significant advances for the development of new psychotropic drugs [1]. In recent years, the findings indicate that CEEG profiles are predictive for psychotropic properties of new compounds. Based on EEG data obtained after administration of known drugs (drug data base), and using newly developed programs, the computer is able to classify new drugs into one of the existing psychotropic drug groups in a fully automated and statistical manner.

The significance of the QEEG was outlined in a series of recent publications [1]. In a recent report to the President of the United States, under the heading "Major advances in the past five years in methodological development" it was reported:

> "The QEEG and animal behavioral techniques extrapolated to man may identify useful drugs for mental illness and distress".

Other publication, "Future directions and developments" reports that:

> "For clinical psychopharmacology we would predict...development of new principles for the design of more specific, less toxic antipsychotic drugs. The QEEG may become a more generally used tool in the evaluation of valuable new compounds."

In the report of the Task Force Committee on bioavailability and bioequivalence of psychotropic drugs of the American College of Neuropsychopharmacology, it was stated:

> "The use of pharmacological end-points to assess bioavailability is still in the developmental stage and there is currently no recognized standard approach, although pharmacodynamic methods such as QEEG..., which are responsive to the effects of drug which cross the blood-brain barrier, provide the most logical opportunity for each development."

The most important method of doing QEEG is CEEG. This method was successfully used to determine the acute pharmacological effects of an active psychotropic drug ingredient at the site of drug action (brain). The quantitative pharmacological EEG (QPEEG) has almost all the requirements of an "ideal" bioavailability method [1].

In order to define CEEG for psychotropic drug development, it is good to compare it with "conventional" EEG for clinical diagnosis [1]. Whereas "conventional" EEG is primarily concerned with abnormal waveforms such as: slow waves (theta or delta), sharp waves, spikes, spikes and waves, focal abnormalities, and paroxysmal activities, the CEEG for pharmacology is primarily concerned with normal waveforms and normal frequency (alpha, beta, and faster activities) and amplitude distributions.

The use of the CEEG in psychotropic drug development is justified: psychotropic drugs (compounds effective in human behavior) produce changes in human brain function; EEG is the simplest, most economical and objective method to study the continuous function of the human brain. The psychotropic drug-induced changes in EEG are relatively small to evaluate visually. Therefore, it is imminent that EEG has to be quantified and the most advanced quantifications are via specific-purpose computers.

There are three most important phases when CEEG is used [1]:

-phase 1, safety clinical pharmacology trials

-phase 2, psychotropic drug studies in humans

-phase 3, preclinical investigations with rats, mice, guinea pigs, and particularly monkeys, to determine the central nervous system (CNS) effects of drugs and to predict their psychotropic properties.

Good results were obtained in recent years by using computerized EEG within quantitative pharmacological EEG, in phase 1, safety clinical pharmacology trials. Quantitative pharmacological EEG is the process involving the use of digitized EEG and various statistical procedures to establish the central effectiveness of a new compound in humans, which consists of

- -quantity of CNS effects
- -estimation of the onset and duration of CNS effects
- -prediction of its clinical usefulness (psychotropic properties) after single oral administration in normal subjects or patient populations.

The most important questions which can be answered with QEEG are the following:

1. Does the drug produce any significant effects on human brain function?
2. What is the minimum CNS-effective dosage?
3. What is the onset and duration of CNS effects?
4. Does the drug have significant time and dose-related CNS-effects? What are the pharmaco-dynamics of the drug at the CNS level?
5. Does the compound have any psychotropic properties for clinical use?

As a conclusion for the importance of CEEG, I would like to enumerate the facts and hypotheses regarding its application in psychopharmacology:

Facts:

1. EEG is the only objective method to study the continuous function of the human brain.
2. All established psychotropic drugs (anxiolytics, antidepressants, psychostimulants, and neuroleptics) produce significant effects in human brain function.

3. CEEG is the simplest method to determine the effects of psychotropic drugs on human brain function.
4. Psychotropic drugs produce significant dose and time related effects on human brain function, which can be demonstrated by CEEG.
5. The same drugs, with similar dosages in the same or even in different populations, produce similar CEEG profile (CNS effects of some drugs are replicable).
6. Psychotropic drugs with similar pharmacological effects <u>in animals</u>, produce similar CEEG profiles.
7. Psychotropic drugs with similar clinical (therapeutic) effects <u>in patients</u> produce similar CEEG profiles. Therapeutical "unequivalent" compounds (antipsychotics versus anxiolytics) produce different CEEG profiles (unequivalent physiological effects).

<u>Hypotheses</u>:

1. There are close correlations between human behavior and EEG changes.
2. The physiological "equivalency" as established by the similarity of the CEEG profiles of a new compound to an established drug, indicates the therapeutic (psychotropic) "equivalency".
3. There are close correlations between the CEEG response (CEEG profiles of an individual to a test dose drug) and the therapeutic efficacy after chronic administration . The more typical the CEEG profile, the better the therapeutic outcome.

CEEG in pharmacology, in the development of new psychotropic drugs, is used in the following way: CEEG profiles for all the known psycho-active compounds are obtained by using period and spectral density EEG analysis programs. These profiles represent from now on the data base and they are stored in the computer memory. The same programs are used to obtain the CEEG profiles of the new compounds. The data base is then used to be compared with these new profiles. Correlation statistics is then performed and the new drugs are classified in one of the well-known categories of psychotropic drugs, of course with a certain acceptable probability.

Before using a computer system to classify new compounds the system has to be validated. This is done by taking the well-established drugs (well-known) and analyse them, compare with the data base, then apply the statistics programs to classify them. To give an example of validation, I would like to speak about the research center of New York Medical College [1]. Its computer data base contains the CEEG effects of 85 clinically well-known psychoactive compounds in 715 male and female volunteer subjects in the age range of 21 to 25, collected in 79 quantitative pharmaco-EEG studies. The results of the validation of this computer system is given in Figure 1.2.

As we can see, the probability to classify incorrectly is 0.096 for anxiolytics, 0.2 for antidepressants, 0.058 for psychostimulants, 0.6 for neuroleptics, and 0.166 for overall situation.

Date Base Drug Classes	No. of Drugs	Correctly Classified	Not Classified	Incorrectly Classified
Anxiolytics	52	45	2	5
Antidepressants	35	17	11	7
Psychostimulants	17	14	2	1
Neuroleptics	10	3	1	6
Overall	114	79	16	19

Fig. 1.2

Computer System Validation Results

The fact that from 10 neuroleptic drugs only 3 were correctly classified, 1 not classified, and 6 incorrectly classified is believed to be due to the fact that the doses were very low. So even though the probability of classifying incorrectly, 0.6 is considered to be big, generally, taking in account that the dose was low and the sample sizes were only 4 to 6 in a variety of studies, the classification of the well-established compounds was considered to be accurate enough.

Certainly, there is no other method available in humans or in animals, to predict the psychotropic properties of different compounds, after single oral dosages in such a systematic and reliable manner.

1.2 The automated EEG spectrum analysis system - generalities

This system is the subject of my thesis which I did at Hoffmann-La Roche Inc., Nutley, New Jersey.

The automated EEG spectrum analysis system was designed to obtain CEEG profiles in the phase of preclinical investigations, with monkeys (phase 3). Spectral analysis profiles are obtained after the administration of the drug, they are stored and at the same time, some spectral parameters are printed out for the user. The system does not have a data base and it is not used yet to classify drugs automatically (by highly statistics software) but these features are not difficult to develop.

To be a little more specific, the automated EEG spectrum analysis system performs on line analysis, simultaneously for 2 channels (2 monkeys).

The EEG data (variation in time) is acquired with an Apple II computer which has a digital oscilloscope in it, providing the user the EEG pattern as it is recorded. The data is then saved on disk and will be analyzed immediately or later, this being the choice of the user and also depending on the type of the experiment. Analysis of the EEG data consists in performing the FFT, saving spectral data on disk without erasing the EEG data, averaging every 8 spectrums (also an average of 64 spectrums can be done) and displaying the averages on the screen in a band of approximately 70 Hz, calculating histogram amplitudes, displaying the histogram, calculating pie chart coefficients and displaying it. Printouts of the histogram and pie chart are done in order to file the information. The comparison between CEEG profiles filed, in order to characterize the psychotropic properties of the drug is done by the user.

Before the automated EEG spectrum analysis system was designed, the same experiments were done with the help of another system designed by Data General Corporation. Both systems consist of 2 stages or 2 main parts: the data acquisition and the data analysis.

For the old system, the data acquisition is performed by storing the data on line, from the electrodes using a magnetic tape recorder unit. The recorder is a very big unit equipped with filters and amplifiers to adjust the signal. The data analysis part contains: UA14 spectrum analyser, 100Q spectrum averager by Federal Scientific, a minicomputer and a hard copy unit.

The spectrum analyzer receives the EEG signal from the tape recorder, played back at a rate 32 times faster than the recording rate (in order to reduce the data analysis time). The frequency analysis is set to be done from 0 to 1000 Hz but the actual range, taking into consideration that the tape was played back faster, will be 31.25 Hz = 1000/32.

The spectrum analyzer linear analog output is fed into the spectrum averager. The average of all the spectrums obtained in a 15 minute period of the experiment (the experiment is practically the data acquisition stage) is performed. By averaging, the amplitude of the deterministic signal is increased with respect to the amplitude of randomly fluctuating noise, so the signal to noise ratio is improved.

The computer used to control the operating modes of the spectrum analyzer and spectrum averager is a Data General "NOVA" computer.

For each value of a spectrum component the computer reads a ten bit value and for each spectrum gives 400 values, which makes the resolution in frequency to be 31.25 Hz/400 = 0.078125 Hz. The 400 values for the average of the spectrums obtained in the 15 minute experiment, are stored on disk.

The hard copy unit gives a printout of a compressed spectrum for the 60 minute experiment (4 spectrums, practically 4 averages, obtained at every 15 minutes in an hour of experiment). Also, another printout is provided, containing the percentage of power in each of the 8 bands of the 31.25 Hz range, for each of the 4 averages/hour experiment.

The system that makes the subject of my thesis has the same purpose and works in the following way:

1. The data acquisition part is performed with the help of the Applescope, digital oscilloscope of the Apple II computer, which acquires 2 sweeps of data at the same time (2 channels, channel A and B). Each sweep has a duration of 3.6 sec and is digitized at a rate of 512 points/sweep. The digitized data is not stored directly on disk, but an intermediate 8 Kbytes of memory are used to hold up to 16 sweeps of data, 8 for each channel (16 x 512 = 8 x 2 x 512 = 8 x 1024 bytes). After the intermediate memory was entirely used, all of its content is stored on the data disk drive 2. (There is another disk drive, 1, for the program disks). The total amount of data stored, at the end of the acquisition process which lasts for approximately 9 minutes, is 64 sweeps for each channel, in other words, 2 x 64 x 512 bytes = 64 Kbytes.

The fact that we have 64 sweeps for each channel at the end of the experiment is due to the possibility of the old system to do averages of 64 spectrums (64 spectrums are obtained with the old system for 15 minutes of experiments). In this way the systems are made compatible (of course the compatibility is not achieved only by this, but at least the systems do the same thing when it comes to averaging).

2. The data analysis part is not so complicated as in the old system. It is formed only by the Apple II computer with the 2 disk drives and an EPSON printer.

   For each sweep the computer calculates the FFT using 512 points for a range of frequency of 0-70 Hz.

   The resolution in frequency is then 70 Hz/512 = 0.269 Hz. The result of the 9 minute experiment was 128 sweeps, 64 for each channel. The 128 spectrum are obtained in a period of approximately 40 minutes. Two final averages (each of 64 spectrums) are then obtained (1 for each channel) and displayed on the screen, also printed out. The following process is the statistical analysis of the data which lasts about 8 minutes and consists in doing 2 histograms and 2 pie charts (1 for each channel).

In order to compare the results of these 2 systems, the new system was equipped with a supplementary program which takes only 31.25 Hz range from the total 70 Hz range, calculates its total power and the relative power of the bands between 0 and 31.25 Hz (8 bands). The new system prints out these relative powers and also the relative powers for the 13 bands included in the 0-70 Hz range.

Comparing the 2 systems, a few observations have to be made. The resolution of the new system is not as high as the resolution of the old one, but it can sweep 2 channels at the same time (this is the trade off). In the same time the noise is minimized because the acquisition of the data is done by software. Another advantage of the present system is that it does not contain so much expensive equipment (all the components of the old system are very expensive), and it does not need a lot of space as the other one. The Apple computer is used with a maximum efficiency, while the NOVA computer is used at only a small part of its total capacity (it only coordinates the other system components).

The big tape recorder unit is not needed any more, the data is stored on disk. However, an ordinary disk can not hold more than 140 Kbytes. 128 Kbytes were necessary for 64 sweeps of data for each channel (2 x 64 x 512 = 64K) and 64 spectrums for each channel (2 x 64 x 512 = 64K). This is the situation for 2 channels. If the application has to be extended to 16 channels, a hard disk is necessary.

Another advantage of the system is that it performs the entire statistical analysis, total spectral power, individual bands power, relative powers, and draws the histogram and pie chart. The old system was able to calculate only the total and the individual power, the rest of the calculations and the drawing being the job of the user. In addition, the 0-31.25 Hz range was extended to 70 Hz which provides the opportunity to obtain new effects of some drugs, which, until now, were not able to be seen.

From this point of statistical analysis, to the classification of the new drugs there are other steps which were not as yet implemented which are still done by the user. The data base (results of the experiments with well-established drugs) is not available in the computer but is manually operated by the user. These are the features of the system and they can be relatively easily developed because the main algorithms of manipulating the EEG information are already working and the results were found to be compatible with the results of the Data General Corporation system.

Chapter 2

DESCRIPTION OF THE SYSTEM

The block diagram of the system is shown in Fig. 2.1. Following the signal from the electrodes, the system is formed by: filters, Apple II computer, disk drives, and printer.

2.1 System components

2.1.1 Filters

The system contains two band pass filters, one for each channel. The bandwidth of the filters is 0.5-110 Hz. The signal coming from the monkeys has a large d.c. component which makes it impossible to visualize it on the Apple II computer screen, even though the Apple-scope offers two resolutions at which the waveform is sampled, and for each one, five possibilities to compress or expand the current waveform being displayed from memory. That is why the 0 Hz frequency is not included in the filter band. After the filters, the EEG waveform has no d.c. component and can be visualized.

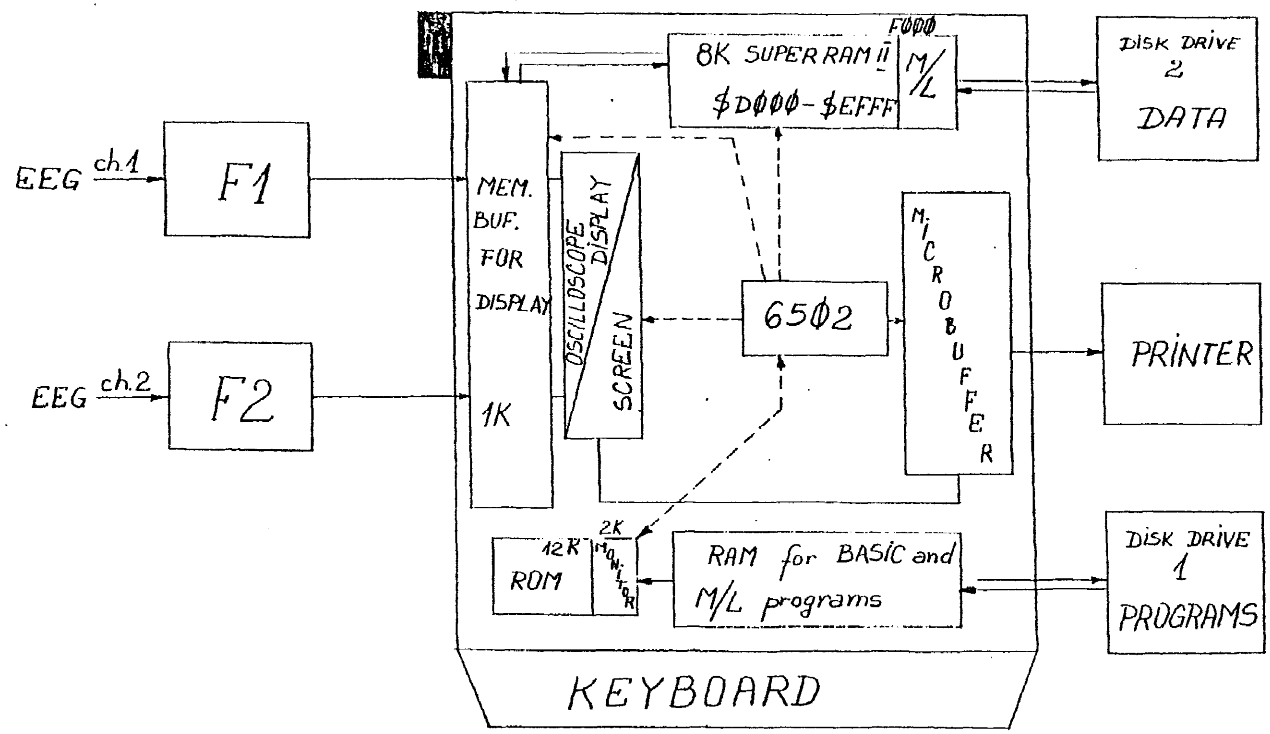


Figure 2.1 Block Diagram of the Automated EEG Spectrum Analysis System

2.1.2 Apple II Computer

a. Introduction

The Apple II computer was chosen to acquire the EEG data and to process it because it has the possibility to be used as both a computer and an oscilloscope. The oscilloscope is called Applescope.

The Applescope system was fortunately designed as a low cost alternative to expensive digital storage oscilloscopes. In addition, the combination of a data acquisition system with a personal computer allows waveform manipulation and different kinds of analysis, which are not available with a digital oscilloscope.

To provide a complete unit for the Apple II computer, a simple operational software package is provided in PROM on the digital circuit card. Once the control is transferred to the operational software, the user is able to manipulate a digital storage oscilloscope, not by switching on a front panel, but by pressing keys on the Apple II keyboard. From now on, the digitized waveform points are stored in the computer RAM memory and are available for the user who wants to use the entire power of the Apple II. Also, an advanced software package is available on floppy disk, "SCOPE DRIVER version 1.3", which contains many of the more common data manipulation routines.

b. Apple II Seen as a Personal Computer

b.1 The Main Board

The main board represents the computer itself and occupies most of the bottom of the computer case. The board contains about eighty integrated circuits and a lot of other components. A general view of the main board is given in Fig. 2.2 in Appendix A. The most important components of the main board are: the peripheral connectors, the microprocessor, the ROMs (Read Only Memory) and the RAMs (Random Access Memory).

b.1.1 Peripheral Connectors

Along the back edge of the Apple's board are situated eight peripheral connectors. The pinout and the signal description for the connectors is given in Appendix A. Fig. 2.2. shows which of these connectors, numbered from 0 to 7, are used. The connectors are called also slots. They are:

Slot 0: SUPER RAM card

Slot 1: Microbuffer card

Slot 2: Scope driver digital card

Slot 3: Free

Slot 4: Scope driver analog card

Slot 5: Free

Slot 6: Disk drives 1,2 card

Slot 7: Free

The other connectors on the main board are:

-Power connector

-USER 1 jumper

-Keyboard connector

-Speaker connector

-Eurapple jumpers

-Game I/O connector

-Video output connectors

-Cassette interface jacks

More information about all the connectors is given in Appendix A.

b.1.2 The Microprocessor

In the center of the main board is situated the brain of the Apple, the 6502 microprocessor which runs at a rate of 1,023,000 machine cycles/second. The 16 bit address bus offers an addressing range of 65,536 bytes. It has 56 instructions available and 13 addressing modes. The description of this microprocessor is given in Fig. 2.3. The instructions and references to the addressing modes are given in Appendix B.

b.1.3 The ROMs

Below the microprocessor are six sockets which may be filled with one to six ROMs containing programs for the Apple, as:

-Apple system monitor

-Apple Autostart monitor

-Apple Integer BASIC

-Applesoft II BASIC

-Apple programmer's aid #1 utility subroutine package

The number of ROM circuits (1 to 6) depends on how many accessories the computer has. In this system, all 6 ROMs are present.

ROM organization is given in Fig. 2.4 in Appendix A. This figure shows that, from page 248 to page 255 (end of the 64 Kbytes of RAM) respectively from \$F800 to \$FFFF, is placed the MONITOR ROM or the AUTOSTART MONITOR ROM. The differences between them are related to editing controls, stop-list and the RESET cycle. Our computer has an AUTOSTART ROM in it and Applesoft BASIC in its

THE MICROPROCESSOR

The 65Ø2 Microprocessor	
Model:	MCS65Ø2/SY65Ø2
Manufactured by:	MOS Technology, Inc. Synertek Rockwell
Number of instructions:	56
Addressing modes:	13
Accumulators:	1 (A)
Index registers:	2 (X,Y)
Other registers:	Stack pointer (S) Processor status (P)
Stack:	256 bytes, fixed
Status flags:	N (sign) C (carry) V (overflow)
Other flags:	I (Interrupt disable) D (Decimal arithmetic) B (Break)
Interrupts:	2 (IRQ, NMI)
Resets:	1 (RES)
Addressing range:	2^{16} (64K) locations
Address bus:	16 bits, parallel
Data bus:	8 bits, parallel Bidirectional
Voltages:	+5 volts
Power dissipation:	.25 watt
Clock frequency:	1.023MHz

Figure 2.3 General Information About the Microprocessor

ROMs. The Integer BASIC is loaded when necessary in the language card. This Apple version is called Apple II Plus.

With respect to the ROM programs, the interest, for the present application, is in Applesoft II BASIC, which is described in "Apple II BASIC programming manual".

In order to understand how the present automated spectrum analysis system was designed, the user should know that the Apple computer offers two versions of BASIC programming language.

-Integer BASIC-fast BASIC suited for many applications in education, game playing, and graphics (see Apple II BASIC programming manual).

-Applesoft floating-point BASIC-better suited for most business and scientific applications.

In Appendix B there is information about the differences between Applesoft and Integer BASIC and about the two versions of the Applesoft BASIC: Firmware Applesoft and Cassette Tape Applesoft.

b.1.4. The RAMs

Going back to Fig 2.2 in Appendix A, we shall now speak about the RAM (Random Access Memory) memory. Below the ROM circuits there are three rows of RAM, of eight sockets each. Totally, this area can hold 24 RAM integrated circuits, and in terms of bytes, up to 49,152 bytes (48K).

Most of the Apple's RAM memory is free to use to store programs or data, but only when the oscilloscope feature is not used. When it is used, some parts of the memory are not available any more, but a lot of space remains for BASIC programs. With respect to the

M/L programs, in this last case, there is not much space available because of the SCOPE DRIVER program. This occupies a part of the space reserved for the M/L. Fig. 2.6 and 2.7 show the RAM organization and , respectively, the system memory map. They are given in Appendix A.

There is a possibility to create more RAM by installing an Apple language card in slot 0, which will create 16K extra RAM in the following manner: 12K are addressed with the ROM addresses and the remaining 4K will be provided by sharing the 4K range $D000-$DFFF.

In our case, the language card installed in slot 0 is called SUPER RAM II. It provides RAM extension and also the possibility to use both Applesoft and Integer BASIC, without switching the control from the ROMs on the firmware card, to the ROMs on the main board and reverse. The characteristics of the SUPER RAM II, provided by R.H. Electronics, are given in Appendix A.

SUPER RAM II works nicely in relation to the Disk Operating System (DOS). Information about how DOS and SUPER RAM II work is given in Appendix A.

b.2 The Apple Video Display

The characteristics of the Apple Video Display are given in Fig. 2.9 in Appendix C.

b.2.1. The video connector allows the connection between the Apple computer and a closed-circuit video monitor. The video signal available at this connector is maximum 1V amplitude, adjustable with a potentiometer. Our computer has a Revision 1 type of main board, which means that the video signal is available also on a single wire-wrap pin, but with 2V maximum in amplitude.

The signal is a NTSC (National Television Standards Committee) compatible, positive composite color video signal. However, the computer can be internally modified to generate a video signal compatible with CCIR standard, which is used in Europe.

b.2.2 Screen Format

If talking with the computer means pressing the keys on the keyboard, receiving the information from the computer means reading the screen. The information is displayed on the screen of the monitor connected to the Apple, in 3 different formats or modes (Fig. 2.10, Appendix C).

1. Text
2. Low-Resolution Graphics (LRG)
3. High-Resolution Graphics (HRG)

More information about how these three modes can be obtained and how they work is given in Appendix C.

Apple II computer has, of course, other input/output features. These special inputs and outputs and also the Apple computer types are presented at the end of Appendix C.

c. Apple II Seen as an Oscilloscope (Applescope)

c.1 System Overview

In order to provide the function of a digital oscilloscope (Fig. 2.15), two high-speed analog to digital converters, controlled by the computer, are used. The screen image is obtained in the mixed text - high resolution graphics mode, text in order to display continuously the trace parameters (4 lines at the bottom of the screen) and graphics in order to display the digitized input signal.

The operational software for the Applescope is stored in a PROM memory in the digital circuit board on slot 2 ($C800-$CFEC; 2028 bytes, around 2 Kbytes of memory). The operational software controls the trace parameters according to the keys pressed and generates the graphics on the monitor display.

There is also another software package available to work with the Applescope, provided on floppy disk and called "Scope Driver, Version 1.3". Using this disk, we have a data acquisition system at our fingertips, a very powerful system, capable of analysing the data (signal averaging, digital filtering, frequency spectrum analysis), to store it on disk or to give a hard copy output to the printer.

The analog and digital card from Fig. 2.15 are exchanging information through a 20 pin connector cable. The analog card contains the analog to digital converters and an 8 bit magnitude comparator. The digital card contains the circuits to control the PROM, the logic circuits used for triggering, the circuits to control the buffer RAM, the sample rate selection and Direct Memory Access (DMA) circuits.

c.2 Data Acquisition and Display Cycle, Operational Commands

Fig. 2.16 shows the way that the information about channel selection, sample rate, scale, and also the result at the converters output circulate between cards during a data acquisition cycle. After a data acquisition cycle the information is stored in the memory buffer for display which appears in Fig. 2.16 under the name of "1024 x 8 buffer RAM". This buffer is located at $1000-$13FF.

Depending on what situation or working mode the user chooses, the Applescope operational software figures out the necessary hardware and selects it by sending 3, 8 bit, control words (Fig. 2.16, 2.17):

CON1 - to the triggering logic block } on analog card
CON2 - to the post trigger delay counter } on analog card
CON3 - to the sample rate control - on digital card

After the hardware selection, the operational software enables the triggering logic block. Then, an interval of time is allocated for the trigger conditions to be met. In this interval, the converter results are stored in the 1K buffer RAM for pre-trigger viewing. In the triggering moment a delay counter is started (called a delay counter because it is set up to count during a specific interval of time given by the triggering moment). When the counter reaches the end of the operation it was set for, it resets the triggering logic and send an interrupt request on the Apple bus. The interrupt request is necessary because now the buffer RAM is full of information and needs to be read. The operational software reads the data and displays the signal sweep on the monitor display. This is the end of a data acquistion and display cycle.

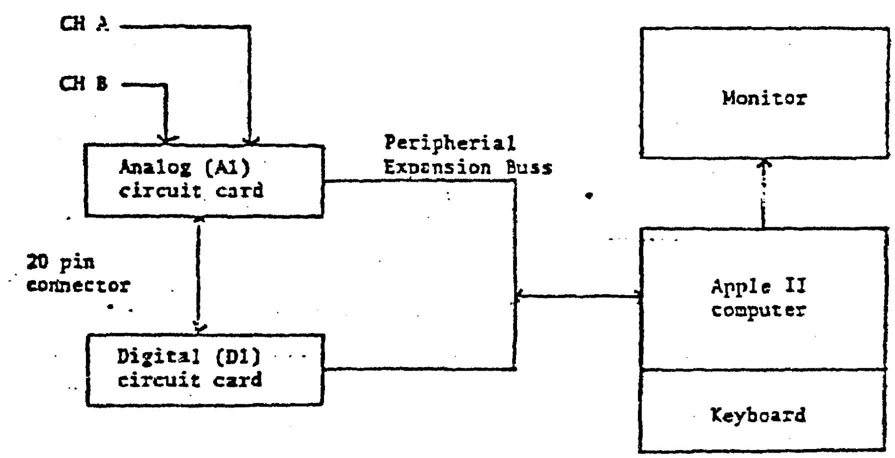


Figure 2.15 Applescope General View

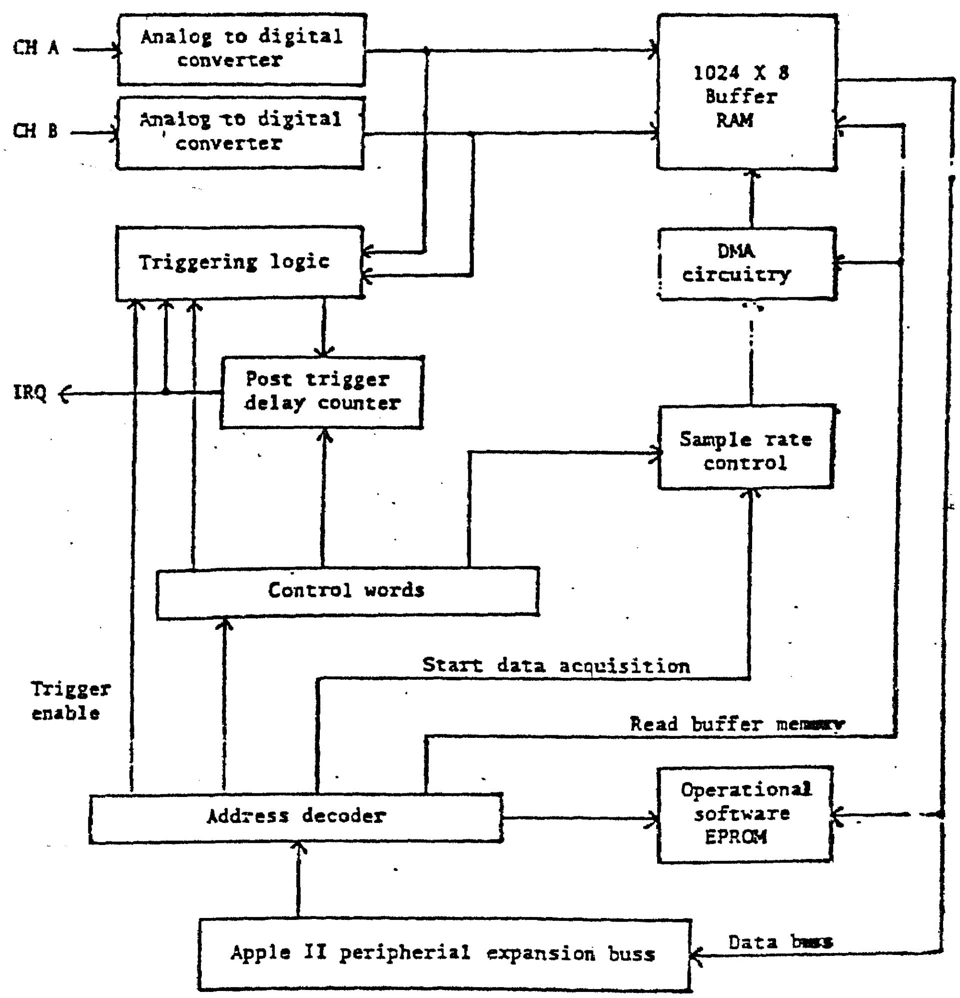


Figure 2.16 Data Acquisition Cycle

CON1		8 bit value used for the trigger level comparison
CON2 -	Bit 7	High enables greater than threshold trigger
	Bit 6	High enables less than threshold trigger
	Bit 5	Low disables channel B trigger
	Bit 4	Low disables channel A trigger
	Bit 3	Low enables channel A data
	Bit 2	Low enables channel B data
	Bit 1	Set the memory cycle and post trigger delay
	Bit 0	
CON3 -	Bit 7	Sets the channel B resolution
	Bit 6	Sets the channel A resolution
	Bit 5	No connection
	Bit 4	No connection
	Bit 3	No connection
	Bit 2	
	Bit 1	Selects the sample rate clock
	Bit 0	

Figure 2.17 Control Words

There is a memory buffer for display at \$1000-\$13FF. The first 256 bytes of this buffer are displayed, not the entire Kbyte. The order of events is:

- transfer the program control to Applescope Operational Software by executing program at location \$C200, the start up vector for our situation (digital card placed in slot No. 2). This program identifies the slot in which the digital card is located and enables the ROM from \$C800 to \$CFFF (locations of the operational software).
- control the Applescope from keyboard using the operational commands (see Appendix E) while in command mode.

Once the control was transferred to the operational software, the Applescope display appears on the monitor as in Fig. 2.18.

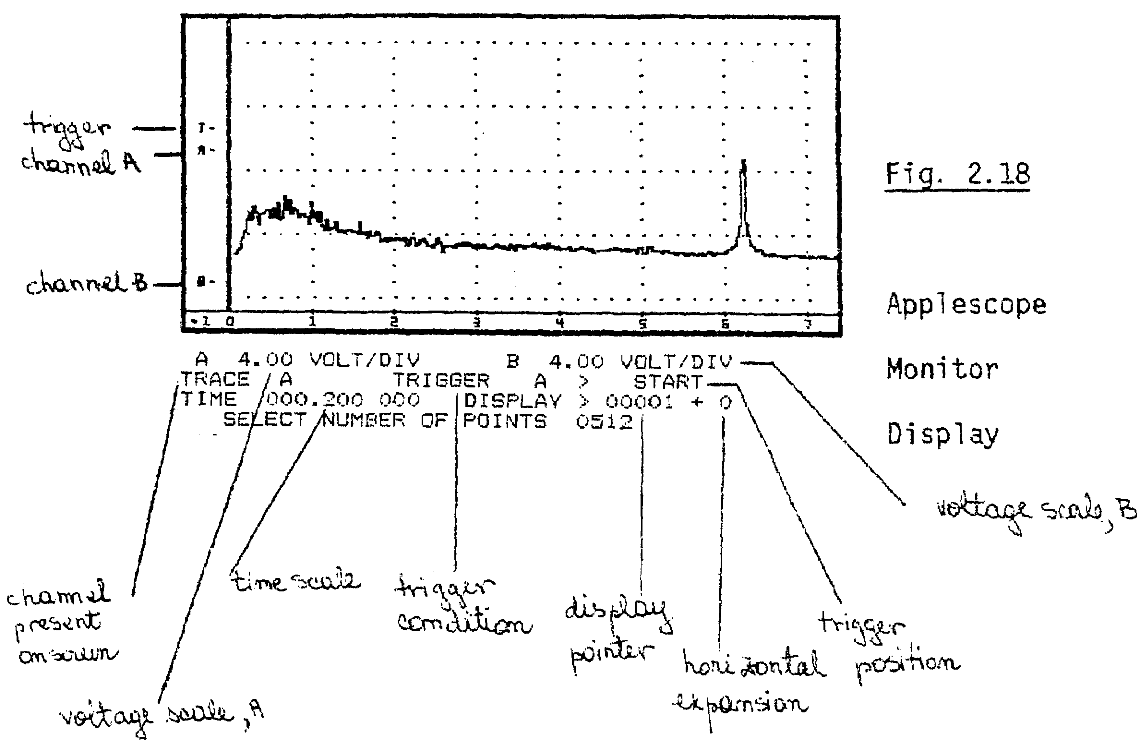


Fig. 2.18

Applescope Monitor Display

The trace paramters for both channels are displayed in the bottom four lines of the display.

The program control can be trasnferred from the Applescope operational software to the master program by pressing "RESET".

c.3 Characteristics of the Applescope

A. Sweep Control. There are 2 ways of acquiring data: single sweep or continuous sweep. The way the scope works in those two modes is described in the flowchart of Fig. 2.19.

In both sweep modes, the Applescope can work with one trace or dual trace. When a data acquisition cycle is finished, when only on trace is used, the result is 1 sweep of 1024 bytes. When dual trace is used, the result is 2 sweeps of 512 bytes each, one for each channel.

It is important to point out that in the dual trace mode the data from channel A and B is alternatively stored in RAM, one sample for A, one for B, until the end of the 512 bytes sweep. This is equivalent with a sampling rate divided by 2 (with respect to the single trace mode sampling rate) and is compensated by the operational software doubling the horizontal scale whenever in dual trace mode. Fig. 2.20 (Appendix D) shows the memory field with data in 2's complement notation, after one data acquisition cycle, for channel A and B.

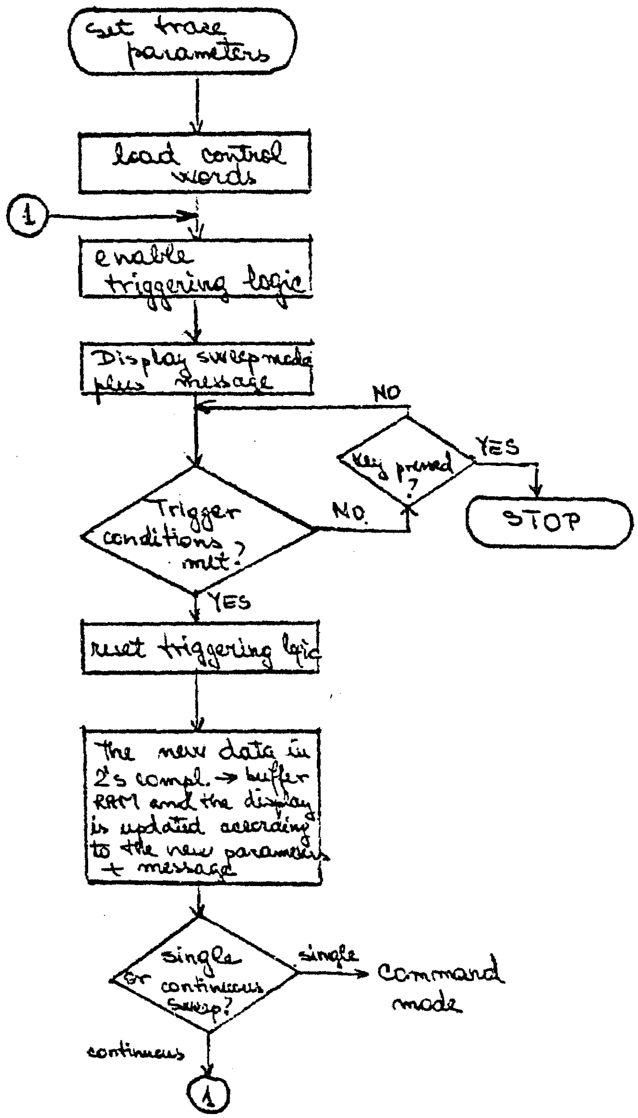


Fig. 2.19 Flowchart for data acquisition cycle.

B. Voltage scale. The voltage scale can be modified in 12 steps, first 6 with a resolution of 7.1 mV/step and the last 6 with 71 mV/step. The possibility to have 2 resolutions is provided by hardware (see Fig. 2.21, Appendix D).

The vertical scale has 28 pixels (dots) per division and 1 step can have 8, 4, 2, or 1 pixels. The vertical expansion (Fig. 2.22) represents the possibility to expand or compress the vertical scale by varying the number of steps/division. Vertical expansion is provided by the operational software. The voltage scale, volts/div., is given by correlating the vertical expansion with the resolution. For example:

$$\left.\begin{array}{ll}\text{vertical expansion} & \text{is "x8"} \\ \text{resolution} & \text{is } 7.1\ \frac{\text{mV}}{\text{step}}\end{array}\right\} \Rightarrow \text{voltage scale} = \frac{7.1\ \text{mV/step}}{\frac{8}{28}\text{div/step}}$$

$$= 24.8\ \frac{\text{mV}}{\text{div.}} = 0.0248\ \frac{\text{V}}{\text{div.}}$$

C. Time Scale. The time scale is adjusted by controlling both the sampling rate and the horizontal expansion (expanding or compressing the horizontal scale). To understand what the horizontal expansion does , it has to be pointed out that the screen offers the image of the first 256 bytes of the current data content of the buffer RAM if the horizontal expansion is zero. However, it can vary between -8 and +8. For negative values we have compression, 256×2^n points appear on the screen (n is the value of the horizontal expansion, the absolute value). For positive values we have expansion,

Vertical Expansion	28 pixels ------- 1 div.		
"x8"	8 pixels ------- 1 step	---	8/28 div./step
"x4"	4 pixels ------- 1 step	---	4/28 div./step
"x2"	2 pixels ------- 1 step	---	2/28 div./step
"x1"	1 pixels ------- 1 step	---	1/28 div./step

Figure 2.22 The vertical expansion

$256/2^n$ points appear on the screen. For example, if n = +2, no. of points = $256/2^2$ = 64 points and if n = -2, no. of points = 256×2^2 = 1024 points.

These examples are illustrated in Figure 2.23.

The horizontal expansion is correlated with the sampling rate to give the horizontal scale. The sampling rate is given by hardware, by dividing the 7MHz clock and by software, making a timing loop. The time scale can be expressed in seconds, milliseconds, and microseconds. For time scales greater than 1 ms, the sampling rate is always controlled by a software timing loop and the horizontal expansion is 1 (software data acquisition). For time scales less than 1 ms, the sampling rate is given by hardware (4 sampling frequencies are combined with the horizontal expansion to obtain the desired scale - hardware data acquisition).

D. Trigger Control. The commands referring to triggering are related to the trigger condition and trigger position. Trigger position and condition are specified in the second of the four lines of text (on the bottom of the screen).

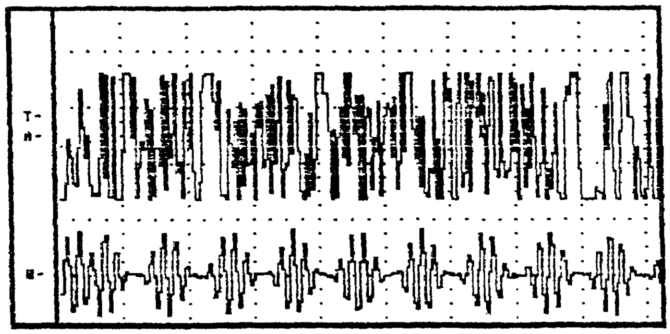
The position of the trigger can be at the START, MIDDLE, or END of the memory buffer for display.

The memory buffer for display has different structures with respect to the trigger position, depending on the sweep rate. The 2 different structures, one for sweep rates faster than 1 msec/div (time < 1 msec/div) and one for sweep rats slower than 1 msec/div (time > 1 msec/div.), are given in Figure 2.24 (Appendix D).

Fig. 2.23

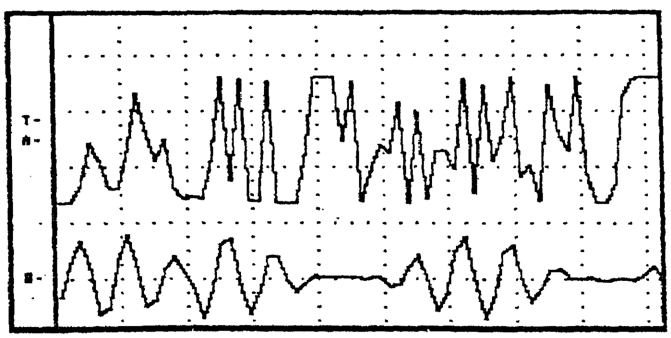
Horizontal Expansion

H.E.=0



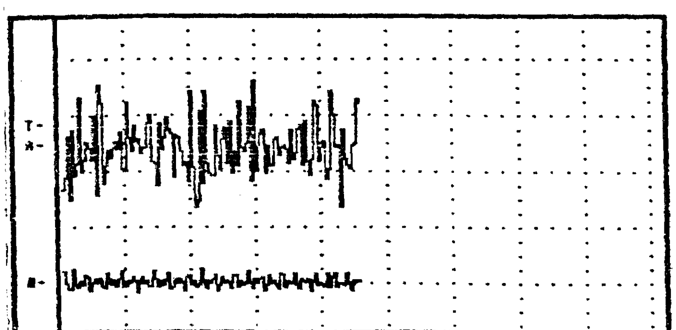
A .800 VOLT/DIV B .400 VOLT/DIV
TRACE A B TRIGGER A > START
TIME 000.200 000 DISPLAY < 00014 + 0
COMMAND :

H.E.=+2



A .800 VOLT/DIV B .400 VOLT/DIV
TRACE A B TRIGGER A > START
TIME 000.200 000 DISPLAY < 00014 + 2
COMMAND :

H.E.=-2



A .800 VOLT/DIV B .400 VOLT/DIV
TRACE A B TRIGGER A > START
TIME 000.200 000 DISPLAY < 00014 - 2
COMMAND :

HE = Horizontal Expansion

The START and the END positions of the trigger are slightly offset from the actual limits of the display buffer in order to allow always some amount of pre- and post-trigger viewing without setting the trigger position in the middle of the signal sweep. There is another possibility of triggering, by using external signal. In this case, the trigger position is at the first point in the buffer.

The display pointer, which appears at the end of line 3 of the text (bottom of screen) is relative to the current trigger position and is given in the number of sample points (not number of display points which can be compressed or expanded). Whenever the trigger position is changed, the relative position of the display pointer changes, but its absolute position in the display pointer is the same.

E. Memory Usage and User Customization

The scope driver programs load from memory locations $0B50 to $1FFF and from $8000 to $95BF. In addition, both pages of the text display, both pages of the high resolution graphics and memory from $6000 to $7FFF are used as working data and display buffers. Memory from $280 to $37F is used to save all the zero page memory whenever the SCOPE DRIVER software is being used.

The SCOPE DRIVER options are added to the foundation program by loading different modules into SCOPE DRIVER RAM area. If the module containing the selected option is not loaded, the software will load the appropriate module from disk when it is first selected. The option will now be available for continual use until a different option module has been loaded. Several SCOPE DRIVER options may be contained in each module depending on program complexity.

The machine language SCOPE DRIVER programs were designed to allow for easy access and customization by BASIC programmers. The approach that this computer has is to designate a command memory buffer where keystrokes corresponding to SCOPE DRIVER commands can be stored. Whenever the command buffer is enabled, the machine language SCOPE DRIVER programs respond to the command buffer keystrokes exactly as if they were being input for the keyboard. Once a signal trace has been characterized, a simple Applesoft BASIC program may be written to totally automate the data acquisition cycle.

Up to 47 commands steps may be stored in memory for execution each time the SCOPE DRIVER software is called. By using these commands to initiate keystrokes from the keyboard, most any kind of signal trace can be acquired for use by an Applesoft program. Successive commands from command buffer will be executed until either the 47th command is reached or a disk supervisor command is encountered.

The raw data from each signal trace is available for user manipulation and may be read directly from the display buffers. The data is in 2's complement notation and must be converted before use in BASIC programs. Appendix E presents another characteristic of the Applescope, the disk supervisor and gives information about the disk drives and Disk Operating System (DOS). At the end of this appendix there is a list with the DOS commands.

2.1.3 The Printer and the Microbuffer

The printer used in this project was EPSON, MX GRAFTRAX PLUS. The information is sent from the computer to the printer via the microbuffer, the card in slot 1. The printer is used to obtain the 8 spectrums average for both channels (or whatever average), the histogram for a frequency range 0-71.1 Hz and the pie chart for both channels.

The microbuffer is a very important part of the printing process. The MICROBUFFER II, used in this project, is an intelligent Centronics-compatible parallel printer interface for the Apple II and Apple II Plus computers. The Microbuffer II has up to 32 Kbytes (16 Kbytes standard) of on-board memory for data buffering and provides useful text control functions. For user with certain "graphics" type printers, as the one used, this microbuffer includes an extensive set of advanced high-resolution graphics dump routines.

Data buffering increases data processing efficiency by freeing the Apple and the operator from the wait normally experienced while printing. The Microbuffer II will allow the Apple to print and process simultaneously. It will accept data as fast as the Apple can send it (up to the buffer size) and return control of the computer to the user while it handles the printing. Additional data may be sent to the Microbuffer II without waiting for previous jobs to be completed.

The Microbuffer II contains an intelligent controller, control software in ROM, and high-speed RAM for data buffering. The

RAM allows the Microbuffer II to accept data from the Apple at up to 4,000 characters per second to the limit of memory available.

If the amount of printed data is less than the buffer size, the Apple will complete its dump in a matter of seconds. When the amount of printed data exceeds the buffer size, the Microbuffer II will respond to the Apple as a normal printer interface taking one line at a time. This will result in approximate time savings of nine minutes for a 80 character per second printer assuming a thru-put of 60 characters per second, when using a 32 Kbytes buffer size. These time savings will vary with line length. For output that consists of very short lines, like assembler listings, the time savings will be greater because printers take longer to print a line feed and carriage return than to print normal characters.

The software in ROM controls all of the functions of the Microbuffer II and is different for each graphic printer. Microbuffer II is typically shipped wiht EPSON firmware.

More information about how the microbuffer can be used can be found in its "users manual".

2.2 System software

2.2.1 Introduction

The frequency analysis of the EEG singal offers important information about the behavior of the monkeys as effects of drug administration. But in order to obtain useful information about the EEG power distribution in certain frequency bands, an average of EEG spectrums has to be done. The drug effects can not be obtained looking at one single spectrum at a time because of many reasons. One of them relates to the fact that EEG patterns are influenced by random events as the changing in the animal position. Doing an average of spectrums, these random effects eventually cancel one another.

Another reason consists in the fact that the drug effect appears during a certain interval of time. Analyzing one single spectrum does not give the cumulative effect of the drug but just a too small, too big, or no effect. The right effect represents the average of these effects during a certain interval of time which is defined by the research experience.

The software is able to control the data acquisition, the spectral components calculation and the statistical analysis of the EEG power.

The data is acquired as sweeps of 3.6 second durations. Then, each sweep spectrum is done using the machine language modules available from R.C. Electronics Incorporated and, of course, the facility of having a command buffer which can store up to 47 commands (to imitate the keyboard). It is important to remember that the data has to be acquired continuously for at least 10 minutes in order to obtain some visible effects of the drug. For this reason, the data has to be first saved, and then, after the experiment is over, retrieved and processed. Storage capability is needed and is accomplished by the data disk in drive 2. Another important thing about timing during the data acquisition process is the duration of each sweep saving process. Saving on a disk using BASIC statements takes one to two minutes for each sweep which is a lot if you think that in those two minutes we loose about 40 sweeps, and may be the most important ones. In this way the acquisition process is not effective. We must use an intermediate storage capability to accumulate more sweeps and then to save all of them on a disk. The intermediate memory used is the Super RAM range $D000-$EFFF and the saving and retrieving process are executed by machine language routines.

In my algorithm, I stored the data in the first part of the data disk and right after that I stored the spectrums. The averages of the spectrums were stored as binary files in the program disk (drive 1) and they can be seen one by one. I tried to acquire as much data as possible in one experiment. The amount of data is limited, however, by the capacity of the data disk (drive 2). If the data is erased after the spectrum is done (replacing data with spectrums) the possibility to store is greater. I did not choose this variant because I considered that the data might be necessary to be visualized, in the case the analysis effects are unusual. The question is, how many sweeps of data can we store on the disk knowing that space is necessary for the same amount of spectrums? The computer is set to work with both channels, so in this case one sweep, for one channel, has 512 points. The disk has a capacity of 35 tracks or 560 sectors. Each sector has 256 bytes and 4 blocks of 256 bytes each represent 1 Kbyte. The disk capacity is then 140 Kbytes.

The intermediate memory has 8 Kbytes and contains 16 sweeps (or spectrums when the spectrums are done), 8 for each channel. Because of the disk's capacity, I can save 8 times the intermediate memory full of data or spectrums, and I will get 64 Kbytes of data and 64 Kbytes of spectrums which means 64 sweeps for each channel and respectively 64 spectrums for each channel. The rest of the disk (140K - 128K = 12K) is unused for now. The map of the disk is given in Fig. 2.25. The figure shows also the map of the intermediate memory.

When the data is to be processed, a block of 8K is retrieved at a time. Whatever the process is, saving on disk or retrieving from disk, it works for 8 Kbytes of memory.

The 8 Kbytes memory block contains information about both channels. Even if we want to process the data for one channel, the entire block is retrieved from the disk in the intermediate memory (Super RAM) and then, other machine language routines do the averaging, move results in to a memory buffer for display, on the screen, or at the printer. The same results are moved in another part of the memory for the statistical analysis.

Because a block of 8K from the disk contains 8 spectrums for channel A and 8 spectrums for channel B, averages of 8 spectrums are done for both channels, 8 averages for each channel. With respect to the average, there are 2 variants of the program:

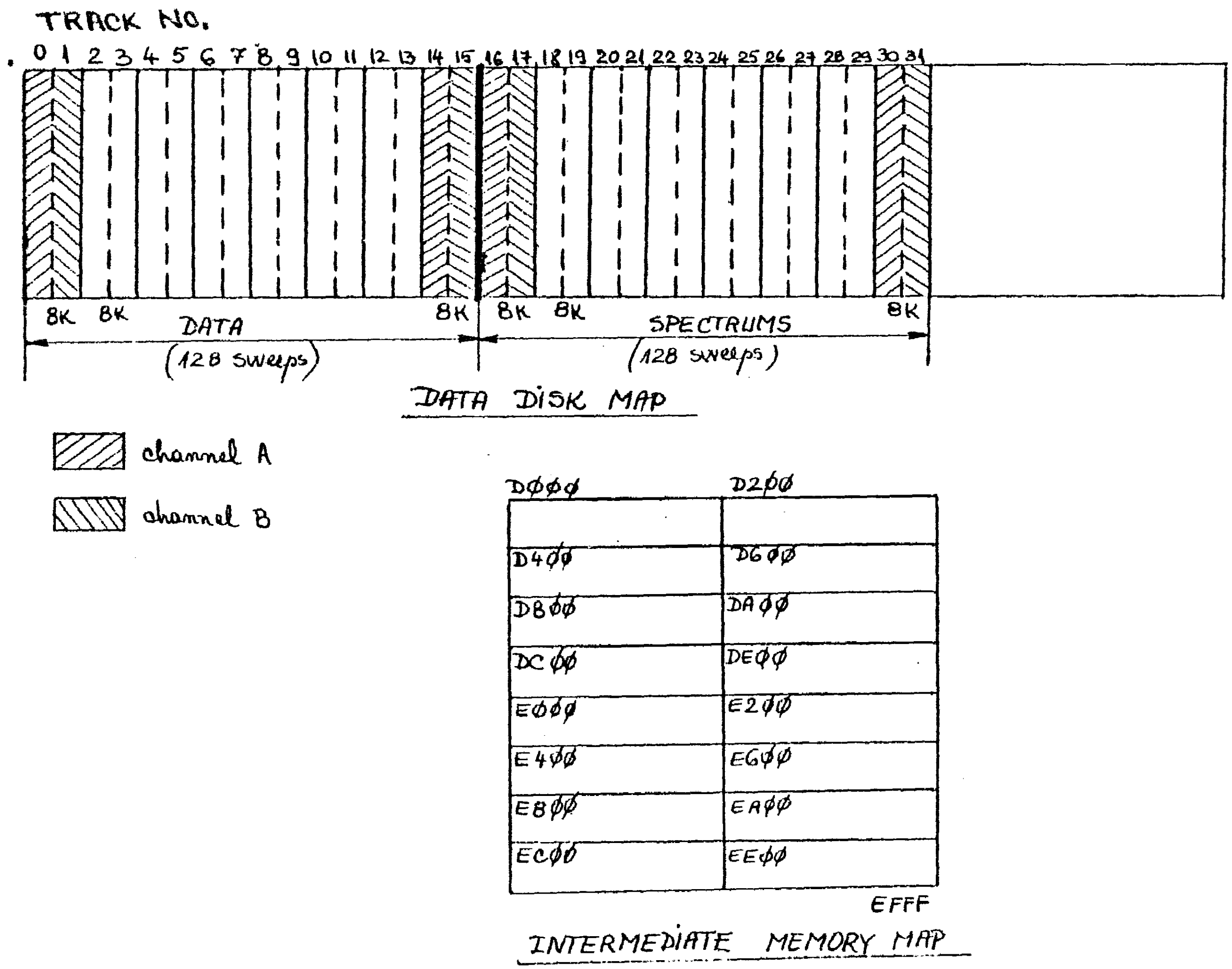


Figure 2.25 Data Disk Map and Intermediate Memory Map

Variant 1, which does eight 8 spectrum averages for each channel and Variant 2, which does a total average for eight 8 spectrum averages for each channel.

