Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation.

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen.
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
Automated EEG Spectrum Analysis System

by

Marinela C. Laguna

Dissertation submitted 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
Title of Thesis: Automated EEG spectrum analysis system

Thesis and Abstract Approved:

_________________________   _______________________
Dr. Stanley Reisman         Date
Associate Professor
Electrical Engineering

_________________________   _______________________

_________________________   _______________________

_________________________   _______________________

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:


<table>
<thead>
<tr>
<th>Collegiate Institutions attended</th>
<th>Date</th>
<th>Degree</th>
<th>Date of Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>N.J.I.T., Newark</td>
<td>1/83-5/85</td>
<td>M.S.B.E.</td>
<td>May, 1985</td>
</tr>
</tbody>
</table>

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.
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

<table>
<thead>
<tr>
<th>Chapter 1: Introduction</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 Electroencephalography (EEG) signals used in diagnosis and pharmacology</td>
<td>1</td>
</tr>
<tr>
<td>1.2 The automated EEG spectrum analysis system - generalities</td>
<td>12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 2: Description of the System</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 System Components</td>
<td>18</td>
</tr>
<tr>
<td>2.1.1 Filters</td>
<td>18</td>
</tr>
<tr>
<td>2.1.2 Apple II computer</td>
<td>20</td>
</tr>
<tr>
<td>a. Introduction</td>
<td>20</td>
</tr>
<tr>
<td>b. Apple II seen as a personal computer</td>
<td>20</td>
</tr>
<tr>
<td>c. Apple II seen as an oscilloscope</td>
<td>26</td>
</tr>
<tr>
<td>2.1.3 The printer and the microbuffer</td>
<td>41</td>
</tr>
<tr>
<td>2.2 System Software</td>
<td>43</td>
</tr>
<tr>
<td>2.2.1 Introduction</td>
<td>43</td>
</tr>
<tr>
<td>2.2.2 General flowchart</td>
<td>47</td>
</tr>
<tr>
<td>2.2.3 Variant 1</td>
<td>49</td>
</tr>
<tr>
<td>2.2.4 Variant 2</td>
<td>74</td>
</tr>
<tr>
<td>2.2.5 Variant 3</td>
<td>78</td>
</tr>
</tbody>
</table>
## Chapter 3: Experimental Results

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 Where is the Hoffmann-La Roche Research situated with respect to the computerized EEG in pharmacology</td>
<td>81</td>
</tr>
<tr>
<td>3.2 Validation of the new system, automated EEG spectrum analysis system</td>
<td>85</td>
</tr>
<tr>
<td>3.3 Conclusions</td>
<td>91</td>
</tr>
</tbody>
</table>

### Appendixes

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Information about the system components</td>
<td>116</td>
</tr>
<tr>
<td>B</td>
<td>Programming</td>
<td>128</td>
</tr>
<tr>
<td>C</td>
<td>The Apple video display, screen formats, other Input/Output features</td>
<td>135</td>
</tr>
<tr>
<td>D</td>
<td>The Applescope</td>
<td>147</td>
</tr>
<tr>
<td>E</td>
<td>Applescope disk supervisor and Disk Operating System (DOS)</td>
<td>152</td>
</tr>
<tr>
<td>F</td>
<td>BASIC programs listing</td>
<td>158</td>
</tr>
<tr>
<td>G</td>
<td>Machine language routines listing</td>
<td>216</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>229</td>
</tr>
</tbody>
</table>
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1.a</td>
<td>Control group reliabilities</td>
<td>2</td>
</tr>
<tr>
<td>1.1.b</td>
<td>Dyslexic group reliabilities</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Computer system validation results</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>Block diagram of the automated spectrum analysis system</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>The Apple main board</td>
<td>121</td>
</tr>
<tr>
<td>2.3</td>
<td>General information about the microprocessor</td>
<td>23</td>
</tr>
<tr>
<td>2.4</td>
<td>ROM memory organization</td>
<td>120</td>
</tr>
<tr>
<td>2.5</td>
<td>Applesoft versions memory map</td>
<td>131</td>
</tr>
<tr>
<td>2.6</td>
<td>RAM memory organization</td>
<td>120</td>
</tr>
<tr>
<td>2.7</td>
<td>System memory map</td>
<td>120</td>
</tr>
<tr>
<td>2.8</td>
<td>Super RAM II usage</td>
<td>118</td>
</tr>
<tr>
<td>2.9</td>
<td>The Apple video display</td>
<td>136</td>
</tr>
<tr>
<td>2.10</td>
<td>Screen formats characteristics</td>
<td>137</td>
</tr>
<tr>
<td>2.11.a</td>
<td>Video display memory ranges</td>
<td>140</td>
</tr>
<tr>
<td>2.11.b</td>
<td>Screen mode combinations</td>
<td>140</td>
</tr>
<tr>
<td>2.12</td>
<td>Deduction of the combinations for the screen</td>
<td>141</td>
</tr>
<tr>
<td>2.13</td>
<td>Input/Output special locations</td>
<td>146</td>
</tr>
<tr>
<td>2.14</td>
<td>Apple II varieties</td>
<td>146</td>
</tr>
<tr>
<td>2.15</td>
<td>Applescope general view</td>
<td>29</td>
</tr>
<tr>
<td>2.16</td>
<td>Data acquisition cycle</td>
<td>30</td>
</tr>
</tbody>
</table>
2.17. Control words--------------------------------------------- 31
2.18. Applescope monitor display--------------------------------- 32
2.19. Flowchart for data acquisition cycle------------------------ 34
2.20. Memory after data acquisition cycle------------------------ 149
2.21. Voltage scales--------------------------------------------- 150
2.22. The vertical expansion------------------------------------- 36
2.23. The horizontal expansion----------------------------------- 38
2.24. Display buffer memory-------------------------------------- 151
2.25. Data disk map and intermediate memory map------------------ 46
2.26. General flowchart------------------------------------------ 48
2.27. Variant 1 flowchart---------------------------------------- 50-51
2.28.a. Flowchart for machine language routines used in
AUTODUAL1------------------------------------------------------- 55
2.28.b. General flowchart for saving or retrieving machine
language routine----------------------------------------------- 56
2.28.c. Flowchart for AUTODUAL1 program------------------------- 57
2.29. Manipulation of information in SPC program----------------- 59
2.30. Flowchart for SPC program---------------------------------- 61-62
2.31.a. Flowchart for AVERAGE program--------------------------- 64
2.31.b. Flowchart for AVELOTAB and its subroutine,
AVEHI.OBJO----------------------------------------------------- 65
2.32.a. Memory map for 8 spectrums sum and average for
channels A and B----------------------------------------------- 67
2.32.b. Zero page pointers used for indexed-indirect
addressing in AVELOTAB and AVELOTABB------------------------ 67
2.33.a. Numbers written by AXIS and AXIS-B programs on
"y" and "x" axis----------------------------- 69
2.33.b. Ideal and real frequency bands limits---------- 69
2.34. Variant 2 flowchart----------------------------- 76-77
2.35. Disks needed to process the EEG signals---------- 80
3.1. Experiments outline------------------------------- 83
3.2. Statistical results------------------------------- 84
3.3. Results outline------------------------------- 88
3.4.1.a. Suriclone control experiment - new system, monkey 461- 93
3.4.1.b. Suriclone drug experiment - new system, monkey 461--- 94
3.4.2.a. Suriclone control experiment - old system, monkey 461- 95
3.4.2.b. Suriclone control results - old system, monkey 461--- 96
3.4.2.c. Suriclone drug experiment - old system, monkey 461--- 97
3.4.2.d. Suriclone drug results - old system, monkey 461------ 98
3.5.1.a. Suriclone control experiment - new system, monkey 514- 99
3.5.1.b. Suriclone drug experiment - new system, monkey 514---- 100
3.5.2.a. Suriclone control experiment - old system, monkey 514- 101
3.5.2.b. Suriclone control results - old system, monkey 514---- 102
3.5.2.c. Suriclone drug experiment - old system, monkey 514---- 103
3.5.2.d. Suriclone drug results - old system, monkey 514------- 104
3.6.1.a. Halazepam control experiment - new system, monkey 461- 105
3.6.1.b. Halazepam drug experiment - new system, monkey 461---- 106
3.6.2.a. Halazepam control experiment - old system, monkey 461- 107
3.6.2.b. Halazepam control results - old system, monkey 461---- 108
3.6.2.c. Halazepam drug results - old system, monkey 461------- 109
3.7.1.a. Halazepam control experiment - new system, monkey 680- 110
3.7.1.b. Halazepam drug experiment - new system, monkey 680---- 111
3.7.2.a. Halazepam control experiment - old system, monkey 680- 112
3.7.2.b. Halazepam control results - old system, monkey 680---- 113
3.7.2.c. Halazepam drug experiment - old system, monkey 680---- 114
3.7.2.d. Halazepam drug results - old system, monkey 680------- 115
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, $\Delta$, $\theta$, $\alpha_1$, $\alpha_2$, $\beta_1$, $\beta_2$, 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**
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:

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

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 psycho-
tropic 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 in ani-
mals, produce similar CEEG profiles.

7. Psychotropic drugs with similar clinical (therapeutic) effects
in patients produce similar CEEG profiles. Therapeutical
"unequivalent" compounds (antipsychotics versus anxiolytics)
produce different CEEG profiles (unequivalent physiological
effects).

Hypotheses:

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 therapeu-
tic 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.
<table>
<thead>
<tr>
<th>Date Base Drug Classes</th>
<th>No. of Drugs</th>
<th>Correctly Classified</th>
<th>Not Classified</th>
<th>Incorrectly Classified</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anxiolytics</td>
<td>52</td>
<td>45</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Antidepressants</td>
<td>35</td>
<td>17</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>Psychostimulants</td>
<td>17</td>
<td>14</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Neuroleptics</td>
<td>10</td>
<td>3</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Overall</td>
<td>114</td>
<td>79</td>
<td>16</td>
<td>19</td>
</tr>
</tbody>
</table>

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 $31.25 \text{ Hz}/400 = 0.078125 \text{ Hz}$. 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 ($16 \times 512 = 8 \times 2 \times 512 = 8 \times 1024 \text{ bytes}$). 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, $2 \times 64 \times 512 \text{ bytes} = 64 \text{ Kbytes}$. 
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 54 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 Kbyte's 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 6502 Microprocessor

<table>
<thead>
<tr>
<th><strong>Model:</strong></th>
<th>MCS6502/SY6502</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Manufactured by:</strong></td>
<td>MOS Technologies, Inc. Syneriiek Rockwell</td>
</tr>
<tr>
<td><strong>Number of instructions:</strong></td>
<td>56</td>
</tr>
<tr>
<td><strong>Addressing modes:</strong></td>
<td>13</td>
</tr>
<tr>
<td><strong>Accumulators:</strong></td>
<td>1 (A)</td>
</tr>
<tr>
<td><strong>Index registers:</strong></td>
<td>2 (X,Y)</td>
</tr>
<tr>
<td><strong>Other registers:</strong></td>
<td>Stack pointer (S) Processor status (P)</td>
</tr>
<tr>
<td><strong>Stack:</strong></td>
<td>256 bytes, fixed</td>
</tr>
<tr>
<td><strong>Status flags:</strong></td>
<td>N (sign) C (carry) V (overflow)</td>
</tr>
<tr>
<td><strong>Other flags:</strong></td>
<td>I (interrupt disable) D (Decimal arithmetic) B (Break)</td>
</tr>
<tr>
<td><strong>Interrupts:</strong></td>
<td>2 (IRQ, NMI)</td>
</tr>
<tr>
<td><strong>Resets:</strong></td>
<td>1 (RES)</td>
</tr>
<tr>
<td><strong>Addressing range:</strong></td>
<td>$2^{16}$ (64K) locations</td>
</tr>
<tr>
<td><strong>Address bus:</strong></td>
<td>16 bits, parallel</td>
</tr>
<tr>
<td><strong>Data bus:</strong></td>
<td>8 bits, parallel Bidirectional</td>
</tr>
<tr>
<td><strong>Voltages:</strong></td>
<td>$\pm 5$ volts</td>
</tr>
<tr>
<td><strong>Power dissipation:</strong></td>
<td>.25 watt</td>
</tr>
<tr>
<td><strong>Clock frequency:</strong></td>
<td>1.023MHz</td>
</tr>
</tbody>
</table>

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
CON2 - to the post trigger delay counter
CON3 - to the sample rate control

} on analog 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 acquisition and display cycle.
Figure 2.15 Applescope General View
Figure 2.16 Data Acquisition Cycle
<table>
<thead>
<tr>
<th>CON1</th>
<th>8 bit value used for the trigger level comparison</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON2 Bit 7</td>
<td>High enables greater than threshold trigger</td>
</tr>
<tr>
<td>Bit 6</td>
<td>High enables less than threshold trigger</td>
</tr>
<tr>
<td>Bit 5</td>
<td>Low disables channel B trigger</td>
</tr>
<tr>
<td>Bit 4</td>
<td>Low disables channel A trigger</td>
</tr>
<tr>
<td>Bit 3</td>
<td>Low enables channel A data</td>
</tr>
<tr>
<td>Bit 2</td>
<td>Low enables channel B data</td>
</tr>
<tr>
<td>Bit 1</td>
<td>Set the memory cycle and post trigger delay</td>
</tr>
<tr>
<td>Bit 0</td>
<td></td>
</tr>
</tbody>
</table>

| 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](image)
The trace parameters for both channels are displayed in the bottom four lines of the display.