Both variants give the user the choice to process channel A, or B, or both, to draw a histogram or pie chart.

There is another variant of the program which processes the 128 sweeps of data in a fixed way in order for the user to not have to supervise the computer during the process.

2.2.2 General Flow Chart Description

Whatever variant is used, the main algorithms and the steps are the same. The general flow chart of the program is given in Fig. 2.26.

Loading the necessary binary files and doing the calibration of the Apple oscilloscope are combined with the data acquisition program on one disk. In this way, every time a new experiment is beginning, the calibration can be done or just checked. There is another disk with the processing program which contains the spectrum components amplitude calculation, the average execution and the statistical analysis (this is practically variant 3). For variant 1 and 2, data acquisiton is combined with spectrum analysis in one disk and there is a separate disk for average and statistical analysis. All variants use a third disk, the initialization disk. This is introduced in the beginning of the experiment in drive 1 in order to initialize new disks (data disks) in drive 2. Variant 3 is the most used because the data acquisition is separate from spectrum analysis and statistical computations. Every time variant 3 runs on the data acquisition disk, we have a new experiment. Variant 3 has

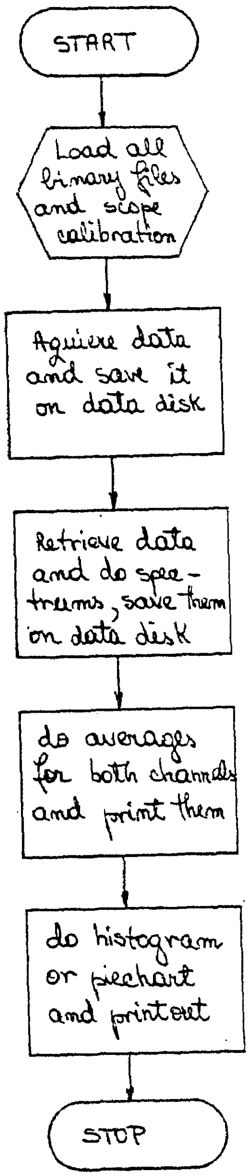


Fig. 2.26 General Flowchart

the advantage of providing whatever number of experiments are needed, one after the other (so the drug effects can be obtained), each one now having its own data disk. Later, the analysis can be done.

2.2.3 Variant 1

The programs included in variant 1 have the possibility to take data, do the spectrums, do 8 averages of 8 spectrums each, save them in the program disk (drive 1), and then use them to draw a histogram or pie chart. The user has the choice to stop or not after the spectrums are done and saved on the data disk (drive 2).

The variant 1 program disk contains 20 BASIC programs, 14 machine language routines, 5 binary files available from R.C. Electronics Incorporated, relative to the Applescope (SCOPE DRIVER and spectrum analysis) and 2 data files relative to the oscilloscope parameters. These 2 files are updated all the time with the new calibration parameters. The variant 1 flow chart is given in Figure 2.27.

In order to understand how variant 1 works, I have to discuss each program or at least the most important ones. From the beginning I would like to point out that all the machine language routines with the title ending in "TAB" are related to a saving or retrieving data process (to disk from the intermediate memory, or to the intermediate memory from the disk). "TAB" means a table which contains all the information about source and destination for data transfer. There is only one machine language routine which is the exception to this rule, the average program (AVELOTAB for channel A and AVELOTABB for channel B). In this program "TAB" means another

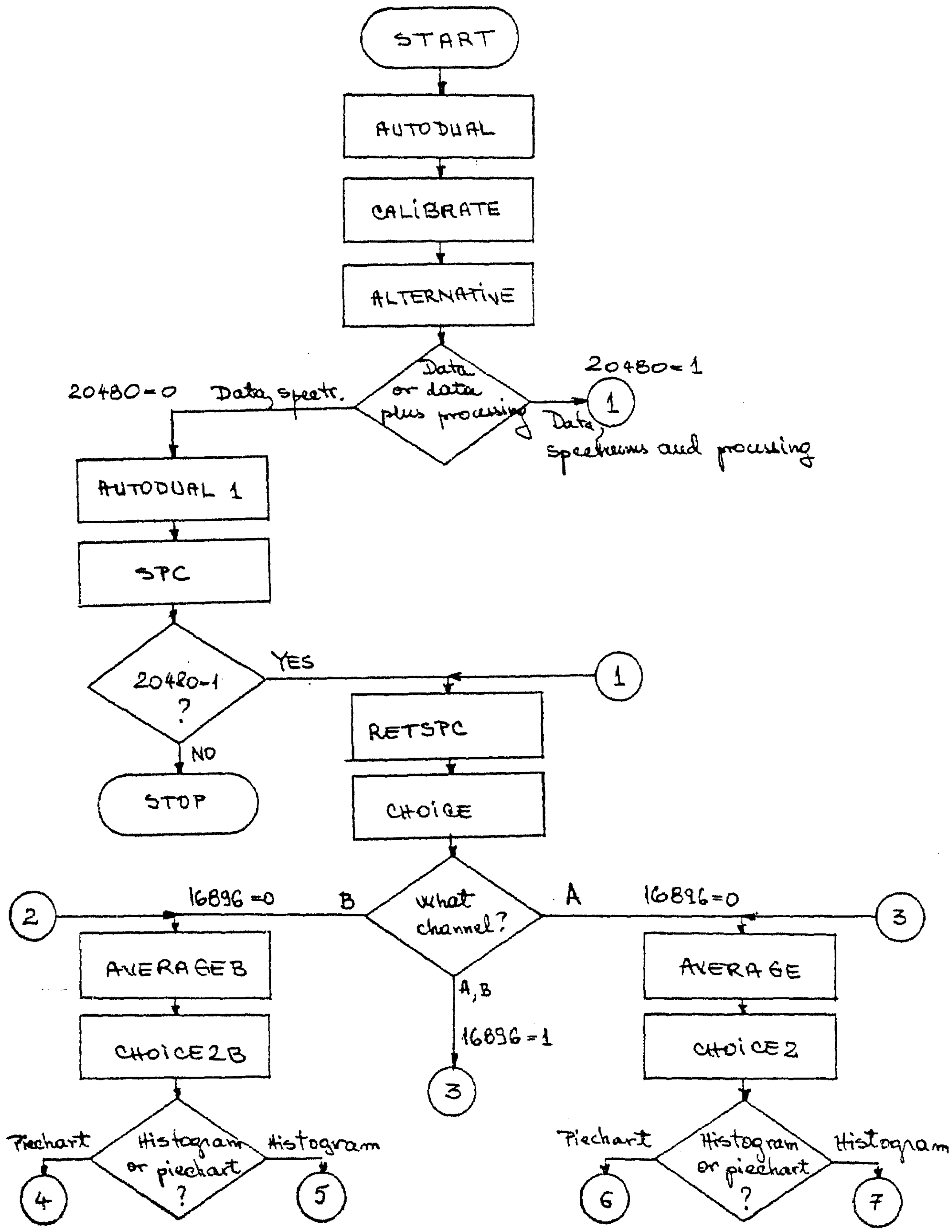


Figure 2.27 Variant 1 Flowchart

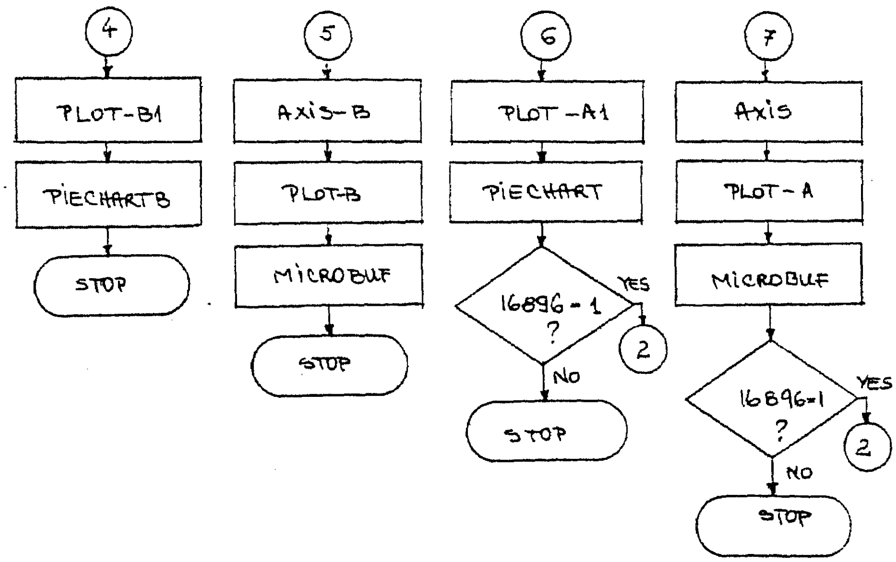


Figure 2.27 continued

Variant 1 Flowchart

table, related to some pointers for indexed-indirect addressing. Another rule that is useful for the programmer to know is that there are similar programs for channel A and B, the programs for channel B have the same name as the programs for channel A but they end with "B".

AUTODUAL - loads the binary files necessary to acquire and process data (scope driver, BPR, spectrum files M4A, M4B, M4C, and the 2 files with the parameters for the digital oscilloscope, CT and CS). The program gives a message to the user with respect to the scope calibration. The program ends by transferring the control to the calibration program.

CALIBRATE - calls the SCOPE DRIVER program in order to give the possibility to the user to check or change the scope parameters and to make sure that the signals are connected correctly to the analog inputs of the computer. By pressing "CTRL SHIFT P" (disk supervisor command) on the keyboard, the program automatically saves the scope parameters (CT and CS files) on disk and then transfers the control to the data acquisition program not before giving the user the choice of acquiring data only or acquiring and processing the data immediately in ALTERNATIVE program.

ALTERNATIVE - loads LOWALLTAB at $300, opens the Super RAM to be written and loads HIALL.OBJ0 at $F000 in Super RAM. Then asks the user if he wants to continue with processing after the data acquisition. Depending on the user's answer, the program set or does not set a variable in location 20480 and transfers the control to AUTODUAL1.

AUTODUAL 1 - gives the user the choice to enter the number of sweeps that he wants to process. In this program "sweep" means a block of 1024 bytes of data, the first half for channel A and the second half for channel B.

As I mentioned in the first paragraph, the maximum number of data sweeps (512 bytes each) for one channel is 64, for both channels 128. Speaking in terms of 1024 byte blocks, the maximum number of blocks is 64. However, the user can enter any number. For larger numbers an error message is generated, for smaller numbers the program works for 8K of data at a time. For a multiple of 8 the program does the work in a loop executed as many times as necessary, for all other cases the number entered is reduced to the closest multiple of 8 and the execution is done with the same algorithm.

The data acquisition is done by the program, not from the keyboard, by placing all the successive commands in the command buffer (47 commands maximum). Once 1K of data is acquired (1 sweep for channel A and 1 sweep for channel B), the memory buffer for display has to be saved because another 1K of data will replace it. Each 1K of data is moved from the memory buffer for display, with LOWALL machine language, in Super RAM II ($D000). This machine language routine is placed at decimal 768 ($300) and is executed by CALL 768. It is used to save 1K at a time as long as the 8K intermediate memory buffer, $D000-$EFFF, is not filled. When it is filled, another machine language, HIALL.OBJ0 is used to save the 8K of Super RAM on disk. HIALL.OBJ0 is placed in Super RAM and not in the low memory, $300-$3FF, reserved for machine language routines because when SCOPE DRIVER program is loaded only $300-$360 is free, so there is no space. HIALL.OBJ0 executes a saving process so it needs a table (see DOS manual) of 21 bytes containing information about the source and the destination of the process (17 bytes represent input/output block, IOB and 4 bytes represent the Device Characteristics Table, DCT). This table contains locations which must be increased during the execution of the saving (or retrieving) process. If the table is placed in Super RAM, this means writing memory. At the same time, execution means reading memory. But Super RAM can not be read from and written on at the same time (see Chapter 2, section 2.1.2.b), there are 2 different soft switches for these 2 operations and they can not be on at the same time. Thus, the table has to be placed in low memory. That is why the LOWALL program, combined with the table, TAB, and with location $346 used as counter, are called LOWALLTAB binary file. HIALL.OBJ0 is a subroutine of LOWALL. Having 2 binary files instead of one is because low memory is reserved for machine language and there is not enough space for one big routine.

Figure 2.28a shows how the machine langauge routines work within AUTODUAL 1 program. Figure 2.28b gives the general steps for saving or retrieving machine langauge routines and figure 2.28c represents the detailed flow chart for AUTODUAL 1 program.

<u>SPC</u> - retrieves the data from disk and does the spectrum of each 512 point sweeps of data. The user has the choice to do the spectrum of a smaller number of sweeps than the one he acquired on disk. However, the program processes 16 spectrums at a time, 8 for channel A and 8 for channel B, alternatively. the number is 16 because a 8K block saved on the disk with HIALL.OBJ0 contains 16 sweeps, 8 for

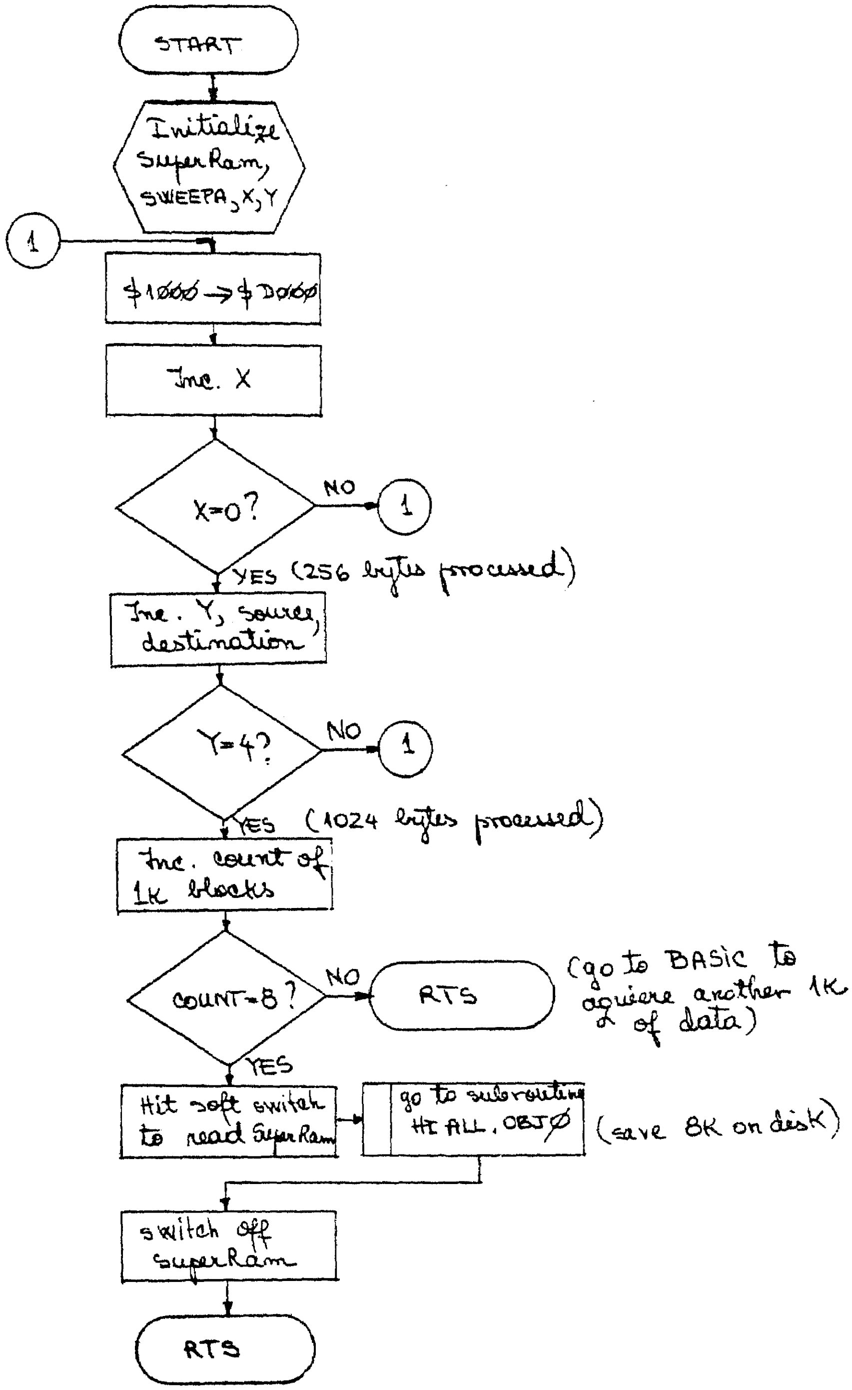
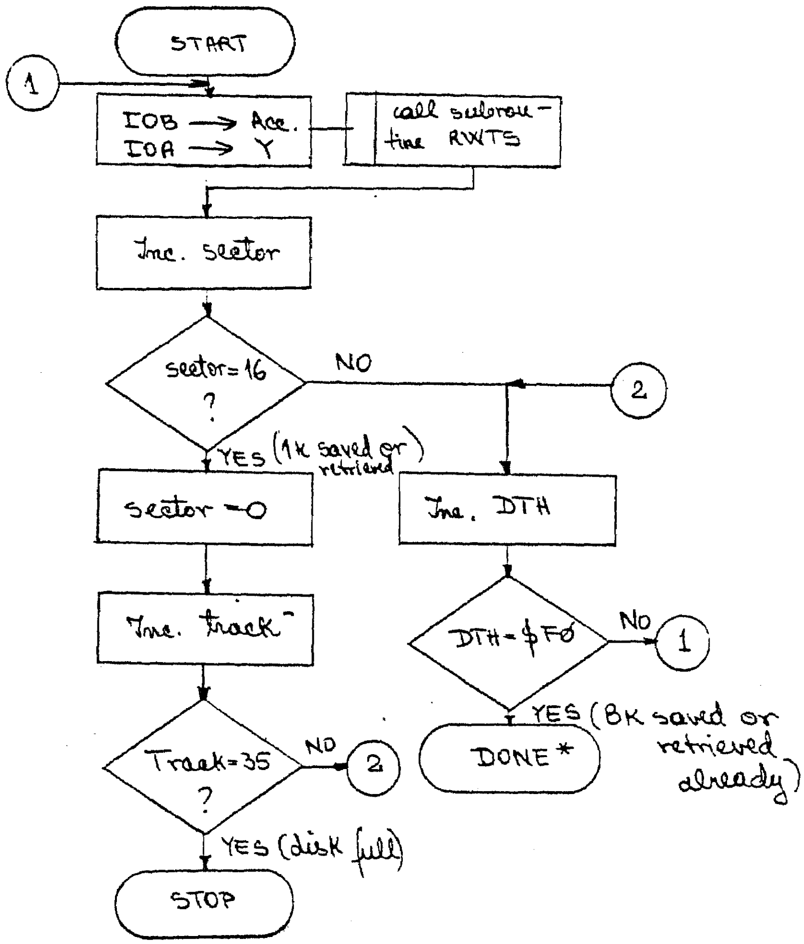


Figure 2.28a Flowchart for Machine Language Routines Used in AUTODUAL1, BASIC Program



IOB hi address = IOB
IOB low address = IOA
DTH = data buffer hi address
RWTS = read-write track and sector subroutine

* This is a general flowchart. However, for HIALL.OBJ0 "DONE" means initialize all the addresses used in the main routine, LOWALL, when control is returned to it to be executed again. Generally, "DONE" does not mean STOP, but reinitializing or activating soft switches before returning to BASIC or machine language main routine.

Figure 2.28b General Flowchart for Saving or Retrieving Machine Language Routine

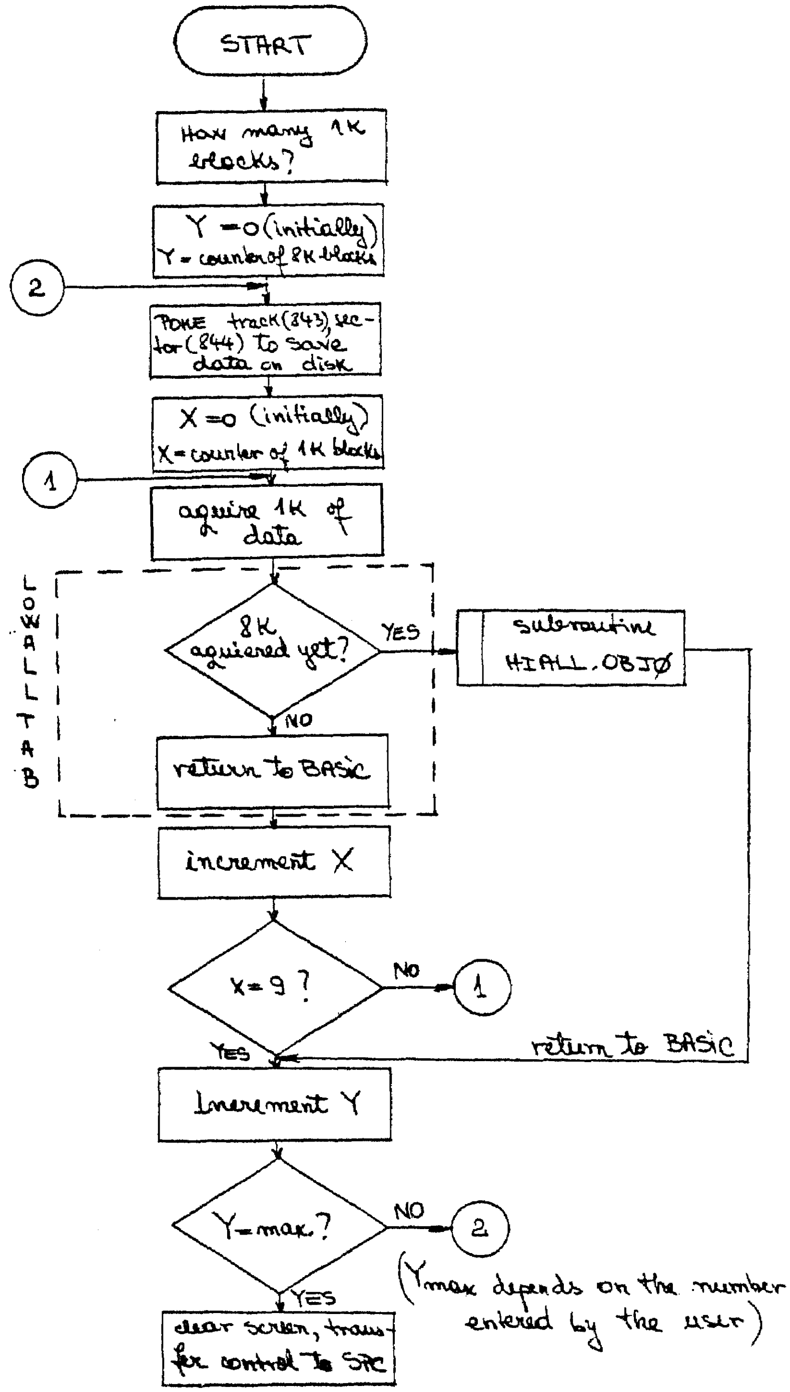


Figure 2.28c Flowchart for AUTODUAL1 Program

each channel, and is retrieved from the disk with the same kind of program as it was saved. The only difference between the machine language routines which save 8K on the disk and retrieve 8K from the disk is in the input/output block (IOB) part where the byte which says "writing" in replaced by "reading". Figure 2.29 shows how the data and spectrums are manipulated by the SPC program. There are 5 machine language routines used by the SPC program.

1. RETTAB is used to retrieve from disk 8K of data, in the Super RAM buffer ($D000-$EFFF). The routine is formed by RET2 program plus IOB/DCT table. This routine works as in Figure 2.28.b and is placed at $0300.

2. OPEN.OBJ0 is placed at $032C and is used to open the Super RAM to be read because there is a routine placed there, SWEEP.OBJ0, which needs to be executed. SWEEP.OBJ0 is a subroutine of the first part of OPEN.OBJ0, located between $032C-$0346. The second part of OPEN.OBJ0 ($0347) is a subroutine of SWEEP.OBJ0. Even if there is enough space in Super RAM to place this part and to not work with a subroutine placed in low memory, this can not be done because of the fact that Super RAM can not be read and written at the same time. Super RAM has to be read in order to execute SWEEP.OBJ0 and has to be written in order to increment pointers because we are dealing with a large amount of data and a fixed pointer can only scan up to 256 bytes. Thus, the pointers have to be modified all the time so they have to be placed in low memory, Super RAM being read at this moment.

3. SWEEP.OBJ0 moves sweeps, one by one, from Super RAM to memory buffer for display.

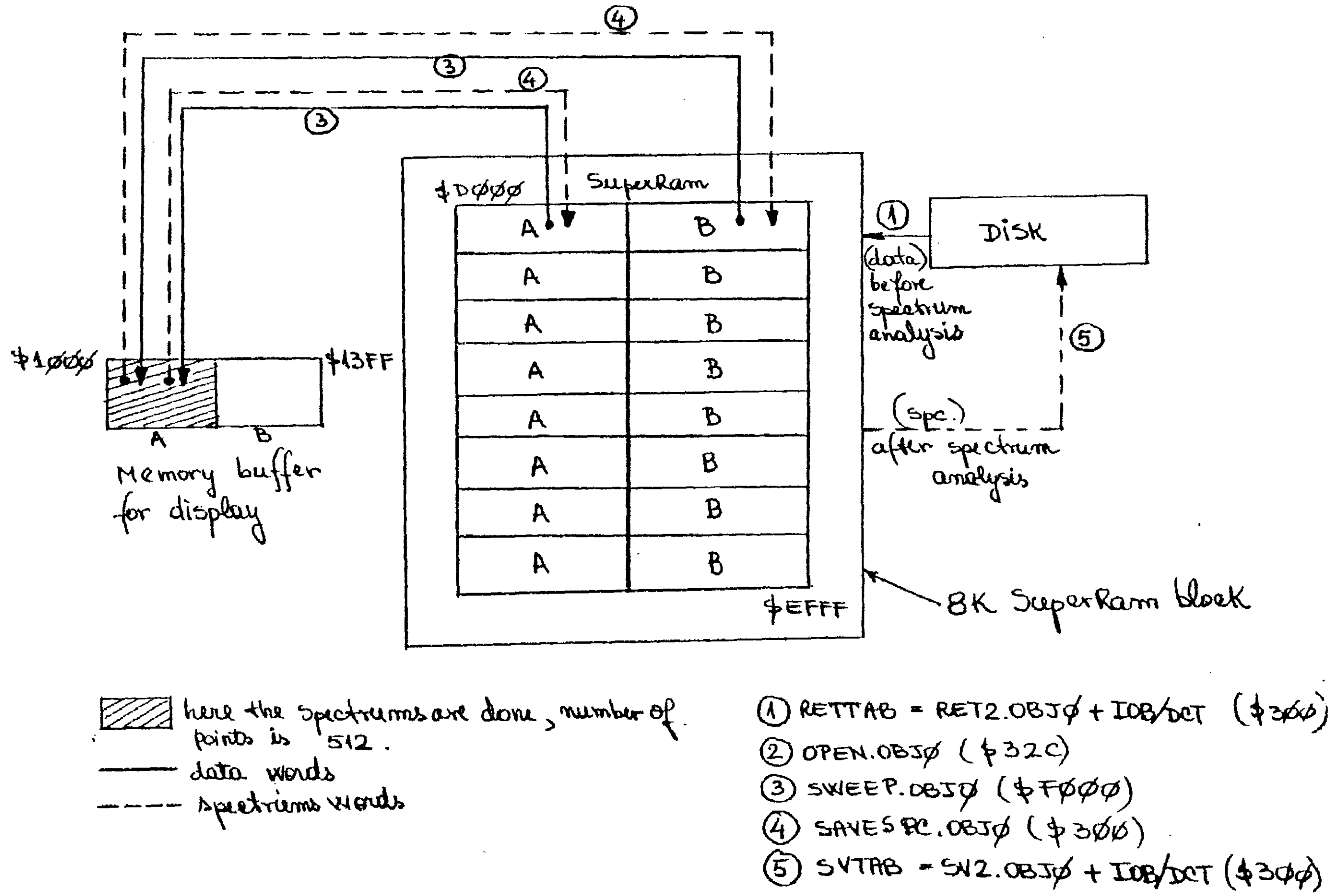


Figure 2.29 Manipulation of Information in SPC Program

4. SAVESPC.OBJ0 moves spectrums after they are done by SPC program, one by one, from the memory buffer for display to Super RAM in the same place where the corresponding data was located.

5. SVTAB saves spectrums from Super RAM on disk in the way shown in figure 2.28b.

Figure 2.30 gives the detailed flow chart of the SPC program and shows when and where each machine language routine is loaded. X, Y, S are variables. X represents the order of the spectrum in a 16 spectrum block, Y is the order of a 16 sweep or spectrum block in the 8 blocks, maximum amount of sweeps or spectrums that this system can work with, and S represents a switch with 2 positions: "1", in case the user wants to take data, do the spectrums, and process the information (statistically) and "0" in case the user wants to stop after the spectrums are done. This switch is set in ALTERNATIVE program.

RETSPC retrieves one 8K block from maximum eight 8K blocks of spectrum that can be stored in the data disk. the user chooses which block has to be retrieved. The 8K block chosen is retrieved in Super RAM with RETTAB machine language routine. Depending on the user's choice, track and sector locations are changed from BASIC, otherwise the retrieving routine works as in Figure 2.28b. After this process, the control is transferred to CHOICE program.

CHOICE gives the user the possibility to process channel A, or channel B, or both of them. If the choice is both channel A and B, a variable is set to 1, location 16896. At the end of channel A processing the variable is tested. If 1, processing continues with

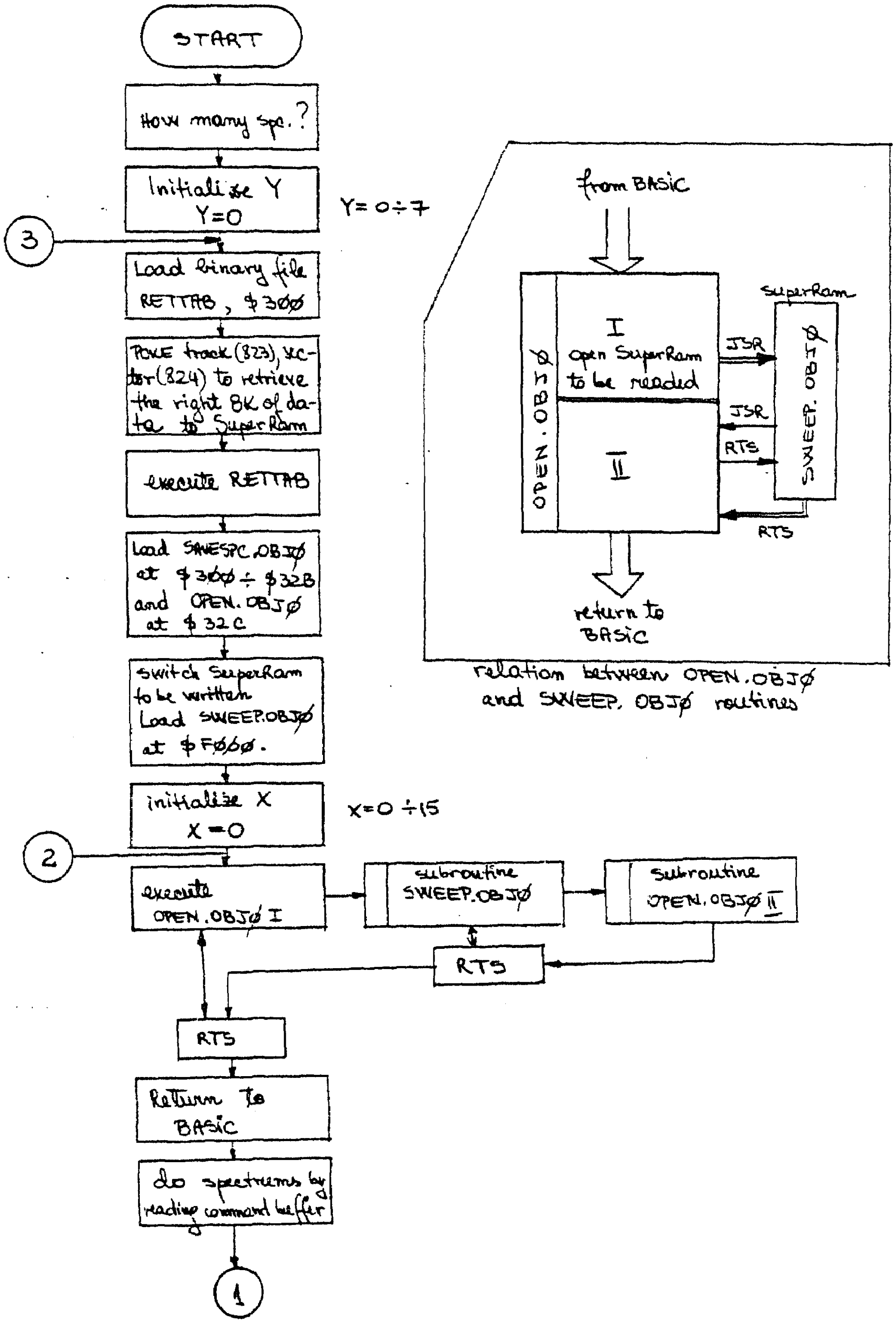


Figure 2.30 Flowchart for SPC Program

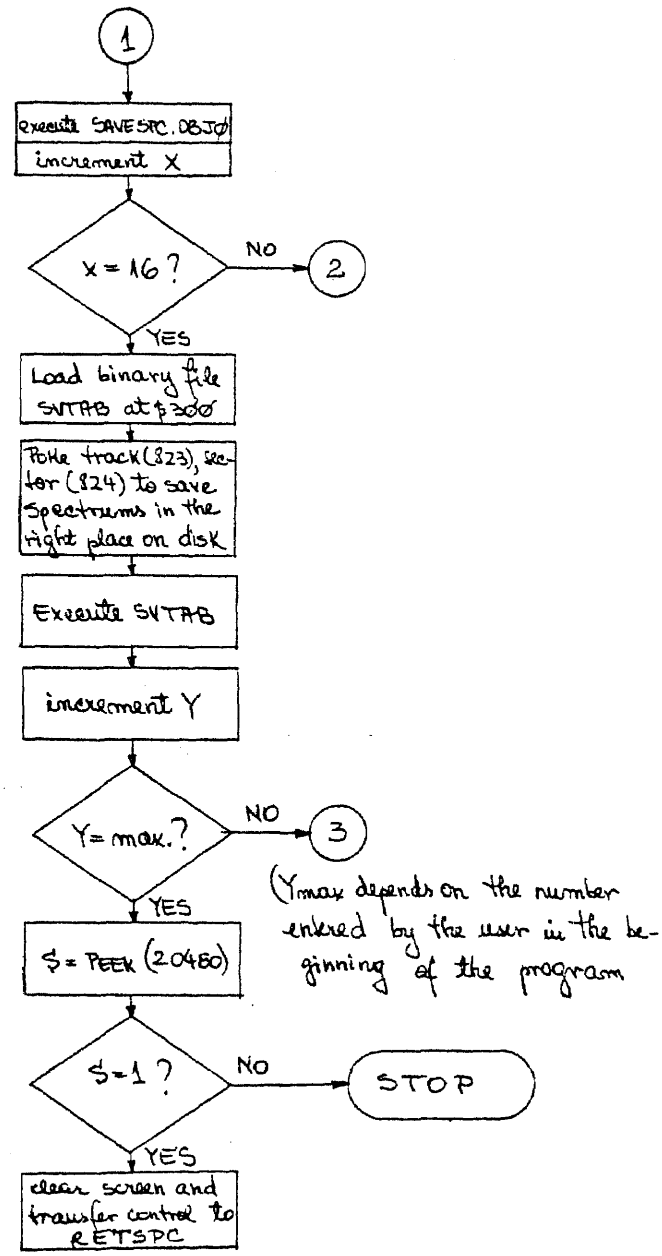


Figure 2.30 (continued) Flowchart for SPC Program

channel B and the variable is reset to zero so that B is not processed more than once. If the choice is channel A or B, the variable is zero, so that at the end of the processing, when the location 16896 is checked, the process stops. In conclusion, 16896 = 0 which means "STOP", and 16896 = 1 means to continue with channel B processing.

When the user chooses channel A or both channels A and B, AVERAGE is the next program, when the user chooses channel B, the next program is AVERAGEB. These 2 programs work in the same way, the difference between them consists of the source of data, destination of data, mainly in what you process and where you store the result.

AVERAGE does the average of 8 spectrums for channel A, from the 8K block retrieved with RETSPC and saves the average in the program disk (drive 1) as a binary file called "AVE.8". At the same time, the program offers the average on the screen and on the printer. The user can enter the desired title for the average. If the time counts and the statistical analysis is more important, the user can skip seeing the average on the screen. He will have it on the printer anyway, meanwhile the statistical analysis can begin (the printer is controlled by the microbuffer, so at the same time the computer microprocessor can do something else).

The detailed flow chart for AVERAGE program is given in Fig. 2.31a.

The machine language routines used by AVERAGE programs are:

1. CLEAR.OBJ0 is a small routine loaded at \$300. It clears the memory areas: \$4000 - \$41FF and \$4400 - \$45FF. This area has to be cleared because the result of the sum of the 8 spectrums (2 bytes

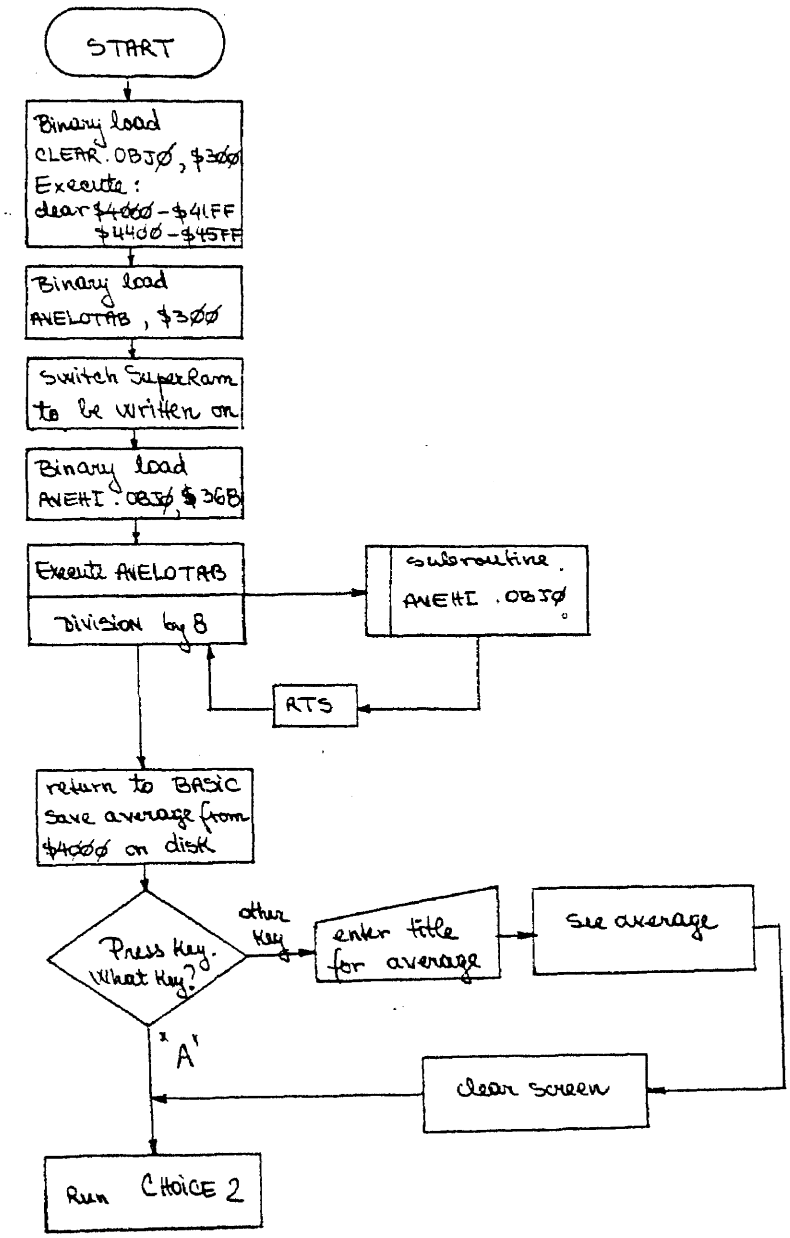


Figure 2.31a Flowchart for AVERAGE program

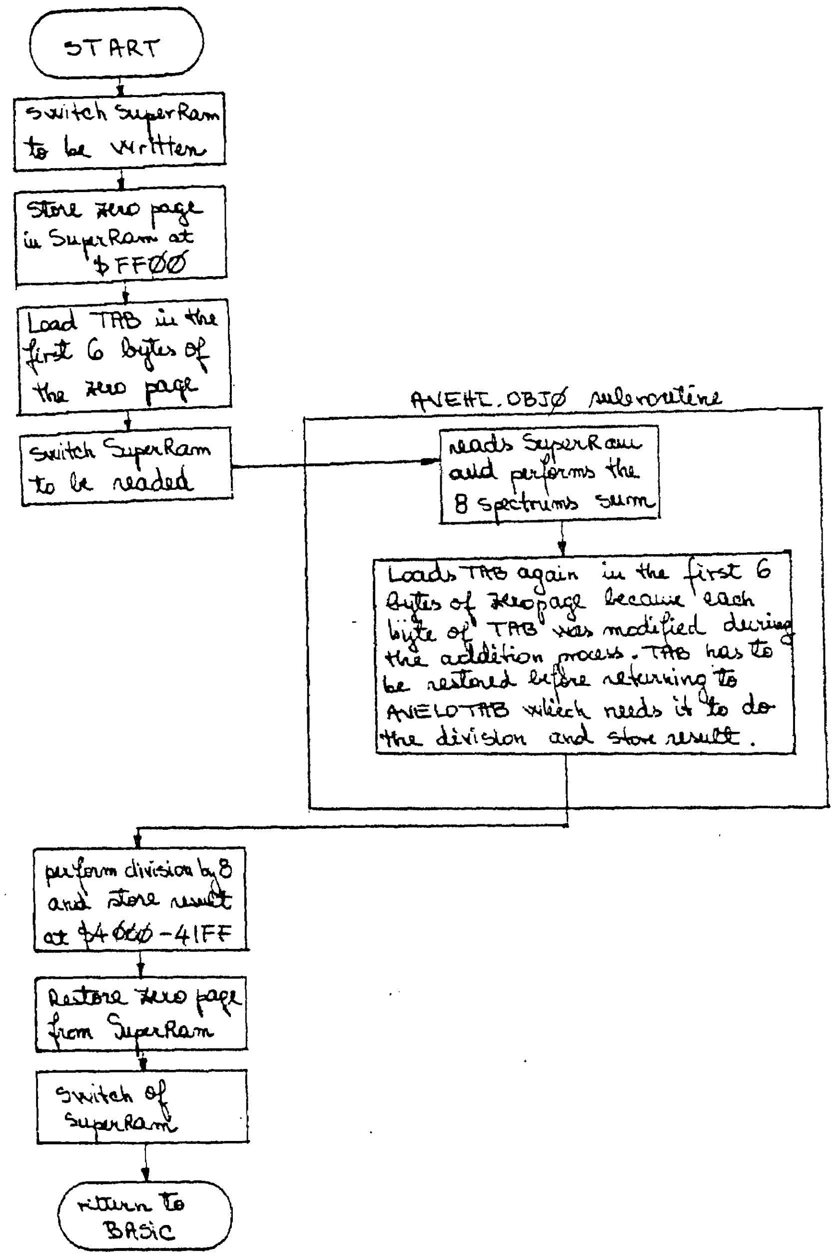


Fig. 2.31b Flowchart for AVELOTAB and its Subroutine, AVEHI.OBJO

result) is going to be stored here by adding it with the content of these memory locations. The area $4000-$41FF is dedicated to the low bytes of the 8 spectrums sum and the area $4400-$45FF, to the high bytes of the sum. The memory map for the results of the 8 spectrums sum for channel A and B is given in Fig. 2.32a.

2. AVELOTAB is placed at $300 and performs a division by 8 of the results of the 8 spectrums sum. The routine uses indexed indirect addressing so needs some space in zero page. This space is obtained by saving zero page to $FF00, in the beginning of the program and restore it back at the end. The control is transferred to AVEHI.OBJ0 subroutine which does the sum of the 8 spectrums. When returning from subroutine, AVELOTAB performs the division by 8 and the 8 spectrums average is obtained. It contains 512 bytes and it is stored at $4000. AVELOTAB is the only routine which "TAB" does not refer to a IOB/DCT table for a saving-retrieving process (same situation for AVELOTABB). "TAB" relates to the pointers used in the indexed-indirect addressing (see Fig. 2.32b).

3. AVEHI.OBJ0 is the routine which does the addition of the 8 spectrums. It is a subroutine of AVELOTAB and is placed at $368, after the AVELO and the TAB. This routine reads the Super RAM data (the spectrums retrieved from disk) and performs the addition of 8 spectrums. Every time 2 bytes are added a 2 bytes result is obtained. The sum of the 8 spectrums will then have 1024 bytes, 512 low bytes placed by AVEHI at $4000-$41FF and 512 high bytes placed at $4400-$45FF.

Fig. 2.31.b shows how these last two routines, AVELOTAB and AVEHI.OBJ0 work together in order to obtain the average.

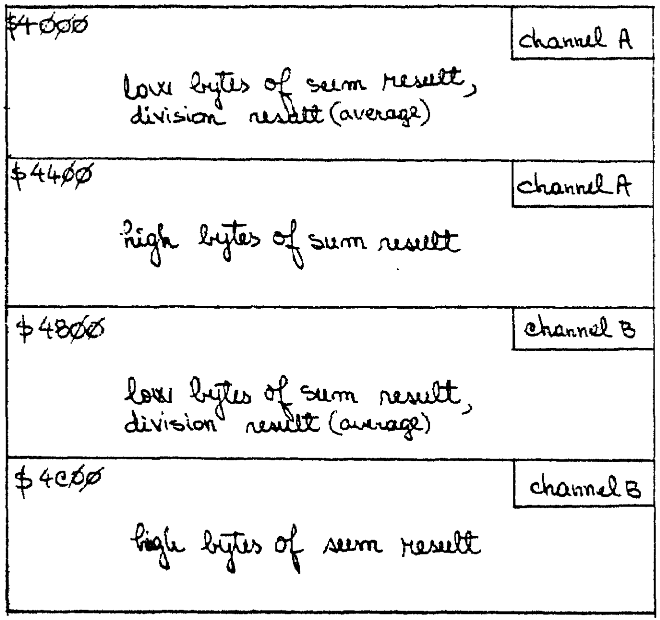


Figure 2.32a Memory Map for 8 Spectrums

Sum and Average for Channels A and B

	ch. A	ch. B
address($)	TAB	TABB
0001	00	00
0002	40	48
0003	00	00
0004	D0	D2
0005	00	00
0006	44	4C

Figure 2.32b Zero Page Pointers Used for Indexed Indirect

Addressing in AVELOTAB and AVELOTABB

AVERAGEB does the average of 8 spectrums for channel B and saves the average in the program disk as a binary file called "AVEB.8". This program does the same thing as AVERAGE program and uses the same type of machine language routines:

1. CLEARB.OBJ0 clears the memory areas $4800-49FF and $4C00-4DFF in order for the 8 spectrum sum to be stored here.

2. AVELOTABB performs the division by 8 and stores the results at $4800-$49FF. The sum of 8 spectrums is provided to AVELOTABB to do the division by 8, by the AVEHIB.OBJ0.

3. AVEHIB.OBJ0 is a subroutine of AVELOTABB and performs the sum of 8 spectrums. This sum has 1024 bytes. The upper 512 bytes are placed at $4C00-$4DFF and the lower bytes are placed at $4800-$49FF.

In conclusion, with respect to the average programs for channel A and channel B, they are practically the same, but they take data from different places and store results in different places. The corresponding machine language routines have to have different names because they deal with different locations.

CHOICE2 is a small BASIC program which gives the user the possibility of choosing between having a histogram done or a pie chart.

CHOICE2B does the same thing as the CHOICE2 program, but for channel B. Two individual programs are needed to make the choice for channel A and channel B because different routines have to be accessed for processing for each channel.

AXIS is the program that draws the axis for the histogram, writes what each axis represents, y axis "% of total power" and x

"y" axis	"x" axis
2	0
4	1.9
6	3.8
8	7.7
10	11.6
12	15.5
14	19.4
16	23.3
18	31.3
20	39.4
22	47.5
24	55.5
26	63.6
28	71.1
30	
32	
34	
36	
38	
40	

a. Numbers Written by AXIS and AXIS-B Programs on "y" and "x" Axis

Ideal limits [Hz]	Real limits [Hz]	Number of points	memory pick up locations (decimal)
0 - 2	0 - 1.94	7	16384 + 0
2 - 4	1.94 - 3.8	7	16384 + N + 0
4 - 8	3.8 - 7.7	14	16384 + 2N + 0
8 - 12	7.7 - 11.6	14	16384 + 2N + M
12 - 16	11.6 - 15.5	14	16384 + 2N + 2M
16 - 20	15.5 - 19.4	14	16384 + 2N + 3M
20 - 24	19.4 - 23.3	14	16384 + 2N + 4M
24 - 32	23.3 - 31.3	29	16384 + 2N + 5M
32 - 40	31.3 - 39.4	29	16384 + 2N + 5M + P
40 - 48	39.4 - 47.5	29	16384 + 2N + 5M + 2P
48 - 56	47.5 - 55.5	29	16384 + 2N + 5M + 3P
56 - 64	55.5 - 63.6	29	16384 + 2N + 5M + 4P
64 - 71.1	63.6 - 71.1	27	16384 + 2N + 5M + 5P

b. Ideal and Real Frequency Bands Limits

Figure 2.33

2.33) and the title of the histogram.

AXIS-B does the same thing but for channel B. The difference with respect to "AXIS" is in the title of the histogram and in the program that the control is transferred to. "AXIS" transfers the control to "PLOT-A" and "AXIS-B" to "PLOT-B".

PLOT-A calculates the coefficients for the histogram and then draws it. These coefficients represent the relative spectral power of each of the thirteen bands, with respect to the total power.

$$[\%]A_i = \frac{\Sigma P_j^2}{\Sigma P_K^2} \times 100$$

$i = 1\text{-}13$

$j = 1\text{-}l$; l = no. of points in each band

$K = 1\text{-}256$

$P_{j,K}$ = spectral amplitude (1 byte in memory)

A_i = histogram coefficients

The spectral amplitudes $P_{j,K}$ are squared because the histogram is related to power, which is proportional with the amplitude squared.

These amplitudes are the 512 bytes resulted from AVERAGE program and are situated at $4000 (decimal 16384). Only the first 256 bytes are the one we are interested in because the other 256 are the mirror image of the first 256. In conclusion, PLOT-A calculates the histogram coefficients for 256 bytes located at $4000. The histogram is done for 13 frequency bands.

The spectrums and their average are between 0 and 71.1 Hz. The bands were chosen by comparing the work done before on these kinds of investigation which show: small bands in the beginning (2 bands of 2 Hz each) followed by 5 bands of 4 Hz each and, at the end one 8 Hz band. These form a total band of only 32 Hz. This was the band used in the previous research. My project does the spectrums in a larger band, 71.1 Hz. The purpose is to quantify the high frequency changes and relate them, eventually, to some drug effects. In order to cover the entire new band, 0-71.1 Hz, I chose to divide it, for the histogram, in 2 small bands of 2 Hz each in the beginning, five 4 Hz bands and others five 8 Hz bands. Of course, these numbers are not exactly the final ones because of the computer resolution.

The real frequency band limits are calculated in the following way:

$$\text{total frequency band} = \frac{\text{256 points (spectrum points)}}{\text{3.6 seconds (1 sweep interval)}} = 71.1 \text{ Hz}$$

256 points represent 71.1 Hz
x points represent 2 Hz (the smallest band)

$$x = \frac{256 \times 2}{71.1} = \frac{256 \times 2}{\frac{256}{3.6}} = 7.2 \text{ points/Hz}$$

But 7.2 is not an integer number. Choosing 7 points/Hz we have: 7 points x frequency resolution [Hz/points] = number of Hz for the first two bands.

$$\text{frequency resolution} = \frac{71.1 \text{ Hz}}{256 \text{ points}} = 0.2\,(7)\ \frac{\text{Hz}}{\text{point}}$$

7 points x 0.2 (7) = 1.94 Hz

Choosing 8 points:

8 points x 0.2 (7) = 2.22 Hz

Taking in consideration that 1.94 Hz is closer to 2 Hz than 2.22 Hz, I choose, for the first 2 bands, 7 points/band. With the same considerations I choose, for the next 5 bands, 14 points/band; for the next 5 bands, 29 points/band; and for the last band, 256-[(2x7)+(5x14)+5x29)] = 256-229 = 27 points/band. There are, in conclusion, 4 numbers of points/band. They are called J, K, T, U.

J = 7 T = 29

K = 14 U = 27

The correspondent frequency intervals are:

$f_J = f_7 =$ 7 x resolution = 1.94 Hz (first 2 bands)
0.2 (7)

$f_K = f_{14}$ = 14 x resolution = 3.(8) Hz (next 5 bands)

$f_T = f_{29}$ = 29 x resolution = 8.05 Hz (next 5 bands)

$f_U = f_{27}$ = 27 x resolution = 7.5 Hz (last band)

The figure 2.33b shows the ideal and real frequency band limits and the memory locations where the correspondent points are located. The number in the "pick up memory locations" column represents the beginning of the interval. The 256 bytes of spectrum amplitudes are located at $4000-$40FF. The next location is

$4100 = decimal 16650 = 16384 + 2J + 5K + 5T + U.

After the histogram coefficients and the frequency limits are calculated the histogram is done on the screen and the control is transferred to MICROBUF.

PLOT-B does the same thing PLOT-A does, but picks up the information from $4800, decimal 18432.

MICROBUF is accessed after both PLOT-A and PLOT-B programs. It sends the screen information to the printer. It also checks the 16986 location. If it is 1 (16896 is set in CHOICE program), then the user chooses to process both channels so the program has to continue to process channel B after channel A results were sent to the printer. If it is 0, the user wants only channel A, so the program can be stopped after the results for channel A are printed out.

PLOT-A1 calculates the histogram coefficients, the percentage of the spectral power of each band relative to the total power, in the same way PLOT-A does. PLOT A-1 does not use these coefficients to draw a histogram but prepares them to be used in a pie chart drawing. PLOT-A1 transfers the control to PIECHART program.

PLOT-B1 does the same thing PLOT-A1 does, but picks up the information from other locations. The control is then transferred to PIECHARTB.

PIECHART draws the pie chart and the necessary explanations on the screen, for channel A. Then checks the location 16986. If it is "1", the user wants to continue to process channel B. If it is "0", the process stops, the user chose channel A only.

PIECHARTB draws the pie chart for channel B. This is the program that ends the processing without any alternative. The process can be ended by MICROBUF or PIECHART (see fig. 2.27) but in these two there is always a chance of continuation, if location 16896 = 1. These are the BASIC and machine language programs contained in Variant 1. They offer the user a lot of choices during the processing. That is why the user has to observe everything from the beginning to the end, in order to answer the questions of the program. axis "HZ", writes the numbers for y and x axis divisions (see Fig. 2.33) and the title of the histogram.

AXIS-B does the same thing but for channel B. The difference with respect to "AXIS" is in the title of the histogram and in the program that the control is transferred to. "AXIS" transfers the control to "PLOT-A" and "AXIS-B" to "PLOT-B".

2.2.4 Variant 2

As opposed to Variant 1, Variant 2 can stop the process after the data is acquired or can do everything, data acquisition, spectrum analysis, and statistical calculations. Another new thing for this variant is that it ends up with a 64 spectrum average for each channel and, again, the histogram or the pie chart can be done. The general flowchart for Variant 2 is given in fig. 2.34.

The first part of the Variant 2 works the same as Variant 1. Beginning with the "CHOICE" program things are changed. From here one, Variant 2 contains the following BASIC programs:

CHOICE1 is practically the same program as "CHOICE" from Variant 1, but accesses other programs than CHOICE does. Location 16896 is set again to zero if the user wants to process only one channel, A or B, and to 1 if the user wants to process A and B.

A-AVERAGE retrieves the spectrums from the disk, 8 times, 8

Kbytes of spectrums (all the spectrums from the disk). Then calculates 8 averages of 8 spectrums each and saves them in the program disk, DRIVE1, with the name "AVE.1", "AVE.2", ..., "AVE.8". The control is then transferred to FINAL-AVE.A.

B-AVERAGE is the analog of A-Average but for channel B. It transfers the control to FINAL-AVE.B.

FINAL-AVE.A retrieves the 8 averages for channel A from disk and places them in Super RAM in the right position so the same machine language routine can be used to average them. A 64 spectrum average is obtained, called "AVE.AVE.A", at $4000-41FF and it is saved in the program disk.

The program control is transferred to SEE FIN.AVE.A.

FINAL-AVE.B calculates 64 spectrums for channel B, "AVE.AVE.B", places it at $4800-$49FF, saves it in the program disk, and than transfers the control to SEE FIN.AVE.B.

SEE FIN.AVE.A offers the final average for channel A on the screen and printed out, asks the user to choose between the pie chart or histogram forms of analyzing this average, and depending on the choice, accesses, respectively, PLOT-A1 or AXIS.

SEE FIN.AVE.B is the analog of SEE FIN.AVE.A, for channel B and transfers the control to PLOT-B1 or AXISB.

From now on, the statistical analysis is done with the same programs as for Variant 1. The machine language routines used in Variant 2 are the same as for the first variant, with small changes or no change at all.

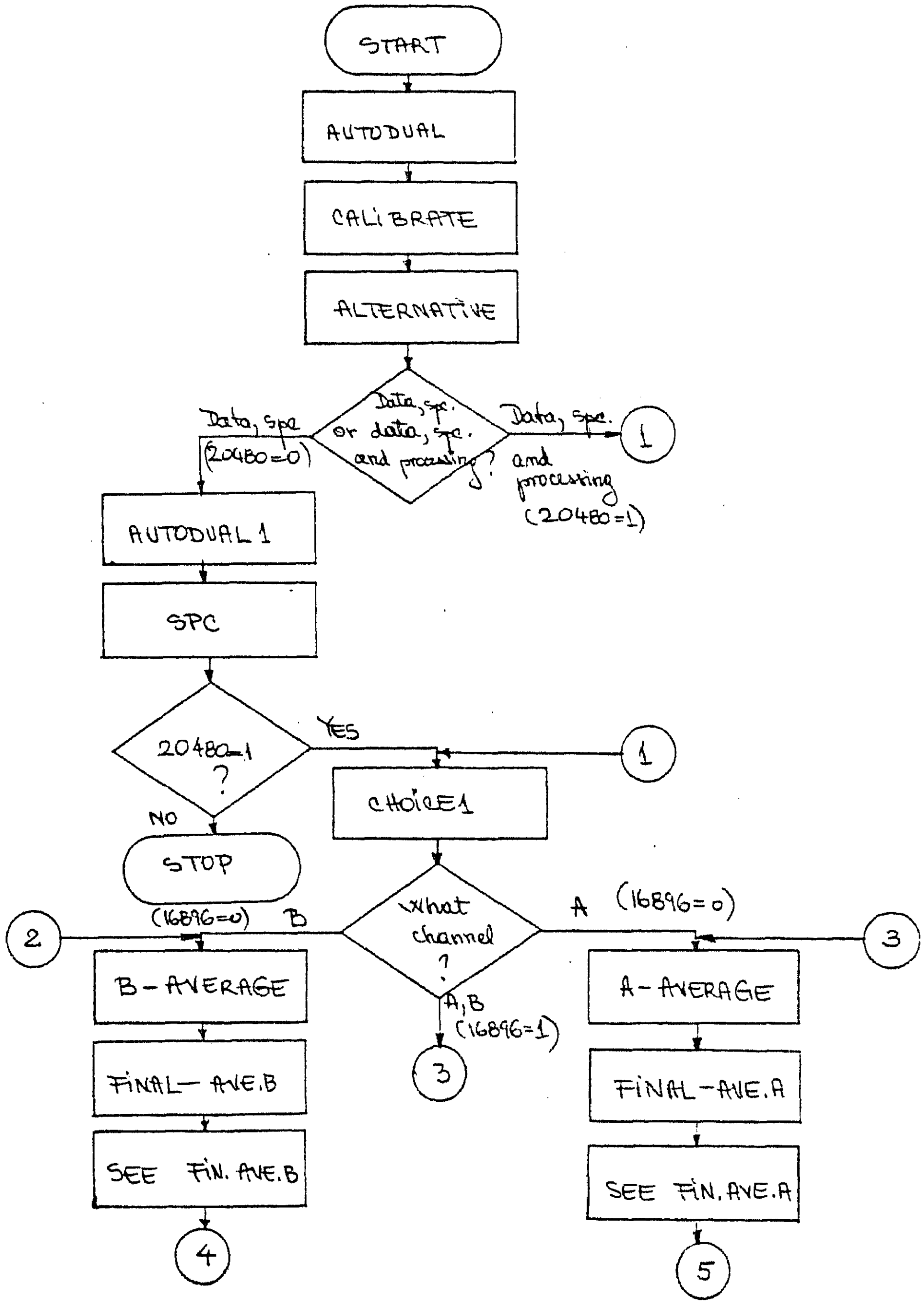


Figure 2.34 Variant 2 Flowchart

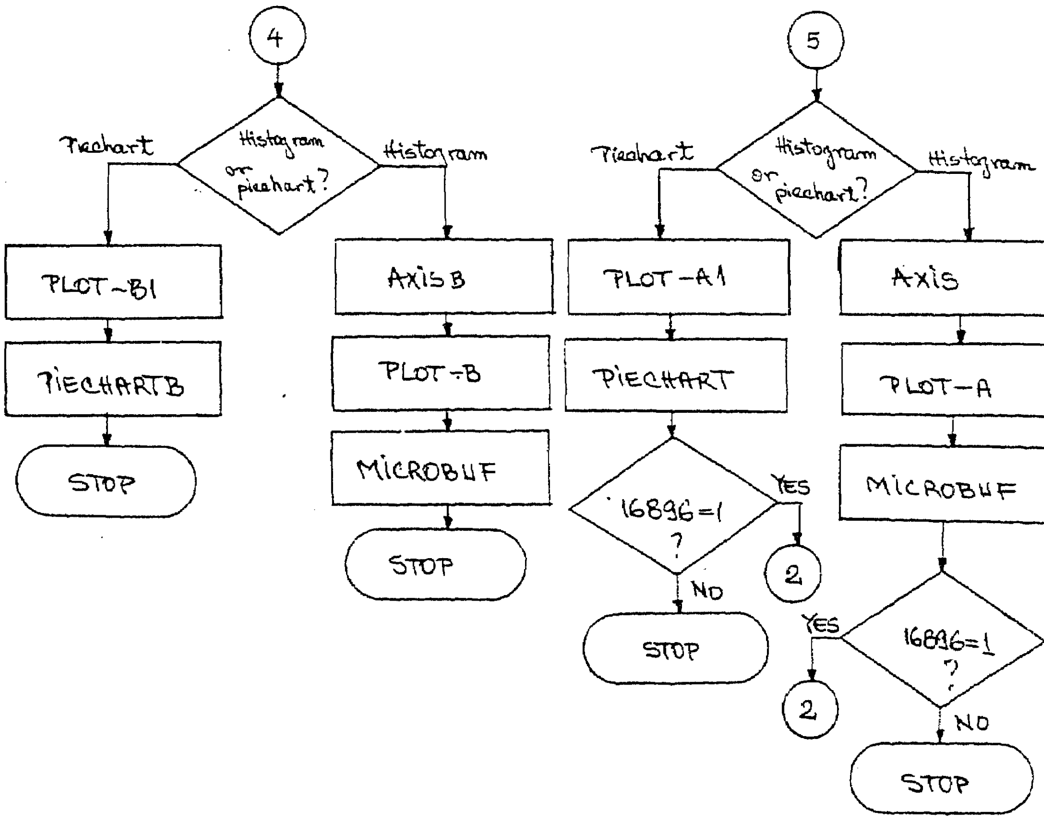


Figure 2.34 (continued)

Variant 2 Flowchart

2.2.5 Variant 3

Variant 3 of processing EEG signals is practically Variant 2 simplified. All the choices are eliminated in order for the user to just press the button for START and come and pick up the results from the printer (which contains all the programs). Variant 3 has 2 program disks, the first contains the data acquisition programs, and the second one contains the processing (spectral and statistical analysis) programs. Every time the first disk is executed, a 9 minute experiment is done and 128 sweeps (64 for each channel) of 512 bytes each are acquired on the data disk (DRIVE 2). Every time the second disk is executed, after 40 minutes of processing, a 64 spectrum average for channel A and one for channel B are obtained at the printer and an alarm goes off to remind the user that the results are ready to be pick up and filed.

All variants have an "INITIALIZING" disk which is a program disk used to initialize data disks for the experiments.

Variant 3 was set up to satisfy the user requirements and is the one most used. All the results obtained for this thesis are the Variant 3 results.

Figure 2.35 gives a general view of all the disks needed for all three variants of processing developed for this thesis.

For Variant 1 and 2, in case the user does not want to acquire data and process immediately, he can stop the program after the data acquisition and the spectral analysis are done and another disk can be made to continue with the processing whenever necessary. Each variant has 2 disks of programs and needs at the same time the initializing program. The total number of program disks made for this thesis was seven (see fig. 2.35).

The listings of all the BASIC programs and machine language routines developed for this thesis are given in Appendix F and G.

Disk Drive / Variant	DRIVE 1 (program disk)	DRIVE 2 (data disk)
VARIANT 1	1-disk - data, spectrums, processing (one 8 spc. average for each channel) (hist. or piechart) 1 disk - processing only	initialized data disks
VARIANT 2	1-disk - data, spectrums, processing (one 64 spc. average for each channel, hist. or piechart) 1- disk - processing only	initialized data disks
VARIANT 3	1-disk - data aquisition 1- disk - spectrums, processing (one 64 spc. average for each channel histogram only)	initialized data disks

INITIALIZING DISK

7 program disks in total

Fig. 2.35 Disks Needed to Process the EEG Signals

Chapter 3

EXPERIMENTAL RESULTS

3.1 Where is the Hoffmann-La Roche Research situated with respect to the computerized EEG in pharmacology

This thesis is based on the work and the experiments done at Hoffmann-La Roche, Nutley, New Jersey. This is a pharmaceutical company and one of its departments is concerned with the testing of the drugs using EEG spectral patterns.

As I described in Chapter 1, the EEG spectrum can give important information about the effects of various drugs on the central nervous system. The experiments are done on animals and humans and the effects of well-known drugs can be verified in this way. More than this, the research groups are trying new compounds, on animals, and observing the various changes in the EEG spectrums, they can classify the new drug in one of the categories of drugs. To do this, different computer systems are used. Chapter 1 describes the system used by the Hoffmann-La Roche company and what it is capable of doing. The experiments are done in the following manner: a group of monkeys is used and experiments are done on them before and after the drug administration (Fig. 3.1).

1° same monkey: a) four 64 EEG spectrums averages are obtained and the percentage of each band power (there are 8 bands, from 0 to 32 Hz) with respect to the total spectral power. All this is done without administering the drug and it is called Control Experiment.

b) A day after the control experiment, the Drug Experiment is done and another four 64 spectrums averages are obtained. The drug experiment is done for different drug doses. The time interval between different dose administration is about one week. The dose increases until the operator considers that the drug had enough effect on the monkey. He can check if it was enough effect by doing in parallel another experiment (the lever pressing experiment).

2° <u>other monkey</u>: The control and drug experiments, for various drug doses., are performed for a group of monkeys in the same way it was described before.

Each dose result is averaged for the n monkeys used in order to see the general effect of the drug, at that amount of drug.

These averages are then compared between them and with the averages obtained from the control experiment and dose related effects are obtained. The results are then plotted as in Figure 3.2. This figure contains three graphs. The left one represents the effect of the drug and contains the results of the lever pressing experiment which was the subject of another thesis. The second graph represents the total power for the drug experiment relative to the total power of the control experiment, in percentage, for different

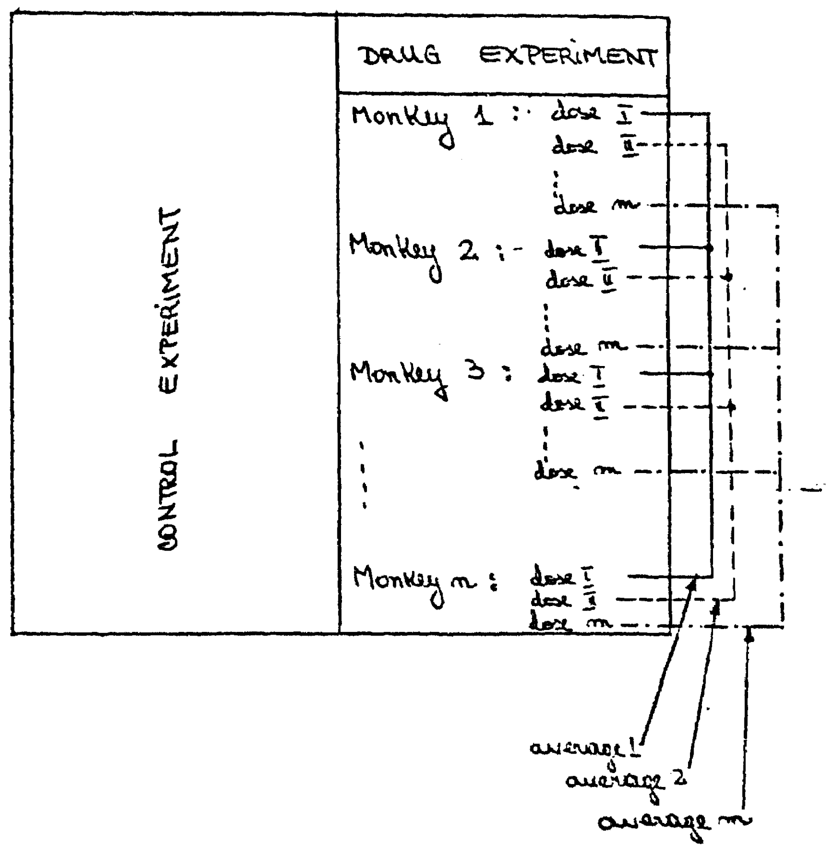


Figure 3.1 Experiments Outline

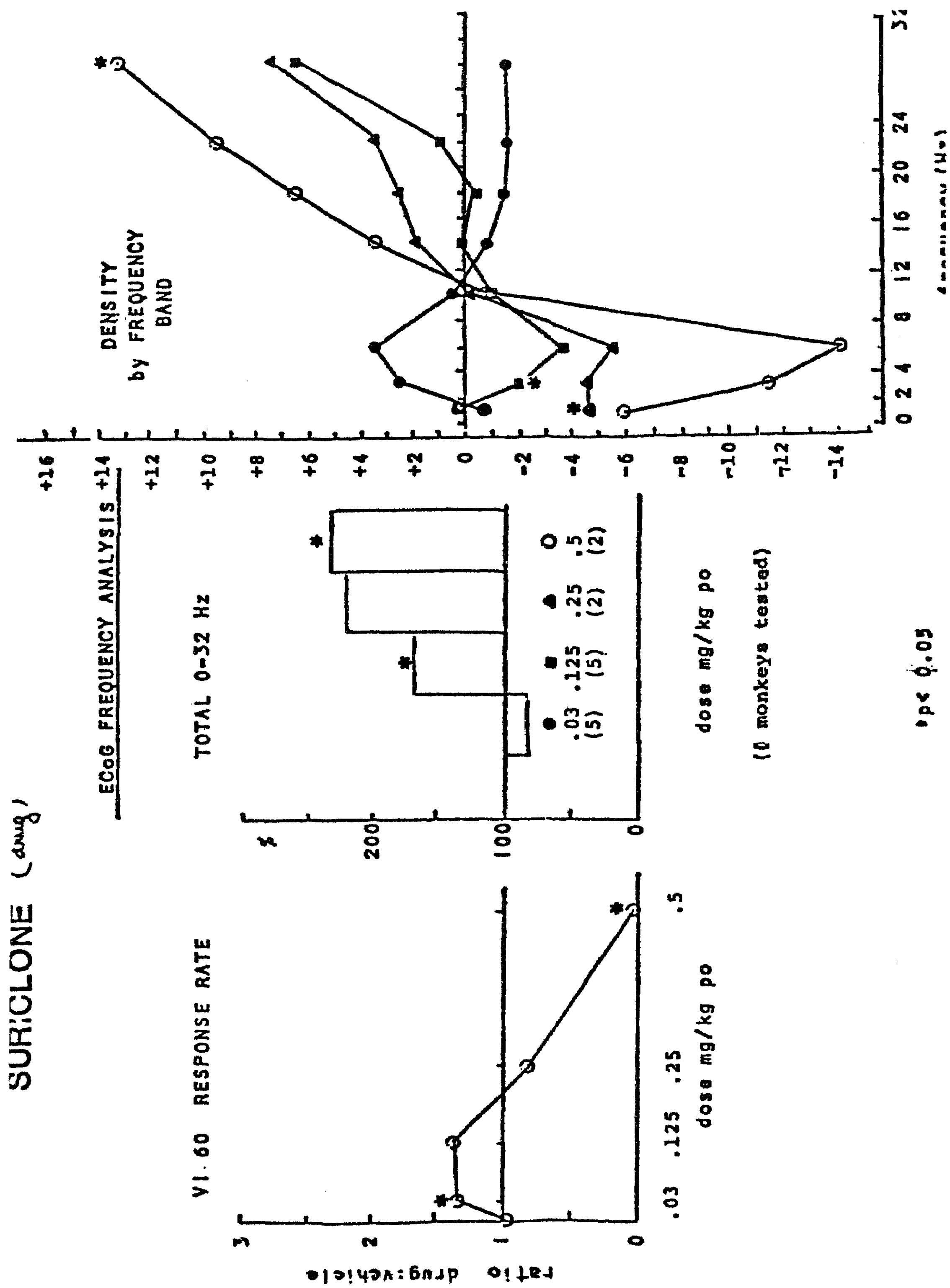


Figure 3.2 Statistical Results

drug dose. The "no-dose" experiment is the one that has 100% power and represents the control experiment. For this particular drug, we can observe that, as the dose increases, the total power increases. Sometimes, however total power increase does not tell too much because this increase can be due to the increase in the various bands powers (there are 8 bands). That is why this graph does not give the complete information about the changes in the EEG spectrum. The third graph provides additional information. It represents y = f (frequency) where y is:

$$y = \left.\frac{\text{band power}}{\text{total power}}\right|_{\substack{\text{control}\\ \text{experiment}}} + \left.\frac{\text{band power}}{\text{total power}}\right|_{\substack{\text{drug}\\ \text{experiment}}}$$

(the ratios are in percentages)

In this way the user can easily read due to which band the total power increases or decreases.

3.2 Validation of the new system, automated EEG spectrum analysis system

After I described how the experiments are carried out at Hoffmann-La Roche with the help of the old system, I have to point out how is the new system used in this research program. The new system is obtaining the spectrums for the control and drug experiment EEG patterns for different monkeys and various doses of drug. It does the 64 spectrums average (or 16 spectrums average), calculates the percentage of the bands power with respect to the total power (for 13 bands, 8 of them the same as for the old system and another 5 bands, up to 71.1 Hz) and draws the histogram or the pie chart to make easier the results interpretation.

All this is done only with the help of the Apple II computer and the results are obtained on the screen and at the printer: Chapter 1 describes the advantages of using this system with respect to the old one. However, the new system results have to be validated by comparing them with the old system results which have been used for seven years. For this reason, I used my variant 3 program to process the EEG signal for various drugs and monkeys (signals already processed with the old system).

The results that I'm going to discuss and compare with the ones of the old system are related to two drugs, suriclone and halazepam, each one administered to two monkeys. We have then four cases (see Fig. 3.3). The results are given in figures 3.4, 3.5, 3.6, 3.7, at the end of this chapter.

I. The "suriclone" drug dose used was 0.5 mg/kg. The drug number is 17-9887 and the number of the monkey is 461 (see fig. 3.4.1, 3.4.2.).

First of all, I would like to compare the control experiment results for the new and old system (fig. 3.4.1.a. and 3.4.2.a., 3.4.2.b). The comparison can not be done looking at the histogram because the frequency band is 0-71.1 Hz and for the old system is only 0-32 Hz. I designed my program to calculate the total power for 0-31.3 Hz and the relative bands power so these numbers can be compared with the results of the control experiment for the old system. For both systems the power is concentrated in the first five bands, the EEG spectrum having a maximum around 4 Hz.

Looking at the drug experiment results (fig. 3.4.1.b. and 3.4.2.c., 3.4.2.d.), both systems shown an increase of the last band power, 23.3-31.3 Hz. Two other aspects have to be observed, which are the difference between the control and drug experiment for the new system and the old system.

Looking at the control and drug averages for the new system (fig. 3.4.1.a. and b.), the amplitudes begin to increase once we are in alpha region (10-12 Hz). Beginning with the REM zone (rapid eye movement sleep, 20 Hz) they increase even more.

Comparing the spectrum averages for control and drug experiments for the old system (fig. 3.4.2.a. and c.), the amplitudes are increasing around 10 Hz also and between 20 and 30 Hz. The old system drug average has, however, very big amplitudes for very slow waves 0-2 Hz (the deep sleep region). These big amplitudes do not appear in the new system results. In order to explain this we have to compare how the experiments are carried out with both systems and see if they are totally compatible.

The drug and control results for the old system represent 1 of 4 possible results obtained at approximately 14-15 minutes intervals. The old system obtains the averages and also records on magnetic tape the EEG signal for about one hour. This EEG signal is the input for the new system and the probability to process the same part of the signal is very small. However, whatever part of the EEG signal recorded is processed, the

Drug	Monkey Number	
Suriclone	461 (channel B)	I
	514 (channel A)	II
Halazepam	461 (channel B)	III
	680 (channel A)	IV

Figure 3.3

Results Outline

spectrum has to be almost the same. But, of course, many small differences can appear depending on which part of the EEG signal is processed. I think the big amplitudes for low frequency couldn't appear, for this reason, in the new system results.

II. The same drug, same dose was used but with monkey 514 (see fig. 3.5.1. and 3.5.2.).

Looking at the control experiment results for both systems we can observe that the spectral energy is concentrated in the theta and delta regions (deep sleep waves) (see fig. 3.5.1.a. and 3.5.2.a., 3.5.2.b.).

If, for the control results, in both cases the amplitudes decrease beginning with 8 Hz, for the drug results, both systems show a constant (or almost constant) amplitude between 8 and 20 Hz and an increase in amplitude between 20 and 30 Hz (fig. 3.5.1.b. and 3.5.2.c., 3.5.2.d.).

In conclusion, the changes from the control to the drug experiment results, are around the alpha band (both systems show a constant amplitude when drug is administered with respect to the "no-drug" case where this band has low amplitude spectrum) and an increase in amplitude in the drug 20-30 Hz interval is shown by both systems.

III. The drug used is "halazepam", the dose is also 0.5 mg/kg and the experiment is done on the monkey numbered 461 (see fig. 3.6.1 and 3.6.2.). Again, first of all I would like to compare the control experiment results for these 2 systems (fig. 3.6.1.a. and 3.6.2.a., 3.6.2.b.). Both of them show a very big activity in the low frequency region (0-10 Hz) and then a decrease in the brain activity.

After the drug is administered (drug experiment results) the slow waves activity is still significant with respect to the total band activity but drops a little bit. Also, the last part of the spectrum, between 25-30 Hz has larger amplitudes (fig. 3.6.1.b. and 3.6.2.c.). Both systems show the same thing. For the drug experiment, a picture of the average is not available, but the comparison can be done looking at the percentage of the power in each band, with respect to the entire band.

IV. Same drug, halazepam, was administered to the monkey numbered 680, same dose being used (see fig. 3.7.1. and 3.7.2.).

The control experiments for both systems (fig. 3.7.1.a. and 3.7.2.a., 3.7.2.b.) show no activity but in the 0-10 Hz frequency band. Both drug results show a small increase in the activity in all bands. In the 0-10 Hz band, the increase is more visible (fig. 3.7.1.b. and 3.7.2.c, 3.7.2.d.).

The histograms obtained for all these cases can not be used for any of these discussions because they consider the total band between 0-71.1 Hz. However, they represent a vary easy method of checking the power distribution in this range.

The validation of the new system can not be done, of course, by considering only 4 cases. Practically, all the well-known drugs, already tested with the old system were tried again with the new one, using the same EEG patterns from the magnetic tape which was kept in files. After about 2 months of running the new system program on the information from the magnetic tape, the new system was considered as good, with nice features and was adopted for the research work. Since this work was completed, the system has been extended to 16 channels.

3.3 Conclusions

The automated EEG spectrum analysis system was considered to have the same results as the old system. The difference is that the new system is very small and almost completely automated, the user has only to change the data disk when the previous disk is full. The system works with two channels simultaneously. However, the time required to acquire and process the data is almost the same as for the old system. The advantage consists in having a small computer system which does everything without the user intervention and which can be programmed to do whatever statistical calculations are needed.

The system has a few limitations. One of them is that it does not improve the interval of time for the data acquisition, but I think this is not so important as the features introduced. Another limitation consists in the method of averaging 64 spectrums. The average is not obtained directly but by averaging 8 averages of 8 spectrums each. This was done because of the small memory space available and might be responsible for some of the differences between the systems results. Of course, the system can be enhanced as follows:

1. The introduction of a hard disk will avoid changing the data disk after each experiment and more experiments could be done consequently. In this way, the effects produced by the drug could be better observed.

   The possibility of having a hard disk provides a lot of storage place. This convenience gives the idea of building a drug data base which could serve as a reference in the classification of new compounds.

2. The system can be modified to work with 16 channels simultaneously. In this way the time problem is solved.

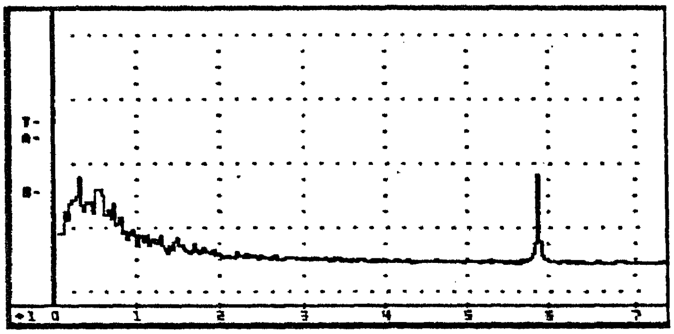
3. The software can be further developed to do all the comparisons between the control and drug experiments results or between the drug results and the data base, to classify automatically new drugs and to file the information.

With all these features, the system seems to introduce a nice and convenient method of analyzing drug effects and classifying new pharmaceutical compounds.

AVERAGEB-64

SURICLONE - CONTROL
17-9887
M461, ch.B

I



A 4.00 VOLT/DIV B .400 VOLT/DIV
TRACE A TRIGGER A > START
TIME 000.200 000 DISPLAY > 00001 + 0
SELECT NUMBER OF POINTS 0512
TOTAL POWER=133788
% OF FIRST BAND=11.6325829
% OF SECOND BAND=23.691213
% OF THIRD BAND=37.0040661
% OF FOURTH BAND=7.93494185
% OF FIFTH BAND=4.25299728
% OF SIXTH BAND=2.19675905
% OF SEVENTH BAND=1.01877597
% OF EIGHTH BAND=1.35886627
% OF NINTH BAND=.682422938
% OF TENTH BAND=.492570335
% OF ELEVENTH BAND=.400633839
% OF TWELTH BAND=7.38481778
% OF THERTEENTH BAND=.267213801
TOTAL % BEFORE ROUND OFF IS T%=98.3178611
% POWER OF THE THIRTEEN BANDS ARE:
A1=11.6
A2=23.6
A3=37
A4=7.9
A5=4.2
A6=2.2
A7=1
A8=1.3
A9=.6
B1=.4
B2=.4
B3=7.3
B4=.2
TOTAL % POWER IS T%=97
TOTAL POWER OF FIRST 8 BANDS=122478.087
NEW % FOR THE FIRST 8 BANDS ARE:

12.6711712	25.7792793
40.4166667	8.6295045
4.58783784	2.40315315
1.09234234	1.42004504

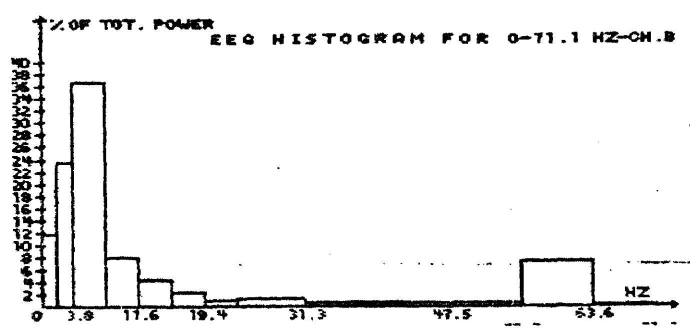
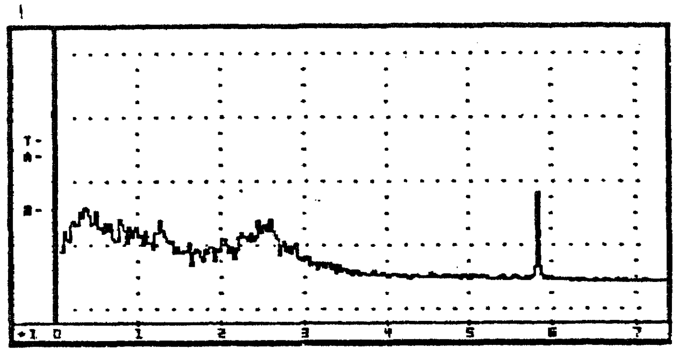


Figure 3.4.1.a. Suriclone control experiment - new system, monkey 461

AVERAGEB—64



SURICLONE —DRUG
17-5337
M461, ch. B

```
 A  4.00 VOLT/DIV        B  .400 VOLT/DIV
TRACE  A        TRIGGER   A  >   START
TIME  000.200 000   DISPLAY > 00001 + 0
   SELECT NUMBER OF POINTS  0512
TOTAL POWER=246974
% OF FIRST BAND=6.08970985
% OF SECOND BAND=13.1766907
% OF THIRD BAND=18.0755869
% OF FOURTH BAND=11.7182376
% OF FIFTH BAND=8.85477824
% OF SIXTH BAND=5.55038183
% OF SEVENTH BAND=9.28316341
% OF EIGHTH BAND=17.5305903
% OF NINTH BAND=1.55441464
% OF TENTH BAND=.653105185
% OF ELEVENTH BAND=.462396852
% OF TWELTH BAND=3.7364257
% OF THERTEENTH BAND=.130540057
TOTAL % BEFORE ROUND OFF IS T%=96.8160211
% POWER OF THE THIRTEEN BANDS ARE:
A1=6
A2=13.1
A3=18
A4=11.7
A5=8.8
A6=5.5
A7=9.2
A8=17.5
A9=1.5
B1=.6
B2=.4
B3=3.7
B4=.1
TOTAL % POWER IS T%=96
TOTAL POWER OF FIRST 8 BANDS=231023.596
NEW % FOR THE FIRST 8 BANDS ARE:
6.4142539        14.0044543
19.2427617       12.5077951
9.40757239       5.87973274
9.83518931       18.7082405
```

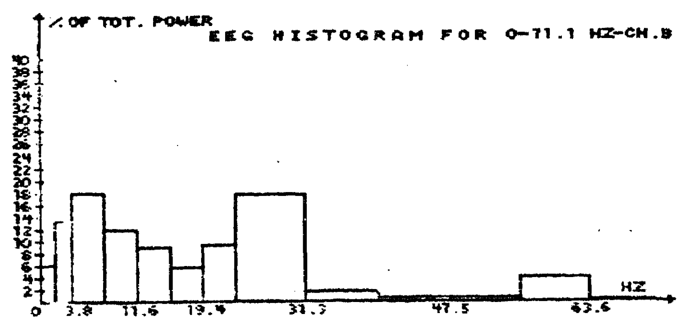


Figure 3.4.1.b. Suriclone drug experiment - new system, monkey 461

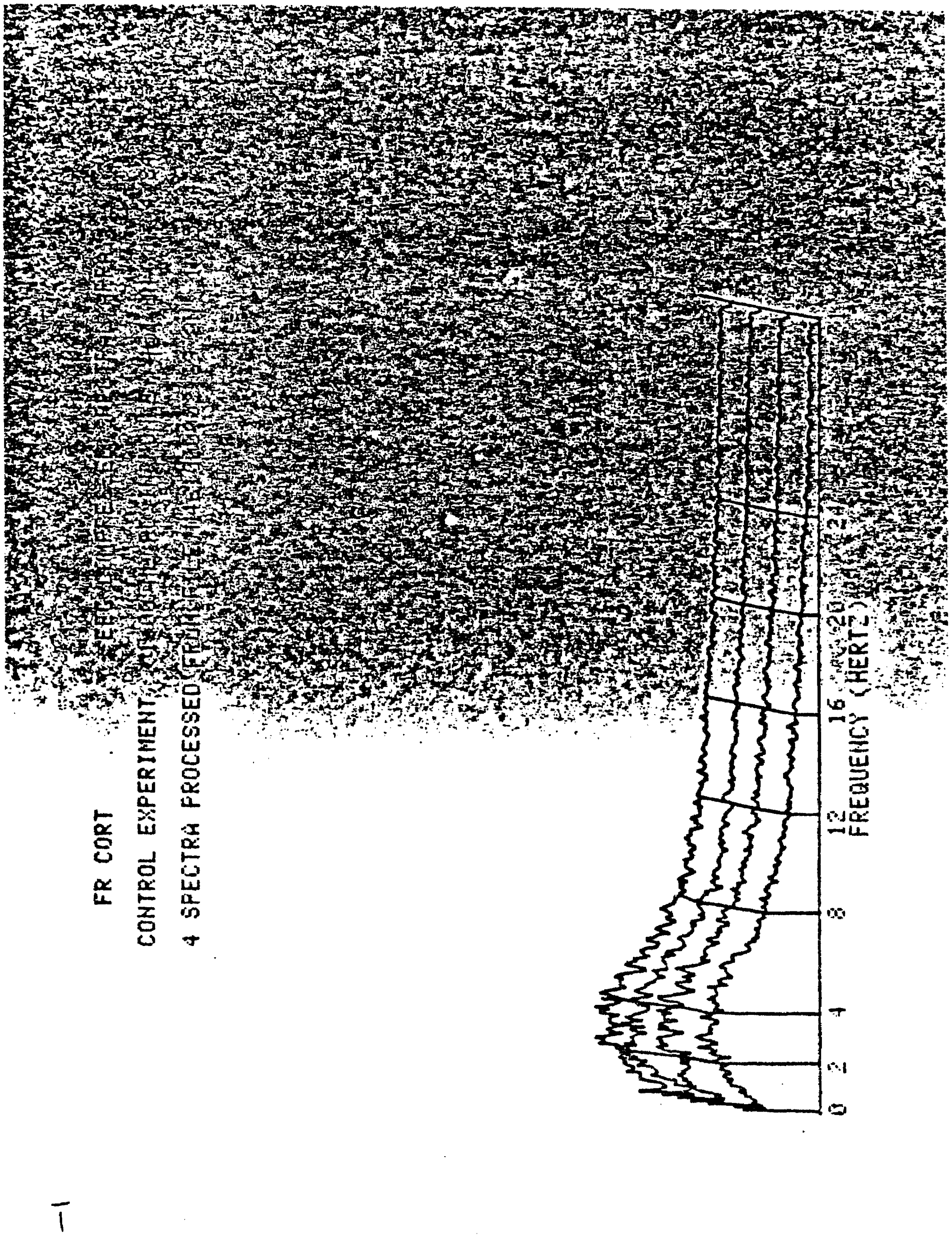


Figure 3.4.2.a. Suriclone control experiment - old system, monkey 461

1-1

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

CONTROL EXPERIMENT, 1.00 ML, IN MONKEY 461. EXPT PERFORMED ON 1/ 4/84

BAND:		1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	CLOCK END TIME	0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
		1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25

		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	13:13	2085	2891	4261	2123	1376	997	726	865	15324.
2	13:27	2453	3234	4549	2209	1438	1020	696	899	16498.
3	13:40	2568	3297	4384	2073	1371	1022	758	1002	16475.
4	13:54	2406	3163	4117	2070	1390	1034	761	1011	15952.

PERCENT FOR EACH SPECTRUM BAND

1	13.61	18.87	27.81	13.85	8.98	6.51	4.74	5.64
2	14.87	19.60	27.57	13.39	8.72	6.18	4.22	5.45
3	15.59	20.01	26.61	12.58	8.32	6.20	4.60	6.08
4	15.08	19.83	25.81	12.98	8.71	6.48	4.77	6.34

SUM OF SPECTRA - PERCENTAGE OF TOTAL

1 AND 2	14.26	19.25	27.69	13.61	8.84	6.34	4.47	5.54
3 AND 4	15.34	19.92	26.22	12.78	8.51	6.34	4.68	6.21
5 AND 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

NOVA3:M461000002.89

Figure 3.4.2.b. Suriclone control results - old system, monkey 461

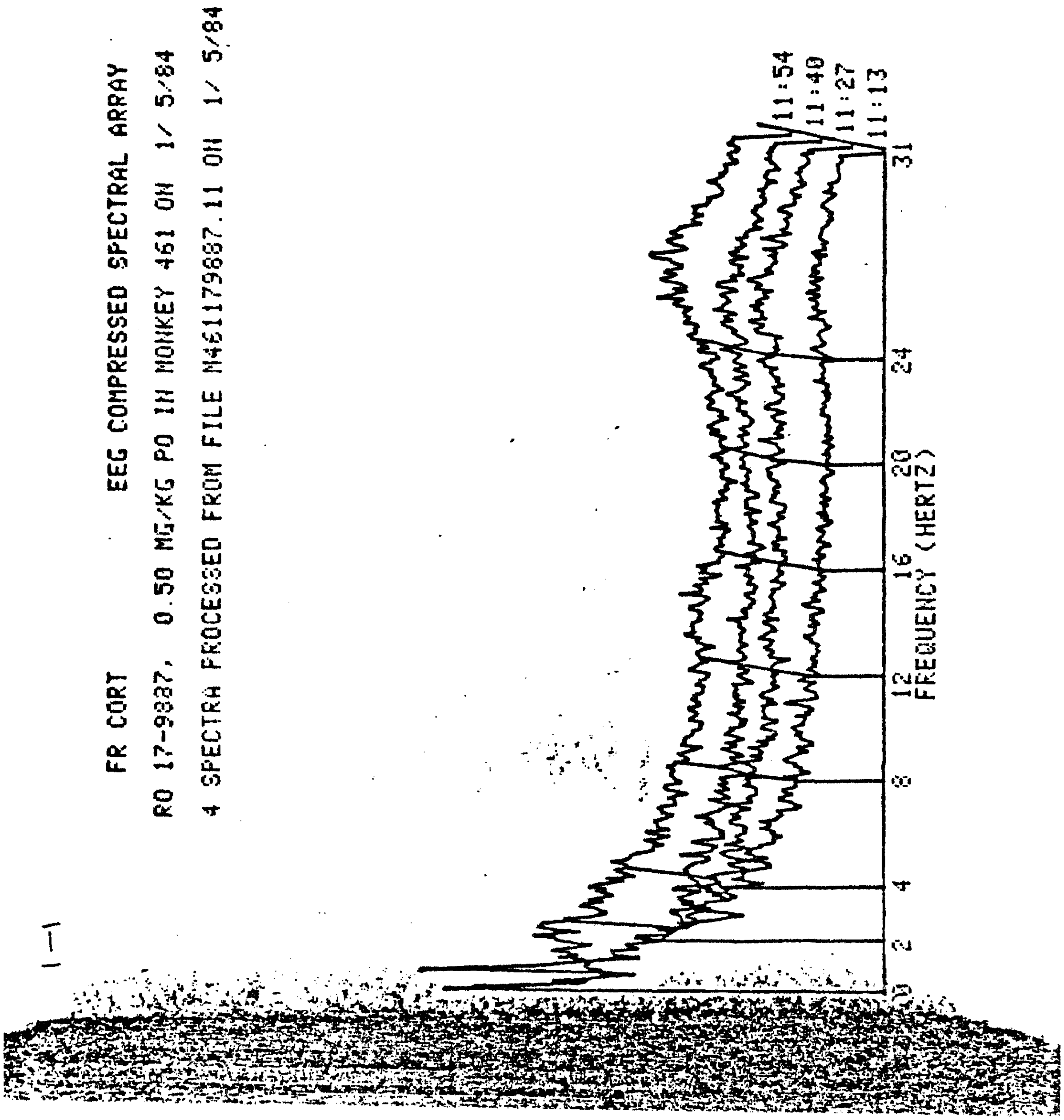


Figure 3.4.2.c. Suriclone drug experiment - old system, monkey 461

|—|

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

R017-9287 AT 0.50 MG/KG, IN MONKEY 461. EXPT PERFORMED ON 1/ 5/84

	BAND	1	2	3	4	5	6	7	8	TOTAL
	CLOCK	<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	END	0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
	TIME	1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25

		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	11:13	6989	4780	5491	3963	3335	2949	2848	5255	35210.
2	11:27	4347	3572	5409	4213	3837	3788	3806	6465	35437.
3	11:40	3028	3261	5053	4027	3595	3573	3892	6393	32822.
4	11:54	5634	4391	5953	4419	4056	3307	3809	7942	39511.

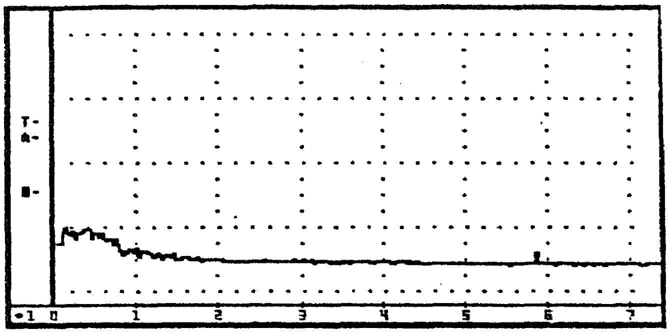
	PERCENT FOR EACH SPECTRUM BAND							
1	19.85	12.44	15.60	11.26	9.47	8.38	8.09	14.92
2	12.27	10.08	15.26	11.89	10.83	10.69	10.74	18.24
3	9.23	9.94	15.40	12.27	10.95	10.89	11.86	19.48
4	14.26	11.11	15.07	11.18	10.27	8.37	9.64	20.10

	SUM OF SPECTRA - PERCENTAGE OF TOTAL							
AND 2	16.05	11.26	15.43	11.57	10.15	9.54	9.42	16.59
AND 4	11.98	10.58	15.22	11.68	10.58	9.51	10.65	19.82
AND 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

NOVA3:M461179887.11

Figure 3.4.2.d. Suriclone drug results - old system, monkey 461

IPR#6
AVERAGEA—64



SURICLONE - CONTROL
17-5887
M 514, ch.A

II

A 4.00 VOLT/DIV B .400 VOLT/DIV
TRACE A TRIGGER A > START
TIME 000.200 000 DISPLAY > 00001 + 0
SELECT NUMBER OF POINTS 0512

TOTAL POWER=30828
% OF FIRST BAND=18.1717919
% OF SECOND BAND=26.1061373
% OF THIRD BAND=34.5205657
% OF FOURTH BAND=8.01219671
% OF FIFTH BAND=3.38653173
% OF SIXTH BAND=1.83923706
% OF SEVENTH BAND=.908265214
% OF EIGHTH BAND=2.29661347
% OF NINTH BAND=1.35915402
% OF TENTH BAND=.956922279
% OF ELEVENTH BAND=.544959128
% OF TWELTH BAND=.827170105
% OF THERTEENTH BAND=.425911509
TOTAL % BEFORE ROUND OFF IS T%=99.3554561
% POWER OF THE THIRTEEN BANDS ARE:
A1=18.1
A2=26.1
A3=34.5
A4=8
A5=3.3
A6=1.8
A7=.9
A8=2.3
A9=1.3
B1=.9
B2=.5
B3=.8
B4=.4
TOTAL % POWER IS T%=98
TOTAL POWER OF FIRST 8 BANDS=29884.2857
NEW % FOR THE FIRST 8 BANDS ARE:

18.6715789	26.9242105
35.5894737	8.25263158
3.40421052	1.8568421
.928421052	2.37263158

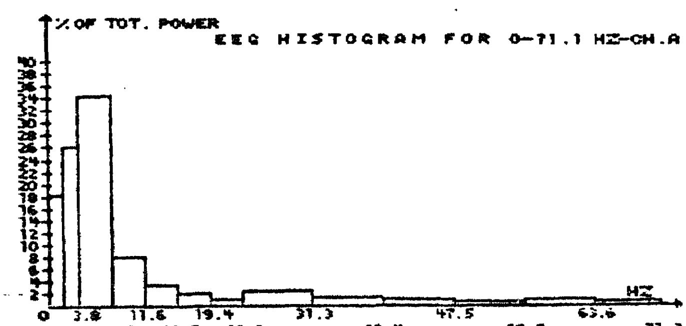
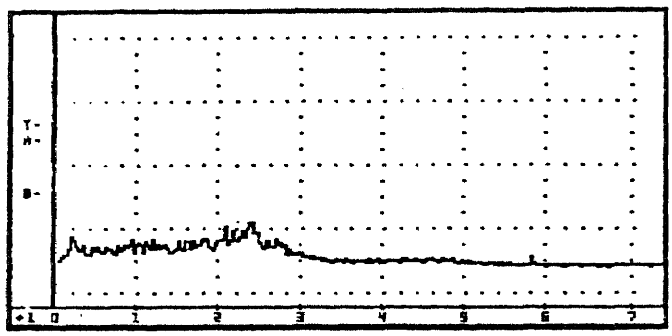


Figure 3.5.1.a. Suriclone control experiment - new system, monkey 514

AVERAGEA—64

SURICLONE —DRUG
17-9887.
M514, ch.A

II



A 4.00 VOLT/DIV B .400 VOLT/DIV
TRACE A TRIGGER A > START
TIME 000.200 000 DISPLAY > 00001 + 0
SELECT NUMBER OF POINTS 0512

TOTAL POWER=58392
% OF FIRST BAND=2.96444719
% OF SECOND BAND=3.25900809
% OF THIRD BAND=7.02836006
% OF FOURTH BAND=11.1590629
% OF FIFTH BAND=8.79915058
% OF SIXTH BAND=12.3441567
% OF SEVENTH BAND=25.7072887
% OF EIGHTH BAND=18.5059597
% OF NINTH BAND=1.83415536
% OF TENTH BAND=2.38731333
% OF ELEVENTH BAND=.810042471
% OF TWELTH BAND=.431565968
% OF THERTEENTH BAND=.220406905
TOTAL % BEFORE ROUND OFF IS T%=95.450918
% POWER OF THE THIRTEEN BANDS ARE:
A1=2.9
A2=3.2
A3=7
A4=11.1
A5=8.8
A6=12.3
A7=25.7
A8=18.5
A9=1.8
B1=2.3
B2=.8
B3=.4
B4=.2
TOTAL % POWER IS T%=95
TOTAL POWER OF FIRST 8 BANDS=55011.4105
NEW % FOR THE FIRST 8 BANDS ARE:

3.07821229	3.39664805
7.4301676	11.7821229
9.34078212	13.0558659
27.2793296	19.6368715

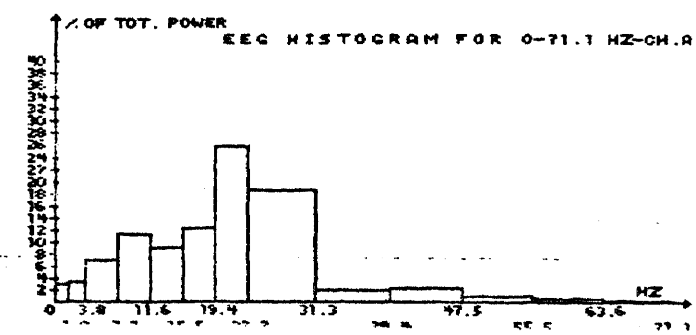


Figure 3.5.1.b. Suriclone drug experiment - new system, monkey 514

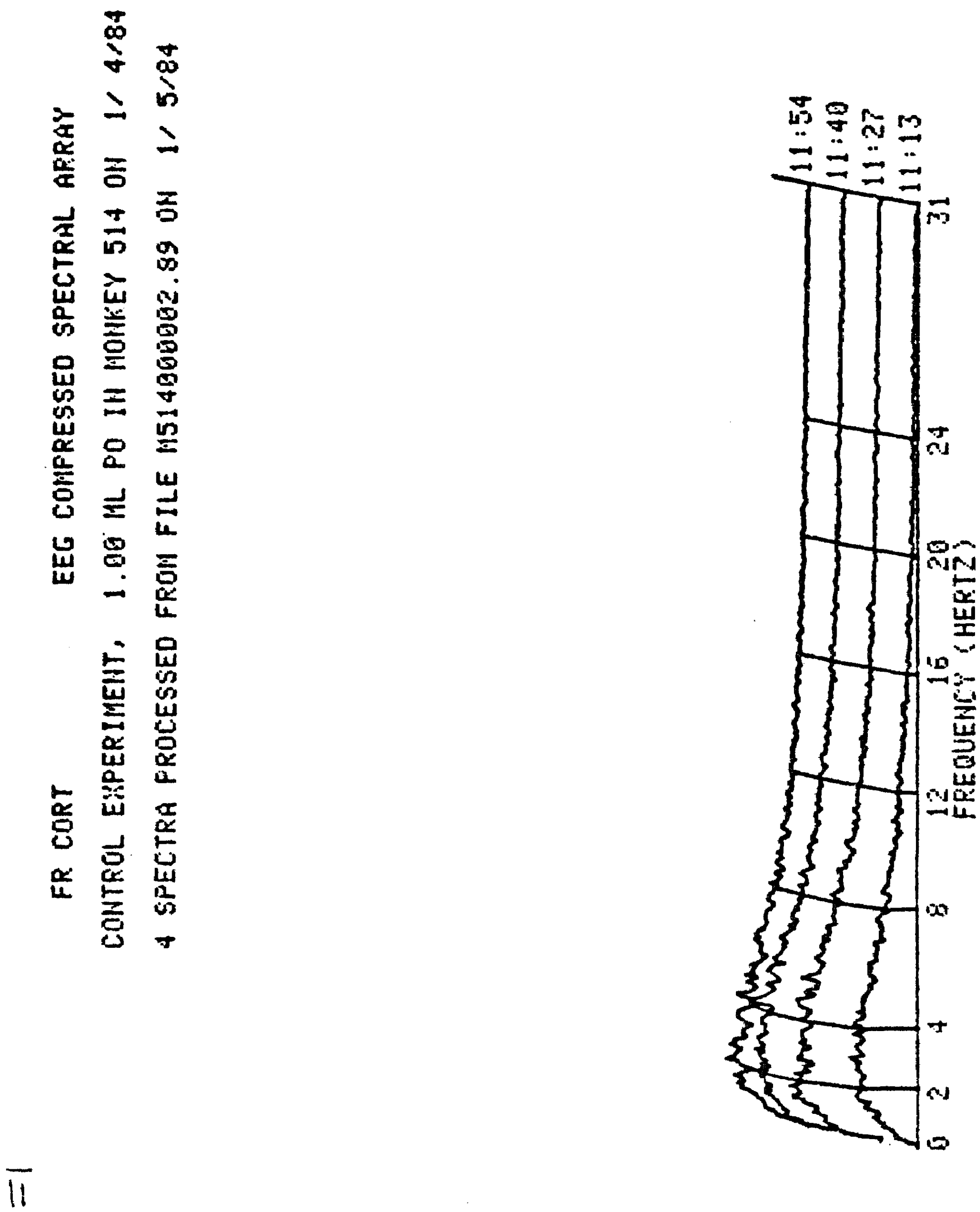


Figure 3.5.2.a. Suriclone control experiment - old system, monkey 514

SPECTRAL DENSITY ESTIMATES BY BAND
EEG FROM THE FR CORT

CONTROL EXPERIMENT, 1.00 ML, IN MONKEY 514. EXP

BAND:	1	2	3	4	5
	<- UPPER AND LOWER FREQUENCY LIMIT OF				
* CLOCK END	0.00	1.99	4.02	8.01	11.99
TIME	1.99	4.02	8.01	11.99	15.98

<- AREA UNDER SPECTRUM WITHIN BAND (ARBI

1	11:13	835	1250	1867	956	565
2	11:27	1336	1668	2555	1276	633
3	11:40	1334	1787	2663	1327	693
4	11:54	1090	1507	2089	992	538

PERCENT FOR EACH SPECTRUM BAND

1	13.45	20.14	30.08	15.40	9.10
2	16.12	20.13	30.83	15.40	7.64
3	15.31	20.51	30.57	15.23	7.95
4	15.82	21.88	30.32	14.40	7.81

SUM OF SPECTRA - PERCENTAGE OF

1 AND 2	14.98	20.13	30.51	15.40	8.27
3 AND 4	15.54	21.11	30.46	14.86	7.89
5 AND 6	0.00	0.00	0.00	0.00	0.00

Figure 3.5.2.b. Suriclone control results - old system, monkey 514

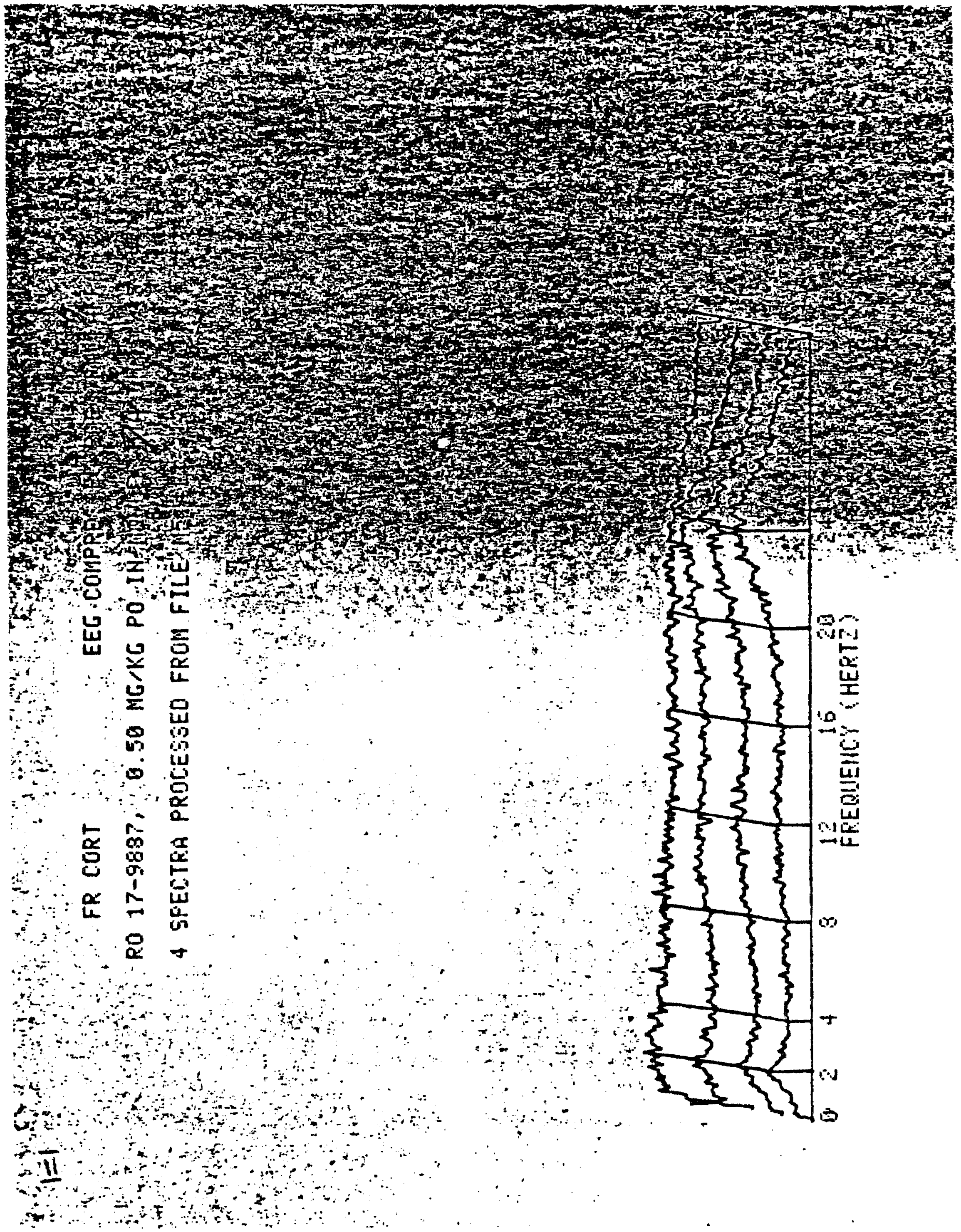


Figure 3.5.2.c. Suriclone drug experiment - old system, monkey 514

1=1

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

RO17-9887 AT 0.50 MG/KG. IN MONKEY 514. EXPT PERFORMED ON 1/ 5/84

BAND		1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	CLOCK END TIME	0.00 1.99	1.99 4.02	4.02 8.01	8.01 11.99	11.99 15.98	15.98 20.04	20.04 24.02	24.02 31.25	0.00 31.25

		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	11:13	666	745	1262	1575	1608	1795	3066	3808	14527.
2	11:27	637	784	1467	2059	1937	2158	3168	3203	15413.
3	11:40	1074	1102	2069	2575	2442	2667	3253	3001	18183.
4	11:54	1501	1733	2992	2644	2480	2601	2541	2816	19508.