The program control can be transferred 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 one 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:

\[
\begin{align*}
\text{vertical expansion is } & "x8" \quad \Rightarrow \text{voltage scale } = \frac{7.1 \text{ mV/step}}{8 \text{ div/step}} \\
\text{resolution is } & 7.1 \text{ mV/step} \\
\end{align*}
\]

\[
\begin{align*}
\Rightarrow & = 24.8 \frac{\text{mV}}{\text{div}} = 0.0248 \frac{\text{V}}{\text{div}}.
\end{align*}
\]

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, \(256 \times 2^n\) points appear on the screen (\(n\) is the value of the horizontal expansion, the absolute value). For positive values we have expansion,
<table>
<thead>
<tr>
<th>Vertical Expansion</th>
<th>Description</th>
<th>28 pixels</th>
<th>1 step</th>
<th>Calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;x8&quot;</td>
<td></td>
<td>8 pixels</td>
<td>1 step</td>
<td>8/28 div./step</td>
</tr>
<tr>
<td>&quot;x4&quot;</td>
<td></td>
<td>4 pixels</td>
<td>1 step</td>
<td>4/28 div./step</td>
</tr>
<tr>
<td>&quot;x2&quot;</td>
<td></td>
<td>2 pixels</td>
<td>1 step</td>
<td>2/28 div./step</td>
</tr>
<tr>
<td>&quot;x1&quot;</td>
<td></td>
<td>1 pixels</td>
<td>1 step</td>
<td>1/28 div./step</td>
</tr>
</tbody>
</table>

*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 $= 256 \times 2^2 = 1024$ 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 rates slower than 1 msec/div (time > 1 msec/div.), are given in Figure 2.24 (Appendix D).
HE = 0

Fig. 2.23

Horizontal Expansion

HE = +2

HE = -2

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 $IFFF 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 with 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 signal 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 cannot 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 lose 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 acquisition 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
START
AUTODUAL
CALIBRATE
ALTERNATIVE

20480=0 Data, spectra
AUTODUAL 1
STC

20480=1 Data, spectra and processing

YES
20480=1 ?
NO
STOP

RETSPC
CHOICE

16836=0
A
what channel?
16836=1

AVERAGE
CHOICE

2

B

4

1

3

3

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.OBJO is used to save the 8K of Super RAM on disk. HIALL.OBJO 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.OBJO 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 LOWALLTABLE binary file. HIALL.OBJO 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 language routines work within AUTODUAL 1 program. Figure 2.28b gives the general steps for saving or retrieving machine language routines and figure 2.28c represents the detailed flow chart for AUTODUAL 1 program.

SPC - 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.OBJO 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.OBJO "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/OCT 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

1. RETTAB = RET2.OBJ/ + IOB/DST (§300)
2. OPEN.OBJ/ (§32c)
3. SWEEP.OBJ/ (§F300)
4. SAVESC.OBJ/ (§300)
5. SAVTAB = 512.OBJ/ + IOB/DST (§304)
4. **SAVESPC.OBJ** 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.

**RETSPEC** 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

1. READ SAVE SPC OBJ
   Increment X

2. YES: ELOT
   NO: 2

3. YES: Y = max?
   NO: 3

4. $ = PEEK (20480)

5. $ = 1?
   NO: STOP
   YES: CLEAR SCREEN AND RETURN CONTROL TO RETSPC
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.OBJO 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.OBJ0
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.OBJO 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.OBJO** 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.OBJO work together in order to obtain the average.
Figure 2.32a Memory Map for 8 Spectrums

Sum and Average for Channels A and B

Figure 2.32b Zero Page Pointers Used for Indexed Indirect Addressing in AVELOTAB and AVELOTABB
AVERAGESB 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.OBJO 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.OBJO.

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

<table>
<thead>
<tr>
<th>&quot;y&quot; axis</th>
<th>&quot;x&quot; axis</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.2</td>
<td>0.9</td>
</tr>
<tr>
<td>4.8</td>
<td>3.8</td>
</tr>
<tr>
<td>4.4</td>
<td>4.4</td>
</tr>
<tr>
<td>4.0</td>
<td>14.0</td>
</tr>
<tr>
<td>3.6</td>
<td>18.5</td>
</tr>
<tr>
<td>3.2</td>
<td>19.4</td>
</tr>
<tr>
<td>2.8</td>
<td>23.3</td>
</tr>
<tr>
<td>2.4</td>
<td>31.3</td>
</tr>
<tr>
<td>2.0</td>
<td>39.4</td>
</tr>
<tr>
<td>1.6</td>
<td>47.5</td>
</tr>
<tr>
<td>1.2</td>
<td>55.5</td>
</tr>
<tr>
<td>0.8</td>
<td>63.6</td>
</tr>
<tr>
<td>0.4</td>
<td>71.1</td>
</tr>
</tbody>
</table>

b. Ideal and Real Frequency Bands Limits

Figure 2.33
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.

\[
\frac{\sum_{j} \sum_{K} P_{j,K}^2}{\sum_{K} P_{j,K}^2} i = 1-13 \\
\times \times 100 \quad j = 1-1; \quad l = \text{no. of points in each band} \\
\text{K} = 1-256 \\
P_{j,K} = \text{spectral amplitude (1 byte in memory)} \\
A_i = \text{histogram coefficients}
\]

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:

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

\[
256 \text{ points represent } 71.1 \text{ Hz} \quad \frac{x \text{ points represent 2 Hz (the smallest band)}}{71.1} = \frac{256 \times 2}{256} = 7.2 \text{ points/Hz}
\]

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.

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

\[
7 \text{ points x 0.2 (7) = 1.94 Hz}
\]
Choosing 8 points:

\[ 8 \text{ points} \times 0.2 \ (7) = 2.22 \text{ Hz} \]

Taking into 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, \( 256 - [(2 \times 7) + (5 \times 14) + 5 \times 29] \) = 256 - 229 = 27 points/band. There are, in conclusion, 4 numbers of points/band. They are called J, K, T, U.

\[ J = 7 \quad T = 29 \]
\[ K = 14 \quad U = 27 \]

The correspondent frequency intervals are:

\[ f_J = f_7 = 7 \times \text{resolution} = 1.94 \text{ Hz} \quad \text{(first 2 bands)} \]
\[ 0.2 \ (7) \]
\[ f_K = f_{14} = 14 \times \text{resolution} = 3.8 \text{ Hz} \quad \text{(next 5 bands)} \]
\[ f_T = f_{29} = 29 \times \text{resolution} = 8.05 \text{ Hz} \quad \text{(next 5 bands)} \]
\[ f_U = f_{27} = 27 \times \text{resolution} = 7.5 \text{ Hz} \quad \text{(last band)} \]

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

\[ $4100 = \text{decimal} \ 16650 = 16384 + 2J + 5K + 5T + U. \]
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:

**CHOICE2** 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 then 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

START
AUTODUAL
CALIBRATE
ALTERNATIVE

START
AUTODUAL 1
SPC

20480=1?

STOP
(16896=0)

A (16896=0)

A - AVERAGE
FINAL - AVERAGE
SEE FIN. AVERAGE

B - AVERAGE
FINAL - AVERAGE
SEE FIN. AVERAGE

1. DATA, SP, and processing?
2. DATA, SP, or DATA, SP, and processing?
3. DATA, SP, (20480=1)
4. DATA, SP, (20480=1)
5. DATA, SP, and processing?
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.
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° other monkey: 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:

$$y = \frac{\text{band power}}{\text{total power}}_{\text{control experiment}} + \frac{\text{band power}}{\text{total power}}_{\text{drug experiment}}$$

(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
<table>
<thead>
<tr>
<th>Drug</th>
<th>Monkey Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suriclone</td>
<td>461 (channel B) I</td>
</tr>
<tr>
<td></td>
<td>514 (channel A) II</td>
</tr>
<tr>
<td>Halazepam</td>
<td>461 (channel B) III</td>
</tr>
<tr>
<td></td>
<td>680 (channel A) IV</td>
</tr>
</tbody>
</table>

**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 very 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.
Figure 3.4.1.a. Suriclone control experiment - new system, monkey 461.
Figure 3.4.1.b. Suriclone drug experiment - new system, monkey 461
Figure 3.4.2.a. Suricione 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 1/4/84

<table>
<thead>
<tr>
<th>BAND: 1 2 3 4 5 6 7 8 TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK END</td>
</tr>
<tr>
<td>0.00 1.99 4.02 8.01 11.99 15.98 20.04 24.02 0.00</td>
</tr>
<tr>
<td>TIME</td>
</tr>
<tr>
<td>1.99 4.02 8.01 11.99 15.98 20.04 24.02 31.25 31.25</td>
</tr>
</tbody>
</table>

<- AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS) -> TOTAL

| 1 13:13 | 2985 2981 4261 2123 1376 997 726 865 15324. |
| 2 13:27 | 2453 3224 4554 2209 1438 1020 696 899 16459. |
| 3 13:40 | 2966 3278 4364 2073 1371 1622 756 1002 16475. |
| 4 13:54 | 2408 3163 4117 2870 1390 1034 761 1011 15952. |

PERCENT FOR EACH SPECTRUM BAND

| 1 13.61 | 18.87 | 27.91 | 13.65 | 8.90 | 6.51 | 4.74 | 5.64 |
| 2 14.87 | 15.65 | 27.57 | 13.58 | 8.72 | 6.18 | 4.22 | 3.45 |
| 3 15.59 | 20.61 | 25.61 | 12.58 | 8.32 | 6.28 | 4.60 | 6.08 |
| 4 15.08 | 15.65 | 25.81 | 12.98 | 8.71 | 6.48 | 4.77 | 6.34 |

SUM OF SPECTRA - PERCENTAGE OF TOTAL

| 1 AND 2 | 14.26 | 19.55 | 27.69 | 13.61 | 8.84 | 6.34 | 4.47 | 5.54 |
| 3 AND 4 | 15.84 | 13.22 | 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 |

NDV43: 461000002.89
Figure 3.4.2.c. Suriclone drug experiment - old system, monkey 461
Figure 3.4.2.d. Suriclone drug results - old system, monkey 461

SPECTRAL DENSITY ESTIMATES BY BANDWIDTH

EFG FROM THE FR CORTEX

017-9887 AT 0.50 MG/KG, IN MONKEY 461. EXPERIMENT PERFORMED ON 1/5/84

<table>
<thead>
<tr>
<th>BAND</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK</td>
<td>0.00</td>
<td>1.39</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
<td>15.98</td>
<td>20.04</td>
<td>24.02</td>
<td>0.00</td>
</tr>
<tr>
<td>TIME</td>
<td>1.99</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
<td>15.98</td>
<td>20.04</td>
<td>24.02</td>
<td>31.25</td>
<td>31.25</td>
</tr>
</tbody>
</table>

< AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS) >

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6288</td>
<td>4304</td>
<td>5491</td>
<td>5263</td>
<td>3335</td>
<td>2943</td>
<td>2848</td>
<td>5255</td>
<td>35210</td>
</tr>
<tr>
<td>2</td>
<td>4347</td>
<td>3572</td>
<td>5409</td>
<td>4513</td>
<td>3637</td>
<td>3788</td>
<td>3806</td>
<td>6465</td>
<td>35437</td>
</tr>
<tr>
<td>3</td>
<td>3251</td>
<td>5053</td>
<td>4027</td>
<td>3595</td>
<td>3573</td>
<td>3892</td>
<td>6393</td>
<td>32922</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4534</td>
<td>4391</td>
<td>5563</td>
<td>4413</td>
<td>4056</td>
<td>3307</td>
<td>3089</td>
<td>7942</td>
<td>39511</td>
</tr>
</tbody>
</table>

PERCENT FOR EACH SPECTRUM BAND

| 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 | 15.82 |
| AND 6 | 0.00  | 0.00  | 0.00  | 0.00  | 0.00  | 0.00 | 0.00 | 0.00  |

SUM OF SPECTRA - PERCENTAGE OF TOTAL

NOVA3: M461179887.11
Figure 3.5.1.a. Suriclone control experiment - new system, monkey 514.
Figure 3.5.1.b. Suriclon drug experiment - new system, monkey 514
Figure 3.5.2.a. Suriclone control experiment - old system, monkey 514

FR CORT            EEG COMPRESSED SPECTRAL ARRAY
CONTROL EXPERIMENT, 1.00 ML PO IN MONKEY 514 ON 1/4/84
4 SPECTRA PROCESSED FROM FILE M514000002.89 ON 1/5/84
Figure 3.5.2.b. Suriclione control results - old system, monkey 514.