PERCENT FOR EACH SPECTRUM BAND

1	4.60	5.13	8.69	10.84	11.07	12.36	21.11	26.21
2	4.13	5.09	9.52	13.36	12.57	14.00	20.55	20.78
3	5.91	6.06	11.38	14.16	13.43	14.67	17.89	16.50
4	7.63	8.88	15.34	14.53	12.71	13.33	13.03	14.44

SUM OF SPECTRA - PERCENTAGE OF TOTAL

1 AND 2	4.36	5.11	9.11	12.14	11.84	13.20	20.82	23.42
3 AND 4	6.83	7.52	13.43	14.33	13.06	13.98	15.37	15.43
5 AND 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

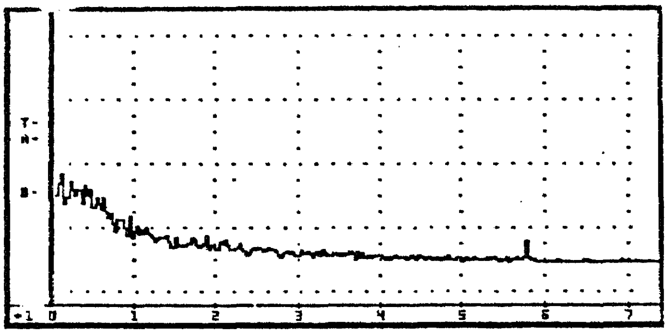
NOVA3:M514179887.11

Figure 3.5.2.d. Suriclone drug results - old system, monkey 514

HALAZEPAM - CONTROL
8-6728
M 461, ch.B

III

AVERAGEB-64



```
 A  4.00 VOLT/DIV       B  .400 VOLT/DIV
TRACE  A        TRIGGER   A  >   START
TIME  000.200 000    DISPLAY > 00001 + 0
   SELECT NUMBER OF POINTS  0512
TOTAL POWER=176033
% OF FIRST BAND=19.1895838
% OF SECOND BAND=21.4681338
% OF THIRD BAND=26.8966614
% OF FOURTH BAND=10.033346
% OF FIFTH BAND=4.51108599
% OF SIXTH BAND=3.72827822
% OF SEVENTH BAND=2.63018866
% OF EIGHTH BAND=3.34085087
% OF NINTH BAND=1.82579403
% OF TENTH BAND=1.09865764
% OF ELEVENTH BAND=.798713878
% OF TWELTH BAND=.761220907
% OF THERTEENTH BAND=.433498265
TOTAL % BEFORE ROUND OFF IS T%=96.7160135
% POWER OF THE THIRTEEN BANDS ARE:
A1=19.1
A2=21.4
A3=26.9
A4=10
A5=4.5
A6=3.7
A7=2.6
A8=3.3
A9=1.8
B1=1.1
B2=.8
B3=.7
B4=.4
TOTAL % POWER IS T%=96
TOTAL POWER OF FIRST 8 BANDS=167781.453
NEW % FOR THE FIRST 8 BANDS ARE:
20.0393443       22.452459
28.2229508       10.4918033
4.72131148       3.88196722
2.72786885       3.46229508
```

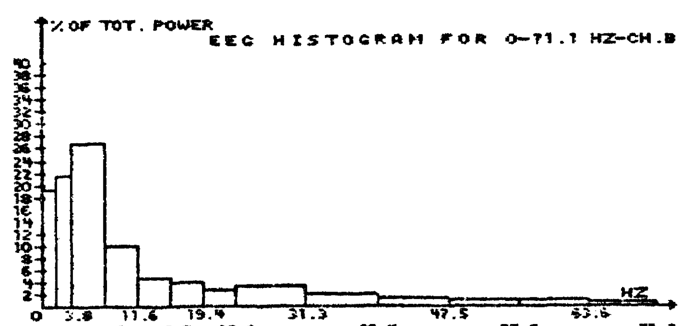
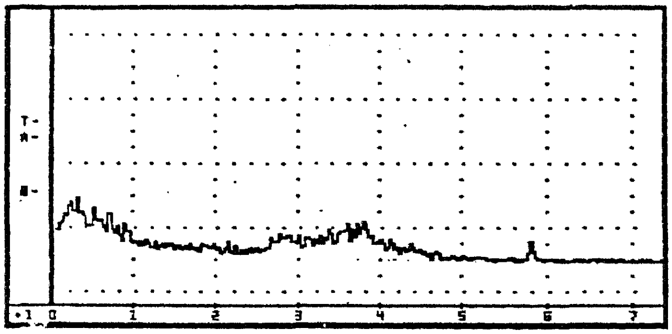


Figure 3.6.1.a. Halazepam control experiment - new system, monkey 461

AVERAGEB-64



HALAZEPAM - DRUG
8-6728
M461, ch. B
III

A 4.00 VOLT/DIV B .400 VOLT/DIV
TRACE A TRIGGER A > START
TIME 000.200 000 DISPLAY > 00001 + 0
SELECT NUMBER OF POINTS 0512
TOTAL POWER=144757
% OF FIRST BAND=9.59193684
% OF SECOND BAND=14.180316
% OF THIRD BAND=18.5421085
% OF FOURTH BAND=6.59449975
% OF FIFTH BAND=3.67167046
% OF SIXTH BAND=3.29034175
% OF SEVENTH BAND=2.36396167
% OF EIGHTH BAND=11.5538454
% OF NINTH BAND=18.3431544
% OF TENTH BAND=4.83707178
% OF ELEVENTH BAND=1.0887211
% OF TWELTH BAND=1.25589781
% OF THERTEENTH BAND=.66456199
TOTAL % BEFORE ROUND OFF IS T%=95.9780874
% POWER OF THE THIRTEEN BANDS ARE:
A1=9.5
A2=14.1
A3=18.5
A4=6.5
A5=3.6
A6=3.2
A7=2.3
A8=11.5
A9=18.3
B1=4.8
B2=1
B3=1.2
B4=.6
TOTAL % POWER IS T%=95
TOTAL POWER OF FIRST 8 BANDS=105444.046
NEW % FOR THE FIRST 8 BANDS ARE:
13.0419075 19.3569364
25.3973988 8.9234104
4.94219653 4.39306358
3.15751445 15.7875723

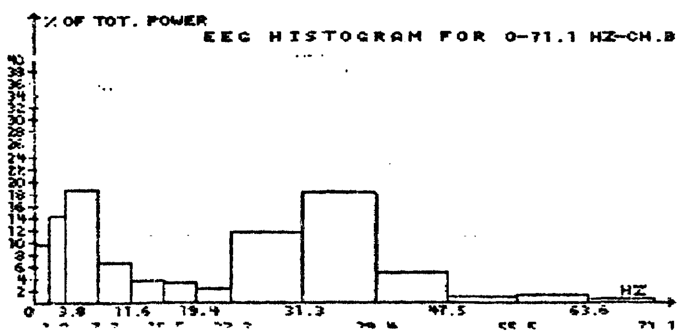


Figure 3.6.1.b. Halazepam drug experiment - new system, monkey 461

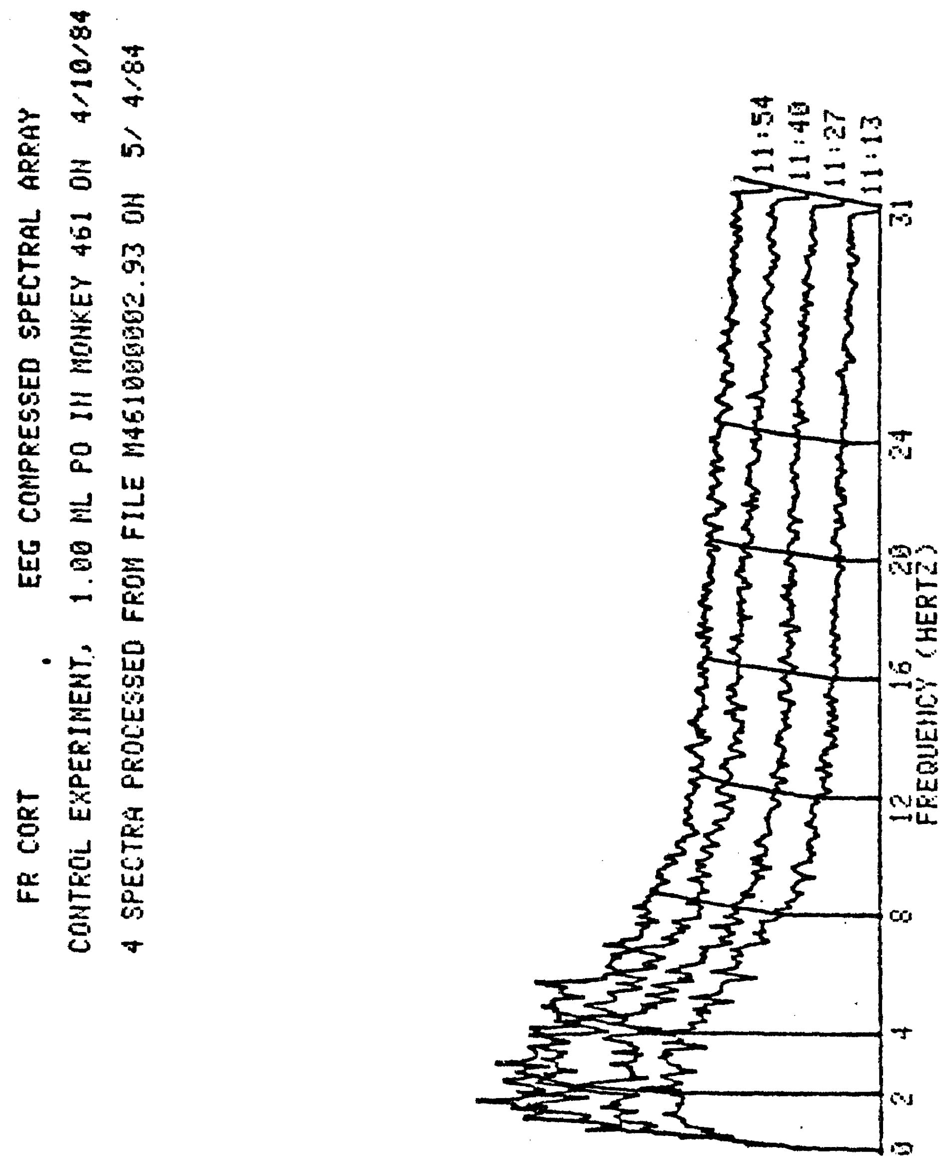


Figure 3.6.2.a. Halazepam control experiment - old system, monkey 461

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

CONTROL EXPERIMENT, 1.00 ML, IN MONKEY 461. EXPT PERFORMED ON 4/10/84

BAND:		1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	CLOCK END TIME	0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
		1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25
		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	11:13	3690	4512	6197	3100	2134	1854	1658	2346	25491.
2	11:27	4345	4693	6457	3417	2544	2325	2038	2863	28682.
3	11:40	4330	5228	6603	4009	3098	2836	2316	3024	31944.
4	11:54	4828	4835	6608	3813	3021	2758	2351	3091	31305.
		PERCENT FOR EACH SPECTRUM BAND								
1		14.48	17.70	24.31	12.16	8.37	7.27	6.50	9.20	
2		15.15	16.36	22.51	11.91	8.87	8.11	7.11	9.98	
3		15.12	16.37	20.67	12.55	9.70	8.88	7.25	9.47	
4		15.42	15.44	21.11	12.18	9.65	8.81	7.51	9.87	
		SUM OF SPECTRA - PERCENTAGE OF TOTAL								
1 AND 2		14.83	16.99	23.36	12.03	8.64	7.71	6.82	9.62	
3 AND 4		15.27	15.91	20.89	12.37	9.67	8.84	7.38	9.67	
5 AND 6		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

NOVA3:M461000002.93

Figure 3.6.2.b. Halazepam control results - old system, monkey 461

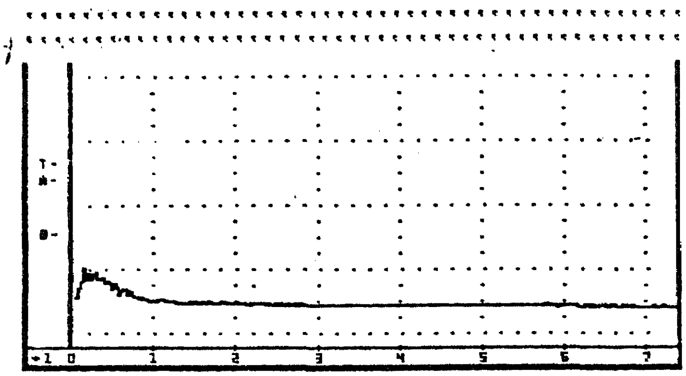
SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

RO 3-6728 AT 10.00 MG/KG, IN MONKEY 461. EXPT PERFORMED ON 4/11/84

#	BAND:	1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	CLOCK END TIME	0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
		1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25
		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	11:13	3401	4370	6057	3224	2202	1985	2177	5222	28638.
2	11:27	3479	4244	5864	3075	2184	2124	2226	5478	28674.
3	11:40	3360	4238	5647	3326	2400	2363	2423	5763	29520.
4	11:54	3778	4518	6096	3403	2463	2377	2392	5710	30737.
		PERCENT FOR EACH SPECTRUM BAND								
1		11.88	15.26	21.15	11.26	7.69	6.93	7.60	18.23	
2		12.13	14.80	20.45	10.72	7.62	7.41	7.76	19.10	
3		11.38	14.36	19.13	11.27	8.13	8.08	8.21	19.52	
4		12.29	14.70	19.83	11.07	8.01	7.73	7.78	18.58	
		SUM OF SPECTRA - PERCENTAGE OF TOTAL								
1 AND 2		12.00	15.03	20.80	10.99	7.65	7.17	7.68	18.67	
3 AND 4		11.85	14.53	19.49	11.17	8.07	7.87	7.99	19.04	
5 AND 6		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	

NOVA3:M461086728.11

Figure 3.6.2.c. Halazepam drug results - old system, monkey 461



HALAZEPAM - CONTROL
8-6728
M680, ch. A

IV

```
A  4.00 VOLT/DIV      B  .400 VOLT/DIV
TRACE  A        TRIGGER   A  >   START
TIME  000.200 000    DISPLAY > 00001 + 0
   SELECT NUMBER OF POINTS  0512

TOTAL POWER=18673
% OF FIRST BAND=26.3053607
% OF SECOND BAND=34.0973598
% OF THIRD BAND=24.206073
% OF FOURTH BAND=4.19857548
% OF FIFTH BAND=1.92791731
% OF SIXTH BAND=2.01895785
% OF SEVENTH BAND=1.11926311
% OF EIGHTH BAND=1.65479569
% OF NINTH BAND=.824720184
% OF TENTH BAND=.856852139
% OF ELEVENTH BAND=.701547689
% OF TWELTH BAND=.519466609
% OF THERTEENTH BAND=.334172334
TOTAL % BEFORE ROUND OFF IS T%=98.7650619
% POWER OF THE THIRTEEN BANDS ARE:
A1=26.3
A2=34.1
A3=24.2
A4=4.2
A5=1.9
A6=2
A7=1.1
A8=1.6
A9=.8
B1=.8
B2=.7
B3=.5
B4=.3
TOTAL % POWER IS T%=98
TOTAL POWER OF FIRST 8 BANDS=18177.5939
NEW % FOR THE FIRST 8 BANDS ARE:
27.0167715       35.0293501
24.8595388       4.31446541
1.95178197       2.05450734
1.12997904       1.64360587
```

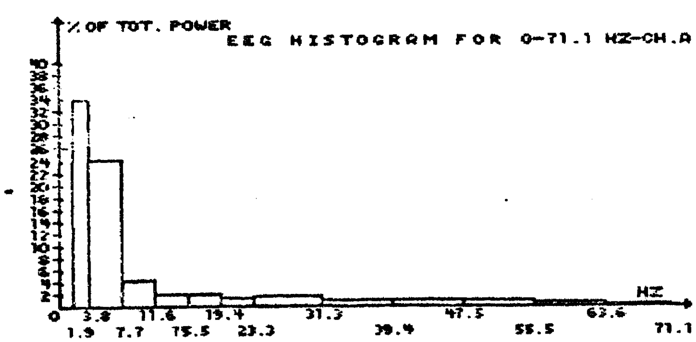
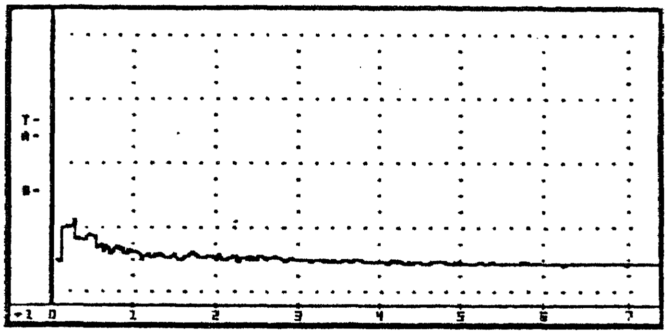


Figure 3.7.1.a. Halazepam control experiment - new system, monkey 680

,AVERAGEA—64



HALAZEPAM —DRUG
8-6728
MG80, ch. A

IV

A 4.00 VOLT/DIV B .400 VOLT/DIV
TRACE A TRIGGER A > START
TIME 000.200 000 DISPLAY > 00001 + 0
SELECT NUMBER OF POINTS 0512

TOTAL POWER=30874
% OF FIRST BAND=20.2144199
% OF SECOND BAND=21.7658872
% OF THIRD BAND=23.2266632
% OF FOURTH BAND=7.18403836
% OF FIFTH BAND=4.0519531
% OF SIXTH BAND=4.60905616
% OF SEVENTH BAND=2.84057783
% OF EIGHTH BAND=5.41556002
% OF NINTH BAND=2.39683877
% OF TENTH BAND=1.52879446
% OF ELEVENTH BAND=.929584764
% OF TWELTH BAND=.663989117
% OF THERTEENTH BAND=.526332837
TOTAL % BEFORE ROUND OFF IS T%=95.3536957
% POWER OF THE THIRTEEN BANDS ARE:
A1=20.2
A2=21.7
A3=23.2
A4=7.1
A5=4
A6=4.6
A7=2.8
A8=5.4
A9=2.4
B1=1.5
B2=.9
B3=.6
B4=.5
TOTAL % POWER IS T%=94
TOTAL POWER OF FIRST 8 BANDS=29231.766
NEW % FOR THE FIRST 8 BANDS ARE:

21.3348315	22.9191011
24.5033708	7.4988764
4.2247191	4.85842697
2.95730337	5.70337079

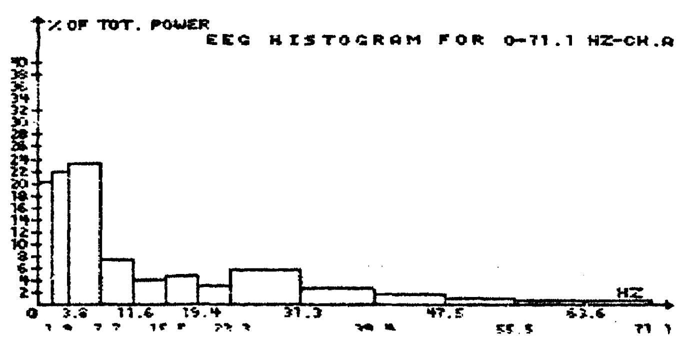


Figure 3.7.1.b. Halazepam drug experiment - new system, monkey 680

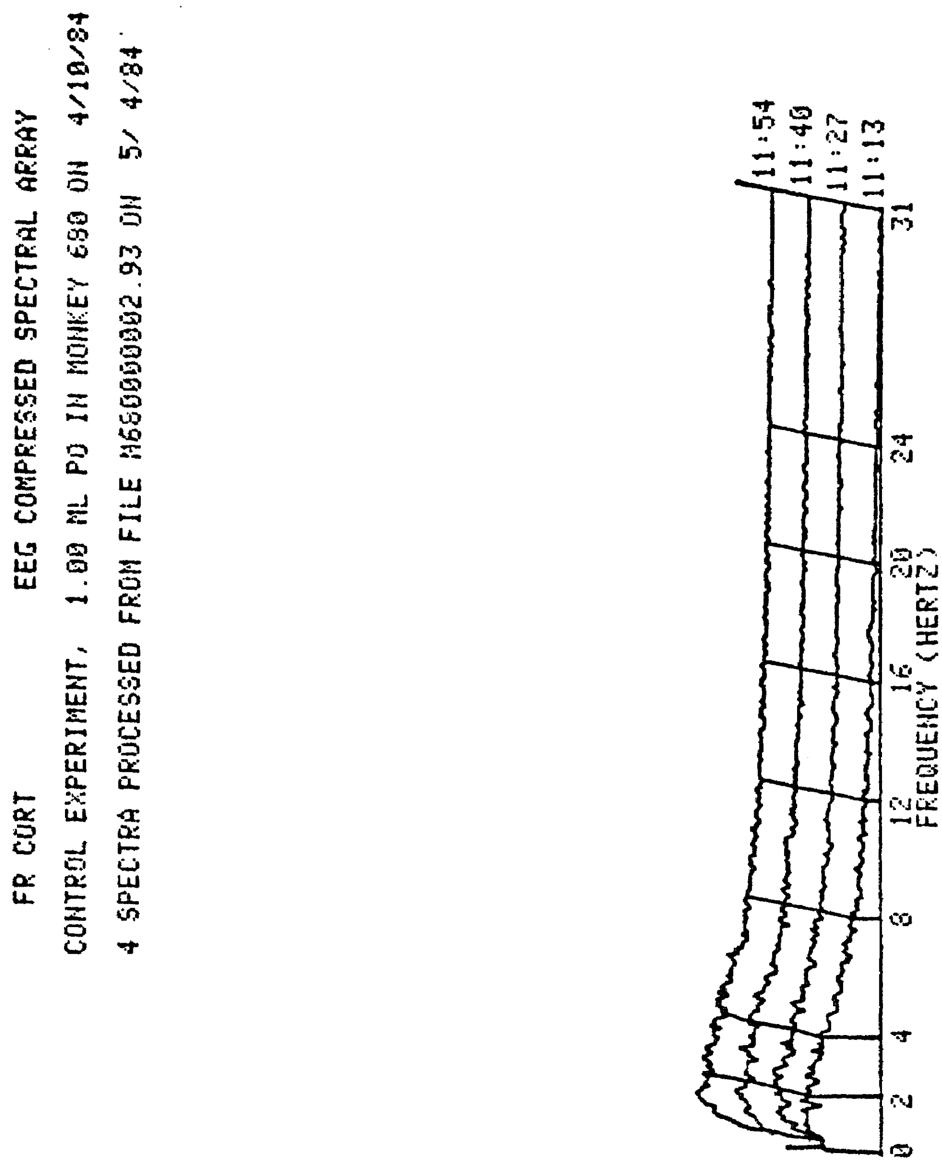


Figure 3.7.2.a. Halazepam control experiment - old system, monkey 680

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

CONTROL EXPERIMENT, 1.00 ML. IN MONKEY 680. EXPT PERFORMED ON 4/10/84

#	CLOCK END TIME	BAND: 1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND ->								
		0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
		1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25

		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS) ->								TOTAL
1	11:13**	1414	1426	1640	794	482	324	194	146	6420.
2	11:27**	1034	1282	1550	722	400	280	192	174	5634.
3	11:40**	1130	1230	1434	756	408	230	142	108	5438.
4	11:54**	1180	1254	1478	680	374	202	114	66	5348.

PERCENT FOR EACH SPECTRUM BAND

1	22.02	22.21	25.55	12.37	7.51	5.05	3.02	2.27
2	19.07	22.55	27.27	12.70	7.04	4.93	3.38	3.06
3	20.59	22.41	27.04	13.78	7.43	4.19	2.59	1.97
4	22.06	23.45	27.64	12.72	6.99	3.78	2.13	1.23

SUM OF SPECTRA - PERCENTAGE OF TOTAL

1 AND 2	20.64	22.37	26.35	12.52	7.29	4.99	3.19	2.64
3 AND 4	21.32	22.92	27.33	13.25	7.22	3.99	2.36	1.61
5 AND 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0 00

WARNING - SPECTRAL VALUES PRECEDED BY A ** MAY BE INVALID

NOVA3:M680000002.93

Figure 3.7.2.b. Halazepam control results - old system, monkey 680

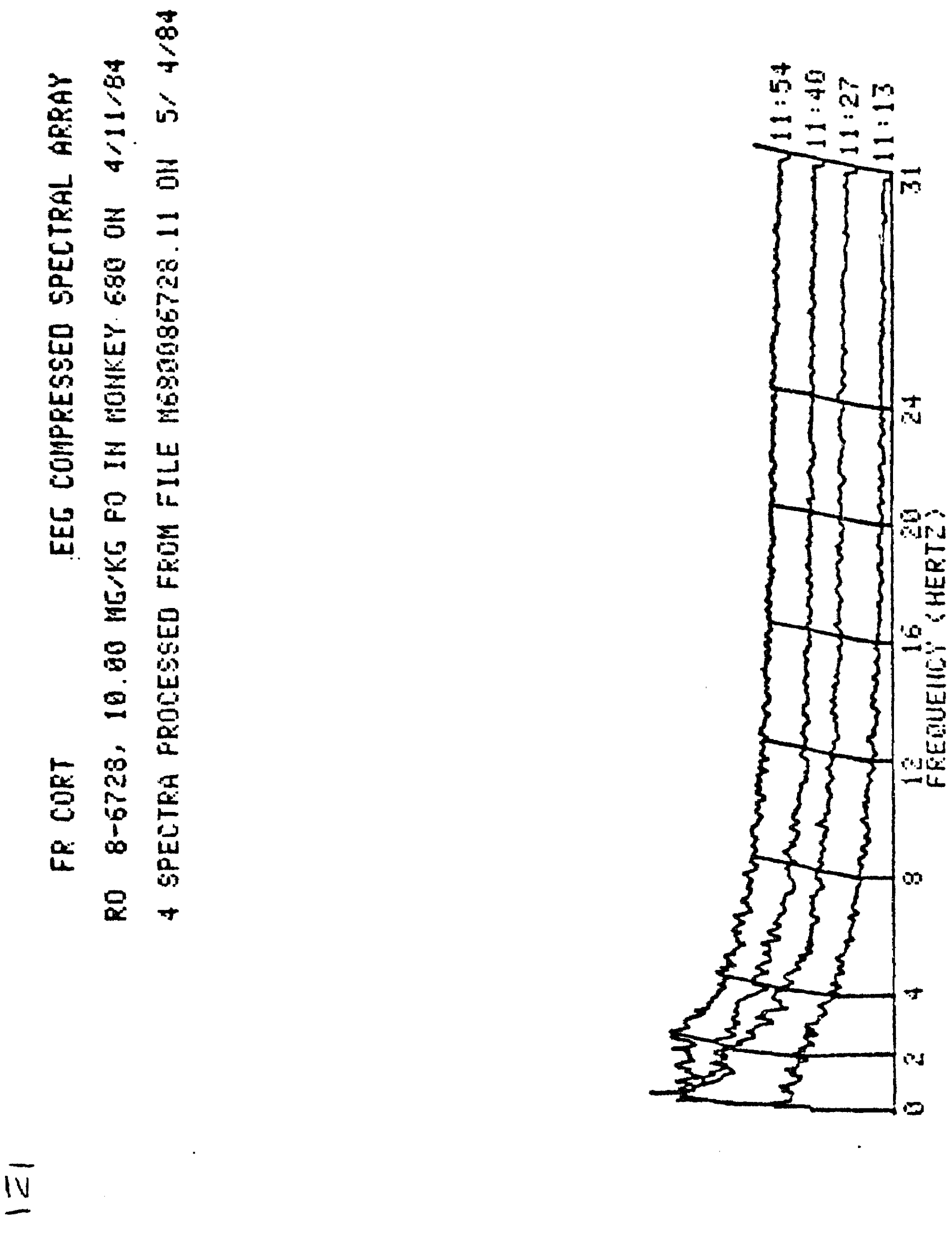


Figure 3.7.2.c. Halazepam drug experiment - old system, monkey 680

|2|

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH
EEG FROM THE FR CORT

RO 8-6728 AT 10.00 MG/KG, IN MONKEY 680. EXPT PERFORMED ON 4/11/84

BAND:		1	2	3	4	5	6	7	8	TOTAL
		<- UPPER AND LOWER FREQUENCY LIMIT OF EACH BAND							->	
#	CLOCK END TIME	0.00	1.99	4.02	8.01	11.99	15.98	20.04	24.02	0.00
		1.99	4.02	8.01	11.99	15.98	20.04	24.02	31.25	31.25

		<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)							->	TOTAL
1	11:13	2224	1809	2026	1150	750	520	452	712	9643.
2	11:27	3296	2199	2283	1465	986	851	769	1066	12825.
3	11:40	2433	1933	2112	1169	817	660	556	853	10533.
4	11:54	2346	1942	2178	1411	1045	868	736	1013	11539.

PERCENT FOR EACH SPECTRUM BAND

1	23.06	18.76	21.01	11.93	7.78	5.39	4.69	7.38
2	25.00	17.15	17.80	11.42	7.69	6.64	6.00	8.31
3	23.10	18.35	20.05	11.10	7.76	6.27	5.28	8.10
4	20.33	16.83	18.88	12.23	9.06	7.52	6.38	8.78

SUM OF SPECTRA - PERCENTAGE OF TOTAL

1 AND 2	24.17	17.84	19.18	11.64	7.73	6.10	5.43	7.91
3 AND 4	21.65	17.56	19.44	11.69	8.44	6.92	5.85	8.45
5 AND 6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

NOVA3:M680086728.11

Figure 3.7.2.d. Halazepam drug results - old system, monkey 680

APPENDIX A

Information About The System Components

How SuperRam Works in Relation to the Disk Operating System (DOS)

The DOS is available by booting DOS 3.3 disk. The message that comes on the screen when this disk is booted is:

"DOS VERSION 3.3 08/25/80

APPLE II PLUS OR ROMCARD SYSTEM MASTER

(LOADING INTEGER BASIC INTO LANGUAGE CARD)"

which means that the HELLO program of this disk loads INTEGER BASIC into the language card (or Super RAM card) automatically. An image of the Autostart ROM is also loaded into the Super RAM, providing reset-to-BASIC, stop-listing, full-escape cursor moves with whatever version of BASIC is loaded into this card.

Because our computer is an APPLE II plus (Revision 1 type of main board, Autostart ROM, and Applesoft BASIC in ROM on the main board), once the DOS 3.3 is booted, it acts as an Apple II with INTEGER BASIC firmware card, except that this version of INTEGER BASIC will operate with Autostart ROM.

DOS 3.3 software performs automatically the bank switching in order to expand the RAM from 48K up to 64K. The switching commands are available for the user to write programs that involve work with Super RAM II. These commands consist of reading once or twice the specified location, from BASIC with PEEK command and from M/L with LDA $XXXX. The specified locations and the effect of reading them are presented in Fig. 2.8 in this Appendix.

SWITCH LOCATION		EFFECT
Decimal	Hexadecimal	
NORMAL BANK $D000—$DFFF		
-16256 (40280)	$C080	*Select SuperRam II for reading only. Write-protect SuperRam II
-16255 (40281)	$C081	*Select main board ROMS. Two or more succesive reds will write-enable the Super-Ram II
-16254 (40282)	$ C082	Select main board ROMS. Write-protect Super Ram II
-16253 (40283)	$C083	*Select Super Ram II. Two or more succesive reads to this address write-enable Super-Ram II.
ALTERNATE BANK $D000—$DFFF		
-16248 (49288)	$C088	Select SuperRam II, write protect SuperRam II.
-16247 (49289)	$C089	Select main board ROMS. Two or more succesive reads will will write-enable SuperRam II
-16245 (49291)	$C08B	Select SuperRam II. Two or more succesive reads will write-enable SuperRam II.

* with $D000 — $DFFF normal bank mapped in.

Figure 2.8 SuperRam II Usage

The configuration of the Super RAM II does not include any circuit in F8 socket. The capacity of a socket is 2K. The place on F8 is empty because the monitor ROM is missing from the Super RAM card (the monitor occupies 2K and is placed between $F800 and $FFFF). But, an image of the Autostart ROM is automatically loaded into Super RAM II. However, sometimes the Old Monitor is needed because of the STEP and TRACE commands available. In this case, the circuit that contains the Autostart ROM, on the main board (socket F8) can be changed with an Old Monitor and its content can be saved as binary file on disk. Then, using a small BASIC program, the Old Monitor can be loaded into Super RAM II for use.

Table 17: ROM Organization and Usage

Page Number: Decimal	Hex	Used By:	
208	$D0	Programmer's Aid #1	Applesoft II BASIC
212	$D4	Programmer's Aid #1	Applesoft II BASIC
216	$D8		Applesoft II BASIC
220	$DC		Applesoft II BASIC
224	$E0	Integer BASIC	Applesoft II BASIC
228	$E4	Integer BASIC	Applesoft II BASIC
232	$E8	Integer BASIC	Applesoft II BASIC
236	$EC	Integer BASIC	Applesoft II BASIC
240	$F0	Integer BASIC	Applesoft II BASIC
244	$F4	Utility Subroutines	Applesoft II BASIC
248	$F8	Monitor ROM	Autostart ROM
252	$FC	Monitor ROM	Autostart ROM

Figure 2.4 ROM Memory Organization

System Memory Map

Page Number: Decimal	Hex	
0	$00	RAM (48K)
1	$01	
2	$02	
.	.	
.	.	
.	.	
190	$BE	
191	$BF	
192	$C0	I/O (2K)
193	$C1	
.	.	
.	.	
.	.	
198	$C6	
199	$C7	
200	$C8	I/O ROM (2K)
201	$C9	
.	.	
.	.	
.	.	
206	$CE	
207	$CF	
208	$D0	ROM (12K)
209	$D1	
.	.	
.	.	
.	.	
254	$FE	
255	$FF	

Figure 2.7 System Memory Map

Table 16: RAM Organization and Usage

Page Number: Decimal	Hex	Used For:	
0	$00	System Programs	
1	$01	System Stack	
2	$02	GETLN Input Buffer	
3	$03	Monitor Vector Locations	
4 through 7	$04–$07	Text and Lo-Res Graphics Primary Page Storage	
8 through 11	$08–$0B	Text and Lo-Res Graphics Secondary Page Storage	FREE RAM
12 through 31	$0C–$1F		FREE RAM
32 through 63	$20–$3F	Hi-Res Graphics Primary Page Storage	FREE RAM
64 through 95	$40–$5F	Hi-Res Graphics Secondary Page Storage	FREE RAM
96 through 191	$60–$BF		FREE RAM

Figure 2.6 RAM Memory Organization

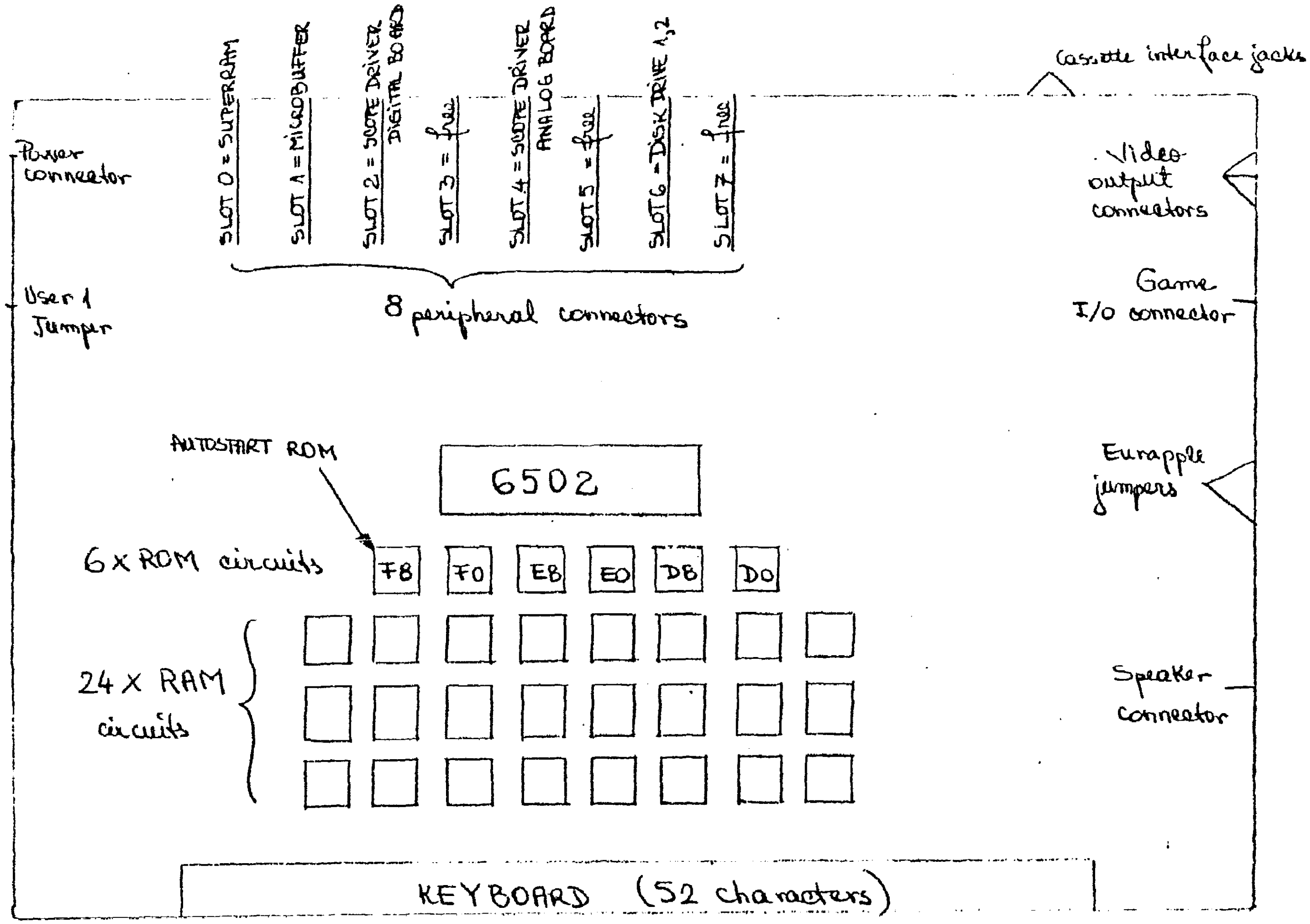
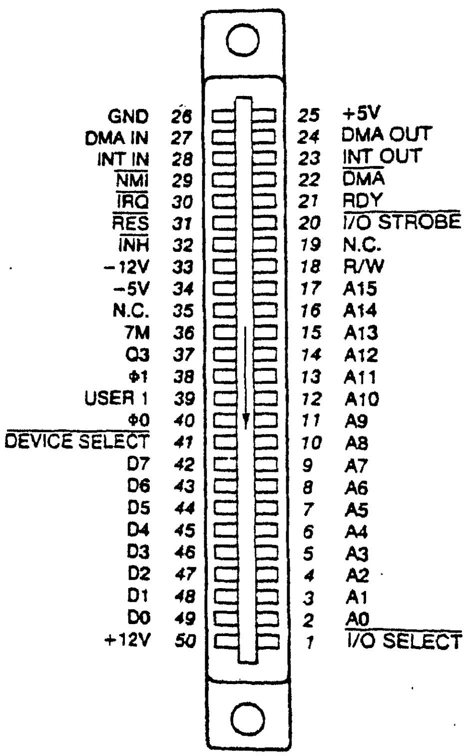


Figure 2.2 The Apple Main Board



Peripheral connector pinout

Table 33: Peripheral Connector Signal Description

Pin:	Name:	Description:
1	$\overline{\text{I/O SELECT}}$	This line, normally high, will become low when the microprocessor references page \$C*n*, where *n* is the individual slot number. This signal becomes active during $\Phi 0$ and will drive 10 LSTTL loads*. This signal is not present on peripheral connector 0.
2-17	A0-A15	The buffered address bus. The address on these lines becomes valid during $\Phi 1$ and remains valid through $\Phi 0$. These lines will each drive 5 LSTTL loads*.
18	R/$\overline{\text{W}}$	Buffered Read/$\overline{\text{Write}}$ signal. This becomes valid at the same time the address bus does, and goes high during a read cycle and low during a write. This line can drive up to 2 LSTTL loads*.
19	SYNC	On peripheral connector 7 *only*, this pin is connected to the video timing generator's SYNC signal.
20	$\overline{\text{I/O STROBE}}$	This line goes low during $\Phi 0$ when the address bus contains an address between \$C800 and \$CFFF. This line will drive 4 LSTTL loads*.
21	RDY	The 6502's RDY input. Pulling this line low during $\Phi 1$ will halt the microprocessor, with the address bus holding the address of the current location being fetched.
22	$\overline{\text{DMA}}$	Pulling this line low disables the 6502's address bus and halts the microprocessor. This line is held high by a 3KΩ resistor to +5v.
23	INT OUT	Daisy-chained interrupt output to lower priority devices. This pin is usually connected to pin 28 (INT IN).
24	DMA OUT	Daisy-chained DMA output to lower priority devices. This pin is usually connected to pin 22 (DMA IN).
25	+5v	+5 volt power supply. 500mA current is available for *all* peripheral cards.
26	GND	System electrical ground.

Table 33 (cont'd): Peripheral Connector Signal Description

Pin:	Name:	Description:
27	DMA IN	Daisy-chained DMA input from higher priority devices. Usually connected to pin 24 (DMA OUT).
26	INT IN	Daisy-chained interrupt input from higher priority devices. Usually connected to pin 23 (INT OUT).
29	$\overline{\text{NMI}}$	Non-Maskable Interrupt. When this line is pulled low the Apple begins an interrupt cycle and jumps to the interrupt handling routine at location $3FB.
30	$\overline{\text{IRQ}}$	Interrupt ReQuest. When this line is pulled low the Apple begins an interrupt cycle only if the 65Ø2's I (Interrupt disable) flag is not set. If so, the 65Ø2 will jump to the interrupt handling subroutine whose address is stored in locations $3FE and $3FF.
31	$\overline{\text{RES}}$	When this line is pulled low the microprocessor begins a RESET cycle (see page 36).
32	$\overline{\text{INH}}$	When this line is pulled low, all ROMs on the Apple board are disabled. This line is held high by a 3KΩ resistor to +5v.
33	−12v	−12 volt power supply. Maxmum current is 200mA for all peripheral boards.
34	−5v	−5 volt power supply. Maximum current is 200mA for all peripheral boards.
35	COLOR REF	On peripheral connector 7 *only*, this pin is connected to the 3.5MHz COLOR REFerence signal of the video generator.
36	7M	7MHz clock. This line will drive 2 LSTTL loads*.
37	Q3	2MHz asymmetrical clock. This line will drive 2 LSTTL loads*.
38	Φ1	Microprocessor's phase one clock. This line will drive 2 LSTTL loads*.
39	USER 1	This line, when pulled low, disables *all* internal I/O address decoding**.
40	ΦØ	Microprocessor's phase zero clock. This line will drive 2 LSTTL loads*.
41	$\overline{\text{DEVICE SELECT}}$	This line becomes active (low) on each peripheral connector when the address bus is holding an address between $CØnØ and $CØnF, where *n* is the slot number plus $8. This line will drive 10 LSTTL loads*.
42-49	DØ-D7	Buffered bidirectional data bus. The data on this line becomes valid 300nS into ΦØ on a write cycle, and should be stable no less than 100ns before the end of ΦØ on a read cycle. Each data line can drive one LSTTL load.
50	+12v	+12 volt power supply. This can supply up to 250mA total for all peripheral cards.

Table 28: Auxiliary Video Output Connector Signal Descriptions

Pin	Name	Description
1	GROUND	System common ground; 0 volts.
2	VIDEO	NTSC compatible positive composite video. Black level is about .75 volt, white level about 2.0 volt, sync tip level is 0 volts. Output level is not adjustable. This is not protected against short circuits.
3	+12v	+12 volt power supply.
4	−5v	−5 volt line from power supply.

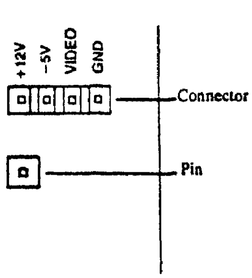


Figure 15. Auxiliary Video Output Connector and Pin.

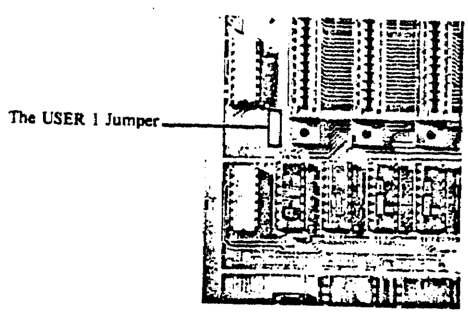


Photo 8. The USER 1 Jumper.

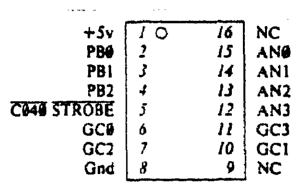


Figure 16.
Game I/O Connector Pinouts

Table 29: Game I/O Connector Signal Descriptions

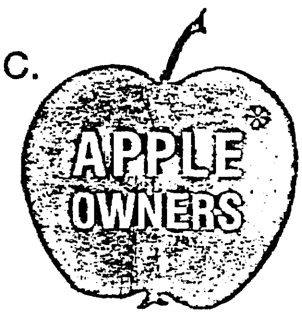
Pin:	Name:	Description:
1	+5v	+5 volt power supply. Total current drain on this pin must be less than 100mA.
2-4	PB0-PB2	Single-bit (Pushbutton) inputs. These are standard 74LS series TTL inputs.
5	C040 STROBE	A general-purpose strobe. This line, normally high, goes low during Φ0 of a read or write cycle to any address from $C040 through $C04F. This is a standard 74LS TTL output.
6,7,10,11	GC0-GC3	Game controller inputs. These should each be connected through a 150K Ohm variable resistor to +5v.
8	Gnd	System electrical ground.
12-15	AN0-AN3	Annunciator outputs. These are standard 74LS series TTL outputs and must be buffered if used to drive other than TTL

Roy Hicks, Owner
DESIGNER/ELECTRONIC ENGINEER

R.H. ELECTRONICS INC.

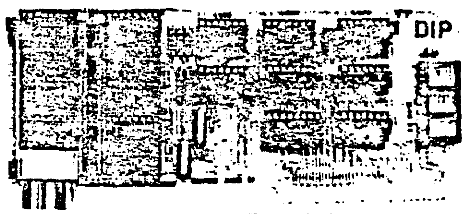
COMPUTER PRODUCTS

MANUFACTURING



SUPER RAM ◦ II™

FOR YOUR APPLE II COMPUTER*



- PLUG IN SLOT Ø
- GOLD PLATED CONTACTS
- INCLUDES 5 RAM-ROM OPTIONS
- THIS IS SOPHISTICATED FIRMWARE
- ENJOY THE BEST OF BOTH WORLDS
- 16K RAM (RANDOM ACCESS MEMORY)
- INCLUDES SELECTABLE DIP SWITCH
- EXPANDS YOUR 48K APPLE TO 64K OF PROGRAMMABLE MEMORY
- ELIMINATES THE NEED FOR APPLESOFT* OR INTEGER BASIC ROM CARD
- ALLOWS YOU TO RUN APPLE'S NEW FORTRAN PACKAGE ALSO PASCAL AND PILOT—CP/M® - COBOL - INTEGER/BASIC - APPLESOFT/BASIC - VISICALC - DOS 3.3
- KEYBOARD CONTROL SELECTION OF RAM OR MOTHER BOARD ROM LANGUAGE
- INCLUDES: INSTALLATION INSTRUCTIONS AND APPLICATIONS NOTES
- THE SOFTWARE DEVELOPED BY VARIOUS VENDORS FOR YOUR (64K) SHOULD NOW WORK AS THEY ADVERTISED
- THE MOST VERSATILE RAM EXPANSION ON THE MARKET TODAY
- ROM SOCKET LET'S YOU CREATE YOUR OWN SPECIAL USES—DESIGN YOUR OWN SOFTWARE PROGRAMS
- LOCKS IN SOFTWARE PROGRAMS SO THEY CAN'T BE COPIED

UNIQUE 1 YEAR WARRANTY!

"When a better product is made, we'll be the ones to make it!"

Table 9: Annunciator Special Locations				
Ann.	State	Address: Decimal		Hex
Ø	off	4924Ø	-16296	SCØ58
	on	49241	-16295	SCØ59
1	off	49242	-16294	SCØ5A
	on	49243	-16293	SCØ5B
2	off	49244	-16292	SCØ5C
	on	49245	-16291	SCØ5D
3	off	49246	-1629Ø	SCØ5E
	on	49247	-16289	SCØ5F

APPENDIX B

Programming

Versions of the Applesoft BASIC

Applesoft BASIC is available in two versions:

a. Firmware Applesoft

Comes with the Applesoft in ROM (in the ROM memory on the main board) or on a Firmware Applesoft card which has to be plugged in slot 0.

If the system works with Firmware Applesoft card installed in slot 0, then there is a possibility to choose between languages using a switch on this board. Practically there is a language choice, and at the same time, a choice of the ROMs on Firmware card for Applesoft, or, on main board for INTEGER BASIC.

To put the computer in Applesoft, the switch has to be in upward position, RESET, CTRL B keys have to be pressed and the prompt character "]" will appear. To put the computer in INTEGER BASIC, the switch has to be in downward position.

RESET and CTRL B keys have to be pressed and the prompt character ">" will appear.

The system can also work with Applesoft and INTEGER BASIC without switching all the time between them. There is another card, a language card, that has to be installed in slot 0 and can be loaded, whenever necessary, with INTEGER BASIC.

The Applesoft can be stored in ROM. The language card has another function also.

It can extend the RAM capacity from 48K to 64K (see RAMs paragraph). The conclusions can be summarized in the following table:

	Firmware card in Slot 0	Language Card in Slot 0
Applesoft BASIC	Switch UP (The card becomes an Applesoft card)	in ROM
Integer BASIC	Switch DOWN (The card becomes an Integer BASIC card)	Loaded Automatically by DOS in the language card

b. Cassette Tape Applesoft

The Applesoft BASIC is loaded, whenever is needed, from a cassette tape.

The system used is a firmware Applesoft version and has the Applesoft BASIC in the ROM circuits on the main board, being available when the computer is turned on. The INTEGER BASIC is loaded, when necessary, from a disk to the language card, placed in slot 0.

The most important advantage of using firmware Applesoft is that it is placed in ROM so that the entire RAM capacity can be used. On the contrary, using the cassette tape version, each time the Applesoft is needed, it has to be loaded from the cassette tape, spending tie and about 10K of RAM, which means that, in computers with small memory, many of the features are not available because the correspondent locations in memory are erased when the content of the tape is loaded.

The memory map given in Fig. 2.5 in this Appendix makes it easier to understand the difference between the two version of Applesoft.