<table>
<thead>
<tr>
<th>Band</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>END</td>
<td>0.00</td>
<td>1.99</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
</tr>
<tr>
<td>TIME</td>
<td>1.99</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
<td>15.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Area under spectrum within band (arb)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Percent for each spectrum band</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sum of spectra - percentage of</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AND 2</td>
</tr>
<tr>
<td>3 AND 4</td>
</tr>
<tr>
<td>5 AND 6</td>
</tr>
</tbody>
</table>
Figure 3.5.2.c. Suriclone drug experiment - old system, monkey 514.

FR CORT EEG COMP
RO 17-9887, 0.50 MG/KG PO IN
4 SPECTRA PROCESSED FROM FILE

FREQUENCY (HERTZ)
## Spectral Density Estimates by Bandwidth

**EEG from the FR CORT**

**ROI7-9887 at 0.50 mg/kg, in Monkey 514.**  
**Exp. performed on 1/5/84**

<table>
<thead>
<tr>
<th>Band</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clock</td>
<td>0.00</td>
<td>1.39</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
<td>15.96</td>
<td>20.04</td>
<td>24.02</td>
<td>31.25</td>
</tr>
<tr>
<td>Time</td>
<td>1.99</td>
<td>4.00</td>
<td>8.01</td>
<td>11.99</td>
<td>15.96</td>
<td>20.04</td>
<td>24.02</td>
<td>31.25</td>
<td></td>
</tr>
</tbody>
</table>

**Area under spectrum within band (Arbitrary units)**

| 1 | 666 | 745 | 1252 | 1575 | 1608 | 1795 | 3666 | 3608 | 14527 |
| 2 | 637 | 784 | 1467 | 2059 | 1937 | 2158 | 3166 | 3203 | 15413 |
| 3 | 1074 | 1102 | 2069 | 2575 | 2442 | 2667 | 3253 | 2001 | 18183 |
| 4 | 1591 | 1733 | 2392 | 2844 | 2480 | 2601 | 2541 | 2816 | 19508 |

**Percent for each spectrum band**

| 1 | 4.69% | 5.13% | 8.69% | 10.84% | 11.07% | 12.36% | 21.11% | 26.21% |
| 2 | 4.13% | 5.06% | 9.52% | 13.36% | 12.57% | 14.00% | 20.55% | 29.78% |
| 3 | 5.91% | 6.06% | 11.38% | 14.16% | 13.43% | 14.87% | 17.89% | 16.50% |
| 4 | 7.69% | 8.88% | 15.74% | 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% | 29.82% | 23.42% |
| 3 and 4 | 6.83% | 7.52% | 13.43% | 14.33% | 13.36% | 13.33% | 15.37% | 15.43% |
| 5 and 5 | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |

**NOVA3: M514179887.11**
Figure 3.6.1.a. Halazepam control experiment - new system, monkey 461
HALAZEPAM - DRUG
8-0426
M461, ch.B

Figure 3.6.1.b. Halazepam drug experiment - new system, monkey 461
Figure 3.6.2.a. Halazepam control experiment - old system, monkey 461

FR CORT, EEG COMPRESSED SPECTRAL ARRAY
CONTROL EXPERIMENT, 1.00 ML PO IN MONKEY 461 ON 4/10/84
4 SPECTRA PROCESSED FROM FILE M461000002.93 ON 5/4/84

Frequency (Hertz)

0 2 4 6 8 10 12 16 20 24 31

11:54 11:40 11:27 11:13
<table>
<thead>
<tr>
<th>BAND</th>
<th>TOT</th>
<th>1.99</th>
<th>8.01</th>
<th>11.99</th>
<th>15.98</th>
<th>18.04</th>
<th>24.04</th>
<th>30.25</th>
<th>36.45</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>23</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>33</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>43</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>53</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>63</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>73</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>83</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
<tr>
<td>93</td>
<td>1.11</td>
<td>0.90</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
<td>0.97</td>
<td>0.98</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Band</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Upper and lower frequency limit of each band**

<table>
<thead>
<tr>
<th>#</th>
<th>TIME</th>
<th>1.99</th>
<th>4.02</th>
<th>8.01</th>
<th>11.99</th>
<th>15.98</th>
<th>20.04</th>
<th>24.02</th>
<th>0.00</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CLOK</td>
<td>0.00</td>
<td>4.02</td>
<td>8.01</td>
<td>11.99</td>
<td>15.98</td>
<td>20.04</td>
<td>24.02</td>
<td>0.00</td>
</tr>
</tbody>
</table>

**Area under spectrum within band (arbitrary units)**

- **Total**

<table>
<thead>
<tr>
<th></th>
<th>11:13</th>
<th>11:27</th>
<th>11:40</th>
<th>11:54</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3491</td>
<td>3479</td>
<td>3360</td>
<td>3778</td>
</tr>
<tr>
<td>2</td>
<td>4378</td>
<td>4244</td>
<td>4238</td>
<td>4518</td>
</tr>
<tr>
<td>3</td>
<td>6057</td>
<td>5864</td>
<td>5647</td>
<td>6096</td>
</tr>
<tr>
<td>4</td>
<td>3224</td>
<td>3075</td>
<td>3326</td>
<td>3403</td>
</tr>
<tr>
<td></td>
<td>2232</td>
<td>2184</td>
<td>2400</td>
<td>2463</td>
</tr>
<tr>
<td></td>
<td>1965</td>
<td>2134</td>
<td>2363</td>
<td>2377</td>
</tr>
<tr>
<td></td>
<td>2177</td>
<td>2226</td>
<td>2423</td>
<td>2392</td>
</tr>
<tr>
<td></td>
<td>5222</td>
<td>5475</td>
<td>5763</td>
<td>5718</td>
</tr>
<tr>
<td></td>
<td>28638</td>
<td>28674</td>
<td>29520</td>
<td>30737</td>
</tr>
</tbody>
</table>

**Percent for each spectrum band**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.88</td>
<td>15.26</td>
<td>21.15</td>
<td>11.26</td>
<td>7.69</td>
<td>6.93</td>
<td>7.68</td>
<td>18.23</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12.13</td>
<td>14.80</td>
<td>20.45</td>
<td>10.72</td>
<td>7.62</td>
<td>7.41</td>
<td>7.76</td>
<td>19.10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>11.36</td>
<td>14.36</td>
<td>19.13</td>
<td>11.27</td>
<td>8.13</td>
<td>8.08</td>
<td>8.21</td>
<td>19.52</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>12.29</td>
<td>14.70</td>
<td>19.83</td>
<td>11.07</td>
<td>8.01</td>
<td>7.73</td>
<td>7.78</td>
<td>18.58</td>
<td></td>
</tr>
</tbody>
</table>

**Sum of spectra - Percentage of total**

<table>
<thead>
<tr>
<th></th>
<th>1 AND 2</th>
<th>3 AND 4</th>
<th>5 AND 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.00</td>
<td>10.89</td>
<td>6.00</td>
</tr>
<tr>
<td>2</td>
<td>14.53</td>
<td>11.17</td>
<td>6.00</td>
</tr>
<tr>
<td>3</td>
<td>14.53</td>
<td>11.17</td>
<td>6.00</td>
</tr>
<tr>
<td>4</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>5</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

NOVA3 M461086728.11
Figure 3.7.1.a. Halazepam control experiment - new system, monkey 680
Figure 3.7.1.b. Halazepam drug experiment - new system, monkey 680
Figure 3.7.2.a. Halazepam control experiment - old system, monkey 680

FR CORT  EEG COMPRESSED SPECTRAL ARRAY
CONTROL EXPERIMENT, 1.00 ML PO IN MONKEY 680 ON 4/10/84
4 SPECTRA PROCESSED FROM FILE H680000002.93 ON 5/4/84.
Figure 3.7.2.b. Halazepam control results - 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

<table>
<thead>
<tr>
<th>BAND</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLOCK</td>
<td>0.00</td>
<td>1.00</td>
<td>4.02</td>
<td>8.01</td>
<td>11.00</td>
<td>15.00</td>
<td>20.00</td>
<td>24.02</td>
<td>0.00</td>
</tr>
<tr>
<td>TIME</td>
<td>1.99</td>
<td>4.02</td>
<td>8.01</td>
<td>11.00</td>
<td>15.00</td>
<td>20.00</td>
<td>24.00</td>
<td>31.25</td>
<td>31.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AREA UNDER SPECTRUM WITHIN BAND (ARBITRARY UNITS)</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 11:13**</td>
<td>1414</td>
</tr>
<tr>
<td>2 11:27**</td>
<td>1034</td>
</tr>
<tr>
<td>3 11:40**</td>
<td>1130</td>
</tr>
<tr>
<td>4 11:54**</td>
<td>1180</td>
</tr>
</tbody>
</table>

PERCENT FOR EACH SPECTRUM BAND

<table>
<thead>
<tr>
<th>BAND</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22.02</td>
<td>22.21</td>
<td>25.55</td>
<td>12.37</td>
<td>7.51</td>
<td>5.05</td>
<td>3.02</td>
<td>2.27</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>19.07</td>
<td>22.55</td>
<td>27.27</td>
<td>12.70</td>
<td>7.04</td>
<td>4.93</td>
<td>3.38</td>
<td>3.06</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20.59</td>
<td>22.41</td>
<td>27.04</td>
<td>13.78</td>
<td>7.43</td>
<td>4.19</td>
<td>2.59</td>
<td>1.97</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>22.06</td>
<td>23.45</td>
<td>27.64</td>
<td>12.72</td>
<td>6.99</td>
<td>3.78</td>
<td>2.13</td>
<td>1.23</td>
<td></td>
</tr>
</tbody>
</table>

SUM OF SPECTRA - PERCENTAGE OF TOTAL

<table>
<thead>
<tr>
<th>ALL AND</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 AND 2</td>
<td>20.64</td>
<td>22.37</td>
<td>28.35</td>
<td>12.52</td>
<td>7.29</td>
<td>4.99</td>
<td>3.19</td>
<td>2.64</td>
<td></td>
</tr>
<tr>
<td>3 AND 4</td>
<td>21.32</td>
<td>22.92</td>
<td>27.33</td>
<td>13.25</td>
<td>7.22</td>
<td>3.93</td>
<td>2.36</td>
<td>1.61</td>
<td></td>
</tr>
<tr>
<td>5 AND 6</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>

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

NOVA3: M680000002.93
Figure 3.7.2.c.
Halazepam drug experiment - old system, monkey 680

FR CORT
EEG COMPRESSED SPECTRAL ARRAY
RO 8-6728, 10.00 MG/KG PO IN MONKEY 680 ON 4/11/84
4 SPECTRA PROCESSED FROM FILE M680086728.11 ON 5/4/84
**Figure 3.7.2.d.** Halazepam drug results - old system, monkey 680

<table>
<thead>
<tr>
<th>Band</th>
<th>Clock Time</th>
<th>Upper and Lower Frequency Limit of Each Band</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.00</td>
<td>11.99 8.61 11.99 15.98 20.94 24.92 29.98 31.25</td>
<td>31.25</td>
</tr>
<tr>
<td></td>
<td>4.82</td>
<td>4.82 8.01 11.89 15.86 20.94 24.92 29.98 31.25</td>
<td>31.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Band</th>
<th>Area Under Spectrum within Band (Arbitrary Units)</th>
<th>Total Area</th>
<th>Total Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.84 6.84 3.64 1.14 0.43 0.16 0.08 0.04 0.02</td>
<td>20.86</td>
<td>9.37</td>
</tr>
<tr>
<td>2</td>
<td>18.41 16.31 13.42 11.53 9.64 7.76 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>3</td>
<td>21.42 17.34 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>4</td>
<td>23.86 18.76 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>5</td>
<td>25.30 18.93 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>6</td>
<td>26.33 18.84 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Band</th>
<th>Area Under Spectrum within Band (Arbitrary Units)</th>
<th>Total Area</th>
<th>Total Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.84 6.84 3.64 1.14 0.43 0.16 0.08 0.04 0.02</td>
<td>20.86</td>
<td>9.37</td>
</tr>
<tr>
<td>2</td>
<td>18.41 16.31 13.42 11.53 9.64 7.76 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>3</td>
<td>21.42 17.34 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>4</td>
<td>23.86 18.76 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>5</td>
<td>25.30 18.93 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
<tr>
<td>6</td>
<td>26.33 18.84 13.46 11.58 9.70 7.82 6.67 5.58 4.49</td>
<td>77.88</td>
<td>34.37</td>
</tr>
</tbody>
</table>
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.
<table>
<thead>
<tr>
<th>Switch Location</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decimal</td>
<td>Hexadecimal</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>NORMAL BANK $2000-$DFFF</strong></td>
<td></td>
</tr>
<tr>
<td>-16256 (40280)</td>
<td>$0080</td>
</tr>
<tr>
<td>-16255 (40281)</td>
<td>$0081</td>
</tr>
<tr>
<td>-16254 (40282)</td>
<td>$0082</td>
</tr>
<tr>
<td>-16253 (40283)</td>
<td>$0083</td>
</tr>
<tr>
<td><strong>ALTERNATE BANK $2000-$DFFF</strong></td>
<td></td>
</tr>
<tr>
<td>-16248 (49288)</td>
<td>$0086</td>
</tr>
<tr>
<td>-16247 (49289)</td>
<td>$0087</td>
</tr>
<tr>
<td>-16245 (49291)</td>
<td>$0089</td>
</tr>
</tbody>
</table>

*With $2000-$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.
Figure 2.4 ROM Memory Organization

Table 16: RAM Organization and Usage

<table>
<thead>
<tr>
<th>Page Number</th>
<th>Decimal Hex</th>
<th>Used For</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>System Programs</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>System Stack</td>
</tr>
<tr>
<td>2</td>
<td>02</td>
<td>GETLN Input Buffer</td>
</tr>
<tr>
<td>3</td>
<td>03</td>
<td>Monitor Vector Locations</td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>Text and Lo-Res Graphics</td>
</tr>
<tr>
<td>5</td>
<td>05</td>
<td>Primary Page Storage</td>
</tr>
<tr>
<td>6</td>
<td>06</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>07</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>08</td>
<td>Text and Lo-Res Graphics</td>
</tr>
<tr>
<td>9</td>
<td>09</td>
<td>Secondary Page Storage</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>FREE</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>Hi-Res Graphics</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td>Hi-Res Graphics</td>
</tr>
<tr>
<td>63</td>
<td>3F</td>
<td>Primary Page Storage</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>Secondary Page Storage</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>SBF</td>
<td></td>
</tr>
</tbody>
</table>

Figure 2.6 RAM Memory Organization

Table 17: ROM Organization and Usage

<table>
<thead>
<tr>
<th>Page Number</th>
<th>Decimal Hex</th>
<th>Used By</th>
</tr>
</thead>
<tbody>
<tr>
<td>208</td>
<td>100</td>
<td>Programmer’s Aid #1</td>
</tr>
<tr>
<td>212</td>
<td>104</td>
<td>Applesoft II Basic</td>
</tr>
<tr>
<td>216</td>
<td>108</td>
<td>INTEGER BASIC</td>
</tr>
<tr>
<td>220</td>
<td>10C</td>
<td></td>
</tr>
<tr>
<td>224</td>
<td>10E</td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>232</td>
<td>116</td>
<td></td>
</tr>
<tr>
<td>236</td>
<td>11A</td>
<td></td>
</tr>
<tr>
<td>240</td>
<td>11E</td>
<td></td>
</tr>
<tr>
<td>244</td>
<td>120</td>
<td>Utility Subroutines</td>
</tr>
<tr>
<td>248</td>
<td>124</td>
<td>Monitor ROM</td>
</tr>
<tr>
<td>252</td>
<td>128</td>
<td>Autostart ROM</td>
</tr>
</tbody>
</table>

Figure 2.7 System Memory Map

Table 16: RAM Organization and Usage

<table>
<thead>
<tr>
<th>Page Number</th>
<th>Decimal Hex</th>
<th>Used For</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>00</td>
<td>System Programs</td>
</tr>
<tr>
<td>1</td>
<td>01</td>
<td>System Stack</td>
</tr>
<tr>
<td>2</td>
<td>02</td>
<td>GETLN Input Buffer</td>
</tr>
<tr>
<td>3</td>
<td>03</td>
<td>Monitor Vector Locations</td>
</tr>
<tr>
<td>4</td>
<td>04</td>
<td>Text and Lo-Res Graphics</td>
</tr>
<tr>
<td>5</td>
<td>05</td>
<td>Primary Page Storage</td>
</tr>
<tr>
<td>6</td>
<td>06</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>07</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>08</td>
<td>Text and Lo-Res Graphics</td>
</tr>
<tr>
<td>9</td>
<td>09</td>
<td>Secondary Page Storage</td>
</tr>
<tr>
<td>10</td>
<td>0A</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0B</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0C</td>
<td>FREE</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td>RAM</td>
</tr>
<tr>
<td>31</td>
<td>1F</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>20</td>
<td>Hi-Res Graphics</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td>Hi-Res Graphics</td>
</tr>
<tr>
<td>63</td>
<td>3F</td>
<td>Primary Page Storage</td>
</tr>
<tr>
<td>64</td>
<td>40</td>
<td>Secondary Page Storage</td>
</tr>
<tr>
<td>through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>5F</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>through</td>
<td></td>
<td></td>
</tr>
<tr>
<td>191</td>
<td>SBF</td>
<td></td>
</tr>
</tbody>
</table>
Figure 2.2 The Apple Main Board
Peripheral connector pinout
<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>D0 SELECT</td>
<td>This line, normally high, will become low when the microprocessor references page SCN, where n is the individual slot number. This signal becomes active during φ0 and will drive 10 LSTTL loads*. This signal is not present on peripheral connector Φ.</td>
</tr>
<tr>
<td>2-17</td>
<td>A8-A15</td>
<td>The buffered address bus. The address on these lines becomes valid during φ1 and remains valid through φ0. These lines will each drive 5 LSTTL loads*.</td>
</tr>
<tr>
<td>18</td>
<td>R/W</td>
<td>Buffered Read/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*.</td>
</tr>
<tr>
<td>19</td>
<td>SYNC</td>
<td>On peripheral connector 7 only, this pin is connected to the video timing generator's SYNC signal.</td>
</tr>
<tr>
<td>20</td>
<td>D0 STROBE</td>
<td>This line goes low during φ0 when the address bus contains an address between S'CR0 and S'CRFF. This line will drive 4 LSTTL loads*.</td>
</tr>
<tr>
<td>21</td>
<td>RDY</td>
<td>The 6582's RDY input. Pulling this line low during φ1 will halt the microprocessor, with the address bus holding the address of the current location being fetched.</td>
</tr>
<tr>
<td>22</td>
<td>DMA</td>
<td>Pulling this line low disables the 6582's address bus and halts the microprocessor. This line is held high by a 3kΩ resistor to +5v.</td>
</tr>
<tr>
<td>23</td>
<td>INT OUT</td>
<td>Daisy-chained interrupt output to lower priority devices. This pin is usually connected to pin 23 (INT IN).</td>
</tr>
<tr>
<td>24</td>
<td>DMA OUT</td>
<td>Daisy-chained DMA output to lower priority devices. This pin is usually connected to pin 22 (DMA IN).</td>
</tr>
<tr>
<td>25</td>
<td>+5v</td>
<td>+5 volt power supply. 500mA current is available for all peripheral cards.</td>
</tr>
<tr>
<td>26</td>
<td>GND</td>
<td>System electrical ground.</td>
</tr>
<tr>
<td>Pin</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>27</td>
<td>DMA IN</td>
<td>Daisy-chained DMA input from higher priority devices. Usually connected to pin 24 (DMA OUT).</td>
</tr>
<tr>
<td>26</td>
<td>INT IN</td>
<td>Daisy-chained interrupt input from higher priority devices. Usually connected to pin 23 (INT OUT).</td>
</tr>
<tr>
<td>29</td>
<td>NMI</td>
<td>Non- Maskable Interrupts. When this line is pulled low the Apple begins an interrupt cycle and jumps to the interrupt handling routine at location $3FB$.</td>
</tr>
<tr>
<td>30</td>
<td>IRQ</td>
<td>Interrupt Requset. When this line is pulled low the Apple begins an interrupt cycle only if the 6522's I (Interrupt disable) flag is not set. If so, the 6522 will jump to the interrupt handling subroutine whose address is stored in locations $3FE$ and $3FF$.</td>
</tr>
<tr>
<td>31</td>
<td>RES</td>
<td>When this line is pulled low the microprocessor begins a RESET cycle (see page 36).</td>
</tr>
<tr>
<td>32</td>
<td>INH</td>
<td>When this line is pulled low, all ROMs on the Apple board are disabled. This line is held high by a $3k\Omega$ resistor to $+5V$.</td>
</tr>
<tr>
<td>33</td>
<td>$-12V$</td>
<td>$-12V$ volt power supply. Maximum current is 200mA for all peripheral boards.</td>
</tr>
<tr>
<td>34</td>
<td>$-5V$</td>
<td>$-5V$ volt power supply. Maximum current is 200mA for all peripheral boards.</td>
</tr>
<tr>
<td>35</td>
<td>COLOR REF</td>
<td>On peripheral connector 3 only, this pin is connected to the 3.5MHz COLOR REFERENCE signal of the video generator.</td>
</tr>
<tr>
<td>36</td>
<td>7M</td>
<td>7MHz clock. This line will drive 2 LSTTL loads*.</td>
</tr>
<tr>
<td>37</td>
<td>Q3</td>
<td>2MHz asymmetrical clock. This line will drive 2 LSTTL loads*.</td>
</tr>
<tr>
<td>38</td>
<td>Q1</td>
<td>Microprocessor's phase one clock. This line will drive 2 LSTTL loads*.</td>
</tr>
<tr>
<td>39</td>
<td>USER 1</td>
<td>This line, when pulled low, disables all internal I/O address decoding**.</td>
</tr>
<tr>
<td>40</td>
<td>Q0</td>
<td>Microprocessor's phase zero clock. This line will drive 2 LSTTL loads*.</td>
</tr>
<tr>
<td>41</td>
<td>DEVICE SELECT</td>
<td>This line becomes active (low) on each peripheral connector when the address bus is holding an address between S0H and S9H, where a is the slot number plus 58. This line will drive 10 LSTTL loads*.</td>
</tr>
<tr>
<td>42-49</td>
<td>D6-D7</td>
<td>Buffered bidirectional data bus. The data on this line becomes valid 300ns into a read or a write cycle, and should be stable no less than 100ns before the end of the read of write cycle. Each data line can drive one LSTTL load.</td>
</tr>
<tr>
<td>50</td>
<td>$+12V$</td>
<td>$+12V$ volt power supply. This can supply up to 250mA total for all peripheral units.</td>
</tr>
</tbody>
</table>
Table 28: Auxiliary Video Output Connector Signal Descriptions

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>GROUND</td>
<td>System common ground; 0 volts.</td>
</tr>
<tr>
<td>2</td>
<td>VIDEO</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>+12v</td>
<td>+12 volt power supply.</td>
</tr>
<tr>
<td>4</td>
<td>-5v</td>
<td>-5 volt line from power supply.</td>
</tr>
</tbody>
</table>

![Connector Diagram]

The USER 1 Jumper

Figure 15. Auxiliary Video Output Connector and Pins.