Appendix I: Memory Map

MEMORY RANGE	DESCRIPTION
Ø.1FF	Program work space; not available to user.
2ØØ.2FF	Keyboard character buffer.
3ØØ.3FF	Available to user for short machine language programs.
4ØØ.7FF	Screen display area for page 1 text or color graphics.
8ØØ.2FFF	In cassette tape version, the APPLESOFT BASIC intepreter.
8ØØ.XXX	If firmware APPLESOFT (Part number A2BØØØ9X) installed, user program and variable space, where XXX is maximum RAM memory to be used by APPLESOFT. This is either total system RAM memory, or less if the user is reserving part of high memory for machine language routines or high-resolution screen buffers.
2ØØØ.3FFF	Firmware APPLESOFT only: high-resolution graphics display page 1.
3ØØØ.XXX	Cassette tape APPLESOFT II; user program and variables where XXX is maximum available RAM memory to be used by APPLESOFT. This is either total system RAM memory, or less if the user is reserving part of high memory for machine language routines or page 2 high-resolution graphics.
4ØØØ.5FFF	High-resolution graphics display page 2.
CØØØ.CFFF	Hardware I/O Addresses.
DØØØ.DFFF	Future ROM expansion.
DØØØ.F7FF	APPLESOFT II firmware version, with select switch "ON" (up).
EØØØ.F7FF	APPLE Integer BASIC.
F8ØØ.FFFF	APPLE System Monitor.

Figure 2.5 Applesoft Versions

6502 MICROPROCESSOR INSTRUCTIONS

ADC	Add Memory to Accumulator with Carry
AND	"AND" Memory with Accumulator
ASL	Shift Left One Bit (Memory or Accumulator)
BCC	Branch on Carry Clear
BCS	Branch on Carry Set
BEQ	Branch on Result Zero
BIT	Test Bits in Memory with Accumulator
BMI	Branch on Result Minus
BNE	Branch on Result not Zero
BPL	Branch on Result Plus
BRK	Force Break
BVC	Branch on Overflow Clear
BVS	Branch on Overflow Set
CLC	Clear Carry Flag
CLD	Clear Decimal Mode
CLI	Clear Interrupt Disable Bit
CLV	Clear Overflow Flag
CMP	Compare Memory and Accumulator
CPX	Compare Memory and Index X
CPY	Compare Memory and Index Y
DEC	Decrement Memory by One
DEX	Decrement Index X by One
DEY	Decrement Index Y by One
EOR	"Exclusive-Or" Memory with Accumulator
INC	Increment Memory by One
INX	Increment Index X by One
INY	Increment Index Y by One
JMP	Jump to New Location
JSR	Jump to New Location Saving Return Address
LDA	Load Accumulator with Memory
LDX	Load Index X with Memory
LDY	Load Index Y with Memory
LSR	Shift Right one Bit (Memory or Accumulator)
NOP	No Operation
ORA	"OR" Memory with Accumulator
PHA	Push Accumulator on Stack
PHP	Push Processor Status on Stack
PLA	Pull Accumulator from Stack
PLP	Pull Processor Status from Stack
ROL	Rotate One Bit Left (Memory or Accumulator)
ROR	Rotate One Bit Right (Memory or Accumulator)
RTI	Return from Interrupt
RTS	Return from Subroutine
SBC	Subtract Memory from Accumulator with Borrow
SEC	Set Carry Flag
SED	Set Decimal Mode
SEI	Set Interrupt Disable Status
STA	Store Accumulator in Memory
STX	Store Index X in Memory
STY	Store Index Y in Memory
TAX	Transfer Accumulator to Index X
TAY	Transfer Accumulator to Index Y
TSX	Transfer Stack Pointer to Index X
TXA	Transfer Index X to Accumulator
TXS	Transfer Index X to Stack Pointer
TYA	Transfer Index Y to Accumulator

HEX OPERATION CODES

00 — BRK
01 — ORA — (Indirect, X)
02 — NOP
03 — NOP
04 — NOP
05 — ORA — Zero Page
06 — ASL — Zero Page
07 — NOP
08 — PHP
09 — ORA — Immediate
0A — ASL — Accumulator
0B — NOP
0C — NOP
0D — ORA — Absolute
0E — ASL — Absolute
0F — NOP
10 — BPL
11 — ORA — (Indirect), Y
12 — NOP
13 — NOP
14 — NOP
15 — ORA — Zero Page, X
16 — ASL — Zero Page, X
17 — NOP
18 — CLC
19 — ORA — Absolute, Y
1A — NOP
1B — NOP
1C — NOP
1D — ORA — Absolute, X
1E — ASL — Absolute, X
1F — NOP
20 — JSR
21 — AND — (Indirect, X)
22 — NOP
23 — NOP
24 — BIT — Zero Page
25 — AND — Zero Page
26 — ROL — Zero Page
27 — NOP
28 — PLP
29 — AND — Immediate
2A — ROL — Accumulator
2B — NOP
2C — BIT — Absolute
2D — AND — Absolute
2E — ROL — Absolute
2F — NOP
30 — BMI
31 — AND — (Indirect), Y
32 — NOP
33 — NOP
34 — NOP
35 — AND — Zero Page, X
36 — ROL — Zero Page, X
37 — NOP
38 — SEC
39 — AND — Absolute, Y
3A — NOP
3B — NOP
3C — NOP
3D — AND — Absolute, X
3E — ROL — Absolute, X
3F — NOP
40 — RTI
41 — EOR — (Indirect, X)
42 — NOP
43 — NOP
44 — NOP
45 — EOR — Zero Page
46 — LSR — Zero Page
47 — NOP
48 — PHA
49 — EOR — Immediate
4A — LSR — Accumulator
4B — NOP
4C — JMP — Absolute
4D — EOR — Absolute
4E — LSR — Absolute
4F — NOP
50 — BVC
51 — EOR (Indirect), Y
52 — NOP
53 — NOP
54 — NOP
55 — EOR — Zero Page, X
56 — LSR — Zero Page, X
57 — NOP
58 — CLI
59 — EOR — Absolute, Y
5A — NOP
5B — NOP
5C — NOP
5D — EOR — Absolute, X
5E — LSR — Absolute, X
5F — NOP
60 — RTS
61 — ADC — (Indirect, X)
62 — NOP
63 — NOP
64 — NOP
65 — ADC — Zero Page
66 — ROR — Zero Page
67 — NOP
68 — PLA
69 — ADC — Immediate
6A — ROR — Accumulator
6B — NOP
6C — JMP — Indirect
6D — ADC — Absolute
6E — ROR — Absolute
6F — NOP
70 — BVS
71 — ADC — (Indirect), Y
72 — NOP
73 — NOP
74 — NOP
75 — ADC — Zero Page, X
76 — ROR — Zero Page, X
77 — NOP
78 — SEI
79 — ADC — Absolute, Y
7A — NOP
7B — NOP
7C — NOP
7D — ADC — Absolute, X NO
7E — ROR — Absolute, X NO
7F — NOP
80 — NOP
81 — STA — (Indirect, X)
82 — NOP
83 — NOP
84 — STY — Zero Page
85 — STA — Zero Page
86 — STX — Zero Page
87 — NOP
88 — DEY
89 — NOP
8A — TXA
8B — NOP
8C — STY — Absolute

Appendix M: Differences Between APPLESOFT and Integer BASIC

DIFFERENCES BETWEEN COMMANDS

These commands are available in APPLESOFT, but not in Integer BASIC:

ATN						
CHR$	COS					
DATA	DEF FN	DRAW				
EXP						
FLASH	FN	FRE				
GET						
HCOLOR	HGR	HGR2	HIMEM:	HOME	HPLOT	
INT	INVERSE					
LEFT$	LOG	LOMEM:				
MID$						
NORMAL						
ON...GOSUB		ON...GOTO		ONERR GOTO		
POS						
READ	RECALL	RESTORE	RESUME	RIGHT$	ROT	
SCALE	SHLOAD	SIN	SPC	SPEED	SQR	STOP
	STORE	STR$				
TAN						
USR						
VAL						
WAIT						
XDRAW						

These commands are available in Integer BASIC, but not in APPLESOFT:

AUTO	
DSP	
MAN	MOD

These are named differently in the languages:

Integer BASIC	APPLESOFT
CLR	CLEAR
CON	CONT
TAB	HTAB (Note: APPLESOFT also has a TAB)
GOTO X*1Ø+1ØØ	ON X GOTO 1ØØ, 11Ø, 12Ø
GOSUB X*1ØØ+1ØØØ	ON X GOSUB 1ØØØ, 11ØØ, 12ØØ
CALL -936	HOME (or CALL -936)
POKE 5Ø,127	INVERSE
POKE 5Ø,255	NORMAL
X	X% (% indicates integer variable)
#	<> <u>or</u> ><

APPENDIX C

The Apple Video Display, Screen Formats,

Other Input/Output Features

THE APPLE VIDEO DISPLAY

The Apple Video Display

Display type:	Memory mapped into system RAM
Display modes:	Text, Low-Resolution Graphics, High-Resolution Graphics
Text capacity:	960 characters (24 lines, 40 columns)
Character type:	5 × 7 dot matrix
Character set:	Upper case ASCII, 64 characters
Character modes:	Normal, Inverse, Flashing
Graphics capacity:	1,920 blocks (Low-Resolution) in a 40 by 48 array 53,760 dots (High-Resolution) in a 280 by 192 array
Number of colors:	16 (Low-Resolution Graphics) 6 (High-Resolution Graphics)

Figure 2.9

TEXT	24 lines, 40 columns The characters can be: numbers, special symbols, upper-case letters One character appears in a matrix of 7 dots high and 5 dots wide The space between lines and columns is 1 dot wide.
LOW REZ GRAPHICS LRG	1,920 colored squares = 40 (vertically) X 48 (horizontally) no. of colors = 16 There is no space between blocks.
HIGH REZ GRAPHICS HRG	192 (vertically) X 280 (horizontally) colored dots. One dot has the size of the dot used to build characters in TEXT mode. no. of colors = 6 for Revision 1 type of main board. (our case) 4 for Revision Ø type of main board.

Figure 2.10 Screen Formats Characteristics

Screen Formats

The user can set the computer for either one of the three screen formats: TEXT, LRG, HRG or can work with text mixed with low or high resolution graphics (when the bottom lines of the display will be occupied by text). The two graphics modes can not be used simultaneously.

The information displayed on the screen is stored in the RAM memory. One location of memory can hold the image of an object on the screen, where objects mains: character, 2 colored blocks or a 7 dot line. The TEXT and LRG modes need 1K of RAM to store the information, and they share this 1K area. HRG needs 8K of memory. The 1K and 8K memory areas are called "pages". Each mode has 2 pages, the primary and the secondary page. The purpose of having 2 pages is to be able to draw on one page while displaying the other and doing animation by flipping pages.

The memory map for the 3 screen formats is given in Figure 2.11.a. It is important to establish:

a. The mixed modes are available in the secondary page.

b. Mixing graphics modes is not possible.

c. Mixing 2 pages on the same screen is not possible.

By software, the user can switch on or off the screen format. The soft switch is activated by referencing (reading from or writing on) a special memory location. The operation is called "throw" the switch.

Figure 2.11.b. contains the 8 memory locations used to set the switches. They work in pairs of 2 in the following manner: when one memory location is referenced, the correspondent soft switch is "on" and its companion is "off".

In order to find the possible combinations for these switches we start with the fact that we can have text or graphics mode (1 or 0) (Figure 2.12).

1. For graphics mode we can have all graphics or mixed text and graphics (2 or 3) and for each of these situations, low or high resolution graphics (6 or 7). More than that, for each of the last cases, primary (4) or secondary (5) page can be used.

2. For text mode we can follow the same rule in order to have the maximum number of combinations that can be obtained with 4 two-way switches. But, not all the 8 combinations obtained for text mode are visible. Knowing that 3 is possible only in graphics mode (see Apple II reference manual, [7]), the last 4 combinations are useless. With respect to the first 4 combinations, 2 things have to be pointed out:

- 2 is not necessary any more because its companion (3) is not used.
- if it is text mode, we are not interested in low or high resolution graphics, so 6 and 7 are useless.

The conclusion is that for text mode we have only two combinations: 1,4 and 1,5. Totally there are 10 useful combinations (given in the table of Figure 2.11.c).

The switches can be manipulated from BASIC or M/L and to set a particular mode, the order in throwing the switches is not important. However, when one of the graphics mode has to be used, the last switch of 4 necessary to throw must be the TEXT/GRAPHICS switch, in order to make invisible all the mode changes and at the end to obtain the finished picture at once.

Table 4: Video Display Memory Ranges					
Screen	Page	Begins at: Hex	Decimal	Ends at:	
Text/Lo-Res	Primary	$400	1024	$7FF	2047
	Secondary	$800	2048	$BFF	3071
Hi-Res	Primary	$2000	8192	$3FFF	16383
	Secondary	$4000	16384	$5FFF	24575

a.

Table 5: Screen Soft Switches			
Location: Hex	Decimal		Description:
$C050	49232	-16304	Display a GRAPHICS mode.
$C051	49233	-16303	Display TEXT mode.
$C052	49234	-16302	Display all TEXT or GRAPHICS.
$C053	49235	-16301	Mix TEXT and a GRAPHICS mode.*
$C054	49236	-16300	Display the Primary page (Page 1).
$C055	49237	-16299	Display the Secondary page (Page 2).
$C056	49238	-16298	Display LO-RES GRAPHICS mode.*
$C057	49239	-16297	Display HI-RES GRAPHICS mode.*

b.

Table 6: Screen Mode Combinations					
Primary Page			Secondary Page		
Screen	Switches		Screen	Switches	
All Text	$C054	$C051	All Text	$C055	$C051
All Lo-Res Graphics	$C054 $C052	$C056 $C050	All Lo-Res Graphics	$C055 $C052	$C056 $C050
All Hi-Res Graphics	$C054 $C052	$C057 $C050	All Hi-Res Graphics	$C055 $C052	$C057 $C050
Mixed Text and Lo-Res	$C054 $C053	$C056 $C050	Mixed Text and Lo-Res	$C055 $C053	$C056 $C050
Mixed Text and Hi-Res	$C054 $C053	$C057 $C050	Mixed Text and Hi-Res	$C055 $C053	$C057 $C050

c.

Figure 2.11

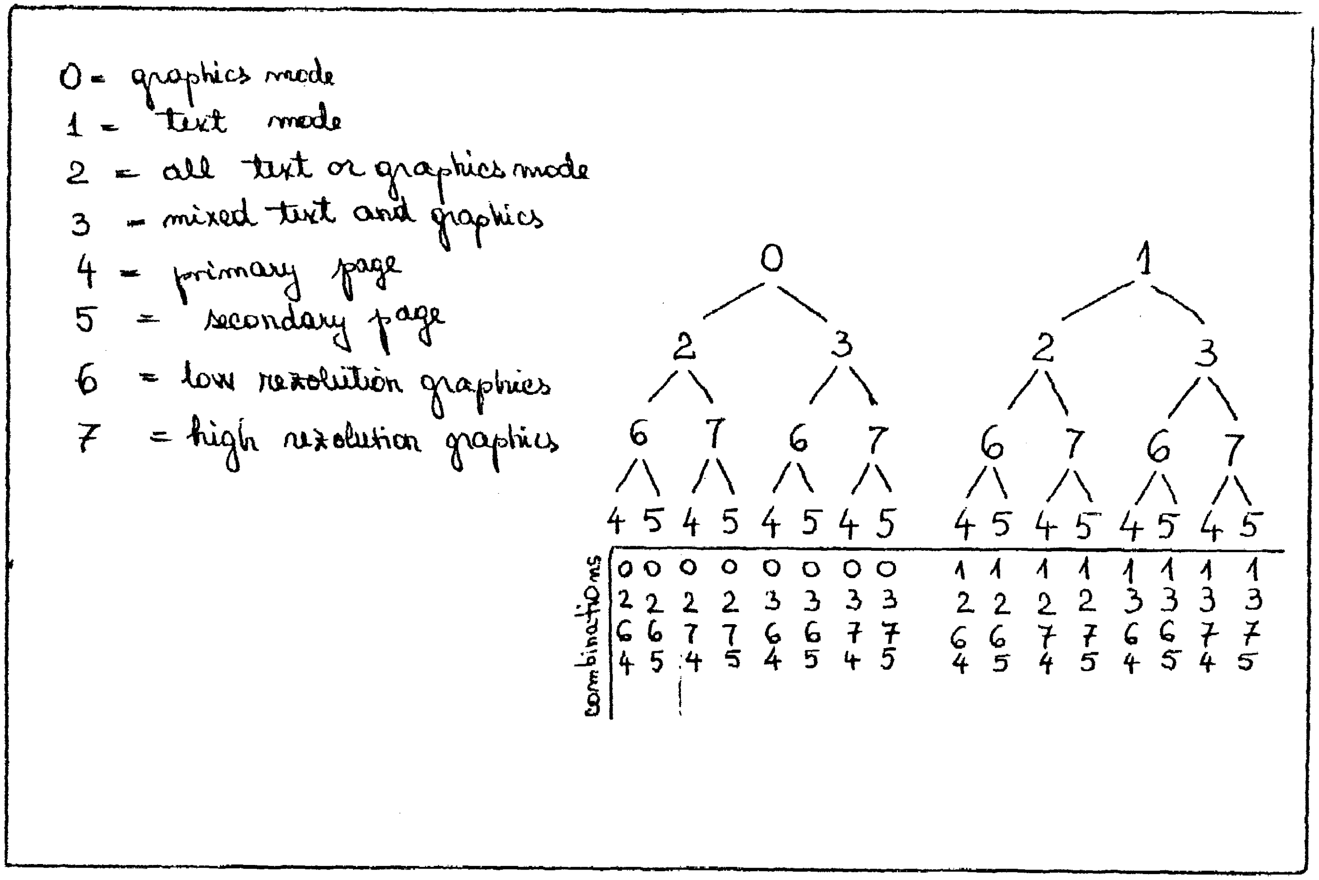


Figure 2.12 Deduction of the Combinations for the Screen Format Soft Switches

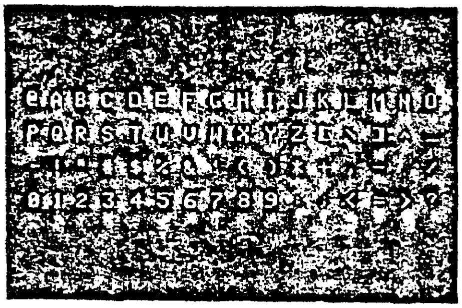


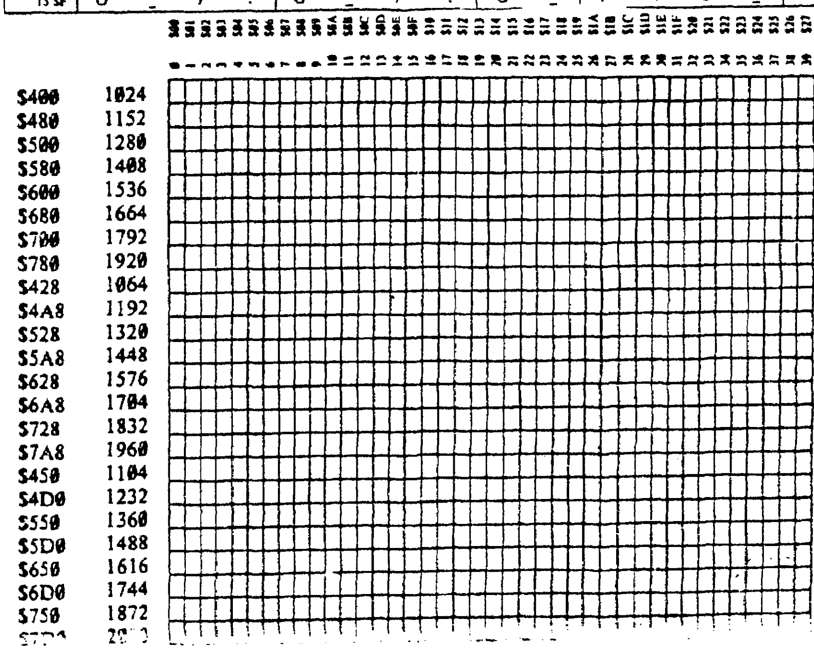
Photo 6. The Apple Character Set.

Table 8: Low-Resolution Graphics Colors

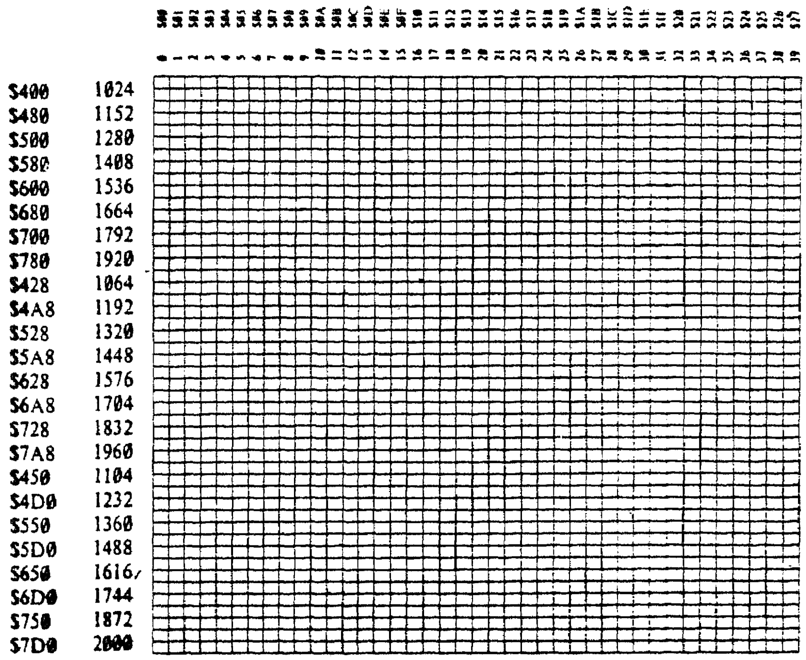
Decimal	Hex	Color	Decimal	Hex	Color
0	$0	Black	8	$8	Brown
1	$1	Magenta	9	$9	Orange
2	$2	Dark Blue	10	$A	Grey 2
3	$3	Purple	11	$B	Pink
4	$4	Dark Green	12	$C	Light Green
5	$5	Grey 1	13	$D	Yellow
6	$6	Medium Blue	14	$E	Aquamarine
7	$7	Light Blue	15	$F	White

Table 7: ASCII Screen Characters

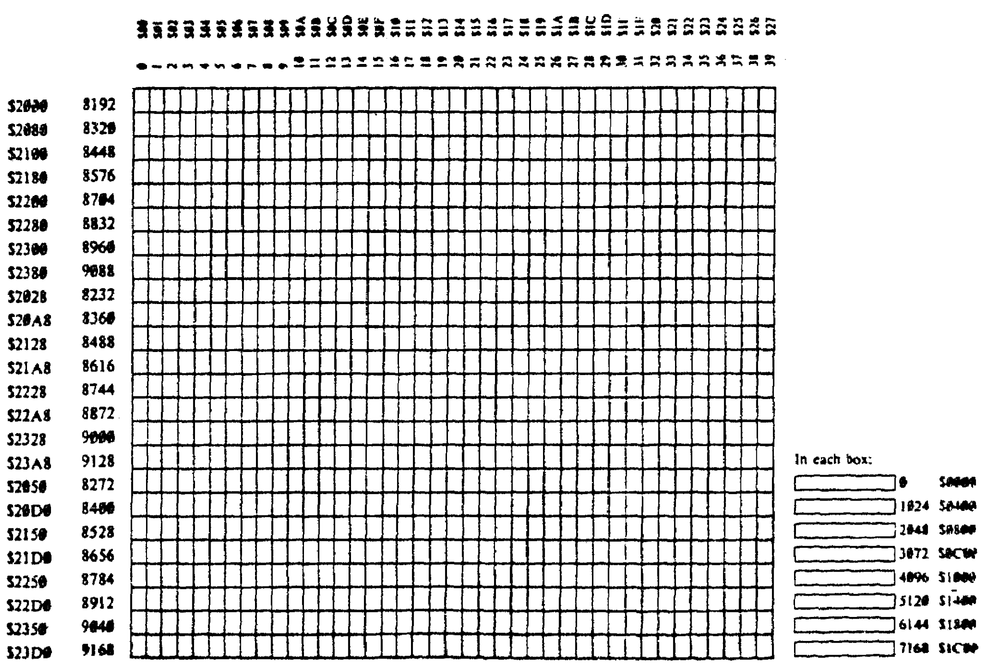
	Inverse				Flashing				Normal (Control)						(Lowercase)	
Decimal	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
Hex	$00	$10	$20	$30	$40	$50	$60	$70	$80	$90	$A0	$B0	$C0	$D0	$E0	$F0
0 $0	@	P		0	@	P		0	@	P		0	@	P		0
1 $1	A	Q	!	1	A	Q	!	1	A	Q	!	1	A	Q	!	1
2 $2	B	R	"	2	B	R	"	2	B	R	"	2	B	R	"	2
3 $3	C	S	#	3	C	S	#	3	C	S	#	3	C	S	#	3
4 $4	D	T	$	4	D	T	$	4	D	T	$	4	D	T	$	4
5 $5	E	U	%	5	E	U	%	5	E	U	%	5	E	U	%	5
6 $6	F	V	&	6	F	V	&	6	F	V	&	6	F	V	&	6
7 $7	G	W	'	7	G	W	'	7	G	W	'	7	G	W	'	7
8 $8	H	X	(	8	H	X	(	8	H	X	(	8	H	X	(	8
9 $9	I	Y	)	9	I	Y	)	9	I	Y	)	9	I	Y	)	9
10 $A	J	Z	*	:	J	Z	*	:	J	Z	*	:	J	Z	*	:
11 $B	K	[	+	;	K	[	+	;	K	[	+	;	K	[	+	;
12 $C	L	\	,	<	L	\	,	<	L	\	,	<	L	\	,	<
13 $D	M	]	–	=	M	]	–	=	M	]	–	=	M	]	–	=
14 $E	N	^	.	>	N	^	.	>	N	^	.	>	N	^	.	>
15 $F	O	_	/	?	O	_	/	?	O	_	/	?	O	_	/	?



Map of the Text Screen



Map of the Low-Resolution Graphics Mode



Map of the High-Resolution Graphics Mode

Other Input/Output Features

In addition to the video connector, the Apple II computer also has other input/output features. These are:

Inputs

1. The cassette input, used to "listen" to a cassette tape recording. In other words, to decode the tones on the tape into data and store them in memory.
2. Tree one-bit digital inputs which can be connected to whatever other electronic device or to a push-button. They are related to addresses 49249, 49250, 49251, ($C061, $C062, $C063).
3. Four analog inputs available to connect to 150KΩ variable resistors. Between each input and the +5V power supply, variable resistances will be created and will be used in timing circuits. (The variation in resistance will produce changes in the timing characteristics of the correspondent timing circuit).

   With M/L, the changes in timing loops can be sensed and, in conclusion, the position of the potentiometer at the analog input can be determined. The memory locations associated with these four inputs are: 49252-49255 ($C064-$C067) and 49264 ($C070) to reset the timing circuits.

Outputs:

1. Cassette Output, connected to a toggle soft switch on the Apple board (location 49184, $C020) and also to the microphone input of a cassette tape recorder. By referencing the soft switch location repeatedly, a tone can be produced and the pitch and duration of the tone can be controlled by software. The characteristics of the tone recorded on tape represent the encoded information.

2. The Speaker, controlled by a toggle soft switch related to address 49200 ($C030).

3. Four "Annunciator" Outputs which can be connected to circuits to drive speakers, relays, lamps. The way the annunciators soft switches work is described in Appendix A.

4. Utility Strobe Output, is called $C040 STROBE, it is normally +5V, but a program can control it to drop it 0V for 0.5 μsec.

Figure 2.13 gives the general view of the memory locations used for input/output.

Table 10: Input/Output Special Locations				
Function:	Address: Decimal		Hex	Read/Write
Speaker	49200	-16336	SC030	R
Cassette Out	49184	-16352	SC020	R
Cassette In	49256	-16288	SC060	R
Annunciators*	49240 through 49247	-16296 through -16289	SC058 through SC05F	R/W
Flag inputs	49249	-16287	SC061	R
	49250	-16286	SC062	R
	49251	-16285	SC063	R
Analog Inputs	49252	-16284	SC064	R
	49253	-16283	SC065	
	49254	-16282	SC066	
	49255	-16281	SC067	
Analog Clear	49264	-16272	SC070	R/W
Utility Strobe	49216	-16320	SC040	R

Figure 2.13 Input/Output Special Locations

With Respect to	Varieties of Apple	
System Monitor	Old Monitor	Autostart Monitor
Main Board	Revision 0 main board	Revision 1 main board
Operating Software	Applesoft and Integer BASIC in a firmware card (slot 0), switch used to get up one of them	Applesoft in ROM and Integer BASIC loaded whenever necessary into a language card (slot 0). Apple II Plus, the one that was used in the present application.

Figure 2.14 Apple II Varieties

APPENDIX D

The Applescope

COMMAND SUMMARY

COMMAND MODE

A 'space'	Trace channel A signal only
A B	Trace both channel A and channel B input signals
A S 'space'	Cycles the channel A voltage scale each time the 'space' bar is pressed
A Z 'space'	Sets the channel A zero voltage to the center of the display
A Z D	Decrements the channel A zero voltage each time the 'D' key is pressed
A Z U	Increments the channel A zero voltage each time the 'U' key is pressed
B 'space'	Trace channel B signal only
B S 'space'	Cycles the channel B voltage scale each time the 'space' bar is pressed
B Z 'space'	Sets the channel B zero voltage to the center of the display
B Z D	Decrements the channel B zero voltage each time the 'D' key is pressed
B Z U	Increments the channel B zero voltage each time the 'U' key is pressed
C	CONTINUOUS - Continuously acquire data until stopped (sweep rates slower than 100 msec/div. o
D N	Enters the DMA mode (Sweep rates slower than 1 msec/div. only)
D F	Exits from the DMA mode
S	SINGLE SWEEP - Acquire a single sweep when triggered
T I	Enters the time adjustment mode
T 'space'	Sets the trigger threshold voltage to zero
T D	Decrements the trigger threshold voltage each time the 'D' key is pressed
T U	Increments the trigger threshold voltage each time the 'U' key is pressed
T S	Sets the trigger position to the START of the signal trace
T M	Sets the trigger position to the MIDDLE of the signal trace
T E	Sets the trigger position to the END of the signal trace
T A	Trigger channel A
T B	Trigger channel B
T >	Trigger input signal above threshold
T <	Trigger input signal below threshold
T R	TRIGGERED SWEEP - Acquire a new signal trace whenever triggered
T X	Trigger on external signal
'left arrow'	Move display window left one sample
'comma'	Move display window left 10 samples
M	Move display window left 100 samples
'right arrow'	Move display window right one sample
'period'	Move display window right 10 samples
/	Move display window right 100 samples
<	Compress the horizontal display base by 1/2
>	Expand the horizontal display base by 2
'ctrl shft P'	Access disk supervisor
'ESC'	Halt and freeze current data acquisition cycle

TIME ADJUSTMENT MODE

'right arrow'	Decreases the sweep rate one step
'left arrow'	Increases the sweep rate one step
>	Increases the sweep rate by 10 second steps (Seconds time scale only)
<	Decreases the sweep rate by 10 second steps (Seconds time scale only)

DISK SUPERVISOR

Q	QUIT saving current parameters and return to control language
L	LOAD data file from disk (enter file name after ?)
S	SAVE current data file to disk (enter file name after ?)
D	Allows input of any DOS command after the "COMMAND ?" prompt
'reset'	Returns directly to the control language.

Entering an invalid input at any time will return the APPLESCOPE to the command mode. Pressing 'reset' while not in the disk supervisor mode will destroy all current files forcing the SCOPE DRIVER to be rebooted.

WARNING - NEVER PRESS 'RESET' UNLESS IN DISK SUPERVISOR.

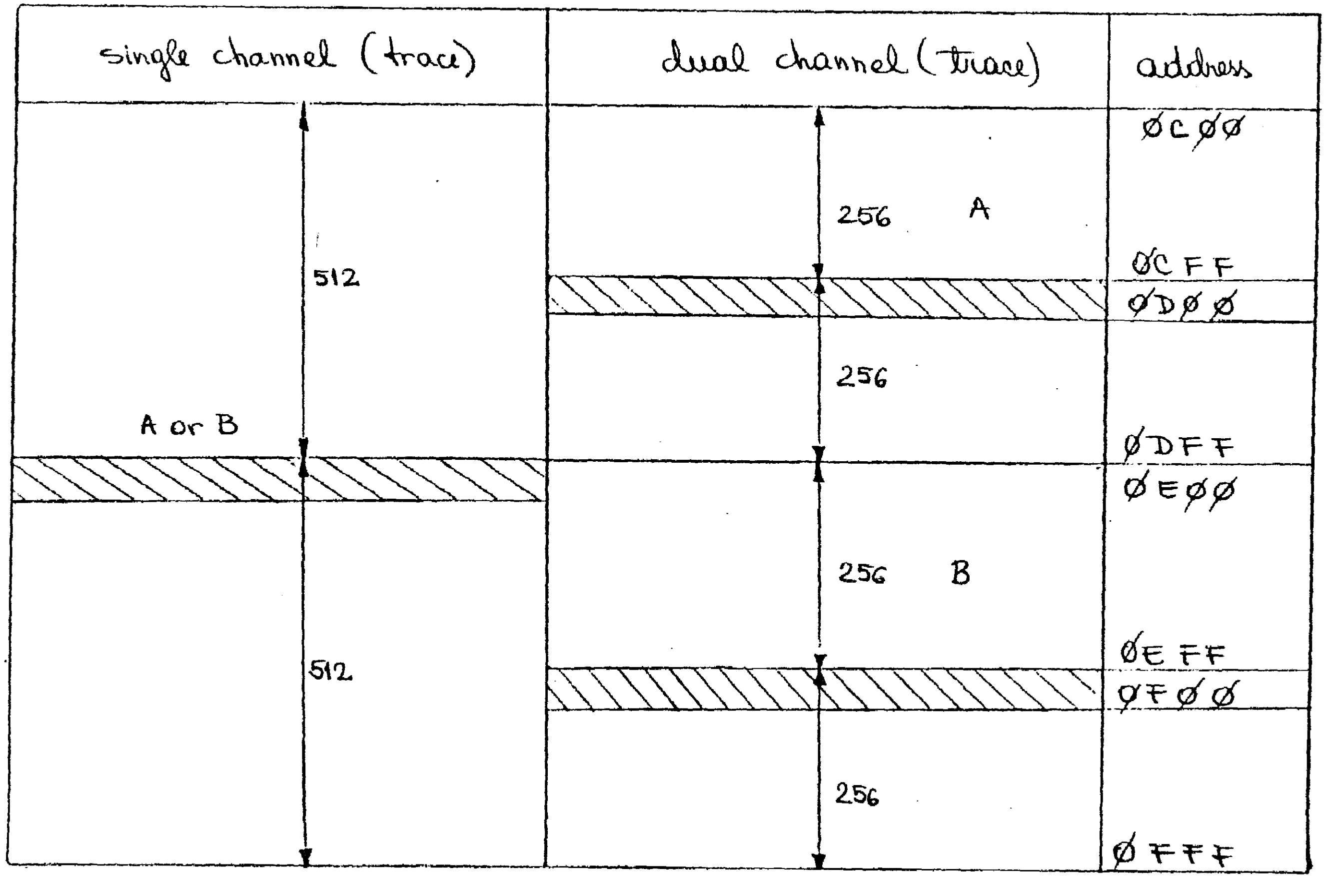


Figure 2.20 Memory After One Data Acquisition Cycle

SCALE (VOLTS/DIV.)	VERTICAL EXPANSION	RESOLUTION (mV/step)
0.025	X 8	7.1
0.050	X 4	7.1
0.100	X 2	7.1
0.200	X 1	7.1
0.400	÷ 2	7.1
0.800	÷ 4	7.1
0.25	X 8	71
0.50	X 4	71
1.00	X 2	71
2.00	X 1	71
4.00	÷ 2	71
8.00	÷ 4	71

Figure 2.21 Voltage Scales

5.1 DISPLAY BUFFER MEMORY – Sweep rates faster than 1 msec./div.

SINGLE CHANNEL		DUAL CHANNEL
Start of buffer memory	$1000	Begin CH A data
Trigger START	$1000	CH A Trigger START
External trigger	$1000	External trigger
	.	
	.	
	.	
	$1100	CH A Trigger MIDDLE
	.	
	.	
	.	
	$11FE	CH A Trigger END
	$11FF	End CH A data
Trigger MIDDLE	$1200	Begin CH B data
	$1200	CH B Treigger START
	.	
	.	
	.	
	$1300	CH B Trigger MIDDLE
	.	
	.	
	.	
Trigger END	$13FC	
	$13FE	CH B Trigger END
End of buffer memory	$13FF	End CH B data

Hardware Data Acquisition

5.2 DISPLAY BUFFER MEMORY – Sweep rates 1 msec./div. and slower

SINGLE CHANNEL		DUAL CHANNEL
Start of buffer memory	$1000	Begin CH A data
External trigger	$1000	External trigger
Trigger START	$1010	CH A Trigger START
	.	
	.	
	$1100	CH A Trigger MIDDLE
	.	
	.	
	.	
	$11F0	CH A Trigger END
	$11FF	End CH A data
Trigger MIDDLE	$1200	Begin CH B data
	$1210	CH B Trigger START
	.	
	.	
	.	
	$1300	CH B Trigger MIDDLE
	.	
	.	
Trigger END	$13F0	CH B Trigger END
End of buffer memory	$13FF	End CH B data

offset from $1000

offset from 13FC

Software Data Acquisition

Figure 2.24 Display Buffer Memory

APPENDIX E

Applescope Disk Supervisor and Disk Operating System (DOS)

The Disk Supervisor

The disk supervisor is a program written in Applesoft and handles all of the disk interface functions, including initial program booting. Once accessed, it provide 4 functions:

1. LOAD ("L"): the previously saved display buffer is loaded into the current display buffer.
2. SAVE ("S"): the current display buffer is saved as a binary file on the disk.
3. DISK ("D"): the screen is changed from mixed graphics with text into text only and any valid disk command may be entered.
4. QUIT ("Q"): represents the exit from the SCOPE DRIVER routine, after saving the current display parameters and data buffers.

Disk Drives and Disk Operating System (DOS)

The disk system used with the Apple II computer as a storage information unit is called DISK II and contains two disk drives, DRIVE1 used for programs (machine language and BASIC programs) and DRIVE2 used for data storage. The two disk drives are connected to "DRIVE1" and "DRIVE2" pins on the card in slot #6.

The DISK II system, unlike the Apple II is a mechanical device, with motors and moving parts. Therefore it is somewhat more delicate than the computer and needs more care.

Learning to use the disk and its operating system (Disk Operating System, DOS) consists of learning a few special instructions, several of which are straightforward extensions of familiar BASIC instructions.

The process of adding the DOS commands to the BASIC in the Apple II is called booting the disk. The disk may be booted from Integer BASIC, from Applesoft, or from the Monitor.

In the DISK II system, information is recorded on a diskette in 35 concentric zones or bands, called tracks. These tracks are numbered from $00, the outermost, through track $22, the innermost. The disk drive's recording and reading head can be moved in or out to stop and hover over each of 35 different zones of the spinning diskettes.

The length of each track is divided into 16 segments, called sectors. These sectors are numbered from $0 through $F and up to 256 bytes of information ($100) can be stored in each sector.

To store information on the diskette, DOS first puts 256 bytes (one sector's worth) of the information in an area of Apple's memory called a file buffer. When this file buffer is full, the information is stored in one sector on the diskette. The DOS fills Apple's file buffer with the next 256 bytes of information and stores that information on the diskette.

In general, DOS begins storing a program or text file whenever it can find an unused sector on the diskette. When that sector is filled with its 265 bytes of information, DOS finds another free sector, perhaps on another track, and continues to record information there. This process continues until the entire file has been stored.

To remember which sectors of which tracks contain information for a particular file, DOS makes up a list of each track and sector used, as it stores the file. The DOS stores that list, called a track-sector list, in another free sector (or sectors) on the diskette.

The file's name, type, length in sectors, and diskette location of the file's track-sector list are recorded in a special area of track $11 called the directory. At this time too, the diskette's track bit map is updated to correctly show which sectors of each track are currently in use.

DOS
QUICK REFERENCE CARD

On this card, DOS commands are grouped into these 5 categories:

Housekeeping commands:

INIT	LOAD	DELETE	VERIFY	MAXFILES
CATALOG	RUN	LOCK	MON	
SAVE	RENAME	UNLOCK	NOMON	

Access Commands:

FP	INT	PR#	IN#	CHAIN

Sequential Text File Commands:

OPEN	READ	APPEND	EXEC
CLOSE	WRITE	POSITION	

Random-Access Text File Commands:

OPEN	CLOSE	READ	WRITE

Machine-Language File Commands:

BLOAD	BRUN	BSAVE

NOTATION AND SYNTAX

A "*parameter*" is a capital letter, usually followed by a number (shown here by a lower-case letter), which gives additional information for executing a command. Multiple parameters may appear in any order, but must be separated from each other by a comma. A parameter shown in square brackets [like this] is optional.

A file name (shown here by X) must immediately follow its command word. File names must begin with a letter; only the first 3Ø characters are used. A comma separates a file name from a following parameter.

CTRL-D (type D while holding down CTRL key) is used in PRINT statements to indicate the start of a deferred-execution DOS command.

Integer BASIC example:

```
    1Ø  D$ = "" : REM  "CTRL-D"
    2Ø  PRINT D$; "CATALOG"
```

Applesoft BASIC example:

```
    1Ø D$ = CHR$(4) : REM  CTRL-D
    2Ø PRINT D$; "CATALOG"
```

The term "BASIC" alone is used to mean either Integer BASIC or Applesoft BASIC. The term "file" alone means any type of diskette file.

COMMAND PARAMETERS

An error message is given if a DOS command quantity is too large or too small.

ALL FILES

Parameter	As shown	Min	Max
Slot	,Ss	S1	S7
Drive	,Dd	D1	D2
Volume	,Vv	VØ *	V254

* Using VØ is like omitting the Vv parameter: the diskette's volume number is ignored. Smallest volume number INIT will actually assign to a diskette is 1.

SEQUENTIAL TEXT FILES

Parameter	As shown	Min	Max
Byte	,Bb	BØ	B32767
Relative Field *	,Rp	RØ	R32767

* With EXEC, always relative to field Ø.

RANDOM-ACCESS TEXT FILES

Parameter	As shown	Min	Max
Record Length	,Lj	L1	L32767
Record Number	,Rr	RØ	R32767

BINARY FILES

Parameter	As shown	Min	Max
Starting Address	,Aa	AØ	A65535
Number of Bytes	,Lj	L1	A32767

DOS COMMANDS

Command	Quantity	As shown	Min	Max
PR#	slot	PR# s	PR#Ø	PR#7
IN#	slot	IN# s	IN#Ø	IN#7
MAXFILES	file buffers	MAXFILES n	n=1	n=16

Commands use Slot or Drive parameters only when changing to a different Slot or Drive.

If a command omits the Volume parameter or uses VØ , the diskette's Volume number is ignored. A command that uses the Volume parameter Vv will not be executed unless the diskette's volume number is v .

HOUSEKEEPING COMMANDS

INIT X [,Vv] [,Ss] [,Dd]
Initializes a blank diskette to form a slave diskette. Assigns greeting program name X and volume number v (if specified). SAVEs the BASIC program currently in memory, under file name X .

CATALOG [,Ss] [,Dd]
Displays volume number and all files on a diskette, with each file's type and sector length. * indicates a LOCKed file.

Type	Description	(How created)
I	Integer BASIC program file	(SAVE)
A	Applesoft BASIC program file	(SAVE)
T	Text File	(OPEN, then WRITE)
B	Binary memory-image file	(BSAVE)

SAVE X [,Ss] [,Dd] [,Vv]
Stores current BASIC program onto diskette, under file name X . Overwrites any previous file of same type and name, without warning.

LOAD X [,Ss] [,Dd] [,Vv]
Loads BASIC program file X into memory, after clearing memory and (if necessary) changing to the correct BASIC.

RUN X [,Ss] [,Dd] [,Vv]
LOADs BASIC program file X , then RUNs the program.

RENAME X, Y [,Ss] [,Dd] [,Vv]
Changes a diskette file's name from X *to* Y

DELETE X [,Ss] [,Dd] [,Vv]
Erases file X *from the diskette.*

LOCK X [,Ss] [,Dd] [,Vv]
Locks file X against accidental change or deletion. LOCKed file shown in CATALOG by *

UNLOCK X [,Ss] [,Dd] [,Vv]
Unlocks previously LOCKed file X to allow change or deletion.

VERIFY X [,Ss] [,Dd] [,Vv]
Checks file X for internal consistency. If X was saved without error, no message is given.

MON [,C] [,I] [,O]
Causes display of disk Commands (C), Input from the disk (I), and Output to the disk (O). With no parameters, MON is ignored.

NOMON [,C] [,I] [,O]
Cancels display of disk Commands (C), Input from the disk (I), and Output to the disk (O). With no parameters, NOMON is ignored.

MAXFILES n
Reserves n file buffers for disk input and output (booting reserves 3 file buffers). Use before LOADing or RUNning a program.

ACCESS COMMANDS

FP [,Ss] [,Dd] [,Vv]
Puts system into Applesoft BASIC, erasing any program in memory.

INT
Puts system into Integer BASIC, erasing any program in memory.

PR# s
Sends subsequent output to slot s . Boots disk if slot s contains disk controller card. PR#0 sends output to TV screen again.

IN# s
Takes subsequent input from slot s . Boots disk if slot s contains disk controller card. IN#0 takes input from keyboard again.

CHAIN Y [,Ss] [,Dd] [,Vv]
RUNs Integer BASIC program file Y , but does not clear variables developed by previous Integer BASIC program.

SEQUENTIAL TEXT FILE COMMANDS

OPEN X [,Ss] [,Dd] [,Vv]
Opens or creates sequential text file X , allocates one file buffer and prepares to WRITE or READ from beginning of file.

CLOSE [X]
Completes WRITE X, if necessary, and de-allocates file buffer assigned to text file X . Without file name, CLOSEs all OPEN files (except an EXEC file).

WRITE X [,Bb]
Subsequent PRINTs send characters to sequential text file X . WRITEing begins at current file position or (if specified) at byte b . Cancelled by any DOS command.

READ X [,Bb]
Subsequent INPUTs and GETs take response characters from sequential text file X . READing begins at current file position or (if specified) at byte b . INPUT response is one field (all characters to next RETURN). Cancelled by any DOS command.

APPEND X [,Ss] [,Dd] [,Vv]
Opens existing sequential text file X , similar to OPEN, but prepares to WRITE at the end of the file.

POSITION X, Rp
In OPEN sequential text file X , subsequent READ or WRITE will proceed from p-th field following current file position.

EXEC X [,Rp] [,Ss] [,Dd] [,Vv]
Executes successive fields in sequential text file X as if typed at keyboard. With Rp parameter, execution begins with p-th field. Fields may include numbered BASIC program lines and direct-execution BASIC or DOS commands to control the Apple.

RANDOM-ACCESS TEXT FILE COMMANDS

OPEN X, Lj [,Ss] [,Dd] [,Vv]
Opens or creates random-access text file X allocates one file buffer, and defines rec length as j bytes. Prepares to WRITE or from beginning of Record 0. Same Length par eter must be used each time file X is OPE

CLOSE [X] [,Ss] [,Dd] [,Vv]
Completes WRITE X, if necessary, and de-allocates file buffer assigned to text fil X . Without file name, CLOSEs all OPEN fi

WRITE X [,Rr] [,Bb]
Subsequent PRINTs send characters to ran access text file X . With no parameters, WRITEing begins at current file position. With Rr parameter alone, WRITEing start byte 0 of Record r . With Bb paramet WRITE starts at byte b of current or sp ified Record. Cancelled by any DOS comma

READ X [,Rr] [,Bb]
Subsequent INPUTs and GETs take response characters from random-access text file With no parameters, READing starts at current file position. With Rr parameter alone, READing starts at byte 0 of Record With Bb parameter, READing starts at by b of current or specified Record. INPUT response is one field (all characters to next RETURN). Cancelled by any DOS comma

MACHINE-LANGUAGE FILE COMMANDS

BSAVE X, Aa, Lj [,Ss] [,Dd] [,Vv]
Stores on diskette, under file name X , the contents of j memory bytes starting at address a .

BLOAD X [,Aa] [,Ss] [,Dd] [,Vv]
Loads binary file X into same memory locations from which file was BSAVEd or (if specified) starting at address a .

BRUN X [,Aa] [,Ss] [,Dd] [,Vv]
BLOADs binary file X , then jumps (JMP) to loaded file's first memory address.

APPENDIX F

BASIC Programs Listing

Variant 1 - Data, Spectrums, Processing

```
]CATALOG

DISK VOLUME 254

*A 004 AUTODUAL
*A 002 CALIBRATE
*A 003 ALTERNATIVE
*A 003 AUTODUAL1
*A 005 SPC
*A 004 RETSPC ✓  — RETTAB
*A 003 CHOICE ✓
*A 005 AVERAGE ✓
*A 004 AVERAGEB ✓
*A 002 CHOICE2  ✓
*A 002 CHOICE2B ✓
*A 013 AXIS
*A 013 AXISB
*A 017 PLOT-A
*A 016 PLOT-B
*A 002 MICROBUF
*A 016 PLOT-A1
*A 012 PLOT-B1
*A 019 PIECHART
*A 019 PIECHARTB
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 LOWALLTAB
*B 002 HIALL.OBJO
*B 002 RETTAB
*B 002 OPEN.OBJO
*B 002 SAVESPC.OBJO
*B 002 SWEEP.OBJO
*B 002 SVTAB
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEHI.OBJO
 B 004 AVE.B
*B 002 CLEARB.OBJO
*B 005 AVELOTABB
*B 002 AVEHIB.OBJO
 B 004 AVEB.B
*B 004 MAGICSPACE#
```

```
]LOAD RETSPC

]LIST

5 D$ =  CHR$ (4)
10  PRINT D$"BLOAD RETTAB1,D1"
15  HOME : PRINT "YOU CAN DO 8 AVERAGES OF 8 SPECTRUMS FOR EACH CHANNEL"
20  VTAB 3: INVERSE : PRINT "ANSWER THE FOLLOWING QUESTION WITH "1" FOR FIRST, "
2" FOR SECOND
     , "3" FOR THIRD,...": NORMAL
25  VTAB 9: INPUT "WHAT IS THE ORDER OF THE AVERAGE THAT YOU WANT TO DO?";T
30  POKE 16897,T
35  IF T <  > 1 GOTO 45
40  GOTO 130
45  IF T <  > 2 GOTO 55
50  GOTO 135
55  IF T <  > 3 GOTO 65
60  GOTO 140
65  IF T <  > 4 GOTO 75
70  GOTO 145
75  IF T <  > 5 GOTO 85
80  GOTO 150
85  IF T <  > 6 GOTO 95
90  GOTO 155
95  IF T <  > 7 GOTO 105
100  GOTO 160
105  IF T <  > 8 GOTO 115
110  GOTO 165
115  HOME : VTAB 8: INVERSE : PRINT "THE NO.OF AVERAGES IS TOO BIG FOR THE POSSI
BILITIES OF T
     HIS DISK!": NORMAL
120  FOR D = 1 TO 2000: NEXT D
125  GOTO 15
130  POKE 823,16: GOTO 170
135  POKE 823.18: GOTO 170
140  POKE 823,20: GOTO 170
145  POKE 823,22: GOTO 170
150  POKE 823,24: GOTO 170
155  POKE 823,26: GOTO 170
160  POKE 823,28: GOTO 170
165  POKE 823,30: GOTO 170
170  CALL 768
175  PRINT D$"RUN CHOICE,D1"
```

```
]LOAD CHOICE

]LIST

5 D$ =  CHR$ (4): HOME
10  PRINT "WHICH CHANNEL DO YOU WANT TO PROCESS(AVERAGE,HISTOGRAM)?
15  PRINT
20  PRINT "****************************************"
25  VTAB 12: PRINT "PRESS "1" FOR CHANNEL A"
30  VTAB 13: PRINT "PRESS "2" FOR CHANNEL B"
35  VTAB 14: PRINT "PRESS "3" FOR BOTH CHAN. A AND B"
40  POKE 16896,0
45  VTAB 15: GET A$: IF  VAL (A$) <  > 1 THEN  GOTO 55
50  GOTO 85
55  IF  VAL (A$) <  > 2 THEN  GOTO 65
60  GOTO 75
65  IF  VAL (A$) <  > 3 THEN  GOTO 45
70  GOTO 95
75  PRINT D$: PRINT D$"RUN AVERAGEB,D1"
80  END
85  PRINT D$: PRINT D$"RUN AVERAGE,D1"
90  END
95  POKE 16896,1
100  PRINT D$: PRINT D$"RUN AVERAGE,D1"
```

```
]LOAD AVERAGE

]LIST

5 D$ =  CHR$ (4)
10  PRINT D$"BLOAD CLEAR.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELGTAB,D1"
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
20  PRINT D$"BLOAD AVEHI.OBJ0,D1": CALL 768: TEXT
25  PRINT D$"BSAVE AVE.B,A$4000,L$200,D1"
30  PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOAD M4C,D1"
35  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
40  PRINT D$"BLOAD AVE.B,A$1000,D1": POKE 37272,1: HOME
45  PRINT "PRESS A TO NOT SEE AVERAGE": PRINT "PRESS OTHER KEY TO SEE AVERAGE
50  GET A$: IF A$ = "A" GOTO 95
55 Y =  PEEK (16897): PRINT
60  PRINT "AVERAGE  J MEANS THAT THE 8K BLOCK NO. J OF SPECTRUMS FROM DISK IS PR
OCESSED (AVER
     AGE CH.A INFORMATION)"
65  PRINT
70  PRINT "TITLE FOR AVERAGE  "Y" ,CH.A": INPUT T$: PRINT D$"PR#1": PRINT  CHR$
(14);T$; CHR$
      (20): PRINT D$"PR#0"
75  POKE 15312,209: POKE 15313,193: POKE 15314,160
80  POKE 15315,207: POKE 15316,181: POKE 15317,160
85  POKE 15318,144: POKE 15319,141: POKE 15320,153: POKE 15321,128
90  POKE 15322,209: CALL 3064: POKE 34,0: TEXT : HOME
95  PRINT D$: PRINT D$"RUN CHOICE2,D1"



]LOAD AVERAGEB

]LIST

5 D$ =  CHR$ (4): PRINT D$"BLOAD CLEARB.OBJO,D1": CALL 768: PRINT D$"BLOAD AVELG
TABB,D1": HOME

10 N =  PEEK ( - 16255):N =  PEEK ( - 16255): PRINT D$"BLOAD AVEHIB.OBJ0,D1": CA
LL 768: TEXT

15  PRINT D$"BSAVE AVEB.B,A$4800,L$200,D1"
20  PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOAD M4C,D1"
25  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
30  PRINT D$"BLOAD AVEB.B,A$1000,D1": POKE 37272,1: HOME
35  PRINT "PRESS A TO NOT SEE AVERAGE": PRINT "PRESS OTHER KEY TO SEE AVERAGE"
40  GET A$: IF A$ = "A" GOTO 85
45 Y =  PEEK (16897): PRINT
50  PRINT "AVERAGE  J MEANS THAT THE 8K BLOCK NO. J OF SPECTRUMS FROM DISK IS PR
OCESSED (AVER
     AGE CH.B INFORMATION)"
55  PRINT
60  PRINT "TITLE FOR AVERAGE  "Y" ,CH.B": INPUT T$: PRINT D$"PR#1": PRINT  CHR$
(14);T$; CHR$
      (20): PRINT D$"PR#0"
65  POKE 15312,209: POKE 15313,193: POKE 15314,160
70  POKE 15315,207: POKE 15316,181: POKE 15317,160
75  POKE 15318,144: POKE 15319,141: POKE 15320,155: POKE 15321,120
80  POKE 15322,209: CALL 3064: POKE 34,0: TEXT : HOME
85  PRINT D$: PRINT D$"RUN CHOICE2B,D1"
```

```
]LOAD CHOICE2

]LIST

5 D$ =  CHR$ (4)
10  HOME : INVERSE : PRINT "WHAT DO YOU WANT FOR CH.A? HISTOGRAM OR PIECHART?":
NORMAL
15  VTAB 8: PRINT "PRESS 1 FOR HISTOGRAM"
20  VTAB 9: PRINT "PRESS 2 FOR PIECHART"
25  VTAB 10: GET A$: IF  VAL (A$) <  > 1 GOTO 35
30  PRINT D$: PRINT D$"RUN AXIS,D1"
35  IF  VAL (A$) <  > 2 GOTO 25
40  PRINT D$: PRINT D$"RUN PLOT-A1,D1"
```

```
]LOAD CHOICE2B

]LIST

5 D$ =  CHR$ (4)
10  HOME : INVERSE : PRINT "WHAT DO YOU WANT FOR CH.B? HISTOGRAM OR PIECHART?":
NORMAL
15  VTAB 8: PRINT "PRESS 1 FOR HISTOGRAM"
20  VTAB 9: PRINT "PRESS 2 FOR PIECHART"
25  VTAB 10: GET A$: IF  VAL (A$) <  > 1 GOTO 35
30  PRINT D$: PRINT D$"RUN AXISB,D1"
35  IF  VAL (A$) <  > 2 GOTO 25
40  PRINT D$: PRINT D$"RUN PLOT-B1,D1"
```

Variant 1 - Processing

```
]CATALOG

DISK VOLUME 254

*A 005 RETSPC
*A 003 CHOICE
*A 005 AVERAGE
*A 004 AVERAGEB
*A 002 CHOICE2
*A 002 CHOICE2B
*A 013 AXIS
*A 013 AXISB
*A 017 PLOT-A
*A 016 PLOT-B
*A 002 MICROBUF
*A 016 PLOT-A1
*A 012 PLOT-B1
*A 019 PIECHART
*A 019 PIECHARTB
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEHI.OBJO
 B 004 AVE.B
*B 002 CLEARB.OBJO
*B 005 AVELOTABB
*B 002 AVEHIB.OBJO
 B 004 AVEB.B
*B 004 MAGICSPACE#
```

```
]LOAD RETSPC

]LIST

5  HOME
10  FOR D = 0 TO 120
15  HTAB 3: VTAB 10: INVERSE : PRINT "E E G  PROCESSING
20  HTAB 3: VTAB 11: PRINT "   M A R I N E L A  L A G U N A -1984 ": NORMAL
25  HTAB 6: VTAB 13: PRINT "V A R I A N T   1"
30  NEXT D: HOME
35 D$ =  CHR$ (4)
40  PRINT D$"BLOAD RETTAB1,D1"
45  HOME : PRINT "YOU CAN DO 8 AVERAGES OF 8 SPECTRUMS FOR EACH CHANNEL"
50  VTAB 5: INVERSE : PRINT "ANSWER THE FOLLOWING QUESTION WITH "1" FOR FIRST,
2" FOR SECOND
    , "3" FOR THIRD,...": NORMAL
55  VTAB 9: INPUT "WHAT IS THE ORDER OF THE AVERAGE THAT YOU WANT TO DO?";T
60  POKE 16897,T
65  IF T <  > 1 GOTO 75
70  GOTO 160
75  IF T <  > 2 GOTO 85
80  GOTO 165
85  IF T <  > 3 GOTO 95
90  GOTO 170
95  IF T <  > 4 GOTO 105
100  GOTO 175
105  IF T <  > 5 GOTO 115
110  GOTO 180
115  IF T <  > 6 GOTO 125
120  GOTO 185
125  IF T <  > 7 GOTO 135
130  GOTO 190
135  IF T <  > 8 GOTO 145
140  GOTO 195
145  HOME : VTAB 8: INVERSE : PRINT "THE NO.OF AVERAGES IS TOO BIG FOR THE POSSI
BILITIES OF T
    HIS DISK!": NORMAL
150  FOR D = 1 TO 2000: NEXT D
155  GOTO 45
160  POKE 823,16: GOTO 200
165  POKE 823,18: GOTO 200
170  POKE 823,20: GOTO 200
175  POKE 823,22: GOTO 200
180  POKE 823,24: GOTO 200
185  POKE 823,26: GOTO 200
190  POKE 823,28: GOTO 200
195  POKE 823,30: GOTO 200
200  CALL 768
205  PRINT D$"RUN CHOICE,D1"
```

Variant 2 - Data, Spectrums, Processing

```
]CATALOG

DISK VOLUME 254

*A 004 AUTODUAL
*A 002 CALIBRATE
*A 003 ALTERNATIVE
*A 005 AUTODUAL1
*A 005 SPC
*A 003 CHOICE1
*A 003 A-AVERAGE
*A 004 B-AVERAGE
*A 005 FINAL-AVE.A
*A 005 FINAL-AVE.B
*A 005 SEE FIN.AVE.A
*A 005 SEE FIN.AVE.B
*A 013 AXIS
*A 013 AXISB
*A 017 PLOT-A
*A 016 PLOT-B
*A 002 MICROBUF
*A 016 PLOT-A1
*A 012 PLOT-B1
*A 019 PIECHART
*A 019 PIECHARTB
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 LOWALLTAB
*B 002 HIALL.OBJ0
*B 002 RETTAB
*B 002 OPEN.OBJ0
*B 002 SAVESPC.OBJ0
*B 002 SWEEP.OBJ0
*B 002 SVTAB
*B 002 RETTAB1
*B 002 CLEAR.OBJ0
*B 002 AVELOTAB
*B 002 AVEHI.OBJ0
*B 002 CLEARB.OBJ0
*B 005 AVELOTABB
*B 002 AVEHIB.OBJ0
*B 004 MAGICSPACE#
```

```
]LOAD AUTODUAL

]LIST

5  HOME
10  HTAB 3: VTAB 10: INVERSE : PRINT "E E G  DATA AQUISITION AND PROCESSING"
15  HTAB 3: VTAB 11: PRINT "    M A R I N E L A   L A G U N A -1984": NORMAL
20  HTAB 6: VTAB 13: PRINT "V A R I A N T  2"
25  HTAB 6: VTAB 15: FLASH : PRINT "LOADING BINARY FILES": NORMAL
30 D$ =  CHR$ (4)
35  PRINT D$"BLOAD BPR,D1"
40  PRINT D$"BLOAD M4A,D1"
45  PRINT D$"BLOAD M4B,D1"
50  PRINT D$"BLOAD M4C,D1"
55  PRINT D$"BLOAD CT,D1"
60  PRINT D$"BLOAD CS,D1"
65  POKE 37375,4: POKE 37272,1
70  POKE 37375,0
75  HOME
80  PRINT "DO CALIBRATION OF THE SCOPE:"
85  PRINT "1.ADJUST TIME BASE TO 0.2 SEC/DIV"
90  PRINT "2.ADJUST SCALE TO 4.0 VOLT/DIV"
95  PRINT "3.ADJUST DISPLAY>00001 BY PRESSING M KEY;ADJUST DISPLAY>00001 BY PRES
SING < CHAR."

100  PRINT "4.PRESS CTRL SHIFT P TO GO OUT FROM CALIBRATION MODE"
105  PRINT "5.I AM READY"
110  PRINT "PRESS 5 IF YOU ARE READY"
115  VTAB (12): GET A$: IF  VAL (A$) <  > 5 THEN  GOTO 115
120  PRINT D$
125  PRINT D$"RUN CALIBRATE,D1"


]LOAD CALIBRATE

]LIST

5 D$ =  CHR$ (4)
10  POKE 37272,1
15  CALL 3064
20  POKE 37272,0
25  PRINT D$"BSAVE CT,A$9190,L$70,D1"
30  PRINT D$"BSAVE CS,A$1F80,L$28,D1"
35  POKE 37272,1
40  TEXT
45  PRINT D$"RUN AUTODUAL1,D1"


]LOAD AUTODUAL1

]LIST

5 D$ =  CHR$ (4)
10  HOME : PRINT "THE MAXIMUM NO. OF SWEEPS CAN BE 64"
15  VTAB 8: INPUT "TOTAL NO.OF SWEEPS";T
20  REM  T=NO.OF BLOCKS OF 1024 BYTES
25  IF T < 65 THEN  GOTO 45
30  HOME : VTAB 8: INVERSE : PRINT "T IS TOO BIG!": NORMAL
35  FOR D = 1 TO 2000: NEXT D: REM  DELAY
40  GOTO 10
45  FOR Y = 0 TO  INT (T / 8) - ((T / 8) =  INT (T / 8))
50  POKE 843,Y * 2: POKE 844,0
55  FOR X = 1 TO 8
60  POKE 15312,209: POKE 15313,193
65  POKE 15314,194: POKE 15315,211
70  POKE 15316,128: POKE 37272,1
75  CALL 3064
80  PRINT D$: POKE 37272,0
85  CALL 768: POKE 37272,1
90  NEXT X: NEXT Y: HOME : POKE 34,0: POKE 49233,0
95  PRINT D$"RUN SPC,D1"
```

```
]LOAD SPC

]LIST

5 D$ =  CHR$ (4)
10  HOME : INPUT "TOTAL NUMRER OF SPECTRUMS":T
15  VTAB 8: PRINT "THE MAXIMUM VALUE OF T CAN BE 128"
20  REM  T=NC.OF BLOCKS OF 512 BYTES
25  IF T < 129 THEN  GOTO 50
30  HOME : VTAB 8: INVERSE : PRINT "T IS TOO BIG!": NORMAL
35  FOR D = 1 TO 2000: NEXT D
40  GOTO 10
45  FOR Y = 0 TO  INT (T / 16) - ((T / 16) =  INT (T / 16))
50  PRINT D$"BLOAD RETTAB,D1"
55  POKE 823,Y * 2: POKE 824,0: CALL 768
60  PRINT D$"BLOAD OPEN.OBJ0,D1"
65  PRINT D$"BLOAD SAVESPC.OBJ0,D1"
70 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
75  PRINT D$"BLOAD SWEEP.OBJ0,D1"
80  FOR X = Y * 16 TO (Y + 1) * 16 - 1
85  CALL 812
90  POKE 15312,209: POKE 15313,193
95  POKE 15314,160: POKE 15315,207
100  POKE 15316,181: POKE 15317,160
105  POKE 15318,141: POKE 15319,211
110  POKE 15320,155: POKE 15321,128
115  POKE 37375,4: POKE 37272,1
120  CALL 3064
125  PRINT D$: POKE 37272,0
130  CALL 768
135  POKE 37272,1: NEXT X
140  PRINT D$"BLOAD SVTAB,D1"
145  POKE 823,Y * 2 + 16: POKE 824,0: CALL 768
150  NEXT Y
155  HOME : POKE 34,0: POKE 49233,0
160 S =  PEEK (20480)
165  IF S <  > 1 THEN  GOTO 175
170  PRINT D$: PRINT D$"RUN CHOICE1,D1"
175  END

]LOAD A-AVERAGE

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.A W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                                    ": NORMAL
15  FOR Y = 0 TO 7: PRINT D$"BLOAD RETTAB1,D1"
20  POKE 823,Y * 2 + 16: CALL 768
25  PRINT D$"BLOAD CLEAR.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTAB,D1"
30 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
35  PRINT D$"BLOAD AVEHI.OBJ0,D1"
40  CALL 768: TEXT
45  PRINT D$"BSAVE AVE."Y",A$4000,L$200 ,D1"
50  NEXT Y
55  PRINT D$"RUN FINAL-AVE.A,D1"

]LOAD B-AVERAGE

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.B W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                                    ": NORMAL
15  FOR Y = 0 TO 7: PRINT D$"BLOAD RETTAB1,D1"
20  POKE 823,Y * 2 + 16: CALL 768
25  PRINT D$"BLOAD CLEARB.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTABB,D1"
30  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.B W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                                    ": NORMAL
35 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
40  PRINT D$"BLOAD AVEHIB.OBJ0,D1"
45  CALL 768: TEXT
50  PRINT D$"BSAVE AVEB."Y",A$4800,L$200,D1"
55  NEXT Y
60  PRINT D$"RUN FINAL-AVE.B,D1"
```

```
]LOAD FINAL-AVE.A

]LIST

5  REM  THIS PROGRAM LOADS CH.A AVERAGES IN SUPERRAM II,  IN THE RIGHT POSITION F
ROM THE POINT
      OF VUE OF THE M/L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.A
10 D$ =  CHR$ (4)
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
20  FOR Y = 0 TO 7
25  REM  53248=$D000
30 Z = 53248 + Y * 1024
35  PRINT D$"BLOAD AVE."Y",A"Z",D1"
40  REM  NOW,SUPERRAM II CONTAINS 8 AVERAGES FOR CH.A,FROM $D000-$D1FF,$D400-$D5
FF,$D800-$D9F
     F,$DC00-$DDFF,$E000-$E1FF,$E400-$E5FF,$E800-$E9FF,$EC00-$EDFF
45  PRINT D$"BLOAD CLEAR.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTAB,D1"
50  HOME : VTAB 15: INVERSE : PRINT "THE FINAL AVERAGE,OF 64 SPECTRUMS,FOR CHAN.
A WILL BE DON
     E NOW AND WILL BE SAVED ON THE DISK                                   ": NORMAL
55  PRINT D$"BLOAD AVEHI.OBJ0,D1"
60  CALL 768: TEXT
65  REM  NOW,THE AVERAGE OF 8 AVERAGES(EACH OF 8 SPC.)OF CH.A,IS LOCATED AT $400
0
70  PRINT D$"BSAVE AVE.AVE.A,A$4000,L$200,D1"
75  PRINT D$"RUN SEE FIN.AVE.A,D1"


]LOAD FINAL-AVE.B

]LIST

5  REM  THIS PROGRAM LOADS CH.B AVERAGES IN SUPERRAM II,IN THE RIGHT POSITION FR
OM THE POINT
      OF VUE OF THE M/L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.B
10 D$ =  CHR$ (4)
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
20  FOR Y = 0 TO 7
25  REM  53760=$D200
30 X = 53760 + Y * 1024
35  PRINT D$"BLOAD AVEB."Y",A"X",D1"
40  REM  NOW,SUPERRAM II CONTAINS 8 AVERAGES FOR CH.B,FROM $D200-$D3FF,$D600-$D7
FF,$DA00-$DBF
     F,$DE00-$DFFF,$E200-$E3FF,$E600-$E7FF,$EA00-$EBFF,$EE00-$EFFF
45  PRINT D$"BLOAD CLEARB.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTABB,D1"
50  HOME : VTAB 15: INVERSE : PRINT "THE FINAL AVERAGE,OF 64 SPECTRUMS,FOR CHAN.
B WILL BE DON
     E NOW AND WILL BE SAVED ON THE DISK                                   ": NORMAL
55  REM  IN LINE 46,THE HOME STATEMENT IS USED TO ERASE THE M/L THAT APPEARS ON
THE SCREEN BY
      LOADING THE BINARY FILE "AVELOTABB"
60  PRINT D$"BLOAD AVEHIB.OBJ0,D1"
65  CALL 768: TEXT
70  REM  NOW,THE AVERAGE OF 8 AVERAGES(EACH OF 8 SPC.)OF CH.B,IS LOCATED AT $480
0
75  PRINT D$"BSAVE AVE.AVE.B,A$4800,L$200,D1"
80  PRINT D$"RUN SEE FIN.AVE.B,D1"
```

```
]LOAD SEE FIN.AVE.A

]LIST

5  REM  THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.A ON THE SCREEN
10 D$ =  CHR$ (4): PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOA
D M4C,D1"
15  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
20  PRINT D$"BLOAD AVE.AVE.A,A$1000,D1"
25  POKE 37272,1: TEXT : HOME : PRINT "PRESS A TO NOT SEE AVERAGE"
30  PRINT "PRESS OTHER KEY TO SEE AVERAGE"
35  GET A$
40  IF A$ = "A" GOTO 80
45  PRINT
50  PRINT "TITLE FOR AVERAGE OF 8 AVERAGES,CH.A": INPUT T$: PRINT D$"PR#1": PRIN
T  CHR$ (14);
    T$; CHR$ (20): PRINT D$"PR#0"
55  POKE 15312,209: POKE 15313,193: POKE 15314,160
60  POKE 15315,207: POKE 15316,181: POKE 15317,160
65  POKE 15318,144: POKE 15319,141: POKE 15320,155
70  POKE 15321,128: POKE 15322,209: POKE 37272,1
75  CALL 3064: PRINT D$: POKE 34,0: TEXT : HOME
80  HOME : PRINT "WHAT DO YOU WANT FOR CH.A? HISTOGRAM OR PIECHART?"
85  VTAB 4: PRINT "PRESS 1-FOR HISTOGRAM"
90  VTAB 5: PRINT "PRESS 2-FOR PIECHART"
95  VTAB 6: GET A$: IF  VAL (A$) <  > 1 GOTO 105
100  PRINT D$: PRINT D$"RUN AXIS,D1"
105  IF  VAL (A$) <  > 2 GOTO 95
110  PRINT D$: PRINT D$"RUN PLOT-A1,D1"

]LOAD SEE FIN.AVE.B

]LIST

5  REM  THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.B ON THE SCREEN
10 D$ =  CHR$ (4): PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOA
D M4C,D1"
15  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
20  PRINT D$"BLOAD AVE.AVE.B,A$1000,D1"
25  POKE 37272,1: TEXT : HOME : PRINT "PRESS A TO NOT SEE AVERAGE"
30  PRINT "PRESS OTHER KEY TO SEE AVERAGE"
35  GET A$
40  IF A$ = "A" GOTO 80
45  PRINT
50  PRINT "TITLE FOR AVERAGE OF 8 AVERAGES,CH.B": INPUT T$: PRINT D$"PR#1": PRIN
T  CHR$ (14);
    T$; CHR$ (20): PRINT D$"PR#0"
55  POKE 15312,209: POKE 15313,193: POKE 15314,160
60  POKE 15315,207: POKE 15316,181: POKE 15317,160
65  POKE 15318,144: POKE 15319,141: POKE 15320,155
70  POKE 15321,128: POKE 15322,209
75  CALL 3064: PRINT D$: POKE 34,0: TEXT : HOME
80  HOME : PRINT "WHAT DO YOU WANT FOR CH.B? HISTOGRAM OR PIECHART?"
85  VTAB 4: PRINT "PRESS 1-FOR HISTOGRAM"
90  VTAB 5: PRINT "PRESS 2-FOR PIECHART"
95  VTAB 6: GET A$: IF  VAL (A$) <  > 1 GOTO 105
100  PRINT D$: PRINT D$"RUN AXISB,D1"
105  IF  VAL (A$) <  > 2 GOTO 95
110  PRINT D$: PRINT D$"RUN PLOT-B1,D1"

]LOAD CHOICE1

]LIST

5 D$ =  CHR$ (4): HOME
10  PRINT "WHICH CHANNEL DO YOU WANT TO PROCESS(AVERAGE,HISTOGRAM)?
15  PRINT
20  PRINT "****************************************"
25  VTAB 12: PRINT "PRESS "1" FOR CHANNEL A"
30  VTAB 13: PRINT "PRESS "2" FOR CHANNEL B"
35  VTAB 14: PRINT "PRESS "3" FOR BOTH CHAN. A AND B"
40  POKE 16876,0
45  VTAB 15: GET A$: IF  VAL (A$) <  > 1 THEN  GOTO 55
50  GOTO 85
55  IF  VAL (A$) <  > 2 THEN  GOTO 65
60  GOTO 75
65  IF  VAL (A$) <  > 3 THEN  GOTO 45
70  GOTO 95
75  PRINT D$: PRINT D$"RUN B-AVERAGE,D1"
80  END
85  PRINT D$: PRINT D$"RUN A-AVERAGE,D1"
90  END
95  POKE 16896,1
100  PRINT D$: PRINT D$"RUN A-AVERAGE,D1"
```

```
]LOAD AXIS

]LIST

5 D$ =  CHR$ (4)
10  REM  DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15  HGR
20  HCOLOR= 3
25  HPLOT 15,144 TO 15,0
30  HPLOT 14,1 TO 16,1: HPLOT 13,2 TO 17,2
35  HPLOT 15,144 TO 279,144
40  HPLOT 277,142 TO 277,146: HPLOT 278,143 TO 278,145
45  PRINT D$"BLOAD MAGICSPACE#,D1"
50  POKE 232,0: POKE 233,96
55  SCALE= 1
60  ROT= 0
65  REM  A=Y CENTER UP;B=Y CENTER DOWN
70  REM  THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75  REM  DRAW UP NUMBERS
80 A = 146:B = 154
85 X = 9
90  FOR N = 1 TO 8
95  READ S
100  DRAW S AT X,A
105 X = X + 4
110  IF N < 4 GOTO 120
115  IF N = 4 THEN X = 23
120  NEXT N
125  DATA  49,37,49,49,49,14,10,19
130 X = 51
135  FOR N = 1 TO 8
140  READ S
145  DRAW S AT X,A
150 X = X + 4
155  IF N < 4 GOTO 165
160  IF N = 4 THEN X = 79
165  NEXT N
170  DATA  12,12,10,17,12,20,10,15
175 X = 122
180  FOR N = 1 TO 12
185  READ S
190  DRAW S AT X,A
195 X = X + 4
200  IF N < 4 GOTO 220
205  IF N = 4 THEN X = 180
210  IF N < 8 GOTO 220
215  IF N = 8 THEN X = 239
220  NEXT N
225  DATA  14,12,10,14,15,18,10,16,17,14,10,17
230  REM  DRAW DOWN NUMBERS
235 X = 16
240  FOR N = 1 TO 4
245  READ S
250  DRAW S AT X,B
255 X = X + 4
260  NEXT N
265  DATA  49,12,10,20
270 X = 37
```

```
275  FOR N = 1 TO 12
280  READ S
285  DRAW S AT X,8
290 X = X + 4
295  IF N < 4 GOTO 315
300  IF N = 4 THEN X = 65
305  IF N < 8 GOTO 315
310  IF N = 8 THEN X = 93
315  NEXT N
320  DATA  49,18,10,18,12,16.10,16,13,14,10,14
325 X = 151
330  FOR N = 1 TO 12
335  READ S
340  DRAW S AT X,3
345 X = X + 4
350  IF N < 4 GOTO 370
355  IF N = 4 THEN X = 209
360  IF N < 8 GOTO 370
365  IF N = 8 THEN X = 266
370  NEXT N
375  DATA  14,20,10,15.16,16,10,16.18,12,10,12
380  REM  THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
IS IS 5
385  REM  DRAW"HZ"ON THE HORIZONTAL AXIS
390 X = 260
395  FOR N = 1 TO 2
400  READ S
405  DRAW S AT X,136
410 X = X + 5
415  NEXT N
420  DATA  30,48
425  REM  MARK THE DIVISIONS ON THE VERTICAL AXIS
430 C = 144 / 240
435 X = 15
440  DIM Y(10): DIM Z(10)
445  FOR I = 1 TO 5
450  READ S
455 Y(I) = 144 - S * C - 2: REM  2 IS BECAUSE THE M/L WHICH DRAWS THE CHAR. CONS
IDERS THE CHA
     R. FROM THE TOP(ONE CHAR. HAS 5 VERT. POINTS)
460 Z(I) = Y(I) + (S * C) / (2 * I)
465  PRINT "S*C="S * C
470  PRINT "Y(I)="Y(I)
475  PRINT "Z(I)="Z(I)
480  NEXT I
485  DATA  40,80,120,160,200
490  FOR I = 1 TO 5
495  DRAW 9 AT X,Y(I)
500  DRAW 9 AT X,Z(I)
505  NEXT I
510  FOR I = 1 TO 5
515 X = 5
520  FOR N = 1 TO 2
525  READ S,U
530  DRAW S AT X,Y(I)
535  DRAW U AT X,Z(I)
540 X = X + 4
545  NEXT N
550  NEXT I
555  REM  S=49.15,49,19,12,13,12,17,13,37 AND U=49,13,49,17,12,37,12,15 12,19
560  DATA  49,49,15,13.49,49,19,17.12,12,13,37,12,12,17,15,13,12,37,19
565  REM   DRAW"% OF TOT. POWER"ON THE VERTICAL AXIS
```

```
570 X = 22:Y = 2
575  DRAW 3 AT X,Y
580 X = X + 8
585  FOR N = 1 TO 2
590  READ S
595  DRAW S AT X,Y
600 X = X + 5
605  NEXT N
610  DATA  37,28
615 X = X + 3
620  FOR N = 1 TO 4
625  READ S
630  DRAW S AT X,Y
635 X = X + 5
640  NEXT N
645  DATA  42,37,42,10
650 X = X + 3
655  FOR N = 1 TO 5
660  READ S
665  DRAW S AT X,Y
670 X = X + 5
675  IF N = 3 THEN X = X + 1
680  NEXT N
685  DATA  38,37,45.27,40
690  REM  DRAW TITLE OF THE GRAPH
695  REM  THE DISTANCE BETWEEN THE NUMBERS IN THE TITLE IS 5
700  REM  THE DISTANCE BETWEEN THE WORDS IN THE TITLE IS 6
705  REM  THE DISTANCE BETWEEN THE LETTERS OF THE TITLE IS 7;EXCEPTION IN THE LA
ST PART,"HZ..
     .",WHERE IS 5
710  SCALE= 1
715 X = 90:Y = 10
720  FOR N = 1 TO 3
725  READ S
730  DRAW S AT X,Y
735 X = X + 7
740  NEXT N
745  DATA  27,27,29
750 X = X + 6
755  FOR N = 1 TO 9
760  READ S
765  DRAW S AT X,Y
770 X = X + 7
775  NEXT N
780  DATA  30,31,41,42,37,29,40,23,35
785 X = X + 6
790  FOR N = 1 TO 3
795  READ S
800  DRAW S AT X,Y
805 X = X + 7
810  NEXT N
815  DATA  28,37,40
820 X = X + 6
825  FOR N = 1 TO 14
830  READ S
835  DRAW S AT X,Y
840 X = X + 5
845  NEXT N
850  DATA  37,9,13,12.10,12,49,30,48,9,25,30,10,23
855  PRINT D$"RUN PLOT-A,D1"
```

```
]LOAD AXISB

]LIST

5 D$ =  CHR$ (4)
10  REM  DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15  HGR
20  HCOLOR= 3
25  HPLOT 15,144 TO 15,0
30  HPLOT 14,1 TO 16,1: HPLOT 13,2 TO 17,2
35  HPLOT 15,144 TO 279,144
40  HPLOT 277,142 TO 277,146: HPLOT 278,143 TO 278,145
45  PRINT D$"BLOAD MAGICSPACE#,D1"
50  POKE 232,0: POKE 233,96
55  SCALE= 1
60  ROT= 0
65  REM  A=Y CENTER UP;B=Y CENTER DOWN
70  REM  THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75  REM  DRAW UP NUMBERS
80 A = 146:B = 154
85 X = 9
90  FOR N = 1 TO 8
95  READ S
100  DRAW S AT X,A
105 X = X + 4
110  IF N < 4 GOTO 120
115  IF N = 4 THEN X = 23
120  NEXT N
125  DATA  49,37,49,49,49,14,10,19
130 X = 51
135  FOR N = 1 TO 8
140  READ S
145  DRAW S AT X,A
150 X = X + 4
155  IF N < 4 GOTO 165
160  IF N = 4 THEN X = 79
165  NEXT N
170  DATA  12,12,10,17,12,20,10,15
175 X = 122
180  FOR N = 1 TO 12
185  READ S
190  DRAW S AT X,A
195 X = X + 4
200  IF N < 4 GOTO 220
205  IF N = 4 THEN X = 180
210  IF N < 8 GOTO 220
215  IF N = 8 THEN X = 239
220  NEXT N
225  DATA  14,12,10,14,15,18,10,16,17,14,10,17
230  REM  DRAW DOWN NUMBERS
235 X = 16
240  FOR N = 1 TO 4
245  READ S
250  DRAW S AT X,B
255 X = X + 4
260  NEXT N
265  DATA  49,12,10,20
270 X = 37
```

```
275  FOR N = 1 TO 12
280  READ S
285  DRAW S AT X,B
290 X = X + 4
295  IF N < 4 GOTO 315
300  IF N = 4 THEN X = 65
305  IF N < 8 GOTO 315
310  IF N = 8 THEN X = 93
315  NEXT N
320  DATA  49,18,10,18,12,16,10,16,13,14,10,14
325 X = 151
330  FOR N = 1 TO 12
335  READ S
340  DRAW S AT X,B
345 X = X + 4
350  IF N < 4 GOTO 370
355  IF N = 4 THEN X = 209
360  IF N < 8 GOTO 370
365  IF N = 8 THEN X = 265
370  NEXT N
375  DATA  14,20,10,15,16,16,10,16,18,12,10,12
380  REM  THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
IS IS 5
385  REM  DRAW"HZ"ON THE HORIZONTAL AXIS
390 X = 260
395  FOR N = 1 TO 2
400  READ S
405  DRAW S AT X,136
410 X = X + 5
415  NEXT N
420  DATA  30,48
425  REM  MARK THE DIVISIONS ON THE VERTICAL AXIS
430 C = 144 / 240
435 X = 15
440  DIM Y(10): DIM Z(10)
445  FOR I = 1 TO 5
450  READ S
455 Y(I) = 144 - S * C - 2: REM  2 IS BECAUSE THE M/L WHICH DRAWS THE CHAR. CONS
IDERS THE CHA
     R. FROM THE TOP(ONE CHAR. HAS 5 VERT. POINTS)
460 Z(I) = Y(I) + (S * C) / (2 * I)
465  PRINT "S*C="S * C
470  PRINT "Y(I)="Y(I)
475  PRINT "Z(I)="Z(I)
480  NEXT I
485  DATA  40,80,120,160,200
490  FOR I = 1 TO 5
495  DRAW 9 AT X,Y(I)
500  DRAW 9 AT-X,Z(I)
505  NEXT I
510  FOR I = 1 TO 5
515 X = 5
520  FOR N = 1 TO 2
525  READ S.U
530  DRAW S AT X,Y(I)
535  DRAW U AT X,Z(I)
540 X = X + 4
545  NEXT N
550  NEXT I
555  REM  S=49,15,49,19,12,13,12,17,13,37 AND U=49,13,49,17,12,37,12,15,12,19
560  DATA  49,49,15,13,49,49,19,17,12,12,13,37,12,12,17,15,13,12,37,19
565  REM   DRAW"% OF TOT. POWER"ON THE VERTICAL AXIS
```

```
570 X = 22:Y = 2
575  DRAW 3 AT X,Y
580 X = X + 8
585  FOR N = 1 TO 2
590  READ S
595  DRAW S AT X,Y
600 X = X + 5
605  NEXT N
610  DATA  37,28
615 X = X + 3
620  FOR N = 1 TO 4
625  READ S
630  DRAW S AT X,Y
635 X = X + 5
640  NEXT N
645  DATA  42,37,42,10
650 X = X + 3
655  FOR N = 1 TO 5
660  READ S
665  DRAW S AT X,Y
670 X = X + 5
675  IF N = 3 THEN X = X + 1
680  NEXT N
685  DATA  38,37,45,27,40
690  REM  DRAW TITLE OF THE GRAPH
695  REM  THE DISTANCE BETWEEN THE NUMBERS IN THE TITLE IS 5
700  REM  THE DISTANCE BETWEEN THE WORDS IN THE TITLE IS 6
705  REM  THE DISTANCE BETWEEN THE LETTERS OF THE TITLE IS 7;EXCEPTION IN THE LA
ST PART,"HZ..
     .",WHERE IS 5
710  SCALE= 1
715 X = 90:Y = 10
720  FOR N = 1 TO 3
725  READ S
730  DRAW S AT X,Y
735 X = X + 7
740  NEXT N
745  DATA  27,27,29
750 X = X + 6
755  FOR N = 1 TO 9
760  READ S
765  DRAW S AT X,Y
770 X = X + 7
775  NEXT N
780  DATA  30,31,41,42,37,29,40,23,33
785 X = X + 6
790  FOR N = 1 TO 3
795  READ S
800  DRAW S AT X,Y
805 X = X + 7
810  NEXT N
815  DATA  28,37,40
820 X = X + 6
825  FOR N = 1 TO 14
830  READ S
835  DRAW S AT X,Y
840 X = X + 5
845  NEXT N
850  DATA  37,9.18,12.10,12.49,30,48,9,25,30,10,24
855  PRINT D$"RUN PLOT-B,D1"
```

```
]LOAD PLOT-A

]LIST

5 D$ =  CHR$ (4)
10  HTAB 10: VTAB 23: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM  O1 IS FOR THE % AND O2 IS FOR HIST.AMPLIT.CALC.
25  FOR Y = 0 TO 255
30 X =  PEEK (16384 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  PRINT D$"PR#1"
55  PRINT "TOTAL POWER="S
60  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ.INTERVALS OF THE HISTOGRAM
65  READ J,K,T,U
70  DATA  7,14,29,27
75  REM  CALC. % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90  FOR Y1 = 0 TO 2 * J
95 X1 =  PEEK (16384 + Y1)
100 P1 = X1 ^ 2
105 S1 = S1 + P1
110 N = N + 1
115  IF N < 7 THEN  GOTO 130
120  IF N = 7 THEN A1 = (S1 / S) * O1
125  IF N = 8 THEN S1 = 0
130  NEXT Y1
135 A2 = (S1 / S) * O1
140  PRINT "% OF FIRST BAND="A1
145  PRINT "% OF SECOND BAND="A2
150  REM  CALC.% POWER OF NEXT FIVE BANDS
155 N = 0
160 S2 = 0
165  FOR Y2 = 0 TO 69
170 X2 =  PEEK (16384 + 2 * J + Y2)
175 P2 = X2 ^ 2
180 S2 = S2 + P2
185 N = N + 1
190  IF N < 14 THEN  GOTO 250
195  IF N = 14 THEN A3 = (S2 / S) * O1
200  IF N = 15 THEN S2 = 0
205  IF N < 28 THEN  GOTO 250
210  IF N = 28 THEN A4 = (S2 / S) * O1
215  IF N = 29 THEN S2 = 0
220  IF N < 42 THEN  GOTO 250
225  IF N = 42 THEN A5 = (S2 / S) * O1
230  IF N = 43 THEN S2 = 0
235  IF N < 56 THEN  GOTO 250
240  IF N = 56 THEN A6 = (S2 / S) * O1
243  IF N = 57 THEN S2 = 0
250  NEXT Y2
255 A7 = (S2 / S) * O1
260  PRINT "% OF THIRD BAND="A3
265  PRINT "% OF FOURTH BAND="A4
270  PRINT "% OF FIFTH BAND="A5
```

```
275  PRINT "% OF SIXTH BAND="A6
280  PRINT "% OF SEVENTH BAND="A7
285  REM  CALC. % POWER OF NEXT FIVE BANDS
290 N = 0
295 S3 = 0
300  FOR Y3 = 0 TO 144
305 X3 =  PEEK (16384 + 2 * J + 5 * K + Y3)
310 P3 = X3 ^ 2
315 S3 = S3 + P3
320 N = N + 1
325  IF N < 29 THEN  GOTO 385
330  IF N = 29 THEN A8 = (S3 / S) * 01
335  IF N = 30 THEN S3 = 0
340  IF N < 56 THEN  GOTO 385
345  IF N = 56 THEN A9 = (S3 / S) * 01
350  IF N = 57 THEN S3 = 0
355  IF N < 87 THEN  GOTO 385
360  IF N = 87 THEN B1 = (S3 / S) * 01
365  IF N = 87 THEN S3 = 0
370  IF N < 116 THEN  GOTO 385
375  IF N = 116 THEN B2 = (S3 / S) * 01
380  IF N = 117 THEN S3 = 0
385  NEXT Y3
390 B3 = (S3 / S) * 01
395  PRINT "% OF EIGHTH BAND="A8
400  PRINT "% OF NINTH BAND="A9
405  PRINT "% OF TENTH BAND="B1
410  PRINT "% OF ELEVENTH BAND="B2
415  PRINT "% OF TWELTH BAND="B3
420  REM  CALC. % POWER OF LAST BAND
425 S4 = 0
430  FOR Y4 = 0 TO 26
435 X4 =  PEEK (16384 + 2 * J + 5 * K + 5 * T + Y4)
440 P4 = X4 ^ 2
445 S4 = S4 + P4
450  NEXT Y4
455 B4 = (S4 / S) * 125
460  PRINT "% OF THERTEENTH BAND="B4
465  PRINT "TOTAL % BEFORE ROUND OFF IS T%="A1 + A2 + A3 + A4 + A5 + A6 + A7 + A
8 + A9 + B1 +
     B2 + B3 + B4
470 A1 =  INT ((A1 * 02) + 0.05) / 02
475 A2 =  INT ((A2 * 02) + 0.05) / 02
480 A3 =  INT ((A3 * 02) + 0.05) / 02
485 A4 =  INT ((A4 * 02) + 0.05) / 02
490 A5 =  INT ((A5 * 02) + 0.05) / 02
495 A6 =  INT ((A6 * 02) + 0.05) / 02
500 A7 =  INT ((A7 * 02) + 0.05) / 02
505 A8 =  INT ((A8 * 02) + 0.05) / 02
510 A9 =  INT ((A9 * 02) + 0.05) / 02
515 B1 =  INT ((B1 * 02) + 0.05) / 02
520 B2 =  INT ((B2 * 02) + 0.05) / 02
525 B3 =  INT ((B3 * 02) + 0.05) / 02
530 B4 =  INT ((B4 * 02) + 0.05) / 02
535  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
540  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
545  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
550  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
555  PRINT "B1="B1: PRINT "B2="B2: PRINT "B3="B3
560  PRINT "B4="B4
565 T% = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
```

```
570  PRINT "TOTAL % POWER IS T%="T%
575  PRINT D$"PR#0"
580  DIM D(20): DIM A(20)
585  DIM N(20): DIM M(20)
590  DIM F(20)
595  REM  CALC. NO.OF HZ/POINT,"S"
600 S = 71.12 / 256
605  PRINT "S="S
610  REM  CALC. NO. OF POINTS/UNIT %,"P"
615 P = 144 / 240:P =  INT ((P * 02) + 0.5) / 02
620  REM  CALC. FREQ. INTERVALS,"D(I)"
625  REM  F(I) ARE THE FREQ. OF THE END OF EACH FREQ. INTERVAL
630 F(0) = J * S
635 F(1) = 2 * F(0)
640 F = F(1)
645  FOR I = 2 TO 6
650 F(I) = F + K * S:F = F(I)
655  NEXT I
660 E = F(6)
665  FOR I = 7 TO 11
670 F(I) = E + T * S:E = F(I)
675  NEXT I
680 F(12) = F(11) + U * S
685  REM  D(I) ARE FREQ. INTERVALS
690 D(0) = F(0)
695 D = D(0)
700  FOR I = 1 TO 12
705 D(I) = F(I) - D:D = F(I)
710  NEXT I
715  FOR I = 0 TO 12
720  PRINT "F(I)="F(I)
725  PRINT "***************"
730  PRINT "D(I)="D(I): NEXT I
735  REM  CALC. AMPLITUDES FOR HISTOGRAM
740 A(0) = A1 * 02:A(1) = A2 * 02
745 A(2) = A3 * 02:A(3) = A4 * 02
750 A(4) = A5 * 02:A(5) = A6 * 02
755 A(6) = A7 * 02:A(7) = A8 * 02
760 A(8) = A9 * 02:A(9) = B1 * 02
765 A(10) = B2 * 02:A(11) = B3 * 02
770 A(12) = B4 * 02
775  FOR I = 0 TO 12
780 D(I) =  INT ((D(I) * 02) + 0.5) / 02
785  PRINT "D(I)="D(I)"HZ"
790  REM  CALC. NO. OF POINTS/INTERVAL FOR FREQ. SCALE,"N(I)"
795 N(I) = D(I) / S:N(I) =  INT ((N(I) * 02) + .5) / 02
800  PRINT "N(I)="N(I)"POINTS"
805  REM  CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
810 M(I) = P * A(I):M(I) =  INT ((M(I) * 02) + .5) / 02
815  NEXT I
820  REM  DRAW THE HISTOGRAM
825 Q = 15:R = 15
830  FOR I = 0 TO 12
835 R = R + N(I): PRINT "R(I)="R
840  FOR Y = 0 TO M(I)
845  HPLOT Q,144 - Y TO Q,144
850  HPLOT R,144 - Y TO R,144
855  NEXT Y
860  FOR X = 0 TO N(I)
865  HPLOT Q,144 - M(I) TO X + Q,144 - M(I)


870  NEXT X
875 Q = Q + N(I)
880  NEXT I
885  PRINT D$"RUN MICROBUF,D1"
```

```
]LOAD PLOT-B

]LIST

5 D$ =  CHR$ (4)
10  HTAB 10: VTAB 23: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM   O1 IS FOR THE % AND O2 IS FOR HIST.AMPLIT.CALC.
25  FOR Y = 0 TO 255
30 X =  PEEK (18432 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  PRINT D$"PR#1"
55  PRINT "TOTAL POWER="S
60  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ.INTERVALS OF THE HISTOGRAM
65  READ J,K,T,U
70  DATA  7,14,29,27
75  REM  CALC. % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90  FOR Y1 = 0 TO 2 * J
95 X1 =  PEEK (18432 + Y1)
100 P1 = X1 ^ 2
105 S1 = S1 + P1
110 N = N + 1
115  IF N < 7 THEN  GOTO 130
120  IF N = 7 THEN A1 = (S1 / S) * O1
125  IF N = 8 THEN S1 = 0
130  NEXT Y1
135 A2 = (S1 / S) * O1
140  PRINT "% OF FIRST BAND="A1
145  PRINT "% OF SECOND BAND="A2
150  REM  CALC.% POWER OF NEXT FIVE BANDS
155 N = 0
160 S2 = 0
165  FOR Y2 = 0 TO 69
170 X2 =  PEEK (18432 + 2 * J + Y2)
175 P2 = X2 ^ 2
180 S2 = S2 + P2
185 N = N + 1
190  IF N < 14 THEN  GOTO 250
195  IF N = 14 THEN A3 = (S2 / S) * O1
200  IF N = 15 THEN S2 = 0
205  IF N < 28 THEN  GOTO 250
210  IF N = 28 THEN A4 = (S2 / S) * O1
215  IF N = 29 THEN S2 = 0
220  IF N < 42 THEN  GOTO 250
225  IF N = 42 THEN A5 = (S2 / S) * O1
230  IF N = 43 THEN S2 = 0
235  IF N < 56 THEN  GOTO 250
240  IF N = 56 THEN A6 = (S2 / S) * O1
245  IF N = 57 THEN S2 = 0
250  NEXT Y2
255 A7 = (S2 / S) * O1
260  PRINT "% OF THIRD BAND="A3
265  PRINT "% OF FOURTH BAND="A4
270  PRINT "% OF FIFTH BAND="A5
```

```
275  PRINT "% OF SIXTH BAND="A6
280  PRINT "% OF SEVENTH BAND="A7
285  REM  CALC. % POWER OF NEXT FIVE BANDS
290 N = 0
295 S3 = 0
300  FOR Y3 = 0 TO 144
305 X3 =  PEEK (18432 + 2 * J + 5 * K + Y3)
310 P3 = X3 ^ 2
315 S3 = S3 + P3
320 N = N + 1
325  IF N < 29 THEN  GOTO 385
330  IF N = 29 THEN A8 = (S3 / S) * 01
335  IF N = 30 THEN S3 = 0
340  IF N < 56 THEN  GOTO 385
345  IF N = 56 THEN A9 = (S3 / S) * 01
350  IF N = 57 THEN S3 = 0
355  IF N < 87 THEN  GOTO 385
360  IF N = 87 THEN B1 = (S3 / S) * 01
365  IF N = 87 THEN S3 = 0
370  IF N < 116 THEN  GOTO 385
375  IF N = 116 THEN B2 = (S3 / S) * 01
380  IF N = 117 THEN S3 = 0
385  NEXT Y3.
390 B3 = (S3 / S) * 01
395  PRINT "% OF EIGHTH BAND="A8
400  PRINT "% OF NINTH BAND="A9
405  PRINT "% OF TENTH BAND="B1
410  PRINT "% OF ELEVENTH BAND="B2
415  PRINT "% OF TWELTH BAND="B3
420  REM  CALC. % POWER OF LAST BAND
425 S4 = 0
430  FOR Y4 = 0 TO 26
435 X4 =  PEEK (18432 + 2 * J + 5 * K + 5 * T + Y4)
440 P4 = X4 ^ 2
445 S4 = S4 + P4
450  NEXT Y4
455 B4 = (S4 / S) * 125
460  PRINT "% OF THERTEENTH BAND="B4
465  PRINT "TOTAL % BEFORE ROUND OFF IS T%="A1 + A2 + A3 + A4 + A5 + A6 + A7 +
8 + A9 + B1 +
     B2 + B3 + B4
470 A1 =  INT ((A1 * 02) + 0.05) / 02
475 A2 =  INT ((A2 * 02) + 0.05) / 02
480 A3 =  INT ((A3 * 02) + 0.05) / 02
485 A4 =  INT ((A4 * 02) + 0.05) / 02
490 A5 =  INT ((A5 * 02) + 0.05) / 02
495 A6 =  INT ((A6 * 02) + 0.05) / 02
500 A7 =  INT ((A7 * 02) + 0.05) / 02
505 A8 =  INT ((A8 * 02) + 0.05) / 02
510 A9 =  INT ((A9 * 02) + 0.05) / 02
515 B1 =  INT ((B1 * 02) + 0.05) / 02
520 B2 =  INT ((B2 * 02) + 0.05) / 02
525 B3 =  INT ((B3 * 02) + 0.05) / 02
530 B4 =  INT ((B4 * 02) + 0.05) / 02
535  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
540  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
545  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
550  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
555  PRINT "B1="B1: PRINT "B2="B2: PRINT "B3="B3
560  PRINT "B4="B4
565 T% = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
```

```
570  PRINT "TOTAL % POWER IS T%="T%
575  PRINT D$"PR#0"
580  DIM D(20): DIM A(20)
585  DIM N(20): DIM M(20)
590  DIM F(20)
595  REM  CALC. NO.OF HZ/POINT,"S"
600 S = 71.12 / 256
605  PRINT "S="S
610  REM  CALC. NO. OF POINTS/UNIT %,"P"
615 P = 144 / 240:P =  INT ((P * O2) + .5) / O2
620  REM  CALC. FREQ. INTERVALS,"D(I)"
625  REM  F(I) ARE THE FREQ. OF THE END OF EACH FREQ. INTERVAL
630 F(0) = J * S
635 F(1) = 2 * F(0)
640 F = F(1)
645  FOR I = 2 TO 6
650 F(I) = F + K * S:F = F(I)
655  NEXT I
660 E = F(6)
665  FOR I = 7 TO 11
670 F(I) = E + T * S:E = F(I)
675  NEXT I
680 F(12) = F(11) + U * S
685  REM  D(I) ARE FREQ. INTERVALS
690 D(0) = F(0)
695 D = D(0)
700  FOR I = 1 TO 12
705 D(I) = F(I) - D:D = F(I)
710  NEXT I
715  FOR I = 0 TO 12
720  PRINT "F(I)="F(I)
725  PRINT "***************"
730  PRINT "D(I)="D(I): NEXT I
735  REM  CALC. AMPLITUDES FOR HISTOGRAM
740 A(0) = A1 * O2:A(1) = A2 * O2
745 A(2) = A3 * O2:A(3) = A4 * O2
750 A(4) = A5 * O2:A(5) = A6 * O2
755 A(6) = A7 * O2:A(7) = A8 * O2
760 A(8) = A9 * O2:A(9) = B1 * O2
765 A(10) = B2 * O2:A(11) = B3 * O2
770 A(12) = B4 * O2
775  FOR I = 0 TO 12
780 D(I) =  INT ((D(I) * O2) + 0.5) / O2
785  PRINT "D(I)="D(I)"HZ"
790  REM  CALC. NO. OF POINTS/INTERVAL FOR FREQ. SCALE,"N(I)"
795 N(I) = D(I) / S:N(I) =  INT ((N(I) * O2) + .5) / O2
800  PRINT "N(I)="N(I)"POINTS"
805  REM  CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
810 M(I) = P * A(I):M(I) =  INT ((M(I) * O2) + .5) / O2
815  NEXT I
820  REM  DRAW THE HISTOGRAM
825 Q = 15:R = 15
830  FOR I = 0 TO 12
835 R = R + N(I)
840  FOR Y = 0 TO M(I)
845  HPLOT Q,144 - Y TO Q,144
850  HPLOT R,144 - Y TO R,144
855  NEXT Y
860  FOR X = 0 TO N(I)
865  HPLOT Q,144 - M(I) TO X + Q,144 - M(I)

870  NEXT X
875 Q = Q + N(I)
880  NEXT I
885  PRINT D$"RUN MICROBUF,D1"
```

```
]LOAD PLOT-A1

]LIST

5 D$ =  CHR$ (4)
10  HOME : INVERSE : PRINT "WAIT FOR THE CALCULATIONS!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM  O1 IS FOR THE % AND O2 IS FOR ROUND-OFF
25  FOR Y = 0 TO 255
30 X =  PEEK (16384 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  HOME
55  PRINT D$"PR#1"
60  PRINT "TOTAL POWER="S
65  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ.INTERVALS OF THE HISTOGRAM
70  READ J,K,T,U
75  DATA  7,14,29,27
80  REM  CALC. % OF FIRST TWO BANDS
85 N = 0
90 S1 = 0
95  FOR Y1 = 0 TO 2 * J
100 X1 =  PEEK (16384 + Y1)
105 P1 = X1 ^ 2
110 S1 = S1 + P1
115 N = N + 1
120  IF N < 7 THEN  GOTO 135
125  IF N = 7 THEN A1 = (S1 / S) * O1
130  IF N = 8 THEN S1 = 0
135  NEXT Y1
140 A2 = (S1 / S) * O1
145  PRINT "% OF FIRST BAND="A1
150  PRINT "% OF SECOND BAND="A2
155  REM  CALC.% POWER OF NEXT FIVE BANDS
160 N = 0
165 S2 = 0
170  FOR Y2 = 0 TO 69
175 X2 =  PEEK (16384 + 2 * J + Y2)
180 P2 = X2 ^ 2
185 S2 = S2 + P2
190 N = N + 1
195  IF N < 14 THEN  GOTO 255
200  IF N = 14 THEN A3 = (S2 / S) * O1
205  IF N = 15 THEN S2 = 0
210  IF N < 28 THEN  GOTO 255
215  IF N = 28 THEN A4 = (S2 / S) * O1
220  IF N = 29 THEN S2 = 0
225  IF N < 42 THEN  GOTO 255
230  IF N = 42 THEN A5 = (S2 / S) * O1
235  IF N = 43 THEN S2 = 0
240  IF N < 56 THEN  GOTO 255
245  IF N = 56 THEN A6 = (S2 / S) * O1
250  IF N = 57 THEN S2 = 0
255  NEXT Y2
260 A7 = (S2 / S) * O1
265  PRINT "% OF THIRD BAND="A3
270  PRINT "% OF FOURTH BAND="A4
```

```
275  PRINT "% OF FIFTH BAND="A5
280  PRINT "% OF SIXTH BAND="A6
285  PRINT "% OF SEVENTH BAND="A7
290  REM  CALC. % POWER OF NEXT FIVE BANDS
295 N = 0
300 S3 = 0
305  FOR Y3 = 0 TO 144
310 X3 =  PEEK (16384 + 2 * J + 5 * K + Y3)
315 P3 = X3 ^ 2
320 S3 = S3 + P3
325 N = N + 1
330  IF N < 29 THEN  GOTO 390
335  IF N = 29 THEN A8 = (S3 / S) * O1
340  IF N = 30 THEN S3 = 0
345  IF N < 56 THEN  GOTO 390
350  IF N = 56 THEN A9 = (S3 / S) * O1
355  IF N = 57 THEN S3 = 0
360  IF N < 87 THEN  GOTO 390
365  IF N = 87 THEN B1 = (S3 / S) * O1
370  IF N = 87 THEN S3 = 0
375  IF N < 116 THEN  GOTO 390
380  IF N = 116 THEN B2 = (S3 / S) * O1
385  IF N = 117 THEN S3 = 0
390  NEXT Y3
395 B3 = (S3 / S) * O1
400  PRINT "% OF EIGHTH BAND="A8
405  PRINT "% OF NINTH BAND="A9
410  PRINT "% OF TENTH BAND="B1
415  PRINT "% OF ELEVENTH BAND="B2
420  PRINT "% OF TWELTH BAND="B3
425  REM  CALC. % POWER OF LAST BAND
430 S4 = 0
435  FOR Y4 = 0 TO 26
440 X4 =  PEEK (16384 + 2 * J + 5 * K + 5 * T + Y4)
445 P4 = X4 ^ 2
450 S4 = S4 + P4
455  NEXT Y4
460 B4 = (S4 / S) * 130
465  PRINT "% OF THERTEENTH BAND="B4
470  PRINT "TOTAL % BEFORE ROUND OFF IS T%="A1 + A2 + A3 + A4 + A5 + A6 + A7 + A
8 + A9 + B1 +
     B2 + B3 + B4
475 A1 =  INT ((A1 * O2) + 0.05) / O2
480 A2 =  INT ((A2 * O2) + 0.05) / O2
485 A3 =  INT ((A3 * O2) + 0.05) / O2
490 A4 =  INT ((A4 * O2) + 0.05) / O2
495 A5 =  INT ((A5 * O2) + 0.05) / O2
500 A6 =  INT ((A6 * O2) + 0.05) / O2
505 A7 =  INT ((A7 * O2) + 0.05) / O2
510 A8 =  INT ((A8 * O2) + 0.05) / O2
515 A9 =  INT ((A9 * O2) + 0.05) / O2
520 B1 =  INT ((B1 * O2) + 0.05) / O2
525 B2 =  INT ((B2 * O2) + 0.05) / O2
530 B3 =  INT ((B3 * O2) + 0.05) / O2
535 B4 =  INT ((B4 * O2) + 0.05) / O2
540  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
545  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
550  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
555  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
560  PRINT "B1="B1: PRINT "B2="B2: PRINT "B3="B3
565  PRINT "B4="B4
```

```
570 T% = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
575  PRINT "TOTAL % POWER IS T%="T%
580  PRINT D$"PR#0"
585  DIM S(20): DIM N(20): DIM R(20)
590 S(0) = A1:S(1) = A2:S(2) = A3:S(3) = A4:S(4) = A5:S(5) = A6
595 S(6) = A7:S(7) = A8:S(8) = A9:S(9) = B1:S(10) = B2:S(11) = B3:S(12) = B4
600  FOR I = 0 TO 12:N(I) =  INT (S(I))
605 R(I) = ((S(I) -  INT (S(I))) + .0005) * 02
610  POKE (16899 + I),N(I): POKE (16912 + I),R(I)
615  NEXT I
620  PRINT D$"RUN PIECHART,D1"


]LOAD MICROBUF

]LIST

5  PRINT  CHR$ (4);"PR#1"
10  PRINT  CHR$ (09);"GEI": PRINT  CHR$ (09);"GDER"
15  PRINT  CHR$ (4);"PR#0"
20  TEXT
25  HOME
30 S =  PEEK (16896)
35  IF S <  > 1 GOTO 45
40  POKE 16896,0: PRINT  CHR$ (4)"RUN B-AVERAGE,D1"
45  END
```

```
]LOAD PLOT-B1

]LIST

5 D$ =  CHR$ (4)
10  HOME : INVERSE : PRINT "WAIT FOR THE CALCULATIONS!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM  O1 IS FOR THE % AND O2 IS FOR ROUND OFF
25  FOR Y = 0 TO 255
30 X =  PEEK (18432 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  HOME
55  PRINT D$"PR#1"
60  PRINT "TOTAL POWER="S
65  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ. INTERVALS OF THE HISTOGRAM
70  READ J,K,T,U
75  DATA  7,14,29,27
80  REM  CALC. % OF FIRST TWO BANDS
85 N = 0
90 S1 = 0
95  FOR Y1 = 0 TO 2 * J
100 X1 =  PEEK (18432 + Y1)
105 P1 = X1 ^ 2
110 S1 = S1 + P1
115 N = N + 1
120  IF N < 7 THEN  GOTO 135
125  IF N = 7 THEN A1 = (S1 / S) * O1
130  IF N = 8 THEN S1 = 0
135  NEXT Y1
140 A2 = (S1 / S) * O1
145  PRINT "% OF FIRST BAND="A1
150  PRINT "% OF SECOND BAND="A2
155  REM  CALC.% POWER OF NEXT FIVE BANDS
160 N = 0
165 S2 = 0
170  FOR Y2 = 0 TO 69
175 X2 =  PEEK (18432 + 2 * J + Y2)
180 P2 = X2 ^ 2
185 S2 = S2 + P2
190 N = N + 1
195  IF N < 14 THEN  GOTO 255
200  IF N = 14 THEN A3 = (S2 / S) * O1
205  IF N = 15 THEN S2 = 0
210  IF N < 28 THEN  GOTO 255
215  IF N = 28 THEN A4 = (S2 / S) * O1
220  IF N = 29 THEN S2 = 0
225  IF N < 42 THEN  GOTO 255
230  IF N = 42 THEN A5 = (S2 / S) * O1
235  IF N = 43 THEN S2 = 0
240  IF N < 56 THEN  GOTO 255
245  IF N = 56 THEN A6 = (S2 / S) * O1
250  IF N = 57 THEN S2 = 0
255  NEXT Y2
260 A7 = (S2 / S) * O1
265  PRINT "% OF THIRD BAND="A3
270  PRINT "% OF FOURTH BAND="A4
```

```
275  PRINT "% OF FIFTH BAND="A5
280  PRINT "% OF SIXTH BAND="A6
285  PRINT "% OF SEVENTH BAND="A7
290  REM  CALC. % POWER OF NEXT FIVE BANDS
295 N = 0
300 S3 = 0
305  FOR Y3 = 0 TO 144
310 X3 =  PEEK (18432 + 2 * J + 5 * K + Y3)
315 P3 = X3 ^ 2
320 S3 = S3 + P3
325 N = N + 1
330  IF N < 29 THEN  GOTO 390
335  IF N = 29 THEN A8 = (S3 / S) * 01
340  IF N = 30 THEN S3 = 0
345  IF N < 56 THEN  GOTO 390
350  IF N = 56 THEN A9 = (S3 / S) * 01
355  IF N = 57 THEN S3 = 0
360  IF N < 87 THEN  GOTO 390
365  IF N = 87 THEN B1 = (S3 / S) * 01
370  IF N = 87 THEN S3 = 0
375  IF N < 116 THEN  GOTO 390
380  IF N = 116 THEN B2 = (S3 / S) * 01
385  IF N = 117 THEN S3 = 0
390  NEXT Y3
395 B3 = (S3 / S) * 01
400  PRINT "% OF EIGHTH BAND="A8
405  PRINT "% OF NINTH BAND="A9
410  PRINT "% OF TENTH BAND="B1
415  PRINT "% OF ELEVENTH BAND="B2
420  PRINT "% OF TWELTH BAND="B3
425  REM  CALC. % POWER OF LAST BAND
430 S4 = 0
435  FOR Y4 = 0 TO 26
440 X4 =  PEEK (18432 + 2 * J + 5 * K + 5 * T + Y4)
445 P4 = X4 ^ 2
450 S4 = S4 + P4
455  NEXT Y4
460 B4 = (S4 / S) * 130
465  PRINT "% OF THERTEENTH BAND="B4
470  PRINT "TOTAL % BEFORE ROUND OFF IS T%="A1 + A2 + A3 + A4 + A5 + A6 + A7 + A
8 + A9 + B1 +
     B2 + B3 + B4
475 A1 =  INT ((A1 * 02) + 0.05) / 02
480 A2 =  INT ((A2 * 02) + 0.05) / 02
485 A3 =  INT ((A3 * 02) + 0.05) / 02
490 A4 =  INT ((A4 * 02) + 0.05) / 02
495 A5 =  INT ((A5 * 02) + 0.05) / 02
500 A6 =  INT ((A6 * 02) + 0.05) / 02
505 A7 =  INT ((A7 * 02) + 0.05) / 02
510 A8 =  INT ((A8 * 02) + 0.05) / 02
515 A9 =  INT ((A9 * 02) + 0.05) / 02
520 B1 =  INT ((B1 * 02) + 0.05) / 02
525 B2 =  INT ((B2 * 02) + 0.05) / 02
530 B3 =  INT ((B3 * 02) + 0.05) / 02
535 B4 =  INT ((B4 * 02) + 0.05) / 02
540  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
545  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
550  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
555  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
560  PRINT "B1="31: PRINT "32="B2: PRINT "B3="B3
565  PRINT "B4="B4

570 T% = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
575  PRINT "TOTAL % POWER IS T%="T%
580  PRINT D$"PR#0"
585  DIM S(20): DIM N(20): DIM R(20)
590 S(0) = A1:S(1) = A2:S(2) = A3:S(3) = A4:S(4) = A5:S(5) = A6
595 S(6) = A7:S(7) = A8:S(8) = A9:S(9) = B1:S(10) = B2:S(11) = B3:S(12) = B4
600  FOR I = 0 TO 12:N(I) =  INT (S(I))
605 R(I) = ((S(I) -  INT (S(I))) + .0005) * 02
610  POKE (16925 + I),N(I): POKE (16938 + I),R(I)
615  NEXT I
620  PRINT D$"RUN PIECHART2.D1"
```

```
]LOAD PIECHART

]LIST

5  REM  *************************
10  REM  *PIECHART  BY.CHIANG    *
15  REM  *************************
20  REM  A=1ST%,B=2ND%.......ETC.
25  DIM S(20): DIM N(20): DIM R(20)
30  FOR I = 0 TO 12
35 N(I) =  PEEK (16899 + I)
40 R(I) =  PEEK (16912 + I)
45 S(I) = N(I) + R(I) / 10
50  NEXT I
55 M = S(0):L = S(1):K = S(2):J = S(3):I = S(4):H = S(5)
60 G = S(6):F = S(7):E = S(8):D = S(9):C = S(10):B = S(11):A = S(12)
65  REM  PORTION#=P
70 P = 13
75 P1 = P:P2 = P
80  PRINT  CHR$ (4);"BLOAD MAGICSPACE#"
85  POKE 232,0: POKE 233,96
90  DIM A(20),B(20),C(20),D(20)
95  DEF  FN X(Z) = RAD *  COS (Z) + XCTR
100  DEF  FN Y(Z) = RAD *  SIN (Z) / .85 + YCTR
105  HGR : HCOLOR= 3:ST = .1
110 XCTR = 70:YCTR = 80:RAD = 65
115  HPLOT XCTR,YCTR
120  FOR Z = 0 TO 6.3 STEP ST
125 X = RAD *  COS (Z) + XCTR
130 Y = RAD *  SIN (Z) / .85 + YCTR
135  HPLOT  TO X,Y
140  NEXT
145  IF P = 1 THEN  GOTO 245
150  IF P = 2 THEN T = A: GOTO 220
155  IF P = 3 THEN T = A + B: GOTO 220
160  IF P = 4 THEN T = A + B + C: GOTO 220
165  IF P = 5 THEN T = A + B + C + D: GOTO 220
170  IF P = 6 THEN T = A + B + C + D + E: GOTO 220
175  IF P = 7 THEN T = A + B + C + D + E + F: GOTO 220
180  IF P = 8 THEN T = A + B + C + D + E + F + G: GOTO 220
185  IF P = 9 THEN T = A + B + C + D + E + F + G + H: GOTO 220
190  IF P = 10 THEN T = A + B + C + D + E + F + G + H + I: GOTO 220
195  IF P = 11 THEN T = A + B + C + D + E + F + G + H + I + J: GOTO 220
200  IF P = 12 THEN T = A + B + C + D + E + F + G + H + I + J + K: GOTO 220
205  IF P = 13 THEN T = A + B + C + D + E + F + G + H + I + J + K + L: GOTO 220
210 Z = (A(P1 + 1) + A(P1)) / 2
215  RETURN
220 Z = 6.3 * T / 100:Z = Z * 10:Z =  INT (Z):Z = Z / 10
225  HPLOT  FN X(Z), FN Y(Z) TO XCTR,YCTR
230 A(P) = Z
235 P = P - 1
240  GOTO 145
245  IF P1 = 13 THEN Z = (6.3 + A(13)) / 2: GOTO 310
250  IF P1 = 12 THEN Z = (6.3 + A(12)) / 2: GOTO 310
255  IF P1 = 11 THEN Z = (6.3 + A(11)) / 2: GOTO 310
260  IF P1 = 10 THEN Z = (6.3 + A(10)) / 2: GOTO 310
265  IF P1 = 9 THEN Z = (6.3 + A(9)) / 2: GOTO 310
270  IF P1 = 8 THEN Z = (6.3 + A(8)) / 2: GOTO 310
```

```
275  IF P1 = 7 THEN Z = (6.3 + A(7)) / 2: GOTO 310
280  IF P1 = 6 THEN Z = (6.3 + A(6)) / 2: GOTO 310
285  IF P1 = 5 THEN Z = (6.3 + A(5)) / 2: GOTO 310
290  IF P1 = 4 THEN Z = (6.3 + A(4)) / 2: GOTO 310
295  IF P1 = 3 THEN Z = (6.3 + A(3)) / 2: GOTO 310
300  IF P1 = 2 THEN Z = (6.3 + A(2)) / 2: GOTO 310
305  GOTO 360
310 RAD = RAD / 2 + 10
315  SCALE= 1
320 F1 = 0
325 SL = 23 + F1
330  DRAW SL AT  FN X(Z), FN Y(Z)
335 F1 = F1 + 1
340 P1 = P1 - 1
345  IF P1 = 0 THEN  GOTO 360
350  GOSUB 210
355  GOTO 325
360  REM  EEG PIECHART
365 Y = 15:X = 155
370  FOR W = 1 TO 3
375  READ S
380  DRAW S AT X,Y
385 X = X + 5
390  NEXT W
395 X = X + 10
400  FOR V = 1 TO 10
405  READ S
410  DRAW S AT X,Y
415 X = X + 5
420  NEXT V
425  REM  DATA FOR TITLE
430  DATA  27,27,29,49,49,38,31,27,25,30,23,40,42
435  IF P2 = 13 THEN 500
440  IF P2 = 12 THEN 505
445  IF P2 = 11 THEN 510
450  IF P2 = 10 THEN 515
455  IF P2 = 9 THEN 520
460  IF P2 = 8 THEN 525
465  IF P2 = 7 THEN 530
470  IF P2 = 6 THEN 535
475  IF P2 = 5 THEN 540
480  IF P2 = 4 THEN 545
483  IF P2 = 3 THEN 550
490  IF P2 = 2 THEN 555
495  IF P2 = 1 THEN 560
500 B(13) =  INT (M / 10):C(13) =  INT (M - B(13) * 10):D(13) = (M - B(13) * 10
- C(13)) * 10

505 B(12) =  INT (L / 10):C(12) =  INT (L - B(12) * 10):D(12) = (L - B(12) * 10
- C(12)) * 10

510 B(11) =  INT (K / 10):C(11) =  INT (K - B(11) * 10):D(11) = (K - B(11) * 10
- C(11)) * 10

515 B(10) =  INT (J / 10):C(10) =  INT (J - B(10) * 10):D(10) = (J - B(10) * 10
- C(10)) * 10

520 B(9) =  INT (I / 10):C(9) =  INT (I - B(9) * 10):D(9) = (I - B(9) * 10 - C(9
)) * 10
525 B(8) =  INT (H / 10):C(8) =  INT (H - B(8) * 10):D(8) = (H - B(8) * 10 - C(8
)) * 10
530 B(7) =  INT (G / 10):C(7) =  INT (G - B(7) * 10):D(7) = (G - B(7) * 10 - C(7
)) * 10
```

```
]LIST 500,

500 B(13) =  INT (M / 10):C(13) =  INT (M - B(13) * 10):D(13) = (M - B(13) * 10
- C(13)) * 10

505 B(12) =  INT (L / 10):C(12) =  INT (L - B(12) * 10):D(12) = (L - B(12) * 10
- C(12)) * 10

510 B(11) =  INT (K / 10):C(11) =  INT (K - B(11) * 10):D(11) = (K - B(11) * 10
- C(11)) * 10

515 B(10) =  INT (J / 10):C(10) =  INT (J - B(10) * 10):D(10) = (J - B(10) * 10
- C(10)) * 10

520 B(9) =  INT (I / 10):C(9) =  INT (I - B(9) * 10):D(9) = (I - B(9) * 10 - C(9
)) * 10
525 B(8) =  INT (H / 10):C(8) =  INT (H - B(8) * 10):D(8) = (H - B(8) * 10 - C(8
)) * 10
530 B(7) =  INT (G / 10):C(7) =  INT (G - B(7) * 10):D(7) = (G - B(7) * 10 - C(7
)) * 10
535 B(6) =  INT (F / 10):C(6) =  INT (F - B(6) * 10):D(6) = (F - B(6) * 10 - C(6
)) * 10
540 B(5) =  INT (E / 10):C(5) =  INT (E - B(5) * 10):D(5) = (E - B(5) * 10 - C(5
)) * 10
545 B(4) =  INT (D / 10):C(4) =  INT (D - B(4) * 10):D(4) = (D - B(4) * 10 - C(4
)) * 10
550 B(3) =  INT (C / 10):C(3) =  INT (C - B(3) * 10):D(3) = (C - B(3) * 10 - C(3
)) * 10
555 B(2) =  INT (B / 10):C(2) =  INT (B - B(2) * 10):D(2) = (B - B(2) * 10 - C(2
)) * 10
560 B(1) =  INT (A / 10):C(1) =  INT (A - B(1) * 10):D(1) = (A - B(1) * 10 - C(1
)) * 10
565  REM  COMPENSATION FOR ROUNDOFF NUMBER
570  FOR N = 1 TO P2
575 D(N) =  INT (D(N) + 0.5)
580  NEXT N
585  PRINT D$"PR#1": PRINT B(8),C(8),D(8)
590  FOR J = 1 TO P2
595  IF B(J) = 0 THEN B(J) = 38
600  IF C(J) = 0 THEN C(J) = 26
605  IF D(J) = 0 THEN D(J) = 26
610  NEXT J
615  PRINT B(8),C(8),D(8): PRINT D$"PR#0"
620 Y = 35
625  REM  THE"FOR-NEXT" STATEMENT IS FROM P2 TO 1 AND NOT FROM 1 TO P2,BECAUSE A
=S(12),B=S(11
     ),C=S(10),...AND, IN THE DISPLAY OF THE RESULT WE WANT "A" FIRST
630  HOME
635  FOR T = P2 TO 1 STEP  - 1
640 X = 145
645  REM  DRAW "A=","B=","C=",...,"M="
650  FOR Q = 1 TO 2
655  READ S
660  DRAW S AT X,Y
665 X = X + 6
670  NEXT Q
675  REM  THE THREE VARIABLES,B(T),C(T),D(T), ARE USED TO DISPLAY THE THREE DIGI
TS OF THE VAL
     UE OF THE PORTION AREA IN THE PIECHART
```

```
]LIST 685,

685 X = X + 5
690  DRAW C(T) + 11 AT X,Y
695 X = X + 5
700  DRAW 10 AT X,Y
705 X = X + 5
710  DRAW D(T) + 11 AT X,Y
715 X = X + 5
720  DRAW 3 AT X,Y
725 X = X + 10
730  REM  DRAW THE NUMBERS TO EXPRESS THE FREQ. INTERVAL AND THE UNITS
735  FOR U = 1 TO 11
740  READ S
745  DRAW S AT X,Y
750 X = X + 5
755  NEXT U
760 Y = Y + 9
765  NEXT T
770  REM  THERE ARE 13 LINES OF DATA, ONE FOR EACH PORTION OF THE PIECHAR
775  DATA  23,21,49,49,49,37,9,49,12,10,20,30,48
780  DATA  24,21,49,12,10,20,9,49,14,10,19,30,48
785  DATA  25,21,49,14,10,19,9,49,18,10,18,30,48
790  DATA  26,21,49,13,10,18,9,12,12,10,17,30,48
795  DATA  27,21,12,12,10,17,9,12,16,10,16,30,48
800  DATA  28,21,12,16,10,16,9,12,20,10,15,30,48
805  DATA  29,21,12,20,10,15,9,13,14,10,14,30,48
810  DATA  30,21,13,14,10,14,9,14,12,10,14,30,48
815  DATA  31,21,14,12,10,14,9,14,20,10,15,30,48
820  DATA  32,21,14,20,10,15,9,15,18,10,16,30,48
825  DATA  33,21,15,18,10,16,9,16,16,10,16,30,48
830  DATA  34,21,16,16,10,16,9,17,14,10,17,30,48
835  DATA  35,21,17,14,10,17,9,18,12,10,12,30,48
840  PRINT  CHR$ (4);"PR#1"
845  PRINT  CHR$ (09);"GMDEL"
850  PRINT  CHR$ (4);"PR#0"
855  TEXT
860  HOME
865 S =  PEEK (16896)
870  IF S <  > 1 GOTO 860
875  POKE 16896,0: PRINT  CHR$ (4)"RUN 3-AVERAGE,D1"
880  END
```

```
]LOAD PIECHART3


]LIST 5-500

5  REM  ***************************
10  REM  *PIECHART  BY.CHIANG     *
15  REM  ***************************
20  REM  A=1ST%.B=2ND%.......ETC.
25  DIM S(20): DIM N(20): DIM R(20)
30  FOR I = 0 TO 12
35 N(I) =  PEEK (16925 + I)
40 R(I) =  PEEK (16938 + I)
45 S(I) = N(I) + R(I) / 10
50  NEXT I
55 M = S(0):L = S(1):K = S(2):J = S(3):I = S(4):H = S(5)
60 G = S(6):F = S(7):E = S(8):D = S(9):C = S(10):B = S(11):A = S(12)
65  REM  PORTION#=P
70 P = 13
75 P1 = P:P2 = P
80  PRINT  CHR$ (4);"BLOAD MAGICSPACE#"
85  POKE 232,0: POKE 233,96
90  DIM A(20),B(20),C(20),D(20)
95  DEF  FN X(Z) = RAD *  COS (Z) + XCTR
100  DEF  FN Y(Z) = RAD *  SIN (Z) / .85 + YCTR
105  HGR : HCOLOR= 3:ST = .1
110 XCTR = 70:YCTR = 80:RAD = 65
115  HPLOT XCTR,YCTR
120  FOR Z = 0 TO 6.3 STEP ST
125 X = RAD *  COS (Z) + XCTR
130 Y = RAD *  SIN (Z) / .85 + YCTR
135  HPLOT  TO X,Y
140  NEXT
145  IF P = 1 THEN  GOTO 245
150  IF P = 2 THEN T = A: GOTO 220
155  IF P = 3 THEN T = A + B: GOTO 220
160  IF P = 4 THEN T = A + B + C: GOTO 220
165  IF P = 5 THEN T = A + B + C + D: GOTO 220
170  IF P = 6 THEN T = A + B + C + D + E: GOTO 220
175  IF P = 7 THEN T = A + B + C + D + E + F: GOTO 220
180  IF P = 8 THEN T = A + B + C + D + E + F + G: GOTO 220
185  IF P = 9 THEN T = A + B + C + D + E + F + G + H: GOTO 220
190  IF P = 10 THEN T = A + B + C + D + E + F + G + H + I: GOTO 220
195  IF P = 11 THEN T = A + B + C + D + E + F + G + H + I + J: GOTO 220
200  IF P = 12 THEN T = A + B + C + D + E + F + G + H + I + J + K: GOTO 220
205  IF P = 13 THEN T = A + B + C + D + E + F + G + H + I + J + K + L: GOTO 220
210 Z = (A(P1 + 1) + A(P1)) / 2
215  RETURN
220 Z = 6.3 + T / 100:Z = Z * 10:Z =  INT (Z):Z = Z / 10
225  HPLOT  FN X(Z), FN Y(Z) TO XCTR,YCTR
230 A(P) = Z
235 P = P - 1
240  GOTO 145
245  IF P1 = 13 THEN Z = (6.3 + A(13)) / 2: GOTO 310
250  IF P1 = 12 THEN Z = (6.3 + A(12)) / 2: GOTO 310
255  IF P1 = 11 THEN Z = (6.3 + A(11)) / 2: GOTO 310
260  IF P1 = 10 THEN Z = (6.3 + A(10)) / 2: GOTO 310
265  IF P1 = 9 THEN Z = (6.3 + A(9)) / 2: GOTO 310
270  IF P1 = 8 THEN Z = (6.3 + A(8)) / 2: GOTO 310
```

```
275  IF P1 = 7 THEN Z = (6.3 + A(7)) / 2: GOTO 310
280  IF P1 = 6 THEN Z = (6.3 + A(6)) / 2: GOTO 310
285  IF P1 = 5 THEN Z = (6.3 + A(5)) / 2: GOTO 310
290  IF P1 = 4 THEN Z = (6.3 + A(4)) / 2: GOTO 310
295  IF P1 = 3 THEN Z = (6.3 + A(3)) / 2: GOTO 310
300  IF P1 = 2 THEN Z = (6.3 + A(2)) / 2: GOTO 310
305  GOTO 360
310 RAD = RAD / 2 + 10
315  SCALE= 1
320 F1 = 0
325 SL = 23 + F1
330  DRAW SL AT  FN X(Z), FN Y(Z)
335 F1 = F1 + 1
340 P1 = P1 - 1
345  IF P1 = 0 THEN  GOTO 360
350  GOSUB 210
355  GOTO 325
360  REM  EEG PIECHART
365 Y = 15:X = 155
370  FOR W = 1 TO 3
375  READ S
380  DRAW S AT X,Y
385 X = X + 5
390  NEXT W
395 X = X + 10
400  FOR V = 1 TO 10
405  READ S
410  DRAW S AT X,Y
415 X = X + 5
420  NEXT V
425  REM  DATA FOR TITLE
430  DATA  27,27,29,49,49,38,31,27,25,30,23,40,42
435  IF P2 = 13 THEN 500
440  IF P2 = 12 THEN 505
445  IF P2 = 11 THEN 510
450  IF P2 = 10 THEN 515
455  IF P2 = 9 THEN 520
460  IF P2 = 8 THEN 525
465  IF P2 = 7 THEN 530
470  IF P2 = 6 THEN 535
475  IF P2 = 5 THEN 540
480  IF P2 = 4 THEN 545
485  IF P2 = 3 THEN 550
490  IF P2 = 2 THEN 555
495  IF P2 = 1 THEN 560
500 B(13) =  INT (M / 10):C(13) =  INT (M - B(13) * 10):D(13) = (M - B(13) * 10
- C(13)) * 10
```

```
]LIST 500-680

500 B(13) =  INT (M / 10):C(13) =  INT (M - B(13) * 10):D(13) = (M - B(13) * 10
 - C(13)) * 10

505 B(12) =  INT (L / 10):C(12) =  INT (L - B(12) * 10):D(12) = (L - B(12) * 10
 - C(12)) * 10

510 B(11) =  INT (K / 10):C(11) =  INT (K - B(11) * 10):D(11) = (K - B(11) * 10
 - C(11)) * 10

515 B(10) =  INT (J / 10):C(10) =  INT (J - B(10) * 10):D(10) = (J - B(10) * 10
 - C(10)) * 10

520 B(9) =  INT (I / 10):C(9) =  INT (I - B(9) * 10):D(9) = (I - B(9) * 10 - C(9
)) * 10
525 B(8) =  INT (H / 10):C(8) =  INT (H - B(8) * 10):D(8) = (H - B(8) * 10 - C(8
)) * 10
530 B(7) =  INT (G / 10):C(7) =  INT (G - B(7) * 10):D(7) = (G - B(7) * 10 - C(7
)) * 10
535 B(6) =  INT (F / 10):C(6) =  INT (F - B(6) * 10):D(6) = (F - B(6) * 10 - C(6
)) * 10
540 B(5) =  INT (E / 10):C(5) =  INT (E - B(5) * 10):D(5) = (E - B(5) * 10 - C(5
)) * 10
545 B(4) =  INT (D / 10):C(4) =  INT (D - B(4) * 10):D(4) = (D - B(4) * 10 - C(4
)) * 10
550 B(3) =  INT (C / 10):C(3) =  INT (C - B(3) * 10):D(3) = (C - B(3) * 10 - C(3
)) * 10
555 B(2) =  INT (B / 10):C(2) =  INT (B - B(2) * 10):D(2) = (B - B(2) * 10 - C(2
)) * 10
560 B(1) =  INT (A / 10):C(1) =  INT (A - B(1) * 10):D(1) = (A - B(1) * 10 - C(1
)) * 10
565  REM  COMPENSATION FOR ROUNDOFF NUMBER
570  FOR N = 1 TO P2
575 D(N) =  INT (D(N) + 0.5)
580  NEXT N
585  FOR J = 1 TO P2
590  IF B(J) = 0 THEN B(J) = 38
595  IF C(J) = 0 THEN C(J) = 26
600  IF D(J) = 0 THEN D(J) = 26
605  NEXT J
610 Y = 35
615  REM  THE"FOR-NEXT" STATEMENT IS FROM P2 TO 1 AND NOT FROM 1 TO P2,BECAUSE A
=S(12),B=S(11
     ),C=S(10),...AND, IN THE DISPLAY OF THE RESULT WE WANT "A" FIRST
620  HOME
625  FOR T = P2 TO 1 STEP  - 1
630 X = 145
635  REM  DRAW "A=","B=","C=",...,"M="
640  FOR Q = 1 TO 2
645  READ S
650  DRAW S AT X,Y
655 X = X + 6
660  NEXT Q
665  REM  THE THREE VARIABLES,B(T),C(T),D(T), ARE USED TO DISPLAY THE THREE DIGI
TS OF THE VAL
     UE OF THE PORTION AREA IN THE PIECHART
670  DRAW B(T) + 11 AT X,Y
675 X = X + 5
680  DRAW C(T) + 11 AT X,Y
```

```
]LIST 685,

685 X = X + 5
690  DRAW 10 AT X,Y
695 X = X + 5
700  DRAW D(T) + 11 AT X,Y
705 X = X + 5
710  DRAW 3 AT X,Y
715 X = X + 10
720  REM  DRAW THE NUMBERS TO EXPRESS THE FREQ. INTERVAL AND THE UNITS
725  FOR U = 1 TO 11
730  READ S
735  DRAW S AT X,Y
740 X = X + 5
745  NEXT U
750 Y = Y + 9
755  NEXT T
760  REM  THERE ARE 13 LINES OF DATA, ONE FOR EACH PORTION OF THE PIECHART
765  DATA  23,21,49,49,49,37,9,49,12,10,20,30,48
770  DATA  24,21,49,12,10,20,9,49,14,10,19,30,48
775  DATA  25,21,49,14,10,19,9,49,18,10,18,30,48
780  DATA  26,21,49,18,10,18,9,12,12,10,17,30,48
785  DATA  27,21,12,12,10,17,9,12,16,10,16,30,48
790  DATA  28,21,12,16,10,16,9,12,20,10,15,30,48
795  DATA  29,21,12,20,10,15,9,13,14,10,14,30,48
800  DATA  30,21,13,14,10,14,9,14,12,10,14,30,48
805  DATA  31,21,14,12,10,14,9,14,20,10,15,30,48
810  DATA  32,21,14,20,10,15,9,15,18,10,16,30,48
815  DATA  33,21,15,18,10,16,9,16,16,10,16,30,48
820  DATA  34,21,16,16,10,16,9,17,14,10,17,30,48
825  DATA  35,21,17,14,10,17,9,18,12,10,12,30,48
830  PRINT  CHR$ (4);"PR#1"
835  PRINT  CHR$ (09);"GMDEL"
840  PRINT  CHR$ (4);"PR#0"
845  TEXT
850  HOME
855  END
```