![Pinout Diagram]

Figure 16. Game I/O Connector Pinouts

Table 29: Game I/O Connector Signal Descriptions

<table>
<thead>
<tr>
<th>Pin</th>
<th>Name,</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>+5v</td>
<td>+5 volt power supply. Total current drain on this pin must be less than 100mA.</td>
</tr>
<tr>
<td>2-4</td>
<td>P00-P82</td>
<td>Single-bit (Pushbutton) inputs. These are standard 74LS series TTL inputs.</td>
</tr>
<tr>
<td>5</td>
<td>C048 STROBE</td>
<td>A general-purpose strobe. This line, normally high, goes low during # of a read or write cycle to any address from 5C46 through 5C4F. This is a standard 74LS TTL output.</td>
</tr>
<tr>
<td>6,7,10,11</td>
<td>GC0-GC3</td>
<td>Game controller inputs. These should each be connected through a 150K Ohm variable resistor to +5v.</td>
</tr>
<tr>
<td>8</td>
<td>Gnd</td>
<td>System electrical ground.</td>
</tr>
<tr>
<td>12-15</td>
<td>AN0-AN3</td>
<td>Annunciator outputs. These are standard 74LS series TTL outputs and must be buffered if used to drive other than TTL.</td>
</tr>
</tbody>
</table>
R.H. ELECTRONICS INC.
COMPUTER PRODUCTS

SUPER RAM II™
FOR YOUR APPLE II COMPUTER*

- Plug in slot 0
- 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
  PLAT-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 lets 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>
<thead>
<tr>
<th>Ann.</th>
<th>State</th>
<th>Address</th>
<th>Decimal</th>
<th>Hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>off</td>
<td>49240</td>
<td>-16296</td>
<td>SC05B</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>49241</td>
<td>-16295</td>
<td>SC059</td>
</tr>
<tr>
<td>1</td>
<td>off</td>
<td>49242</td>
<td>-16294</td>
<td>SC05A</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>49243</td>
<td>-16293</td>
<td>SC05B</td>
</tr>
<tr>
<td>2</td>
<td>off</td>
<td>49244</td>
<td>-16292</td>
<td>SC05C</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>49245</td>
<td>-16291</td>
<td>SC05D</td>
</tr>
<tr>
<td>3</td>
<td>off</td>
<td>49246</td>
<td>-16290</td>
<td>SC05E</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>49247</td>
<td>-16289</td>
<td>SC05F</td>
</tr>
</tbody>
</table>
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:
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 time 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

<table>
<thead>
<tr>
<th>MEMORY RANGE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1FF</td>
<td>Program work space; not available to user.</td>
</tr>
<tr>
<td>200.2FF</td>
<td>Keyboard character buffer.</td>
</tr>
<tr>
<td>300.3FF</td>
<td>Available to user for short machine language programs.</td>
</tr>
<tr>
<td>400.7FF</td>
<td>Screen display area for page 1 text or color graphics.</td>
</tr>
<tr>
<td>800.2OFF</td>
<td>In cassette tape version, the APPLESOFT BASIC interpreter.</td>
</tr>
<tr>
<td>800.XXX</td>
<td>If firmware APPLESOFT (Part number A289009X) 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.</td>
</tr>
<tr>
<td>2000.4FF</td>
<td>Firmware APPLESOFT only; high-resolution graphics display page 1.</td>
</tr>
<tr>
<td>3000.XXX</td>
<td>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.</td>
</tr>
<tr>
<td>4000.5FFF</td>
<td>High-resolution graphics display page 2.</td>
</tr>
<tr>
<td>5000.CFFF</td>
<td>Hardware I/O Addresses.</td>
</tr>
<tr>
<td>6000.DFFF</td>
<td>Future ROM expansion.</td>
</tr>
<tr>
<td>7000.F7FF</td>
<td>APPLESOFT II firmware version, with select switch &quot;ON&quot; (up).</td>
</tr>
<tr>
<td>E000.F7FF</td>
<td>APPLE Integer BASIC.</td>
</tr>
<tr>
<td>FFFF.FFFF</td>
<td>APPLE System Monitor.</td>
</tr>
</tbody>
</table>

Figure 2.5 AppleSoft Versions
### 6502 Microprocessor Instructions

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Add Memory to Accumulator with Carry</td>
</tr>
<tr>
<td>AND</td>
<td>&quot;AND&quot; Memory with Accumulator</td>
</tr>
<tr>
<td>ASL</td>
<td>Shift Left One Bit Memory or Accumulator</td>
</tr>
<tr>
<td>BCC</td>
<td>Branch on Carry Cleared</td>
</tr>
<tr>
<td>BCS</td>
<td>Branch on Carry Set</td>
</tr>
<tr>
<td>BEO</td>
<td>Branch on Result Zero</td>
</tr>
<tr>
<td>BIT</td>
<td>Test Bit in Memory with Accumulator</td>
</tr>
<tr>
<td>BMI</td>
<td>Branch on Result Negative</td>
</tr>
<tr>
<td>BNE</td>
<td>Branch on Result Not Zero</td>
</tr>
<tr>
<td>BPL</td>
<td>Branch on Result Plus</td>
</tr>
<tr>
<td>BRK</td>
<td>Force Break</td>
</tr>
<tr>
<td>BVC</td>
<td>Branch on Overflow Cleared</td>
</tr>
<tr>
<td>BVS</td>
<td>Branch on Overflow Set</td>
</tr>
<tr>
<td>CLC</td>
<td>Clear Carry Flag</td>
</tr>
<tr>
<td>CLD</td>
<td>Clear Decimal Mode</td>
</tr>
<tr>
<td>CLI</td>
<td>Clear Interrupt Disable Bit</td>
</tr>
<tr>
<td>CLV</td>
<td>Clear Overflow Flag</td>
</tr>
<tr>
<td>CMP</td>
<td>Compare Memory and Accumulator</td>
</tr>
<tr>
<td>CPX</td>
<td>Compare Memory and Index X</td>
</tr>
<tr>
<td>CPY</td>
<td>Compare Memory and Index Y</td>
</tr>
<tr>
<td>DECB</td>
<td>Decrement Memory by One</td>
</tr>
<tr>
<td>DECB</td>
<td>Decrement Index X by One</td>
</tr>
<tr>
<td>DECB</td>
<td>Decrement Index Y by One</td>
</tr>
<tr>
<td>DECB</td>
<td>&quot;Exclusive-Or&quot; Memory with Accumulator</td>
</tr>
<tr>
<td>INCB</td>
<td>Increment Memory by One</td>
</tr>
<tr>
<td>INCB</td>
<td>Increment Index X by One</td>
</tr>
<tr>
<td>INCB</td>
<td>Increment Index Y by One</td>
</tr>
<tr>
<td>JMP</td>
<td>Jump to New Location</td>
</tr>
<tr>
<td>JSR</td>
<td>Jump to New Location Saving Return Address</td>
</tr>
<tr>
<td>LDA</td>
<td>Load Accumulator with Memory</td>
</tr>
<tr>
<td>LDX</td>
<td>Load Index X with Memory</td>
</tr>
<tr>
<td>LDY</td>
<td>Load Index Y with Memory</td>
</tr>
<tr>
<td>LSR</td>
<td>Shift Right One Bit Memory or Accumulator</td>
</tr>
<tr>
<td>NOP</td>
<td>No Operation</td>
</tr>
<tr>
<td>ORA</td>
<td>&quot;OR&quot; Memory with Accumulator</td>
</tr>
<tr>
<td>PHA</td>
<td>Push Accumulator on Stack</td>
</tr>
<tr>
<td>PHP</td>
<td>Push Processor Status on Stack</td>
</tr>
<tr>
<td>PLA</td>
<td>Pull Accumulator from Stack</td>
</tr>
<tr>
<td>PLP</td>
<td>Pull Processor Status from Stack</td>
</tr>
<tr>
<td>ROL</td>
<td>Rotate One Bit Left (Memory or Accumulator)</td>
</tr>
<tr>
<td>ROR</td>
<td>Rotate One Bit Right (Memory or Accumulator)</td>
</tr>
<tr>
<td>RTS</td>
<td>Return from Interrupt</td>
</tr>
<tr>
<td>RTS</td>
<td>Return from Subroutine</td>
</tr>
<tr>
<td>SBC</td>
<td>Subtract Memory from Accumulator with Borrow</td>
</tr>
<tr>
<td>SEC</td>
<td>Set Carry Flag</td>
</tr>
<tr>
<td>SED</td>
<td>Set Decimal Mode</td>
</tr>
<tr>
<td>SEI</td>
<td>Set Interrupt Disable Status</td>
</tr>
<tr>
<td>STA</td>
<td>Store Accumulator in Memory</td>
</tr>
<tr>
<td>STX</td>
<td>Store Index X in Memory</td>
</tr>
<tr>
<td>STY</td>
<td>Store Index Y in Memory</td>
</tr>
<tr>
<td>TAX</td>
<td>Transfer Accumulator to Index X</td>
</tr>
<tr>
<td>TAY</td>
<td>Transfer Accumulator to Index Y</td>
</tr>
<tr>
<td>TXS</td>
<td>Transfer Stack Pointer to Index X</td>
</tr>
<tr>
<td>TXA</td>
<td>Transfer Index X to Accumulator</td>
</tr>
<tr>
<td>TXS</td>
<td>Transfer Index X to Stack Pointer</td>
</tr>
<tr>
<td>TYA</td>
<td>Transfer Index Y to Accumulator</td>
</tr>
</tbody>
</table>
### HEX OPERATION CODES

<table>
<thead>
<tr>
<th>Code</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>BRK</td>
</tr>
<tr>
<td>01</td>
<td>ORA</td>
</tr>
<tr>
<td>02</td>
<td>NOP</td>
</tr>
<tr>
<td>03</td>
<td>NOP</td>
</tr>
<tr>
<td>04</td>
<td>NOP</td>
</tr>
<tr>
<td>05</td>
<td>ORA</td>
</tr>
<tr>
<td>06</td>
<td>NOP</td>
</tr>
<tr>
<td>07</td>
<td>NOP</td>
</tr>
<tr>
<td>08</td>
<td>NOP</td>
</tr>
<tr>
<td>09</td>
<td>NOP</td>
</tr>
<tr>
<td>0A</td>
<td>NOP</td>
</tr>
<tr>
<td>0B</td>
<td>NOP</td>
</tr>
<tr>
<td>0C</td>
<td>NOP</td>
</tr>
<tr>
<td>0D</td>
<td>NOP</td>
</tr>
<tr>
<td>0E</td>
<td>NOP</td>
</tr>
<tr>
<td>0F</td>
<td>NOP</td>
</tr>
<tr>
<td>10</td>
<td>REP</td>
</tr>
<tr>
<td>11</td>
<td>NOP</td>
</tr>
<tr>
<td>12</td>
<td>NOP</td>
</tr>
<tr>
<td>13</td>
<td>NOP</td>
</tr>
<tr>
<td>14</td>
<td>NOP</td>
</tr>
<tr>
<td>15</td>
<td>NOP</td>
</tr>
<tr>
<td>16</td>
<td>NOP</td>
</tr>
<tr>
<td>17</td>
<td>NOP</td>
</tr>
<tr>
<td>18</td>
<td>NOP</td>
</tr>
<tr>
<td>19</td>
<td>NOP</td>
</tr>
<tr>
<td>1A</td>
<td>NOP</td>
</tr>
<tr>
<td>1B</td>
<td>NOP</td>
</tr>
<tr>
<td>1C</td>
<td>NOP</td>
</tr>
<tr>
<td>1D</td>
<td>NOP</td>
</tr>
<tr>
<td>1E</td>
<td>NOP</td>
</tr>
<tr>
<td>1F</td>
<td>NOP</td>
</tr>
<tr>
<td>20</td>
<td>JSP</td>
</tr>
<tr>
<td>21</td>
<td>NOP</td>
</tr>
<tr>
<td>22</td>
<td>NOP</td>
</tr>
<tr>
<td>23</td>
<td>NOP</td>
</tr>
<tr>
<td>24</td>
<td>NOP</td>
</tr>
<tr>
<td>25</td>
<td>NOP</td>
</tr>
<tr>
<td>26</td>
<td>NOP</td>
</tr>
<tr>
<td>27</td>
<td>NOP</td>
</tr>
<tr>
<td>28</td>
<td>NOP</td>
</tr>
<tr>
<td>29</td>
<td>NOP</td>
</tr>
<tr>
<td>2A</td>
<td>NOP</td>
</tr>
<tr>
<td>2B</td>
<td>NOP</td>
</tr>
<tr>
<td>2C</td>
<td>NOP</td>
</tr>
<tr>
<td>2D</td>
<td>NOP</td>
</tr>
<tr>
<td>2E</td>
<td>NOP</td>
</tr>
</tbody>
</table>

Note: The operations are described in the context of a microcontroller's instruction set, where `NOP` indicates no operation, `ORA` indicates OR with Accumulator, `NOP` indicates no operation, etc.
Appendix M: Differences Between APPLESOFT and Integer BASIC

DIFFERENCES BETWEEN COMMANDS

These commands are available in APPLESOFT, but not in Integer BASIC:
ATH
CHR$  COS
DATA   DEF FN  DRAW
EXP
FLASH  FN     FRE
GET
COLOR  HGR     HGM2  HIMEM:  HOME  HPILOT
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
TAN
USR
VAL
WAIT
XDRAW

These commands are available in Integer BASIC, but not in APPLESOFT:
AUTO
DSF
RAN
MOD

These are named differently in the languages:

<table>
<thead>
<tr>
<th>Integer BASIC</th>
<th>APPLESOFT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLR</td>
<td>CLEAR</td>
</tr>
<tr>
<td>CON</td>
<td>CONT</td>
</tr>
<tr>
<td>TAB</td>
<td>HTAB (Note: APPLESOFT also has a TAB)</td>
</tr>
<tr>
<td>GOTO X*19+199</td>
<td>ON X GOTO 199, 119, 129</td>
</tr>
<tr>
<td>GOSUB X*199+199</td>
<td>ON X GOSUB 199, 119, 129</td>
</tr>
<tr>
<td>CALL -936</td>
<td>HOME (or CALL -936)</td>
</tr>
<tr>
<td>POKE 59,127</td>
<td>INVERSE</td>
</tr>
<tr>
<td>POKE 59,255</td>
<td>NORMAL</td>
</tr>
<tr>
<td>X</td>
<td>XZ (Z indicates integer variable)</td>
</tr>
<tr>
<td>#</td>
<td>&lt; or &gt;&lt;</td>
</tr>
</tbody>
</table>
APPENDIX C

The Apple Video Display, Screen Formats,

Other Input/Output Features
THE APPLE VIDEO DISPLAY

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

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 GRAPHICS | 1920 colored squares = 40 x 48 vertically horizontally  
no. of colors = 16  
There is no space between blocks. |
| HRG | 192 x 280 colored dots  
vertically horizontally  
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 2 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

<table>
<thead>
<tr>
<th>Screen</th>
<th>Page</th>
<th>Begins at:</th>
<th>Ends at:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Hex</td>
<td>Decimal</td>
</tr>
<tr>
<td>Text/Lo-Res</td>
<td>Primary</td>
<td>$400</td>
<td>1024</td>
</tr>
<tr>
<td></td>
<td>Secondary</td>
<td>$800</td>
<td>2048</td>
</tr>
<tr>
<td>Hi-Res</td>
<td>Primary</td>
<td>$2000</td>
<td>8192</td>
</tr>
<tr>
<td></td>
<td>Secondary</td>
<td>$4000</td>
<td>16384</td>
</tr>
</tbody>
</table>

### Table 5: Screen Soft Switches

<table>
<thead>
<tr>
<th>Location:</th>
<th>Hex</th>
<th>Decimal</th>
<th>Description:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC050</td>
<td>49232</td>
<td>16304</td>
<td>Display a GRAPHICS mode.</td>
</tr>
<tr>
<td>SC051</td>
<td>49233</td>
<td>16305</td>
<td>Display TEXT mode.</td>
</tr>
<tr>
<td>SC052</td>
<td>49234</td>
<td>16302</td>
<td>Display all TEXT or GRAPHICS.</td>
</tr>
<tr>
<td>SC053</td>
<td>49235</td>
<td>16301</td>
<td>Mix TEXT and a GRAPHICS mode.*</td>
</tr>
<tr>
<td>SC054</td>
<td>49236</td>
<td>16300</td>
<td>Display the Primary page (Page 1).</td>
</tr>
<tr>
<td>SC055</td>
<td>49237</td>
<td>16299</td>
<td>Display the Secondary page (Page 2).</td>
</tr>
<tr>
<td>SC056</td>
<td>49238</td>
<td>16298</td>
<td>Display LO-RES GRAPHICS mode.*</td>
</tr>
<tr>
<td>SC057</td>
<td>49239</td>
<td>16297</td>
<td>Display HI-RES GRAPHICS mode.*</td>
</tr>
</tbody>
</table>

### Table 6: Screen Mode Combinations

<table>
<thead>
<tr>
<th>Screen</th>
<th>Primary Page</th>
<th>Secondary Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Switches</td>
<td>Screen</td>
</tr>
<tr>
<td>All Text</td>
<td>SC054</td>
<td>SC051</td>
</tr>
<tr>
<td>All Lo-Res</td>
<td>SC054</td>
<td>SC056</td>
</tr>
<tr>
<td>Graphics</td>
<td>SC052</td>
<td>SC055</td>
</tr>
<tr>
<td>All Hi-Res</td>
<td>SC054</td>
<td>SC057</td>
</tr>
<tr>
<td>Graphics</td>
<td>SC052</td>
<td>SC050</td>
</tr>
<tr>
<td>Mixed Text</td>
<td>SC054</td>
<td>SC056</td>
</tr>
<tr>
<td>and Lo-Res</td>
<td>SC053</td>
<td>SC050</td>
</tr>
<tr>
<td>Mixed Text</td>
<td>SC054</td>
<td>SC057</td>
</tr>
<tr>
<td>and Hi-Res</td>
<td>SC053</td>
<td>SC050</td>
</tr>
</tbody>
</table>

Figure 2.11
Figure 2.12 Deduction of the Combinations for the Screen Format Soft Switches
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 $CO40 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 2.10: Input/Output Special Locations

<table>
<thead>
<tr>
<th>Function</th>
<th>Address:</th>
<th>Hex</th>
<th>Read/Write</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Decimal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speaker</td>
<td>49280</td>
<td>SC030</td>
<td>R</td>
</tr>
<tr>
<td>Cassette Out</td>
<td>49184</td>
<td>SC020</td>
<td>R</td>
</tr>
<tr>
<td>Cassette In</td>
<td>49256</td>
<td>SC060</td>
<td>R</td>
</tr>
<tr>
<td>Annunciators*</td>
<td>49240 through</td>
<td>SC058</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td>49247 through</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49249 through</td>
<td>SC05F</td>
<td></td>
</tr>
<tr>
<td>Flag inputs</td>
<td>49249</td>
<td>SC061</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>49250</td>
<td>SC062</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>49251</td>
<td>SC063</td>
<td>R</td>
</tr>
<tr>
<td>Analog Inputs</td>
<td>49252</td>
<td>SC064</td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>49253</td>
<td>SC065</td>
<td></td>
</tr>
<tr>
<td></td>
<td>49254</td>
<td>SC066</td>
<td></td>
</tr>
<tr>
<td></td>
<td>49255</td>
<td>SC067</td>
<td></td>
</tr>
<tr>
<td>Analog Clear</td>
<td>49264</td>
<td>SC070</td>
<td>R/W</td>
</tr>
<tr>
<td>Utility Strobe</td>
<td>49216</td>
<td>SC040</td>
<td>R</td>
</tr>
</tbody>
</table>

**Figure 2.13 Input/Output Special Locations**

---

<table>
<thead>
<tr>
<th>With Respect to</th>
<th>Varieties of Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>System Monitor</td>
<td>Old Monitor</td>
</tr>
<tr>
<td>Main Board</td>
<td>Revision 0 main board</td>
</tr>
</tbody>
</table>

**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
B Trace both channel A and channel B input signals
C 'space' Cycles the channel A voltage scale each time the 'space' bar is pressed
D 'space' Sets the channel A zero voltage to the center of the display
E 'space' Decreases the channel A zero voltage each time the 'E' key is pressed
F 'space' Increases the channel A zero voltage each time the 'F' key is pressed
G 'space' Trace channel B signal only
H 'space' Cycles the channel B voltage scale each time the 'space' bar is pressed
I 'space' Sets the channel B zero voltage to the center of the display
J 'space' Decreases the channel B zero voltage each time the 'J' key is pressed
K 'space' Increases the channel B zero voltage each time the 'K' key is pressed
L CONTINUOUS - Continuously acquire data until stopped (sweep rates slower than 100 sec/div)
M D DATA mode (sweep rates slower than 1 sec/div, only)
N Exits from the DATA mode
O SINGLE SWEET - Acquire a single sweep when triggered
P Exits from the DDATA mode
Q Enters the TIME adjustment mode
R TIME adjustment mode

TIME ADJUSTMENT MODE

'A. left arrow' Decreases the sweep rate one step
'B. right arrow' Increases the sweep rate one step
'C. left arrow' Decreases the sweep rate by 10 second steps (Seconds time scale only)
'D. right arrow' Increases the sweep rate by 10 second steps (Seconds time scale only)

DISK SUPERVISOR

E 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 DISK SUPERVISOR mode
F Returns directly to the control language.

* Entering an invalid input at any time will return the APPLESANE to the command mode. Pressing "reset" will not let us do the disk supervisor mode will remove all current files forcing the ESCAPE button to be repeated.

WARNING - NEVER PRESS "RESET" UNLESS IN DISK SUPERVISOR.
<table>
<thead>
<tr>
<th>Single channel (trace)</th>
<th>Dual channel (trace)</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>ØC ØØ</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>ØC FF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ØD Ø Ø</td>
</tr>
<tr>
<td>A or B</td>
<td>256</td>
<td>ØD FF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ØE Ø Ø</td>
</tr>
<tr>
<td></td>
<td>256</td>
<td>ØE FF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ØF Ø Ø</td>
</tr>
</tbody>
</table>

**Figure 2.20** Memory After One Data Acquisition Cycle
<table>
<thead>
<tr>
<th>SCALE (VOLTS/DIV)</th>
<th>VERTICAL EXPANSION</th>
<th>RESOLUTION (mV/Div)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.025</td>
<td>x 8</td>
<td>7.1</td>
</tr>
<tr>
<td>0.050</td>
<td>x 4</td>
<td>7.1</td>
</tr>
<tr>
<td>0.100</td>
<td>x 2</td>
<td>7.1</td>
</tr>
<tr>
<td>0.200</td>
<td>x 1</td>
<td>7.1</td>
</tr>
<tr>
<td>0.400</td>
<td>÷ 2</td>
<td>7.1</td>
</tr>
<tr>
<td>0.800</td>
<td>÷ 4</td>
<td>7.1</td>
</tr>
<tr>
<td>0.25</td>
<td>x 8</td>
<td>7.1</td>
</tr>
<tr>
<td>0.50</td>
<td>x 4</td>
<td>7.1</td>
</tr>
<tr>
<td>1.00</td>
<td>x 2</td>
<td>7.1</td>
</tr>
<tr>
<td>2.00</td>
<td>x 1</td>
<td>7.1</td>
</tr>
<tr>
<td>4.00</td>
<td>÷ 2</td>
<td>7.1</td>
</tr>
<tr>
<td>8.00</td>
<td>÷ 4</td>
<td>7.1</td>
</tr>
</tbody>
</table>

*Figure 2.21 Voltage Scales*
5.1 DISPLAY BUFFER MEMORY – Sweep rates faster than 1 msec./div.