Variant 2 - Processing

Before Running

```
]CATALOG

DISK VOLUME 254

*A 004 CHOICE1 ✓
*A 003 A-AVERAGE
*A 004 B-AVERAGE
*A 005 FINAL-AVE.A
*A 005 FINAL-AVE.B
*A 005 SEE FIN.AVE.A
*A 005 SEE FIN.AVE.B
*A 013 AXIS
*A 013 AXISB
*A 017 PLOT-A
*A 016 PLOT-B
*A 002 MICROBUF
*A 016 PLOT-A1
*A 012 PLOT-B1
*A 019 PIECHART
*A 019 PIECHARTB
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEHI.OBJO
 B 004 AVE.8
*B 002 CLEARB.OBJO
*B 005 AVELOTABB
*B 002 AVEHIB.OBJO
 B 004 AVEB.8
*B 004 MAGICSPACE#
```

After Running

```
]CATALOG

DISK VOLUME 254

*A 004 CHOICE1 ✓
*A 003 A-AVERAGE
*A 004 B-AVERAGE
*A 005 FINAL-AVE.A
*A 005 FINAL-AVE.B
*A 005 SEE FIN.AVE.A
*A 005 SEE FIN.AVE.B
*A 013 AXIS
*A 013 AXISB
*A 017 PLOT-A
*A 016 PLOT-B
*A 002 MICROBUF
*A 016 PLOT-A1
*A 012 PLOT-B1
*A 019 PIECHART
*A 019 PIECHARTB
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEHI.OBJO
*B 002 CLEARB.OBJO
*B 005 AVELOTABB
*B 002 AVEHIB.OBJO
*B 004 MAGICSPACE#
 B 004 AVE.0
 B 004 AVE.1
 B 004 AVE.2
 B 004 AVE.3
 B 004 AVE.4
 B 004 AVE.5
 B 004 AVE.6
 B 004 AVE.7
 B 004 AVE.AVE.A
 B 004 AVEB.0
 B 004 AVEB.1
 B 004 AVEB.2
 B 004 AVEB.3
 B 004 AVEB.4
 B 004 AVEB.5
 B 004 AVEB.6
 B 004 AVEB.7
 B 004 AVE.AVE.B
```

```
]LOAD CHOICE1

]LIST

5  HOME
10  FOR D = 0 TO 120
15  HTAB 3: VTAB 10: INVERSE : PRINT "E E G  PROCESEING
20  HTAB 3: VTAB 11: PRINT "   M A R I N E L A  L A G U N A -1984 ": NORMAL
25  HTAB 6: VTAB 13: PRINT "V A R I A N T  2"
30  NEXT D: HOME
35 D$ =  CHR$ (4)
40  PRINT "WHICH CHANNEL DO YOU WANT TO PROCESS(AVERAGE,HISTOGRAM)?
45  PRINT
50  PRINT "****************************************"
55  VTAB 12: PRINT "PRESS "1" FOR CHANNEL A"
60  VTAB 13: PRINT "PRESS "2" FOR CHANNEL B"
65  VTAB 14: PRINT "PRESS "3" FOR BOTH CHAN. A AND B"
70  POKE 16896,0
75  VTAB 15: GET A$: IF  VAL (A$) <  > 1 THEN  GOTO 85
80  GOTO 115
85  IF  VAL (A$) <  > 2 THEN  GOTO 95
90  GOTO 105
95  IF  VAL (A$) <  > 3 THEN  GOTO 75
100  GOTO 125
105  PRINT D$: PRINT D$"RUN B-AVERAGE,D1"
110  END
115  PRINT D$: PRINT D$"RUN A-AVERAGE,D1"
120  END
125  POKE 16896,1
130  PRINT D$: PRINT D$"RUN A-AVERAGE,D1"
```

Variant 3 - Data Acquisition

```
]CATALOG

DISK VOLUME 254

*A 004 AUTODUAL
*A 002 CALIBRATE
*A 003 AUTODUAL1
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 LOWALLTAB
*B 002 HIALL.OBJ0
```

```
]LOAD CALIBRATE

]LIST

5 D$ =  CHR$ (4)
10  POKE 37272,1
15  CALL 3064
20  POKE 37272,0
25  PRINT D$"BSAVE CT,A$9190,L$70,D1"
30  PRINT D$"BSAVE CS,A$1F80,L$28,D1"
35  POKE 37272,1
40  TEXT : HOME
45  PRINT D$"RUN AUTODUAL1,D1"

]LOAD AUTODUAL

]LIST

5  HOME
10  HTAB 3: VTAB 10: INVERSE : PRINT "E E 3  DATA AQUISITION
15  HTAB 3: VTAB 11: PRINT "    M A R I N E L A  L A G U N A -1984": NORMAL
20  HTAB 6: VTAB 13: PRINT "V A R I A N T  1"
25  HTAB 6: VTAB 15: FLASH : PRINT "LOADING BINARY FILES": NORMAL
30 D$ =  CHR$ (4)
35  PRINT D$"BLOAD BPR,D1"
40  PRINT D$"BLOAD M4A,D1"
45  PRINT D$"BLOAD M4B,D1"
50  PRINT D$"BLOAD M4C,D1"
55  PRINT D$"BLOAD CT,D1"
60  PRINT D$"BLOAD CS,D1"
65  POKE 37375,4: POKE 37272,1
70  POKE 37375,0
75  HOME
80  PRINT "DO CALIBRATION OF THE SCOPE:"
85  PRINT "1.ADJUST TIME BASE TO 0.2 SEC/DIV"
90  PRINT "2.ADJUST SCALE TO 4.0 VOLT/DIV"
95  PRINT "3.ADJUST DISPLAY>00001 BY PRESSING M KEY;ADJUST DISPLAY>00001 BY PRES
SING < CHAR."

100  PRINT "4.PRESS CTRL SHIFT P TO GO OUT FROM CALIBRATION MODE"
105  PRINT "5.I AM READY"
110  PRINT "PRESS 5 IF YOU ARE READY"
115  VTAB (12): GET A$: IF  VAL (A$) <  > 5 THEN  GOTO 115
120  PRINT D$
125  PRINT D$"RUN CALIBRATE,D1"

]LOAD AUTODUAL1

]LIST

5 D$ =  CHR$ (4)
10  PRINT D$"BLOAD LOWALLTAB,D1"
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255): PRINT D$"BLOAD HIALL.OBJ0,D1"
20  HOME : PRINT "THE MAXIMUM NO. OF SWEEPS CAN BE 64"
25  VTAB 8: INPUT "TOTAL NO.OF SWEEPS";T
30  REM  T=NO.OF BLOCKS OF 1024 BYTES
35  IF T < 65 THEN  GOTO 55
40  HOME : VTAB 8: INVERSE : PRINT "T IS TOO BIG!": NORMAL
45  FOR D = 1 TO 2000: NEXT D: REM  DELAY
50  GOTO 20
55  FOR Y = 0 TO  INT (T / 8) - ((T / 8) =  INT (T / 8))
60  POKE 843,Y * 2: POKE 844,0
65  FOR X = 1 TO 8
70  POKE 15312,209: POKE 15313,193
73  POKE 15314,194: POKE 15315,211
80  POKE 15316,128: POKE 37272,1
85  CALL 3064
90  PRINT D$: POKE 37272,0
95  CALL 768: POKE 37272,1
100  NEXT X: NEXT Y: HOME : POKE 34,0: POKE 49233,0
105  END
```