<table>
<thead>
<tr>
<th>SINGLE CHANNEL</th>
<th>DUAL CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of buffer memory</td>
<td>$1000</td>
</tr>
<tr>
<td>Trigger START</td>
<td>$1000</td>
</tr>
<tr>
<td>External trigger</td>
<td>$1000</td>
</tr>
<tr>
<td></td>
<td>Begin CH A data</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger START</td>
</tr>
<tr>
<td></td>
<td>External trigger</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger MIDDLE</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger END</td>
</tr>
<tr>
<td></td>
<td>End CH A data</td>
</tr>
<tr>
<td></td>
<td>$1200</td>
</tr>
<tr>
<td>Trigger MIDDLE</td>
<td>$1200</td>
</tr>
<tr>
<td></td>
<td>Begin CH B data</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger START</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger MIDDLE</td>
</tr>
<tr>
<td>Trigger END</td>
<td>$1300</td>
</tr>
<tr>
<td>End of buffer memory</td>
<td>$1300</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger END</td>
</tr>
<tr>
<td></td>
<td>End CH B data</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>SINGLE CHANNEL</th>
<th>DUAL CHANNEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start of buffer memory</td>
<td>$1000</td>
</tr>
<tr>
<td>External trigger</td>
<td>$1000</td>
</tr>
<tr>
<td>Trigger START</td>
<td>$1010</td>
</tr>
<tr>
<td></td>
<td>Offset from $1000</td>
</tr>
<tr>
<td></td>
<td>Offset from $1010</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger MIDDLE</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger END</td>
</tr>
<tr>
<td></td>
<td>End CH A data</td>
</tr>
<tr>
<td></td>
<td>$110F0</td>
</tr>
<tr>
<td>Trigger MIDDLE</td>
<td>$110F0</td>
</tr>
<tr>
<td></td>
<td>CH A Trigger END</td>
</tr>
<tr>
<td></td>
<td>End CH A data</td>
</tr>
<tr>
<td></td>
<td>$1200</td>
</tr>
<tr>
<td></td>
<td>Begin CH B data</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger START</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger MIDDLE</td>
</tr>
<tr>
<td>Trigger END</td>
<td>$1300</td>
</tr>
<tr>
<td>End of buffer memory</td>
<td>$130F0</td>
</tr>
<tr>
<td></td>
<td>CH B Trigger END</td>
</tr>
<tr>
<td></td>
<td>End CH B data</td>
</tr>
</tbody>
</table>

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 provides 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 8 categories:

Housekeeping commands:
- INIT
- LOAD
- DELETE
- VERIFY
- MAXFILES
- CATALOG
- RUN
- LOCK
- MOM
- SAVE
- BBACK
- UNLOCK
- MOM

Access Commands:
- FP
- INT
- IN
- CHAIN

Sequential Text File Commands:
- OPEN
- READ
- APPEND
- EXEC
- CLOSE
- WRITE
- POSITION

Random-Access Text File Commands:
- OPEN
- READ
- WRITE

Machine-Language File Commands:
- RDLD
- BMOV
- 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 32 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 examples:
- 10 DB = "SET "CHILD"
- 20 PRINT DB; "CATALOG"

Applesoft BASIC example:
- 10 DB = CHR$(4) + VRM CTRL-D 20 PRINT DB; "CATALOG"

The term "BASIC" alias is used to mean either Integer BASIC or Applesoft BASIC. The term "file" alias 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
--- | --- | --- | ---
Name | A | 1 | 255
Drive | .D | 0 | 1
Volume | .V | 0 | 254

* Using VD is like omitting the V 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 | .B | 1 | 4095
Relative Field | .R | 0 | 832767

* With EXEC, always relative to field 0.

RANDOM-ACCESS TEXT FILES

Parameter | As shown | Min | Max
--- | --- | --- | ---
Record Length | .L | 1 | 255
Record Number | .R | 0 | 255

BINARY FILES

Parameter | As shown | Min | Max
--- | --- | --- | ---
Starting Address | .A | 0 | 25335
Number of Bytes | .B | 0 | 4095

DOS COMMANDS

Command | Quantity | As shown | Min | Max
--- | --- | --- | --- | ---
FD | slot | F4 | 0 | 3
LD | slot | F5 | 0 | 3
MD | slot | F6 | 0 | 3
MAXFILES | e | 0 | 16

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

If a command unlike the Volume parameter or value VD, the diskette's volume number is ignored. A command that uses the Volume parameter VD will not be executed unless the diskette's volume number is VD.

HOUSEKEEPING COMMANDS

INIT X [.V|L [.S|D [.D]

Initializes a blank diskette to form a slave diskette. Assigns gloating program name X and volume number VD (if specified). SAVES the BASIC program currently in memory, under file name X.

CATALOG [.S|D [.D]

Displays volume number and all files on a diskette, each file’s type and sector length. * indicates a LOCKED file.

Type | Description | Example
--- | --- | ---
A | Integer BASIC program file | SAVE A [.S|D [.D]
B | Apple with BASIC program file | SAVE B [.S|D [.D]
C | Text file | SAVE C [.T|D [.D]
D | Binary memory-image file | SAVE D [.S|D [.D]

SAVE X [.S|D [.D] [.V]

Saves current BASIC program onto diskette, under file name X. Overwrites any previous file of same type and name, without warning.

LOAD X [.S|D [.D] [.V]

Loads BASIC program file X into memory, after clearing memory and (if necessary) changing to the correct BASIC.

RUN X [.S|D [.D] [.V]

Runs BASIC program file X, then BEND the program.

RENAME X, Y [.S|D [.D] [.V]

Changes a diskette file's name from X to Y.

DELETE X [.S|D [.D] [.V]

Erases file X from the diskette.

LOCK X [.S|D [.D] [.V]

Locks file X against accidental changes or deletion. LOCKED file shown in CATALOG by *. *

UNLOCK X [.S|D [.D] [.V]

Unlocks previously LOCKED file X to allow change or deletion.
SEQUENTIAL TEXT FILE COMMANDS

OPEN X [.,a] [.,d] [.,v]  
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 [.,b]  
Subsequent PRINTs send characters to
sequential text file X. WRITing begins at current file position or (if specified)
at byte b. Cancelled by any DOS command.

READ X [.,b]  
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
in one field (all characters to next
RETURN). Cancelled by any DOS command.

APPEND X [.,a] [.,d] [.,v]  
Opens existing sequential text file X,
similar to OPEN, but prepares to WRITE
at the end of the file.

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

EXEC X [.,p] [.,a] [.,d] [.,v]  
Executes sequence of BASIC statements
in text file X as if typed at keyboard.
With b-th parameter, execution begins with
p-th field. Fields may include numbered
BASIC program lines and direct-execution
BASIC or DOS commands to control the Appl.

RANDOM-ACCESS TEXT FILE COMMANDS

OPEN X, t, j [.,a] [.,d] [.,v]  
Opens or creates random-access text file X
allocates one file buffer, and defines read-
length as j bytes. Prepares to WRITE or
READ from beginning of file. Same length par-
meter must be used each time file X is OPEN

CLOSE [x] [.,a] [.,d] [.,v]  
Completes WRITE X, if necessary, and de-
allocates file buffer assigned to text file
X. Without file name, CLOSES all OPEN files.

WRITE X [.,a] [.,b]  
Subsequent PRINTs send characters to ran-
dom-access text file X. With no parameter,
WRITing begins at current file position.
With b-th parameter, WRITing starts at
byte b of record X. With a-th parameter,
WRITE starts at byte a of current or spe-
cified record. Cancelled by any DOS command.

READ X [.,a] [.,b]  
Subsequent INPUTs and GETs take response
characters from random-access text file X,
with no parameter, READing starts at cur-
rent file position. With a-th parameter,
READing starts at byte a or specified record.
READing starts at byte b of current or specified record. INPUT
response in one field (all characters to next
RETURN). Cancelled by any DOS command.

MACHINE-LANGUAGE FILE COMMANDS

SAVE X, A, j, L [.,a] [.,d] [.,v]  
Stores on disks, under file name X,
the contents of j memory bytes starting
at address A.

LOAD X [.,a] [.,b] [.,d] [.,v]  
Loads binary file X into memory locations
from which file was SAVEd or
(if specified) starting at address A.

B Sản phẩm 157
APPENDIX F

BASIC Programs Listing
Variant 1 - Data, Spectrums, Processing

ICATALOG

DISK VOLUME 25:

* A 004 AUTO DUAL
* A 002 CALIBRATE
* A 003 ALTERNATIVE
* A 003 AUTO DUAL 1
* A 003 SPC
* A 004 RET SPC 1
* A 003 CHOICE
* A 005 AVERAGE
* A 004 AVERAGE 1
* A 002 CHOICE 2
* A 002 CHOICE 2 B
* A 013 AXIS
* A 013 AXISS
* A 017 PLOT A
* A 016 PLOT B
* A 002 MICRO BUF
* A 016 PLOT A 1
* A 012 PLOT B 1
* A 019 PIE CHART
* A 019 PIE CHART B
* B 031 BPR
* B 022 M4 A
* B 018 M4 B
* B 025 MAC
* B 002 CT
* B 002 CS
* B 002 LOW ALL TAB
* B 002 MALL . O BJO
* B 002 RET TAB
* B 002 O PEN . O BJO
* B 002 SAVE SPC . O BJO
* B 002 SWEEP . O BJO
* B 002 SVTAB
* B 002 RET TAB 1
* B 002 CLEAR . O BJO
* B 002 AVE LOT AB
* B 002 AVE H1 . O BJO
* B 004 AVE . 3
* B 002 CLEAR B . O BJO
* B 005 AVE LOT AB 2
* B 002 AVE H1 B . O BJO
* B 004 AVE 3 B
* B 004 MAG I SPACE 4
LOAD RETPC

LIST

5 D$ = CHR$ (4) : HOME
10 PRINT "CH$ (4)"
15 PRINT "YOU CAN DO 8 AVERAGES OF 8 SPECTRUMS FOR EACH CHANNEL."
20 PRINT "AVERAGE INVERSE, CH$ (4)."
25 PRINT "FIRST, PRINT "WHAT IS THE ORDER OF THE AVERAGE\n30 PRINT "THAT YOU WANT TO DO?":"
35 INPUT T
40 IF T < > 1 GOTO 55
45 IF T < > 2 GOTO 75
50 GOTO 125
55 IF T < > 3 GOTO 65
60 GOTO 140
65 IF T < > 4 GOTO 75
70 GOTO 145
75 IF T < > 5 GOTO 95
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 145
115 HOME : VTAB B: INVERSE, PRINT "THE NO. OF AVERAGES IS TOO BIG FOR THE POS\n120 WILITIES OF T"
125 DISK": NORMAL
130 FOR D = 1 TO 2000: NEXT D
135 GOTO 15
140 POKE 823.14: GOTO 170
145 POKE 823.18: GOTO 170
150 POKE 823.20: GOTO 170
155 POKE 823.24: GOTO 170
160 POKE 823.26: GOTO 170
165 POKE 823.30: GOTO 170
170 CALL 7448
175 PRINT D$"RUN CHOICE.D1"

LOAD CHOICE

LIST

5 D$ = CHR$ (4) : HOME
10 PRINT "CH$ (4)"
15 PRINT "YOU CAN DO 8 AVERAGES OF 8 SPECTRUMS FOR EACH CHANNEL."
20 PRINT "AVERAGE INVERSE, CH$ (4)."
25 PRINT "FIRST, PRINT "WHAT IS THE ORDER OF THE AVERAGE\n30 PRINT "THAT YOU WANT TO DO?":"
35 INPUT T
40 IF T < > 1 GOTO 55
45 IF T < > 2 GOTO 75
50 GOTO 125
55 IF T < > 3 GOTO 65
60 GOTO 140
65 IF T < > 4 GOTO 75
70 GOTO 145
75 IF T < > 5 GOTO 95
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 145
115 HOME : VTAB B: INVERSE, PRINT "THE NO. OF AVERAGES IS TOO BIG FOR THE POS\n120 WILITIES OF T"
125 DISK": NORMAL
130 FOR D = 1 TO 2000: NEXT D
135 GOTO 15
140 POKE 823.14: GOTO 170
145 POKE 823.18: GOTO 170
150 POKE 823.20: GOTO 170
155 POKE 823.24: GOTO 170
160 POKE 823.26: GOTO 170
165 POKE 823.30: GOTO 170
170 CALL 7448
175 PRINT D$"RUN CHOICE.D1"
JLOAD AVERAGES