Variant 3 - Data Processing

```
]CATALOG

DISK VOLUME 254

*A 005 SPC
*A 004 SPC1
*A 003 A-AVERAGE
*A 005 FINAL-AVE.A
*A 004 SEE FIN.AVE.A
*A 013 AXIS
*A 016 PLOT-A
*A 003 MICROBUF
*A 004 B-AVERAGE
*A 005 FINAL-AVE.B
*A 004 SEE FIN.AVE.B
*A 013 AXISB
*A 016 PLOT-B
*B 031 BPR
*B 022 M4A
*B 018 M4B
*B 023 M4C
 B 002 CT
 B 002 CS
*B 002 RETTAB
*B 002 OPEN.OBJO
*B 002 SAVESPC.OBJO
*B 002 SWEEP.OBJO
*B 002 SVTAB
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEHI.OBJO
*B 002 CLEARB.OBJO
*B 005 AVELOTABB
*B 002 AVEHIB.OBJO
*B 004 MAGICSPACE#
 B 004 AVE.0
 B 004 AVE.1
 B 004 AVE.2
 B 004 AVE.3
 B 004 AVE.4
 B 004 AVE.5
 B 004 AVE.6
 B 004 AVE.7
 B 004 AVE.AVE.A
 B 004 AVEB.0
 B 004 AVEB.1
 B 004 AVEB.2
 B 004 AVEB.3
 B 004 AVEB.4
 B 004 AVEB.5
 B 004 AVEB.6
 B 004 AVEB.7
 B 004 AVE.AVE.B
```

```
]LOAD SPC

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 10: INVERSE : PRINT "E E G  DATA PROCESSING
15  VTAB 11: PRINT "   M A R I N E L A  L A G U N A -1984   ": NORMAL
20  HTAB 6: VTAB 15: FLASH : PRINT "LOADING BINARY FILES": NORMAL
25  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRIN
T D$"DLOAD M4
     C,D1"
30  PRINT D$"BLOAD CT,D1": PRINT D$"ELOAD CS,D1"
35  PRINT D$"RUN SPC1,D1"

]LOAD SPC1

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 10: INVERSE : PRINT "128 SPECTRUMS,64 FOR EACH CHANNEL,WILL BE D
ONE NOW AND W
     ILL BE SAVED ON THE DATA DISK                                ": NORMAL
15  FOR Y = 0 TO 7
20  PRINT D$"BLOAD RETTAB,D1": POKE 823,Y * 2: POKE 824,0: CALL 768
25  PRINT D$"BLOAD OPEN.OBJ0,D1": PRINT D$"BLOAD SAVESPC.OBJ0,D1"
30 N =  PEEK ( - 16255):N =  PEEK ( - 16255): PRINT D$"BLOAD SWEEP.OBJ0,D1"
35  FOR X = 0 TO 15: CALL 812
40  POKE 15312,209: POKE 15313,193: POKE 15314,160: POKE 15315,207
45  POKE 15316,181: POKE 15317,160: POKE 15318,141: POKE 15319,211
50  POKE 15320,155: POKE 15321,128: POKE 37375,4: POKE 37272,1
55  CALL 3064: PRINT D$: POKE 37272,0: CALL 768
60  POKE 37272,1: NEXT X
65  PRINT D$"DLOAD SVTAB,D1"
70  POKE 823,Y * 2 + 16: POKE 824,0: CALL 768: NEXT Y
75  HOME : POKE 34,0: POKE 49233,0
80  PRINT D$: PRINT D$"RUN A-AVERAGE,D1"

]LOAD A-AVERAGE

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.A W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                              ": NORMAL
15  FOR Y = 0 TO 7: PRINT D$"BLOAD RETTAB1,D1"
20  POKE 823,Y * 2 + 16: CALL 768
25  PRINT D$"BLOAD CLEAR.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTAB,D1"
30 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
35  PRINT D$"BLOAD AVEHI.OBJ0,D1"
40  CALL 768: TEXT
45  PRINT D$"BSAVE AVE."Y",A$4000,L$200 ,D1"
50  NEXT Y
55  PRINT D$"RUN FINAL-AVE.A,D1"
```

```
]LOAD FINAL-AVE.A

]LIST

5  REM  THIS PROGRAM LOADS CH.A AVERAGES IN SUPERRAM II, IN THE RIGHT POSITION F
ROM THE POINT
      OF VUE OF THE M/L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.A
10 D$ =  CHR$ (4)
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
20  FOR Y = 0 TO 7
25  REM  53248=$D000
30 Z = 53248 + Y * 1024
35  PRINT D$"BLOAD AVE."Y",A"Z",D1"
40  REM  NOW,SUPERRAM II CONTAINS 8 AVERAGES FOR CH.A,FROM $D000-$D1FF,$D400-$D5
FF,$D800-$D9F
     F,$DC00-$DDFF,$E000-$E1FF,$E400-$E5FF,$E800-$E9FF,$EC00-$EDFF
45  PRINT D$"BLOAD CLEAR.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTAB,D1"
50  HOME : VTAB 15: INVERSE : PRINT "THE FINAL AVERAGE,OF 64 SPECTRUMS,FOR CHAN.
A WILL BE DON
     E NOW AND WILL BE SAVED ON THE DISK                                 ": NORMAL
55  PRINT D$"BLOAD AVEHI.OBJ0,D1"
60  CALL 768: TEXT
65  REM  NOW,THE AVERAGE OF 8 AVERAGES(EACH OF 8 SPC.)OF CH.A,IS LOCATED AT $400
0
70  PRINT D$"BSAVE AVE.AVE.A,A$4000,L$200,D1"
75  PRINT D$"RUN SEE FIN.AVE.A,D1"


]LOAD SEE FIN.AVE.A

]LIST

5  REM  THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.A ON THE SCREEN
10 D$ =  CHR$ (4): PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOA
D M4C,D1"
15  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
20  PRINT D$"BLOAD AVE.AVE.A,A$1000,D1"
25  POKE 37272,1: TEXT : HOME
30  POKE 15312,209: POKE 15313,193: POKE 15314,160: POKE 15315,207
35  POKE 15316,181: POKE 15317,160: POKE 15318,144
40  REM  POKE TITLE CHARACTERS
45  POKE 15319,193: POKE 15320,214: POKE 15321,197: POKE 15322,210
50  POKE 15323,193: POKE 15324,199: POKE 15325,197: POKE 15326,193: POKE 15327,1
73
55  POKE 15328,182: POKE 15329,180: POKE 15330,141: POKE 15331,155: POKE 15332,1
28
60  CALL 3064
65  PRINT D$: POKE 34,0: TEXT : HOME : POKE 16896,1: REM  LOC.16896 IS USED TO R
ECOGNIZE WHEN
      TO STOP
70  PRINT D$"RUN AXIS,D1"
```

```
]LOAD AXIS

]LIST

5 D$ =  CHR$ (4)
10  REM  DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15  HGR
20  HCOLOR= 3
25  HPLOT 15,144 TO 15,0
30  HPLOT 14,1 TO 16,1: HPLOT 13,2 TO 17,2
35  HPLOT 15,144 TO 279,144
40  HPLOT 277,142 TO 277,146: HPLOT 278,143 TO 278,145
45  PRINT D$"BLOAD MAGICSPACE#,D1"
50  POKE 232,0: POKE 233,96
55  SCALE= 1
60  ROT= 0
65  REM  A=Y CENTER UP:B=Y CENTER DOWN
70  REM  THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75  REM  DRAW UP NUMBERS
80 A = 146:B = 154
85 X = 9
90  FOR N = 1 TO 8
95  READ S
100  DRAW S AT X,A
105 X = X + 4
110  IF N < 4 GOTO 120
115  IF N = 4 THEN X = 23
120  NEXT N
125  DATA  49,37,49,49,49,14,10,19
130 X = 51
135  FOR N = 1 TO 8
140  READ S
145  DRAW S AT X,A
150 X = X + 4
155  IF N < 4 GOTO 165
160  IF N = 4 THEN X = 79
165  NEXT N
170  DATA  12,12,10,17,12,20,10,15
175 X = 122
180  FOR N = 1 TO 12
185  READ S
190  DRAW S AT X,A
195 X = X + 4
200  IF N < 4 GOTO 220
205  IF N = 4 THEN X = 180
210  IF N < 8 GOTO 220
215  IF N = 8 THEN X = 239
220  NEXT N
225  DATA  14,12,10,14,15,18,10,16,17,14,10,17
230  REM  DRAW DOWN NUMBERS
235 X = 16
240  FOR N = 1 TO 4
245  READ S
250  DRAW S AT X,B
255 X = X + 4
260  NEXT N
265  DATA  49,12,10,20
270 X = 37
```

```
275  FOR N = 1 TO 12
280  READ S
285  DRAW S AT X,B
290 X = X + 4
295  IF N < 4 GOTO 315
300  IF N = 4 THEN X = 65
305  IF N < 8 GOTO 315
310  IF N = 8 THEN X = 93
315  NEXT N
320  DATA  49,18,10,18,12,16,10,16,13,14,10,14
325 X = 151
330  FOR N = 1 TO 12
335  READ S
340  DRAW S AT X,B
345 X = X + 4
350  IF N < 4 GOTO 370
355  IF N = 4 THEN X = 209
360  IF N < 8 GOTO 370
365  IF N = 8 THEN X = 266
370  NEXT N
375  DATA  14,20,10,15,16,16,10,16,18,12,10,12
380  REM  THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
IS IS 3
385  REM  DRAW"HZ"ON THE HORIZONTAL AXIS
390 X = 260
395  FOR N = 1 TO 2
400  READ S
405  DRAW S AT X,136
410 X = X + 5
415  NEXT N
420  DATA  30,48
425  REM  MARK THE DIVISIONS ON THE VERTICAL AXIS
430 C = 144 / 240
435 X = 15
440  DIM Y(10): DIM Z(10)
445  FOR I = 1 TO 5
450  READ S
455 Y(I) = 144 - S * C - 2: REM  2 IS BECAUSE THE M/L WHICH DRAWS THE CHAR. CONS
IDERS THE CHA
     R. FROM THE TOP(ONE CHAR. HAS 5 VERT. POINTS)
460 Z(I) = Y(I) + (S * C) / (2 * I)
465  PRINT "S*C="S * C
470  PRINT "Y(I)="Y(I)
475  PRINT "Z(I)="Z(I)
480  NEXT I
485  DATA  40,80,120,160,200
490  FOR I = 1 TO 5
495  DRAW 9 AT X,Y(I)
500  DRAW 9 AT X,Z(I)
505  NEXT I
510  FOR I = 1 TO 5
515 X = 5
520  FOR N = 1 TO 2
525  READ S,U
530  DRAW S AT X,Y(I)
535  DRAW U AT X,Z(I)
540 X = X + 4
545  NEXT N
550  NEXT I
555  REM  S=49,15,49,19,12,13,12,17,13,37 AND U=49,13,49,17,12,37,12,15,12,19
560  DATA  49,49,15,13,49,49,19,17,12,12,13,37,12,12,17,15,13,12,37,19
565  REM   DRAW"% OF TOT. POWER"ON THE VERTICAL AXIS
```

```
570 X = 22:Y = 2
575  DRAW 3 AT X,Y
580 X = X + 8
585  FOR N = 1 TO 2
590  READ S
595  DRAW S AT X,Y
600 X = X + 5
605  NEXT N
610  DATA  37,28
615 X = X + 3
620  FOR N = 1 TO 4
625  READ S
630  DRAW S AT X,Y
635 X = X + 5
640  NEXT N
645  DATA  42,37,42,10
650 X = X + 3
655  FOR N = 1 TO 5
660  READ S
665  DRAW S AT X,Y
670 X = X + 5
675  IF N = 3 THEN X = X + 1
680  NEXT N
685  DATA  38,37,45,27,40
690  REM  DRAW TITLE OF THE GRAPH
695  REM  THE DISTANCE BETWEEN THE NUMBERS IN THE TITLE IS 5
700  REM  THE DISTANCE BETWEEN THE WORDS IN THE TITLE IS 6
705  REM  THE DISTANCE BETWEEN THE LETTERS OF THE TITLE IS 7;EXCEPTION IN THE LA
ST PART,"HZ..
    .",WHERE IS 5
710  SCALE= 1
715 X = 90:Y = 10
720  FOR N = 1 TO 3
725  READ S
730  DRAW S AT X,Y
735 X = X + 7
740  NEXT N
745  DATA  27,27,29
750 X = X + 6
755  FOR N = 1 TO 9
760  READ S
765  DRAW S AT X,Y
770 X = X + 7
775  NEXT N
780  DATA  30,31,41,42,37,29,40,23,35
785 X = X + 6
790  FOR N = 1 TO 3
795  READ S
800  DRAW S AT X,Y
805 X = X + 7
810  NEXT N
815  DATA  28,37,40
820 X = X + 6
825  FOR N = 1 TO 14
830  READ S
835  DRAW S AT X,Y
840 X = X + 5
845  NEXT N
850  DATA  37,9,18,12,10,12,49,30,48,9,25,30,10,23
855  PRINT D$"RUN PLOT-A,D1"
```

```
]LOAD PLOT-A

]LIST

5 D$ =  CHR$ (4)
10  HTAB 10: VTAB 23: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM  O1 IS FOR THE % AND O2 IS FOR HIST.AMPLIT.CALC.
25  FOR Y = 0 TO 255
30 X =  PEEK (16384 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  PRINT D$"PR#1"
55  PRINT : PRINT : PRINT "TOTAL POWER="S
60  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ.INTERVALS OF THE HISTOGRAM
65  READ J,K,T,U
70  DATA  7,14,29,27
75  REM  CALC. % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90  FOR Y1 = 0 TO 2 * J
95 X1 =  PEEK (16384 + Y1)
100 P1 = X1 ^ 2
105 S1 = S1 + P1
110 N = N + 1
115  IF N < 7 THEN  GOTO 130
120  IF N = 7 THEN A1 = (S1 / S) * O1
125  IF N = 8 THEN S1 = 0
130  NEXT Y1
135 A2 = (S1 / S) * O1
140  PRINT "% OF FIRST BAND="A1
145  PRINT "% OF SECOND BAND="A2
150  REM  CALC.% POWER OF NEXT FIVE BANDS
155 N = 0
160 S2 = 0
165  FOR Y2 = 0 TO 69
170 X2 =  PEEK (16384 + 2 * J + Y2)
175 P2 = X2 ^ 2
180 S2 = S2 + P2
185 N = N + 1
190  IF N < 14 THEN  GOTO 250
195  IF N = 14 THEN A3 = (S2 / S) * O1
200  IF N = 15 THEN S2 = 0
205  IF N < 28 THEN  GOTO 250
210  IF N = 28 THEN A4 = (S2 / S) * O1
215  IF N = 29 THEN S2 = 0
220  IF N < 42 THEN  GOTO 250
225  IF N = 42 THEN A5 = (S2 / S) * O1
230  IF N = 43 THEN S2 = 0
235  IF N < 56 THEN  GOTO 250
240  IF N = 56 THEN A6 = (S2 / S) * O1
245  IF N = 57 THEN S2 = 0
250  NEXT Y2
255 A7 = (S2 / S) * O1
260  PRINT "% OF THIRD BAND="A3
265  PRINT "% OF FOURTH BAND="A4
270  PRINT "% OF FIFTH BAND="A5
```

```
275  PRINT "% OF SIXTH BAND="A6
280  PRINT "% OF SEVENTH BAND="A7
285  REM  CALC. % POWER OF NEXT FIVE BANDS
290 N = 0
295 S3 = 0
300  FOR Y3 = 0 TO 144
305 X3 =  PEEK (16384 + 2 * J + 5 * K + Y3)
310 P3 = X3 ^ 2
315 S3 = S3 + P3
320 N = N + 1
325  IF N < 29 THEN  GOTO 385
330  IF N = 29 THEN A8 = (S3 / S) * 01
335  IF N = 30 THEN S3 = 0
340  IF N < 56 THEN  GOTO 385
345  IF N = 56 THEN A9 = (S3 / S) * 01
350  IF N = 57 THEN S3 = 0
355  IF N < 87 THEN  GOTO 385
360  IF N = 87 THEN B1 = (S3 / S) * 01
365  IF N = 87 THEN S3 = 0
370  IF N < 116 THEN  GOTO 385
375  IF N = 116 THEN B2 = (S3 / S) * 01
380  IF N = 117 THEN S3 = 0
385  NEXT Y3
390 B3 = (S3 / S) * 01
395  PRINT "% OF EIGHTH BAND="A8
400  PRINT "% OF NINTH BAND="A9
405  PRINT "% OF TENTH BAND="B1
410  PRINT "% OF ELEVENTH BAND="B2
415  PRINT "% OF TWELTH BAND="B3
420  REM  CALC. % POWER OF LAST BAND
425 S4 = 0
430  FOR Y4 = 0 TO 26
435 X4 =  PEEK (16384 + 2 * J + 5 * K + 5 * T + Y4)
440 P4 = X4 ^ 2
445 S4 = S4 + P4
450  NEXT Y4
455 B4 = (S4 / S) * 125
460  PRINT "% OF THERTEENTH BAND="B4
465  PRINT "TOTAL % BEFORE ROUND OFF IS T%="A1 + A2 + A3 + A4 + A5 + A6 + A7 + A
8 + A9 + B1 +
     B2 + B3 + B4
470 A1 =  INT ((A1 * 02) + 0.05) / 02
475 A2 =  INT ((A2 * 02) + 0.05) / 02
480 A3 =  INT ((A3 * 02) + 0.05) / 02
485 A4 =  INT ((A4 * 02) + 0.05) / 02
490 A5 =  INT ((A5 * 02) + 0.05) / 02
495 A6 =  INT ((A6 * 02) + 0.05) / 02
500 A7 =  INT ((A7 * 02) + 0.05) / 02
505 A8 =  INT ((A8 * 02) + 0.05) / 02
510 A9 =  INT ((A9 * 02) + 0.05) / 02
515 B1 =  INT ((B1 * 02) + 0.05) / 02
520 B2 =  INT ((B2 * 02) + 0.05) / 02
525 B3 =  INT ((B3 * 02) + 0.05) / 02
530 B4 =  INT ((B4 * 02) + 0.05) / 02
535  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
540  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
545  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
550  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
555  PRINT "B1="B1: PRINT "B2="B2: PRINT "B3="B3
560  PRINT "B4="B4
565 T% = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
```

```
570  PRINT "TOTAL % POWER IS T%="T%
575  PRINT D$"FREQ"
580  DIM D(20): DIM A(20)
585  DIM N(20): DIM M(20)
590  DIM F(20)
595  REM  CALC. NO.OF HZ/POINT,"S"
600 S = 71.12 / 256
605  PRINT "S="S
610  REM  CALC. NO. OF POINTS/UNIT %,"P"
615 P = 144 / 240:P =  INT ((P * 02) + 0.5) / 02
620  REM  CALC. FREQ. INTERVALS,"D(I)"
625  REM  F(I) ARE THE FREQ. OF THE END OF EACH FREQ. INTERVAL
630 F(0) = J * S
635 F(1) = 2 * F(0)
640 F = F(1)
645  FOR I = 2 TO 6
650 F(I) = F + K * S:F = F(I)
655  NEXT I
660 E = F(6)
665  FOR I = 7 TO 11
670 F(I) = E + T * S:E = F(I)
675  NEXT I
680 F(12) = F(11) + U * S
685  REM  D(I) ARE FREQ. INTERVALS
690 D(0) = F(0)
695 D = D(0)
700  FOR I = 1 TO 12
705 D(I) = F(I) - D:D = F(I)
710  NEXT I
715  FOR I = 0 TO 12
720  PRINT "F(I)="F(I)
725  PRINT "***************"
730  PRINT "D(I)="D(I): NEXT I
735  REM  CALC. AMPLITUDES FOR HISTOGRAM
740 A(0) = A1 * 02:A(1) = A2 * 02
745 A(2) = A3 * 02:A(3) = A4 * 02
750 A(4) = A5 * 02:A(5) = A6 * 02
755 A(6) = A7 * 02:A(7) = A8 * 02
760 A(8) = A9 * 02:A(9) = B1 * 02
765 A(10) = B2 * 02:A(11) = B3 * 02
770 A(12) = B4 * 02
775  FOR I = 0 TO 12
780 D(I) =  INT ((D(I) * 02) + 0.5) / 02
785  PRINT "D(I)="D(I)"HZ"
790  REM  CALC. NO. OF POINTS/INTERVAL FOR FREQ. SCALE,"N(I)"
795 N(I) = D(I) / S:N(I) =  INT ((N(I) * 02) + .5) / 02
800  PRINT "N(I)="N(I)"POINTS"
805  REM  CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
810 M(I) = P * A(I):M(I) =  INT ((M(I) * 02) + .5) / 02
815  NEXT I
820  REM  DRAW THE HISTOGRAM
825 Q = 15:R = 15
830  FOR I = 0 TO 12
835 R = R + N(I): PRINT "R(I)="R
840  FOR Y = 0 TO M(I)
845  HPLOT Q,144 - Y TO Q,144
850  HPLOT R,144 - Y TO R,144
855  NEXT Y
860  FOR X = 0 TO N(I)
865  HPLOT Q,144 - M(I) TO X + Q,144 - M(I)
870  NEXT X
875 Q = Q + N(I)
880  NEXT I
885  PRINT D$"RUN MICROBUF,D1"
```

```
]LOAD MICRODUF
]LIST

5  PRINT  CHR$ (4);"PR#1"
10  PRINT  CHR$ (09);"GE": PRINT  CHR$ (09);"GDER"
15  PRINT  CHR$ (4);"PR#0"
20  TEXT
25  HOME
30 S =  PEEK (16896)
35  IF S <  > 1 GOTO 45
40  POKE 16896,0: PRINT  CHR$ (4)"RUN B-AVERAGE,D1"
45 G$ =  CHR$ (7): FOR X = 1 TO 3: PRINT G$
50  FOR X1 = 1 TO 2: PRINT : NEXT X1: FOR Y1 = 1 TO 4: PRINT G$: NEXT Y1
55  FOR Y2 = 1 TO 3: PRINT : NEXT Y2: NEXT X
60  HOME : HTAB 6: VTAB 7: PRINT "INSERT ANOTHER DATA DISK (DRIVE 2)"
65  HTAB 9: VTAB 5: INVERSE : PRINT "PROCESSING DONE!": NORMAL
70  HTAB 6: VTAB 9: PRINT "RUN AGAIN THE PROGRAM (DRIVE 1) BY"
75  HTAB 9: VTAB 11: PRINT "PRESSING"
80  HTAB 18: VTAB 11: INVERSE : PRINT "PR#6": NORMAL
85  END


]LOAD B-AVERAGE

]LIST

5 D$ =  CHR$ (4)
10  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.B W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                                       ": NORMAL
15  FOR Y = 0 TO 7: PRINT D$"BLOAD RETTAB1,D1"
20  POKE 823,Y * 2 + 16: CALL 768
25  PRINT D$"BLOAD CLEARB.OBJO,D1": CALL 768: PRINT D$"BLOAD AVELOTABB,D1"
30  HOME : VTAB 15: INVERSE : PRINT "8 AVERAGES,OF 8 SPECTRUMS EACH,FOR CHAN.B W
ILL BE DONE N
     OW AND WILL BE SAVED ON THE DISK                                       ": NORMAL
35 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
40  PRINT D$"BLOAD AVEHIB.OBJO,D1"
45  CALL 768: TEXT
50  PRINT D$"BSAVE AVEB."Y",A$4800,L$200,D1"
55  NEXT Y
60  PRINT D$"RUN FINAL-AVE.B,D1"
```

```
]LOAD FINAL-AVE.B

]LIST

5  REM  THIS PROGRAM LOADS CH.B AVERAGES IN SUPERRAM II,IN THE RIGHT POSITION FR
OM THE POINT
     OF VUE OF THE M/L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.B
10 D$ =  CHR$ (4)
15 N =  PEEK ( - 16255):N =  PEEK ( - 16255)
20  FOR Y = 0 TO 7
25  REM  53760=$D200
30 X = 53760 + Y * 1024
35  PRINT D$"BLOAD AVEB."Y",A"X",D1"
40  REM  NOW,SUPERRAM II CONTAINS 8 AVERAGES FOR CH.B,FROM $D200-$D3FF,$D600-$D7
FF,$DA00-$DBF
     F,$DE00-$DFFF,$E200-$E3FF,$E600-$E7FF,$EA00-$EBFF,$EE00-$EFFF
45  PRINT D$"BLOAD CLEARB.OBJ0,D1": CALL 768: PRINT D$"BLOAD AVELOTABB,D1"
50  HOME : VTAB 15: INVERSE : PRINT "THE FINAL AVERAGE,OF 64 SPECTRUMS,FOR CHAN.
B WILL BE DON
     E NOW AND WILL BE SAVED ON THE DISK                              ": NORMAL
55  REM  IN LINE 46,THE HOME STATEMENT IS USED TO ERASE THE M/L THAT APPEARS ON
THE SCREEN BY
      LOADING THE BINARY FILE "AVELOTABB"
60  PRINT D$"BLOAD AVEHIB.OBJ0,D1"
65  CALL 768: TEXT
70  REM  NOW,THE AVERAGE OF 8 AVERAGES(EACH OF 8 SPC.)OF CH.B,IS LOCATED AT $480
0
75  PRINT D$"BSAVE AVE.AVE.B,A$4800,L$200,D1"
80  PRINT D$"RUN SEE FIN.AVE.B,D1"


]LOAD SEE FIN.AVE.B

]LIST

5  REM   THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.B ON THE SCREEN
10 D$ =  CHR$ (4): PRINT D$"BLOAD M4A,D1": PRINT D$"BLOAD M4B,D1": PRINT D$"BLOA
D M4C,D1"
15  PRINT D$"BLOAD BPR,D1": PRINT D$"BLOAD CT,D1": PRINT D$"BLOAD CS,D1"
20  PRINT D$"BLOAD AVE.AVE.B,A$1000,D1"
25  POKE 37272,1: TEXT : HOME
30  POKE 15312,209: POKE 15313,193: POKE 15314,160: POKE 15315,207
35  POKE 15316,181: POKE 15317,160: POKE 15318,144
40  REM  POKE TITLE CHARACTERS
45  POKE 15319,193: POKE 15320,214: POKE 15321,197: POKE 15322,210
50  POKE 15323,193: POKE 15324,199: POKE 15325,197: POKE 15326,194: POKE 15327,1
73
55  POKE 15328,182: POKE 15329,180: POKE 15330,141: POKE 15331,155: POKE 15332,1
28
60  CALL 3064
65  PRINT D$: POKE 34,0: TEXT : HOME
70  PRINT D$"RUN AXISB,D1"
```

```
]LOAD AXIS3

]LIST

5 D$ =  CHR$ (4)
10  REM  DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15  HGR
20  HCOLOR= 3
25  HPLOT 15,144 TO 15,0
30  HPLOT 14,1 TO 16,1: HPLOT 13,2 TO 17,2
35  HPLOT 15,144 TO 279,144
40  HPLOT 277,142 TO 277,146: HPLOT 278,143 TO 278,145
45  PRINT D$"BLOAD MAGICSPACE#,D1"
50  POKE 232,0: POKE 233,96
55  SCALE= 1
60  ROT= 0
65  REM  A=Y CENTER UP;B=Y CENTER DOWN
70  REM  THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75  REM  DRAW UP NUMBERS
80 A = 146:B = 154
85 X = 9
90  FOR N = 1 TO 8
95  READ S
100  DRAW S AT X,A
105 X = X + 4
110  IF N < 4 GOTO 120
115  IF N = 4 THEN X = 23
120  NEXT N
125  DATA  49,37,49,49,49,14,10,19
130 X = 51
135  FOR N = 1 TO 8
140  READ S
145  DRAW S AT X,A
150 X = X + 4
155  IF N < 4 GOTO 165
160  IF N = 4 THEN X = 79
165  NEXT N
170  DATA  12,12,10,17,12,20,10,15
175 X = 122
180  FOR N = 1 TO 12
185  READ S
190  DRAW S AT X,A
195 X = X + 4
200  IF N < 4 GOTO 220
205  IF N = 4 THEN X = 180
210  IF N < 8 GOTO 220
215  IF N = 8 THEN X = 239
220  NEXT N
225  DATA  14,12,10,14,15,18,10,16,17,14,10,17
230  REM  DRAW DOWN NUMBERS
235 X = 16
240  FOR N = 1 TO 4
245  READ S
250  DRAW S AT X,B
255 X = X + 4
260  NEXT N
265  DATA  49,12,10,20
270 X = 37
```

```
275  FOR N = 1 TO 12
280  READ S
285  DRAW S AT X,B
290 X = X + 4
295  IF N < 4 GOTO 315
300  IF N = 4 THEN X = 65
305  IF N < 8 GOTO 315
310  IF N = 8 THEN X = 93
315  NEXT N
320  DATA  49,18,10,18,12,16,10,16,13,14,10,14
325 X = 151
330  FOR N = 1 TO 12
335  READ S
340  DRAW S AT X,B
345 X = X + 4
350  IF N < 4 GOTO 370
355  IF N = 4 THEN X = 209
360  IF N < 8 GOTO 370
365  IF N = 8 THEN X = 266
370  NEXT N
375  DATA  14,20,10,15,16,16,10,16,18,12,10,12
380  REM  THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
IS IS 5
385  REM  DRAW"HZ"ON THE HORIZONTAL AXIS
390 X = 260
395  FOR N = 1 TO 2
400  READ S
405  DRAW S AT X,136
410 X = X + 5
415  NEXT N
420  DATA  30,48
425  REM  MARK THE DIVISIONS ON THE VERTICAL AXIS
430 C = 144 / 240
435 X = 15
440  DIM Y(10): DIM Z(10)
445  FOR I = 1 TO 5
450  READ S
455 Y(I) = 144 - S * C - 2: REM  2 IS BECAUSE THE M/L WHICH DRAWS THE CHAR. CONS
IDERS THE CHA
     R. FROM THE TOP(ONE CHAR. HAS 5 VERT. POINTS)
460 Z(I) = Y(I) + (S * C) / (2 * I)
465  PRINT "S*C="S * C
470  PRINT "Y(I)="Y(I)
475  PRINT "Z(I)="Z(I)
480  NEXT I
485  DATA  40,80,120,160,200
490  FOR I = 1 TO 5
495  DRAW 9 AT X,Y(I)
500  DRAW 9 AT X,Z(I)
505  NEXT I
510  FOR I = 1 TO 5
515 X = 5
520  FOR N = 1 TO 2
525  READ S,U
530  DRAW S AT X,Y(I)
535  DRAW U AT X,Z(I)
540 X = X + 4
545  NEXT N
550  NEXT I
555  REM  S=49,15,49,19,12,13,12,17,13,37 AND U=49,13,49,17,12,37,12,15,12,19
560  DATA  49,49,15,13,49,49,19,17,12,12,13,37,12,12,17,15,13,12,37,19
565  REM   DRAW"% OF TOT. POWER"ON THE VERTICAL AXIS
```

```
570 X = 22:Y = 2
575  DRAW 3 AT X,Y
580 X = X + 8
585  FOR N = 1 TO 2
590  READ S
595  DRAW S AT X,Y
600 X = X + 5
605  NEXT N
610  DATA  37,28
615 X = X + 3
620  FOR N = 1 TO 4
625  READ S
630  DRAW S AT X,Y
635 X = X + 5
640  NEXT N
645  DATA  42,37,42,10
650 X = X + 3
655  FOR N = 1 TO 5
660  READ S
665  DRAW S AT X,Y
670 X = X + 5
675  IF N = 3 THEN X = X + 1
680  NEXT N
685  DATA  38,37,45,27,40
690  REM  DRAW TITLE OF THE GRAPH
695  REM  THE DISTANCE BETWEEN THE NUMBERS IN THE TITLE IS 5
700  REM  THE DISTANCE BETWEEN THE WORDS IN THE TITLE IS 6
705  REM  THE DISTANCE BETWEEN THE LETTERS OF THE TITLE IS 7;EXCEPTION IN THE LA
ST PART,"HZ..
     .",WHERE IS 5
710  SCALE= 1
715 X = 90:Y = 10
720  FOR N = 1 TO 3
725  READ S
730  DRAW S AT X,Y
735 X = X + 7
740  NEXT N
745  DATA  27,27,29
750 X = X + 6
755  FOR N = 1 TO 9
760  READ S
765  DRAW S AT X,Y
770 X = X + 7
775  NEXT N
780  DATA  30,31,41,42,37,29,40,23,35
785 X = X + 6
790  FOR N = 1 TO 3
795  READ S
800  DRAW S AT X,Y
805 X = X + 7
810  NEXT N
815  DATA  28,37,40
820 X = X + 6
825  FOR N = 1 TO 14
830  READ S
835  DRAW S AT X,Y
840 X = X + 5
845  NEXT N
850  DATA  37,9,18,12,10,12,49,30,48,9,25,30,10,24
855  PRINT D$"RUN PLOT-B,D1"
```

```
]LOAD PLOT-B

]LIST

5 D$ =  CHR$ (4)
10  HTAB 10: VTAB 23: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15  REM  CALC. TOTAL POWER
20 S = 0:O1 = 100:O2 = 10: REM   O1 IS FOR THE % AND O2 IS FOR HIST.AMPLIT.CALC.
25  FOR Y = 0 TO 255
30 X =  PEEK (18432 + Y)
35 P = X ^ 2
40 S = S + P
45  NEXT Y
50  PRINT D$"PR#1"
55  PRINT : PRINT : PRINT "TOTAL POWER="S
60  REM  J,K,T,U ARE NO.OF POINTS FOR THE FREQ.INTERVALS OF THE HISTOGRAM
65  READ J,K,T,U
70  DATA  7,14,29,27
75  REM  CALC. % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90  FOR Y1 = 0 TO 2 * J
95 X1 =  PEEK (18432 + Y1)
100 P1 = X1 ^ 2
105 S1 = S1 + P1
110 N = N + 1
115  IF N < 7 THEN  GOTO 130
120  IF N = 7 THEN A1 = (S1 / S) * O1
125  IF N = 8 THEN S1 = 0
130  NEXT Y1
135 A2 = (S1 / S) * O1
140  PRINT "% OF FIRST BAND="A1
145  PRINT "% OF SECOND BAND="A2
150  REM  CALC.% POWER OF NEXT FIVE BANDS
155 N = 0
160 S2 = 0
165  FOR Y2 = 0 TO 69
170 X2 =  PEEK (18432 + 2 * J + Y2)
175 P2 = X2 ^ 2
180 S2 = S2 + P2
185 N = N + 1
190  IF N < 14 THEN  GOTO 250
195  IF N = 14 THEN A3 = (S2 / S) * O1
200  IF N = 15 THEN S2 = 0
205  IF N < 28 THEN  GOTO 250
210  IF N = 28 THEN A4 = (S2 / S) * O1
215  IF N = 29 THEN S2 = 0
220  IF N < 42 THEN  GOTO 250
225  IF N = 42 THEN A5 = (S2 / S) * O1
230  IF N = 43 THEN S2 = 0
235  IF N < 56 THEN  GOTO 250
240  IF N = 56 THEN A6 = (S2 / S) * O1
245  IF N = 57 THEN S2 = 0
250  NEXT Y2
255 A7 = (S2 / S) * O1
260  PRINT "% OF THIRD BAND="A3
265  PRINT "% OF FOURTH BAND="A4
270  PRINT "% OF FIFTH BAND="A5
```

```
275  PRINT "% OF SIXTH BAND="A6
280  PRINT "% OF SEVENTH BAND="A7
285  REM  CALC. % POWER OF NEXT FIVE BANDS
290 N = 0
295 S3 = 0
300  FOR Y3 = 0 TO 144
305 X3 =  PEEK (18432 + 2 * J + 5 * K + Y3)
310 P3 = X3 ^ 2
315 S3 = S3 + P3
320 N = N + 1
325  IF N < 29 THEN  GOTO 385
330  IF N = 29 THEN A8 = (S3 / S) * O1
335  IF N = 30 THEN S3 = 0
340  IF N < 56 THEN  GOTO 385
345  IF N = 56 THEN A9 = (S3 / S) * O1
350  IF N = 57 THEN S3 = 0
355  IF N < 87 THEN  GOTO 385
360  IF N = 87 THEN B1 = (S3 / S) * O1
365  IF N = 87 THEN S3 = 0
370  IF N < 116 THEN  GOTO 385
375  IF N = 116 THEN B2 = (S3 / S) * O1
380  IF N = 117 THEN S3 = 0
385  NEXT Y3
390 B3 = (S3 / S) * O1
395  PRINT "% OF EIGHTH BAND="A8
400  PRINT "% OF NINTH BAND="A9
405  PRINT "% OF TENTH BAND="B1
410  PRINT "% OF ELEVENTH BAND="B2
415  PRINT "% OF TWELTH BAND="B3
420  REM  CALC. % POWER OF LAST BAND
425 S4 = 0
430  FOR Y4 = 0 TO 26
435 X4 =  PEEK (18432 + 2 * J + 5 * K + 5 * T + Y4)
440 P4 = X4 ^ 2
445 S4 = S4 + P4
450  NEXT Y4
455 B4 = (S4 / S) * 125
460  PRINT "% OF THERTEENTH BAND="B4
465  PRINT "TOTAL % BEFORE ROUND OFF IS TX="A1 + A2 + A3 + A4 + A5 + A6 + A7 + A
8 + A9 + B1 +
     B2 + B3 + B4
470 A1 =  INT ((A1 * O2) + 0.05) / O2
475 A2 =  INT ((A2 * O2) + 0.05) / O2
480 A3 =  INT ((A3 * O2) + 0.05) / O2
485 A4 =  INT ((A4 * O2) + 0.05) / O2
490 A5 =  INT ((A5 * O2) + 0.05) / O2
495 A6 =  INT ((A6 * O2) + 0.05) / O2
500 A7 =  INT ((A7 * O2) + 0.05) / O2
505 A8 =  INT ((A8 * O2) + 0.05) / O2
510 A9 =  INT ((A9 * O2) + 0.05) / O2
515 B1 =  INT ((B1 * O2) + 0.05) / O2
520 B2 =  INT ((B2 * O2) + 0.05) / O2
525 B3 =  INT ((B3 * O2) + 0.05) / O2
530 B4 =  INT ((B4 * O2) + 0.05) / O2
535  PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
540  PRINT "A1="A1: PRINT "A2="A2: PRINT "A3="A3
545  PRINT "A4="A4: PRINT "A5="A5: PRINT "A6="A6
550  PRINT "A7="A7: PRINT "A8="A8: PRINT "A9="A9
555  PRINT "B1="B1: PRINT "B2="B2: PRINT "B3="B3
560  PRINT "B4="B4
565 TX = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
```

```
570  PRINT "TOTAL % POWER IS T%="T%
575  PRINT D$"PR#0"
580  DIM D(20): DIM A(20)
585  DIM N(20): DIM M(20)
590  DIM F(20)
595  REM  CALC. NO.OF HZ/POINT,"S"
600 S = 71.12 / 256
605  PRINT "S="S
610  REM  CALC. NO. OF POINTS/UNIT %,"P"
615 P = 144 / 240:P =  INT ((P * 02) + .5) / 02
620  REM  CALC. FREQ. INTERVALS,"D(I)"
625  REM  F(I) ARE THE FREQ. OF THE END OF EACH FREQ. INTERVAL
630 F(0) = J * S
635 F(1) = 2 * F(0)
640 F = F(1)
645  FOR I = 2 TO 6
650 F(I) = F + K * S:F = F(I)
655  NEXT I
660 E = F(6)
665  FOR I = 7 TO 11
670 F(I) = E + T * S:E = F(I)
675  NEXT I
680 F(12) = F(11) + U * S
685  REM  D(I) ARE FREQ. INTERVALS
690 D(0) = F(0)
695 D = D(0)
700  FOR I = 1 TO 12
705 D(I) = F(I) - D:D = F(I)
710  NEXT I
715  FOR I = 0 TO 12
720  PRINT "F(I)="F(I)
725  PRINT "***************"
730  PRINT "D(I)="D(I): NEXT I
735  REM  CALC. AMPLITUDES FOR HISTOGRAM
740 A(0) = A1 * 02:A(1) = A2 * 02
745 A(2) = A3 * 02:A(3) = A4 * 02
750 A(4) = A5 * 02:A(5) = A6 * 02
755 A(6) = A7 * 02:A(7) = A8 * 02
760 A(8) = A9 * 02:A(9) = B1 * 02
765 A(10) = B2 * 02:A(11) = B3 * 02
770 A(12) = B4 * 02
775  FOR I = 0 TO 12
780 D(I) =  INT ((D(I) * 02) + 0.5) / 02
785  PRINT "D(I)="D(I)"HZ"
790  REM  CALC. NO. OF POINTS/INTERVAL FOR FREQ. SCALE,"N(I)"
795 N(I) = D(I) / S:N(I) =  INT ((N(I) * 02) + .5) / 02
800  PRINT "N(I)="N(I)"POINTS"
805  REM  CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
810 M(I) = P * A(I):M(I) =  INT ((M(I) * 02) + .5) / 02
815  NEXT I
820  REM  DRAW THE HISTOGRAM
825 Q = 15:R = 15
830  FOR I = 0 TO 12
835 R = R + N(I)
840  FOR Y = 0 TO M(I)
845  HPLOT Q,144 - Y TO Q,144
850  HPLOT R,144 - Y TO R,144
855  NEXT Y
860  FOR X = 0 TO N(I)
865  HPLOT Q,144 - M(I) TO X + Q,144 - M(I)
```

APPENDIX G

Machine Language Routines Listing

```
:LOAD AVEHI

END OF DATA

:L

  TAB       EQU   $361
  HIMEMH    EQU   $0006

  HIMEML    EQU   $0005
  SOURCE1H  EQU   $0004

  SOURCE1L  EQU   $0003
  LOMEMH    EQU   $0002
  LOMEML    EQU   $0001
            ORG   $0368
  START     LDX   #$00
            LDY   #$00
  LOOP      LDA   ($03),Y
  ADD       CLC
            ADC   ($01),Y
            STA   ($01),Y
            LDA   #$00
            ADC   ($05),Y
            STA   ($05),Y
            INY
            CPY   #$00
            BNE   LOOP
            INX
            CPX   #$02
            BEQ   END1
            INC   SOURCE1H
            INC   LOMEMH
            INC   HIMEMH
            JMP   LOOP
  END1      LDA   SOURCE1H
            CMP   #$ED
            BEQ   END2
            LDA   SOURCE1H
            CLC
            ADC   #$03
            STA   SOURCE1H
            LDA   #$40
            STA   LOMEMH
            LDA   #$44
            STA   HIMEMH
            JMP   START
  END2      LDX   #$00
  ENDLOOP   LDA   TAB,X
            STA   $00,X
            INX
            CPX   #$07
            BNE   ENDLOOP
            RTS
```

```
*BLOAD AVEHI.OBJ0,A$300 00 01 EF D8 00
*300LLL

0300-   A2 00       LDX   #$00
0302-   A0 00       LDY   #$00
0304-   B1 03       LDA   ($03),Y
0306-   18          CLC
0307-   71 01       ADC   ($01),Y
0309-   91 01       STA   ($01),Y
030B-   A9 00       LDA   #$00
030D-   71 05       ADC   ($05),Y
030F-   91 05       STA   ($05),Y
0311-   C8          INY
0312-   C0 00       CPY   #$00
0314-   D0 EE       BNE   $0304
0316-   E8          INX
0317-   E0 02       CPX   #$02
0319-   F0 09       BEQ   $0324
031B-   E6 04       INC   $04
031D-   E6 02       INC   $02
031F-   E6 06       INC   $06
0321-   4C 6C 03    JMP   $036C
0324-   A5 04       LDA   $04
0326-   C9 ED       CMP   #$ED
0328-   F0 12       BEQ   $033C
032A-   A5 04       LDA   $04
032C-   18          CLC
032D-   69 03       ADC   #$03
032F-   85 04       STA   $04
0331-   A9 40       LDA   #$40
0333-   85 02       STA   $02
0335-   A9 44       LDA   #$44
0337-   85 06       STA   $06
0339-   4C 68 03    JMP   $0368
033C-   A2 00       LDX   #$00
033E-   BD 61 03    LDA   $0361,X
0341-   95 00       STA   $00,X
0343-   E8          INX
0344-   E0 07       CPX   #$07
0346-   D0 F6       BNE   $033E
0348-   60          RTS
0349-   00          BRK
034A-   C9 02       CMP   #$02
034C-   D0 D7       BNE   $0325
034E-   A2 00       LDX   #$00
0350-   BD 00 FF    LDA   $FF00,X
0353-   95 00       STA   $00,X
0355-   E8          INX
0356-   E0 00       CPX   #$00
0358-   D0 F6       BNE   $0350
035A-   AD 82 C0    LDA   $C082
035D-   AD 82 C0    LDA   $C082
0360-   60          RTS
0361-   00          BRK
0362-   00          BRK
0363-   48          PHA
0364-   00          BRK
0365-   D2          ???
0366-   00          BRK
0367-   4C 04 04    JMP   $0404
036A-   00          BRK
036B-   00          BRK
036C-   00          BRK
```

```
:LOAD AVEHIB

END OF DATA

:L

TAB       EQU   $361
HIMEMH    EQU   $0006

HIMEML    EQU   $0005
SOURCE1H  EQU   $0004

SOURCE1L  EQU   $0003
LOMEMH    EQU   $0002
LOMEML    EQU   $0001
          ORG   $0368
START     LDX   #$00
          LDY   #$00
LOOP      LDA   ($03),Y
ADD       CLC
          ADC   ($01),Y
          STA   ($01),Y
          LDA   #$00
          ADC   ($05),Y
          STA   ($05),Y
          INY
          CPY   #$00
          BNE   LOOP
          INX
          CPX   #$02
          BEQ   END1
          INC   SOURCE1H
          INC   LOMEMH
          INC   HIMEMH
          JMP   LOOP
END1      LDA   SOURCE1H
          CMP   #$EF
          BEQ   END2
          LDA   SOURCE1H
          CLC
          ADC   #$03
          STA   SOURCE1H
          LDA   #$48
          STA   LOMEMH
          LDA   #$4C
          STA   HIMEMH
          JMP   START
END2      LDX   #$00
ENDLOOP   LDA   TAB,X
          STA   $00,X
          INX
          CPX   #$07
          BNE   ENDLOOP
          RTS
```

```
*BLOAD AVEHIB.OBJ0,A$300 00 01 EF D8 00
*300LLL

0300-   A2 00       LDX   #$00
0302-   A0 00       LDY   #$00
0304-   B1 03       LDA   ($03),Y
0306-   18          CLC
0307-   71 01       ADC   ($01),Y
0309-   91 01       STA   ($01),Y
030B-   A9 00       LDA   #$00
030D-   71 05       ADC   ($05),Y
030F-   91 05       STA   ($05),Y
0311-   C8          INY
0312-   C0 00       CPY   #$00
0314-   D0 EE       BNE   $0304
0316-   E8          INX
0317-   E0 02       CPX   #$02
0319-   F0 09       BEQ   $0324
031B-   E6 04       INC   $04
031D-   E6 02       INC   $02
031F-   E6 06       INC   $06
0321-   4C 6C 03    JMP   $036C
0324-   A5 04       LDA   $04
0326-   C9 EF       CMP   #$EF
0328-   F0 12       BEQ   $033C
032A-   A5 04       LDA   $04
032C-   18          CLC
032D-   69 03       ADC   #$03
032F-   85 04       STA   $04
0331-   A9 48       LDA   #$48
0333-   85 02       STA   $02
0335-   A9 4C       LDA   #$4C
0337-   85 06       STA   $06
0339-   4C 68 03    JMP   $0368
033C-   A2 00       LDX   #$00
033E-   BD 61 03    LDA   $0361,X
0341-   95 00       STA   $00,X
0343-   E8          INX
0344-   E0 07       CPX   #$07
0346-   D0 F6       BNE   $033E
0348-   60          RTS
0349-   00          BRK
034A-   C9 02       CMP   #$02
034C-   D0 D7       BNE   $0325
034E-   A2 00       LDX   #$00
0350-   BD 00 FF    LDA   $FF00,X
0353-   95 00       STA   $00,X
0355-   E8          INX
0356-   E0 00       CPX   #$00
0358-   D0 F6       BNE   $0350
035A-   AD 82 C0    LDA   $C082
035D-   AD 82 C0    LDA   $C082
0360-   60          RTS
0361-   00          BRK
0362-   00          BRK
0363-   48          PHA
0364-   00          BRK
0365-   D2          ???
0366-   00          BRK
0367-   4C 04 04    JMP   $0404
036A-   00          BRK
036B-   00          BRK
036C-   00          BRK
```

```
:LOAD AVELO

END OF DATA

:L

                ORG   $0300
        WON     EQU   $C081
        RON     EQU   $C080
        OFF     EQU   $C082
        ZERO    EQU   $0000
        LAST    EQU   $FF00
        HI      EQU   $368
        TAB     EQU   $361
                LDA   WON
                LDA   WON
                LDX   #$00
        MOVE    LDA   ZERO,X
                STA   LAST,X
                INX
                CPX   #$00
                BNE   MOVE
        MTAB    LDA   TAB,X
                STA   $00,X
                INX
                CPX   #$07
                BNE   MTAB
                LDA   RON
                LDA   RON
                JSR   HI
        LOOP    LDY   #$00
        START   LDX   #$00
        BEG     CLC
                LDA   (05),Y
                ROR   A
                STA   (05),Y
                LDA   (01),Y
                ROR   A
                STA   (01),Y
                INX
                CPX   #$03
                BNE   BEG
                ADC   #$00
                STA   (01),Y
                INY
                CPY   #$00
                BNE   START
                INC   $02
                INC   $06
                INC   $00
                LDA   $00
                CMP   #$02
                BNE   LOOP
                LDX   #$00
        RETR    LDA   LAST,X
                STA   ZERO,X
                INX
                CPX   #$00
                BNE   RETR
                LDA   OFF
                LDA   OFF
                RTS
```

```
]BLOAD AVELOTAB,A$300
]CALL-151

*300LLL

0300-   AD 81 C0    LDA   $C081
0303-   AD 81 C0    LDA   $C081
0306-   A2 00       LDX   #$00
0308-   B5 00       LDA   $00,X
030A-   9D 00 FF    STA   $FF00,X
030D-   E8          INX
030E-   E0 00       CPX   #$00
0310-   D0 F6       BNE   $0308
0312-   BD 61 03    LDA   $0361,X
0315-   95 00       STA   $00,X
0317-   E8          INX
0318-   E0 07       CPX   #$07
031A-   D0 F6       BNE   $0312
031C-   AD 80 C0    LDA   $C080
031F-   AD 80 C0    LDA   $C080
0322-   20 68 03    JSR   $0368
0325-   A0 00       LDY   #$00
0327-   A2 00       LDX   #$00
0329-   18          CLC
032A-   B1 05       LDA   ($05),Y
032C-   6A          ROR
032D-   91 05       STA   ($05),Y
032F-   B1 01       LDA   ($01),Y
0331-   6A          ROR
0332-   91 01       STA   ($01),Y
0334-   E8          INX
0335-   E0 03       CPX   #$03
0337-   D0 F0       BNE   $0329
0339-   69 00       ADC   #$00
033B-   91 01       STA   ($01),Y
033D-   C8          INY
033E-   C0 00       CPY   #$00
0340-   D0 E5       BNE   $0327
0342-   E6 02       INC   $02
0344-   E6 06       INC   $06
0346-   E6 00       INC   $00
0348-   A5 00       LDA   $00
034A-   C9 02       CMP   #$02
034C-   D0 D7       BNE   $0325
034E-   A2 00       LDX   #$00
0350-   BD 00 FF    LDA   $FF00,X
0353-   95 00       STA   $00,X
0355-   E8          INX
0356-   E0 00       CPX   #$00
0358-   D0 F6       BNE   $0350
035A-   AD 82 C0    LDA   $C082
035D-   AD 82 C0    LDA   $C082
0360-   60          RTS
0361-   00          BRK
0362-   00          BRK
0363-   40          RTI
0364-   00          BRK
0365-   D0 00       BNE   $0367
0367-   44          ???
0368-   A2 00       LDX   #$00
036A-   FE 03 A0    INC   $A003,X
036D-   B4 B0       LDY   $B0,X
036F-   B0 B0       BCS   $0321
0371-   B0 B0       BCS   $0323
0373-   A9 00       LDA   #$00
```

```
*BLOAD AVELOTABB,A$300 00 01 EF D8 00
*300LLL

0300-    AD 81 C0      LDA   $C081
0303-    AD 81 C0      LDA   $C081
0306-    A2 00         LDX   #$00
0308-    B5 00         LDA   $00,X
030A-    9D 00 FF      STA   $FF00,X
030D-    E8            INX
030E-    E0 00         CPX   #$00
0310-    D0 F6         BNE   $0308
0312-    BD 61 03      LDA   $0361,X
0315-    95 00         STA   $00,X
0317-    E8            INX
0318-    E0 07         CPX   #$07
031A-    D0 F6         BNE   $0312
031C-    AD 80 C0      LDA   $C080
031F-    AD 80 C0      LDA   $C080
0322-    20 68 03      JSR   $0368
0325-    A0 00         LDY   #$00
0327-    A2 00         LDX   #$00
0329-    18            CLC
032A-    B1 05         LDA   ($05),Y
032C-    6A            ROR
032D-    91 05         STA   ($05),Y
032F-    B1 01         LDA   ($01),Y
0331-    6A            ROR
0332-    91 01         STA   ($01),Y
0334-    E8            INX
0335-    E0 03         CPX   #$03
0337-    D0 F0         BNE   $0329
0339-    69 00         ADC   #$00
033B-    91 01         STA   ($01),Y
033D-    C8            INY
033E-    C0 00         CPY   #$00
0340-    D0 E5         BNE   $0327
0342-    E6 02         INC   $02
0344-    E6 06         INC   $06
0346-    E6 00         INC   $00
0348-    A5 00         LDA   $00
034A-    C9 02         CMP   #$02
034C-    D0 D7         BNE   $0325
034E-    A2 00         LDX   #$00
0350-    BD 00 FF      LDA   $FF00,X
0353-    95 00         STA   $00,X
0355-    E8            INX
0356-    E0 00         CPX   #$00
0358-    D0 F6         BNE   $0350
035A-    AD 82 C0      LDA   $C082
035D-    AD 82 C0      LDA   $C082
0360-    60            RTS
0361-    00            BRK
0362-    00            BRK
0363-    48            PHA
0364-    00            BRK
0365-    D2            ???
0366-    00            BRK
0367-    4C 04 04      JMP   $0404
036A-    00            BRK
036B-    00            BRK
036C-    00            BRK
036D-    01 00         ORA   ($00,X)
036F-    00            BRK
```

```
:LOAD SAVESPC

END OF DATA

:L

BUFF      EQU   $D000
SWEEP     EQU   $1000
COUNT     EQU   $0343
BUFFA     EQU   $0314
SWEEPA    EQU   $0311
WON       EQU   $C081
OFF       EQU   $C082
          ORG   $0300
BEG       LDA   #$10
          STA   SWEEPA
          LDA   WON
          LDA   WON
          LDX   #$00
          LDY   #$00
START     LDA   SWEEP,X
          STA   BUFF,X
          INX
          CPX   #$00
          BNE   START
          INC   SWEEPA
          INC   BUFFA
          INY
          CPY   #$02
          BNE   START
          LDA   OFF
          LDA   OFF
          RTS

:LOAD SWEEP

END OF DATA

:L

BUFF      EQU   $D000
SWEEP     EQU   $1000
BUFFA     EQU   $F006
SWEEPA    EQU   $F009
RON       EQU   $C080
ADD       EQU   $347
OFF       EQU   $C082
          ORG   $F000
          LDX   #$00
          LDY   #$00
START     LDA   BUFF,X
          STA   SWEEP,X
          INX
          CPX   #$00
          BNE   START
MID       CLC
          LDA   BUFFA
          ADC   #$01
          STA   $353
          JSR   ADD
          INY
          CPY   #$02
          BNE   START
STOP      RTS
```

```
*BLOAD SAVESPC.OBJO,A$300 00 01 EF D8 00

*300LL

0300-   A9 10       LDA   #$10
0302-   8D 11 03    STA   $0311
0305-   AD 81 C0    LDA   $C081
0308-   AD 81 C0    LDA   $C081
030B-   A2 00       LDX   #$00
030D-   A0 00       LDY   #$00
030F-   BD 00 10    LDA   $1000,X
0312-   9D 00 D0    STA   $D000,X
0315-   E8          INX
0316-   E0 00       CPX   #$00
0318-   D0 F5       BNE   $030F
031A-   EE 11 03    INC   $0311
031D-   EE 14 03    INC   $0314
0320-   C8          INY
0321-   C0 02       CPY   #$02
0323-   D0 EA       BNE   $030F
0325-   AD 82 C0    LDA   $C082
0328-   AD 82 C0    LDA   $C082
032B-   60          RTS
032C-   80          ???
032D-   C0 AD       CPY   #$AD
032F-   80          ???
0330-   C0 60       CPY   #$60
0332-   60          RTS
0333-   01 60       ORA   ($60,X)
0335-   02          ???
0336-   00          BRK
0337-   00          BRK
0338-   00          BRK
0339-   44          ???
033A-   03          ???
033B-   00          BRK
033C-   D0 00       BNE   $033E
033E-   00          BRK
033F-   01 00       ORA   ($00,X)
0341-   00          BRK
0342-   60          RTS
0343-   01 00       ORA   ($00,X)
0345-   01 EF       ORA   ($EF,X)
0347-   D8          CLD
```

```
:LOAD CLEAR

END OF DATA

:L

LOW      EQU   $4000
         ORG   $300
         LDY   #$00
         LDX   #$00
LOOP     LDA   #$00
         STA   LOW,X
         INX
         CPX   #$00
         BNE   LOOP
         INY
         CPY   #$02
         BEQ   END1
         INC   $308
         JMP   LOOP
END1     CPY   #$04
         BEQ   END
         LDA   #$04
         STA   $310
         LDA   #$44
         STA   $308
         JMP   LOOP
END      RTS
```

```
*BLOAD CLEAR.OBJ0,A$300 00 01 EF DB 00

*300LL

0300-   A0 00       LDY   #$00
0302-   A2 00       LDX   #$00
0304-   A9 00       LDA   #$00
0306-   9D 00 40    STA   $4000,X
0309-   E8          INX
030A-   E0 00       CPX   #$00
030C-   D0 F6       BNE   $0304
030E-   C8          INY
030F-   C0 02       CPY   #$02
0311-   F0 06       BEQ   $0319
0313-   EE 08 03    INC   $0308
0316-   4C 04 03    JMP   $0304
0319-   C0 04       CPY   #$04
031B-   F0 0D       BEQ   $032A
031D-   A9 04       LDA   #$04
031F-   8D 10 03    STA   $0310
0322-   A9 44       LDA   #$44
0324-   8D 08 03    STA   $0308
0327-   4C 04 03    JMP   $0304
032A-   60          RTS
032B-   03          ???
032C-   AD 82 C0    LDA   $C082
032F-   AD 82 C0    LDA   $C082
0332-   60          RTS
0333-   01 60       ORA   ($60,X)
0335-   02          ???
```

```
*BLOAD SWEEP.OBJ0,A$300 00 01 EF DB 00

*300LL

0300-   A2 00       LDX   #$00
0302-   A0 00       LDY   #$00
0304-   BD 00 D0    LDA   $D000,X
0307-   9D 00 10    STA   $1000,X
030A-   E8          INX
030B-   E0 00       CPX   #$00
030D-   D0 F5       BNE   $0304
030F-   18          CLC
0310-   AD 06 F0    LDA   $F006
0313-   69 01       ADC   #$01
0315-   8D 53 03    STA   $0353
0318-   20 47 03    JSR   $0347
031B-   C8          INY
031C-   C0 02       CPY   #$02
031E-   D0 E4       BNE   $0304
0320-   60          RTS
0321-   C0 02       CPY   #$02
0323-   D0 EA       BNE   $030F
0325-   AD 82 C0    LDA   $C082
0328-   AD 82 C0    LDA   $C082
032B-   60          RTS
032C-   80          ???
```

```
LOAD HIALL

END OF DATA

:L
               ORG    $F000
     TRACK     EQU    $34B
     SECTOR    EQU    $34C
     DTH       EQU    $350
     BUFF      EQU    $D000
     SWEEP     EQU    $1000
     COUNT     EQU    $346
     BUFFA     EQU    $0314
     SWEEPA    EQU    $0311
     WON       EQU    $C081
     RON       EQU    $C080
     OFF       EQU    $C082
     IOB       EQU    $03
     IOA       EQU    $47
     STOREA    EQU    $F000
     RWTS      EQU    $3D9
     STAR      LDA    #IOB
               LDY    #IOA
               JSR    RWTS
               INC    SECTOR
               LDA    #$10
               CMP    SECTOR
               BNE    DATA
               LDX    #$00
               STX    SECTOR
               LDA    #$44
               INC    TRACK
               CMP    TRACK
               BNE    DATA
               BRK
     DATA      INC    DTH
               LDA    #$F0
               CMP    DTH
               BEQ    STOP
               JMP    STAR
     STOP      LDA    #$D0
               STA    DTH
               STA    BUFFA
               LDA    #$00
               STA    COUNT
               RTS
```

```
*BLOAD HIALL.OBJO,A$300 00 01 EF DB 00

*300LL
0300-   A9 03       LDA   #$03
0302-   A0 47       LDY   #$47
0304-   20 D9 03    JSR   $03D9
0307-   EE 4C 03    INC   $034C
030A-   A9 10       LDA   #$10
030C-   CD 4C 03    CMP   $034C
030F-   D0 10       BNE   $0321
0311-   A2 00       LDX   #$00
0313-   8E 4C 03    STX   $034C
0316-   A9 44       LDA   #$44
0318-   EE 4B 03    INC   $034B
031B-   CD 4B 03    CMP   $034B
031E-   D0 01       BNE   $0321
0320-   00          BRK
0321-   EE 50 03    INC   $0350
0324-   A9 F0       LDA   #$F0
0326-   CD 50 03    CMP   $0350
0329-   F0 03       BEQ   $032E
032B-   4C 00 F0    JMP   $F000
032E-   A9 D0       LDA   #$D0
0330-   8D 50 03    STA   $0350
0333-   8D 14 03    STA   $0314
0336-   A9 00       LDA   #$00
0338-   8D 46 03    STA   $0346
033B-   60          RTS
033C-   20 00 F0    JSR   $F000
033F-   AD 82 C0    LDA   $C082
0342-   AD 82 C0    LDA   $C082
0345-   60          RTS
0346-   00          BRK
0347-   01 60       ORA   ($60,X)
0349-   02          ???
034A-   00          BRK
034B-   00          BRK
034C-   00          BRK
034D-   58          CLI
034E-   03          ???
034F-   00          BRK
0350-   D0 00       BNE   $0352
0352-   00          BRK
```

```
:LOAD CLEARB

END OF DATA
                                    *BLOAD CLEARB.OBJ0 00 01 EF D8 00
:L

   LOW     EQU   $4800     *300LL
           ORG   $300
           LDY   #$00      0300-   A0 00          LDY    #$00
           LDX   #$00      0302-   A2 00          LDX    #$00
   LOOP    LDA   #$00      0304-   A9 00          LDA    #$00
           STA   LOW,X     0306-   9D 00 48       STA    $4800,X
           INX             0309-   E8             INX
           CPX   #$00      030A-   E0 00          CPX    #$00
           BNE   LOOP      030C-   D0 F6          BNE    $0304
           INY             030E-   C8             INY
           CPY   #$02      030F-   C0 02          CPY    #$02
           BEQ   END1      0311-   F0 06          BEQ    $0319
           INC   $308      0313-   EE 08 03       INC    $0308
           JMP   LOOP      0316-   4C 04 03       JMP    $0304
   END1    CPY   #$04      0319-   C0 04          CPY    #$04
           BEQ   END       031B-   F0 0D          BEQ    $032A
           LDA   #$04      031D-   A9 04          LDA    #$04
           STA   $310      031F-   8D 10 03       STA    $0310
           LDA   #$4C      0322-   A9 4C          LDA    #$4C
           STA   $308      0324-   8D 08 03       STA    $0308
           JMP   LOOP      0327-   4C 04 03       JMP    $0304
   END     RTS             032A-   60             RTS
                           032B-   05 6A          ORA    $6A
```

LOWALL

```
END OF DATA

:L

               ORG   $300
      TRACK    EQU   $34B
      SECTOR   EQU   $34C
      DTH      EQU   $350
      BUFF     EQU   $D000
      SWEEP    EQU   $1000
      COUNT    EQU   $346
      BUFFA    EQU   $0314
      SWEEPA   EQU   $0311
      WON      EQU   $C081
      RON      EQU   $C080
      OFF      EQU   $C082
      IOB      EQU   $03
      IOA      EQU   $47
      STOREA   EQU   $F000
      RWTS     EQU   $3D9
      INIT     LDA   WON
               LDA   WON
      BEG      LDA   #$10
               STA   SWEEPA
               LDY   #$00
               LDX   #$00
      START    LDA   SWEEP,X
               STA   BUFF,X
               INX
               CPX   #$00
               BEQ   MID
               JMP   START
      MID      INC   SWEEPA
               INC   BUFFA
               INY
               CPY   #$04
               BEQ   END
               JMP   START
      END      INC   COUNT
               LDA   #$08
               CMP   COUNT
               BEQ   STORE
               RTS
      STORE    LDA   RON
               LDA   RON
               JSR   STOREA
               LDA   OFF
               LDA   OFF
               RTS
```

```
*BLOAD LOWALLTAB,A$300 00 01 EF D8 00

*300LL

0300-   AD 81 C0    LDA   $C081
0303-   AD 81 C0    LDA   $C081
0306-   A9 10       LDA   #$10
0308-   8D 11 03    STA   $0311
030B-   A0 00       LDY   #$00
030D-   A2 00       LDX   #$00
030F-   BD 00 10    LDA   $1000,X
0312-   9D 00 D0    STA   $D000,X
0315-   E8          INX
0316-   E0 00       CPX   #$00
0318-   F0 03       BEQ   $031D
031A-   4C 0F 03    JMP   $030F
031D-   EE 11 03    INC   $0311
0320-   EE 14 03    INC   $0314
0323-   C8          INY
0324-   C0 04       CPY   #$04
0326-   F0 03       BEQ   $032B
0328-   4C 0F 03    JMP   $030F
032B-   EE 46 03    INC   $0346
032E-   A9 08       LDA   #$08
0330-   CD 46 03    CMP   $0346
0333-   F0 01       BEQ   $0336
0335-   60          RTS
0336-   AD 80 C0    LDA   $C080
0339-   AD 80 C0    LDA   $C080
033C-   20 00 F0    JSR   $F000
033F-   AD 82 C0    LDA   $C082
0342-   AD 82 C0    LDA   $C082
0345-   60          RTS
0346-   00          BRK
0347-   01 60       ORA   ($60,X)
0349-   02          ???
034A-   00          BRK
034B-   00          BRK
034C-   00          BRK
034D-   58          CLI
034E-   03          ???
034F-   00          BRK
0350-   D0 00       BNE   $0352
0352-   00          BRK
```

```
LOAD RET2

END OF DATA

:L

               ORG   $300
     IOB       EQU   $03
     IDA       EQU   $33
     RWTS      EQU   $3D9
     DTH       EQU   $33C
     TRACK     EQU   $337
     SECTOR    EQU   $338
     WON       EQU   $C081
     WOFF      EQU   $C082
               LDA   WON
               LDA   WON
     START     LDA   #IOB
               LDY   #IDA
               JSR   RWTS
     UP        INC   DTH
               INC   SECTOR
               LDX   #$10
               CPX   SECTOR
               BNE   NIN
               LDX   #$00
               STX   SECTOR
               INC   TRACK
     NIN       LDX   #$F0
               CPX   DTH
               BEQ   DONE
               JMP   START
     DONE      LDA   WOFF
               LDA   WOFF
               RTS
```

```
*BLOAD RETTAB,A$300 00 01 EF D8 00
*300LL

0300-   AD 81 C0      LDA   $C081
0303-   AD 81 C0      LDA   $C081
0306-   A9 03         LDA   #$03
0308-   A0 33         LDY   #$33
030A-   20 D9 03      JSR   $03D9
030D-   EE 3C 03      INC   $033C
0310-   EE 38 03      INC   $0338
0313-   A2 10         LDX   #$10
0315-   EC 38 03      CPX   $0338
0318-   D0 08         BNE   $0322
031A-   A2 00         LDX   #$00
031C-   8E 38 03      STX   $0338
031F-   EE 37 03      INC   $0337
0322-   A2 F0         LDX   #$F0
0324-   EC 3C 03      CPX   $033C
0327-   F0 03         BEQ   $032C
0329-   4C 06 03      JMP   $0306
032C-   AD 82 C0      LDA   $C082
032F-   AD 82 C0      LDA   $C082
0332-   60            RTS
0333-   01 60         ORA   ($60,X)
0335-   02            ???
0336-   00            BRK
0337-   00            BRK
0338-   00            BRK
0339-   44            ???
033A-   03            ???
033B-   00            BRK
033C-   D0 00         BNE   $033E
033E-   00            BRK
033F-   01 00         ORA   ($00,X)
0341-   00            BRK
0342-   60            RTS
0343-   01 00         ORA   ($00,X)
0345-   01 EF         ORA   ($EF,X)
0347-   D8            CLD
```

```
:LOAD SV2

END OF DATA

:L

                ORG   $300
        IOB     EQU   $03
        IOA     EQU   $33
        RWTS    EQU   $3D9
        DTH     EQU   $33C
        TRACK   EQU   $337
        SECTOR  EQU   $338
        RON     EQU   $C080
        WOFF    EQU   $C082
                LDA   RON
                LDA   RON
        START   LDA   #IOB
                LDY   #IOA
                JSR   RWTS
        UP      INC   DTH
                INC   SECTOR
                LDX   #$10
                CPX   SECTOR
                BNE   NIN
                LDX   #$00
                STX   SECTOR
                INC   TRACK
        NIN     LDX   #$F0
                CPX   DTH
                BEQ   DONE
                JMP   START
        DONE    LDA   WOFF
                LDA   WOFF
                RTS
```

```
*BLOAD SVTAB,A$300 00 01 EF D8 00

*300LL

0300-   AD 80 C0     LDA   $C080
0303-   AD 80 C0     LDA   $C080
0306-   A9 03        LDA   #$03
0308-   A0 33        LDY   #$33
030A-   20 D9 03     JSR   $03D9
030D-   EE 3C 03     INC   $033C
0310-   EE 38 03     INC   $0338
0313-   A2 10        LDX   #$10
0315-   EC 38 03     CPX   $0338
0318-   D0 08        BNE   $0322
031A-   A2 00        LDX   #$00
031C-   8E 38 03     STX   $0338
031F-   EE 37 03     INC   $0337
0322-   A2 F0        LDX   #$F0
0324-   EC 3C 03     CPX   $033C
0327-   F0 03        BEQ   $032C
0329-   4C 06 03     JMP   $0306
032C-   AD 82 C0     LDA   $C082
032F-   AD 82 C0     LDA   $C082
0332-   60           RTS
0333-   01 60        ORA   ($60,X)
0335-   02           ???
0336-   00           BRK
0337-   00           BRK
0338-   00           BRK
0339-   44           ???
033A-   03           ???
033B-   00           BRK
033C-   D0 00        BNE   $033E
033E-   00           BRK
033F-   02           ???
0340-   00           BRK
0341-   00           BRK
0342-   60           RTS
0343-   01 00        ORA   ($00,X)
0345-   01 EF        ORA   ($EF,X)
0347-   D8           CLD
0348-   60           RTS
0349-   02           ???
034A-   00           BRK
```

```
LOAD OPEN

END OF DATA

:L

  WON      EQU  $C081
  RON      EQU  $C080
  SWEEPHI  EQU  $F000
  OFF      EQU  $C082
  BUFFA    EQU  $F006
  SWEEPA   EQU  $F009
           ORG  $032C
  OPEN     LDA  RON
           LDA  RON
           JSR  SWEEPHI
           LDA  WON
           LDA  WON
           LDA  #$10
           STA  SWEEPA
           LDA  OFF
           LDA  OFF
           RTS
           LDA  WON
           LDA  WON
           LDA  #$11
           STA  SWEEPA
           LDA  #$00
           STA  BUFFA
           LDA  RON
           LDA  RON
           RTS
```

```
*BLOAD OPEN.OBJO,A$300 00 01 EF D8 00

*300LL

0300-   AD 80 C0    LDA   $C080
0303-   AD 80 C0    LDA   $C080
0306-   20 00 F0    JSR   $F000
0309-   AD 81 C0    LDA   $C081
030C-   AD 81 C0    LDA   $C081
030F-   A9 10       LDA   #$10
0311-   8D 09 F0    STA   $F009
0314-   AD 82 C0    LDA   $C082
0317-   AD 82 C0    LDA   $C082
031A-   60          RTS
031B-   AD 81 C0    LDA   $C081
031E-   AD 81 C0    LDA   $C081
0321-   A9 11       LDA   #$11
0323-   8D 09 F0    STA   $F009
0326-   A9 00       LDA   #$00
0328-   8D 06 F0    STA   $F006
032B-   AD 80 C0    LDA   $C080
032E-   AD 80 C0    LDA   $C080
0331-   60          RTS
0332-   60          RTS
0333-   01 60       ORA   ($60,X)
0335-   02          ???
0336-   00          BRK
0337-   00          BRK
0338-   00          BRK
0339-   44          ???
033A-   03          ???
033B-   00          BRK
033C-   D0 00       BNE   $033E
033E-   00          BRK
033F-   01 00       ORA   ($00,X)
0341-   00          BRK
0342-   60          RTS
0343-   01 00       ORA   ($00,X)
0345-   01 EF       ORA   ($EF,X)
0347-   D8          CLD
0348-   60          RTS
0349-   02          ???
034A-   C0          BRK
034B-   00          BRK
```

References

1) Turan, M., Itil, M.D. Research Professor and Director, Division of Biological Psychiatry, New York Medical College, "Computer EEG drug data base, a new method for psychotropic drug development in man".

2) George Fein and David Galin, "EEG power spectra in normal and dyslexic children. I. Reliability during passive conditions". Electroencephalography and Clinical Neurophysiology, 1983, 55, 399-405.

3) The Gasser, "Correlating EEG and IQ: A new look at an old problem using computerized EEG parameters". Electroencephalography and Clinical Neurophysiology, 1983, 55, 493-504.

4) Lawrence A. Coben, "Frequency analysis of the resting awake EEG in mild senile dementia of Alzheimer type". Electroencephalography and Clinical Neurophysiology, 1983, 55, 372-380.

5) H. Moldofsky and F.A. Lue, "The relationship of alpha and delta EEG frequencies to pain and mood in"fibrositis" patients treated with chlorpromazine and L-tryptophan". Electroencephalography and Clinical Neurophysiology, 1980, 50, 71-80.

6) Applesoft II BASIC programming reference manual.

7) Apple II reference manual.

8) Applescope reference manual and scope driver addendum to the Applescope reference manual.

9) Apple DOS manual.

10) Apple 6502 assembler - Editor