JLIST

3 DS = CHR$(4)
10 PRINT DS"$LOAD CLEAR OBJ0, DI1": CALL 748: PRINT DS"$LOAD AVELTAB, DI1"
15 N = Peek (-16225): N = Peek (-16225): PRINT DS"$LOAD AVELTAB, DI1"
20 PRINT DS"$LOAD AVE1, DI1": CALL 748: TEXT
25 PRINT DS"$SAVE AVE1, A0$200, L$200, DI1"
30 PRINT DS"$LOAD MAX, DI1": PRINT DS"$LOAD F45, DI1": PRINT DS"$LOAD MAX, DI1"
35 PRINT DS"$LOAD BPR, DI1": PRINT DS"$LOAD CT, DI1": PRINT DS"$LOAD CT, DI1"
40 PRINT DS"$LOAD AVE2, A0$400, DI1": POKE 37372, 1: HOME
45 PRINT "PRESS A TO NOT SEE AVERAGE": PRINT "PRESS OTHER KEY TO SEE AVERAGE"
50 SET A$: IF A$ = "A" GOTO 55
55 Y = Peek (16897): PRINT
60 PRINT "AVERAGE J MEANS THAT THE BK BLOCK NO. J OF SPECTRUMS FROM DISK IS FILLED OMISSION"
65 PRINT X = Peek (16897): PRINT
70 PRINT "TITLE FOR AVERAGE y", CH, A": INPUT "$": PRINT DS"$PR1": PRINT CHR$(14) (T) CHR$(20): PRINT DS"$PR1"
75 POKE 15210, 209: POKE 15312, 160: POKE 15314, 160
80 POKE 15315, 207: POKE 15316, 161: POKE 15317, 160
85 POKE 15318, 164: POKE 15319, 164: POKE 15320, 162: POKE 15321, 128
90 POKE 15322, 209: CALL 30464: POKE 15, 0:1 TEXT : HOME
95 PRINT DS = PRINT DS"$RUN CHOICE2, DI1"

JLOAD AVERAGES

JLIST

3 DS = CHR$(4): PRINT DS"$LOAD CLEAR OBJ0, DI1": CALL 748: PRINT DS"$LOAD AVELTAB, DI1"
10 N = Peek (-16225): N = Peek (-16225): PRINT DS"$LOAD AVELTAB, DI1": CALL 748: TEXT
15 PRINT DS"$SAVE AVE1, A0$400, L$200, DI1"
20 PRINT DS"$LOAD AVE1, DI1": PRINT DS"$LOAD MAX, DI1": PRINT DS"$LOAD F45, DI1"
25 PRINT DS"$LOAD BPR, DI1": PRINT DS"$LOAD CT, DI1": PRINT DS"$LOAD CT, DI1"
30 PRINT DS"$LOAD AVE2, A0$400, DI1": POKE 37372, 1: HOME
35 PRINT "PRESS A TO NOT SEE AVERAGE": PRINT "PRESS OTHER KEY TO SEE AVERAGE"
40 SET A$: IF A$ = "A" GOTO 45
45 Y = Peek (16897): PRINT
50 PRINT "AVERAGE J MEANS THAT THE BK BLOCK NO. J OF SPECTRUMS FROM DISK IS FILLED OMISSION"
55 PRINT X = Peek (16897): PRINT
60 PRINT "TITLE FOR AVERAGE y", CH, A": INPUT "$": PRINT DS"$PR1": PRINT CHR$(14) (T) CHR$(20): PRINT DS"$PR1"
65 POKE 15210, 209: POKE 15312, 160: POKE 15314, 160
70 POKE 15315, 207: POKE 15316, 161: POKE 15317, 160
75 POKE 15318, 164: POKE 15319, 164: POKE 15320, 162: POKE 15321, 128
80 POKE 15322, 209: CALL 30464: POKE 15, 0:1 TEXT : HOME
85 PRINT DS = PRINT DS"$RUN CHOICE2, DI1"
162

LOAD CHOICE2

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

LOAD CHOICE2

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

JCATALOG

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 AXIB
*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 CB
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELTAB
*B 002 AVEHI.OBJO
B 004 AVE.B
*B 002 CLEAR.B.OBJO
*B 005 AVELTAB.B
*B 002 AVEHIB.OBJO
B 004 AVEB.B
*B 004 MAGICSFACE#
LOAD RETSPC

LIST

5 HOME
10 FOR D = 0 TO 120
15 HTAB 31: VTAB 10: INVERSE: PRINT "E E G PROCESSING"
25 HTAB 31: VTAB 13: PRINT "V A R I A N T 1"
30 NEXT D: HME
35 D$ = CHR$(4)
40 PRINT D$: "LOAD RETSPC, D$"
45 HOME: PRINT "YOU CAN DO 8 AVERAGES OF 8 SPECTRUMS FOR EACH CHANNEL."
50 VTAB 31: INVERSE: PRINT "ANSWER THE FOLLOWING QUESTION WITH "1" FOR FIRST, "2" FOR SECOND, "3" FOR THIRD,...": NCANAL
55 VTAB 31: INPUT "WHAT IS THE ORDER OF THE AVERAGE THAT YOU WANT TO DO?"$ IT
60 POKE 1997, IT
65 IF T < > 1 GOTO 75
70 GOTO 160
75 IF T < > 2 GOTO 95
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 31: INVERSE: PRINT "THE NO. OF AVERAGES IS TOO BIG FOR THE POSSIBILITIES OF:"$ IT
150 HIS DISK": NCANAL
155 FOR D = 1 TO 2000: NEXT D
160 POKE 822, 16: GOTO 200
165 POKE 822, 18: GOTO 200
170 POKE 823, 20: GOTO 200
175 POKE 823, 22: GOTO 200
180 POKE 824, 24: GOTO 200
185 POKE 825, 26: GOTO 200
190 POKE 825, 20: GOTO 200
195 POKE 821, 10: GOTO 200
200 CALL 768
205 PRINT D$: "RUN CHOICE, D$"
Variant 2 - Data, Spectrums, Processing
JLOAD AUTODUAL

JLIST

5 HOME
10 VTAB 11: VTAB 10: INVERSE: PRINT "E E G DATA ACQUISITION AND PROCESSING"
20 VTAB 11: VTAB 11: PRINT "A M A N - I":
25 VTAB 11: VTAB 11: PRINT "LOADING DATA FILES": NORMAL
30 D$ = CHR$(4)
35 PRINT D$"LOAD BR, D1";
40 PRINT D$"LOAD M4, D1"
45 PRINT D$"LOAD MC, D1"
50 PRINT D$"LOAD CG, D1"
55 PRINT D$"LOAD CS, D1"
60 PRINT D$"LOAD CS, D1"
65 POKE 3775.4: POKE 3772.1
70 POKE 3775.0
75 HOME
80 PRINT "DO CALIBRATION OF THE CROPS:";
85 PRINT "1. ADJUST TIME BASE TO 0.2 SEC/DIV"
90 PRINT "2. ADJUST SCALE TO 4.0 VOLT/DIV"
95 PRINT "3. ADJUST DISPLAY TO 00001 BY PRESSING M KEY: ADJUST DISPLAY TO 00001 BY PRESSING < CHAR:";
100 PRINT "4. PRESS CTRL SHIFT P TO OSS OUT FROM CALIBRATION MODE"
105 PRINT "5. I AM READY"
110 PRINT "PRESS 5 IF YOU ARE READY"
115 VTAB (12): SET A$: IF VAL (A$) > 5 THEN GOTO 115
120 PRINT D$
125 PRINT D$"RUN CALIBRATE, D1"

JLOAD CALIBRATE

JLIST

5 D$ = CHR$(4)
10 POKE 3772.1
15 CALL 3044
20 POKE 3772.0
25 PRINT D$"SAVE CT, A#190, L#70, D1"
30 PRINT D$"SAVE CS, A#180, L#28, D1"
35 POKE 3772.1
40 TEXT
45 PRINT D$"RUN AUTODUAL, D1"

JLOAD AUTODUAL

JLIST

5 D$ = CHR$(4)
10 HOME: PRINT "THE MAXIMUM NUM. OF SWEEPS CAN BE 64"
15 VTAB 0: INPUT "TOTAL NO. OF SWEEPS": T
20 REN TOD, OF BLOCKS OF 1024 BYTES
25 IF T <= 65 THEN GOTO 45
30 HOME: VTAB 0: INVERSE: PRINT "T IS TOO BIG": NORMAL
35 FOR D = 1 TO 2000: NEXT D: REM DELAY
40 GOTO 10
45 FOR Y = 0 TO INT (T / 8) - (T / 8) = INT (T / 8)
50 POKE 842, Y + 2: POKE 844, 0
55 FOR X = 1 TO 8
60 POKE 15112 + 8 * X: POKE 15120 + 8 * X, 195
65 POKE 15114 + 8 * X: POKE 15122 + 8 * X, 111
70 POKE 15116 + 8 * X: POKE 15124 + 8 * X, 111
75 CALL 3044
80 PRINT D$: POKE 3772.1
85 CALL 3048: POKE 3772.1
90 NEXT X: NEXT Y: HOME: POKE 34, 0: POKE 49233, 0
95 PRINT D$"RUN SPEC, D1"
JLOAD SPC

JLIST

3 DE = CHR$ (4)
10 HOME : INPUT "TOTAL NUMBER OF SPECTRUMS" : T
15 VTab 6: PRINT "THE MAXIMUM VALUE OF T CAN BE 123"
20 REM THIS OF BLOCKS OF 512 BYTES
25 IF T < 129 THEN GOTO 50
30 HOME : VTab 6: 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) - (INT (T / 16) = INT (T / 16))
50 PRINT DS"BLOAD RETTAB,01"
55 POKE 821,Y = 2: POKE 924,0: CALL 768
60 PRINT DS"BLOAD CBF,0B0,01"
65 PRINT DS"BLOAD SAVE.SPC,0B0,01"
70 N = PEEK (-16255): N = PEEK (-16255)
75 PRINT DS"BLOAD SWEEP,0B0,01"
80 FOR X = Y + 16 TO (Y + 1) * 16 + 1
85 CALL 812
90 POKE 15312,209: POKE 15313,193
95 POKE 15314,60: POKE 15315,60
100 POKE 15314,181: POKE 15315,180
105 POKE 15318,141: POKE 15319,211
110 POKE 15320,105: POKE 15321,128
115 POKE 37271,4: POKE 37272,1
120 CALL 3044
125 PRINT D4: POKE 37272,0
130 CALL 768
135 POKE 37272,1: NEXT X
140 PRINT DS"BLOAD SAVETAB,01"
145 POKE 821,Y = 2 + 16: POKE 924,0: CALL 768
150 NEXT Y
155 HOME : POKE 34,0 : POKE 49233,0
160 S = PEEK (20480)
165 IF S = 0 THEN GOTO 175
170 PRINT D4: PRINT DS"RUN CHOOCH,01"
175 END

JLOAD A-AVERAGE

JLIST

3 DE = CHR$ (4)
10 HOME : VTab 15: INVERSE : PRINT "A AVERAGES,OF B SPECTRUMS EACH,FOR CHAN.A "
15 FOR Y = 0 TO 71: PRINT DS"BLOAD RETTAB,01"
20 POKE 821,Y = 2 + 16: CALL 768
25 PRINT DS"BLOAD CBF,0B0,01": CALL 768: PRINT DS"BLOAD AVE.STAB,01"
30 N = PEEK (-16255): IN = PEEK (-16255)
35 PRINT DS"BLOAD AVERAGE,0B0,01"
40 CALL 762: TEXT
45 PRINT DS"SAVE AVE." Y, A#A400, L#200, 01"
50 NEXT Y
55 PRINT DS"RUN FINAL-AVE,A,DI"

JLOAD B-AVERAGE

JLIST

3 DE = CHR$ (4)
10 HOME : VTab 15: INVERSE : PRINT "B AVERAGES,OF B SPECTRUMS EACH,FOR CHAN.B "
15 FOR Y = 0 TO 71: PRINT DS"BLOAD RETTAB,01"
20 POKE 821,Y = 2 + 16: CALL 768
25 PRINT DS"BLOAD CBF,0B0,01": CALL 768: PRINT DS"BLOAD AVE.STAB,01"
30 HOME : VTab 15: INVERSE : PRINT "B AVERAGES,OF B SPECTRUMS EACH,FOR CHAN.B "
35 N = PEEK (-16255): N = PEEK (-16255)
40 PRINT DS"BLOAD BLURS,0B0,01"
45 CALL 762: TEXT
50 PRINT DS"SAVE AVE." Y, A#A400, L#200, 01"
55 NEXT Y
60 PRINT DS"RUN FINAL-AVE,B,DI"
LOAD FINAL-AVE.A

JLIST

5 REM THIS PROGRAM LOADS CH.A AVERAGES IN SUPERROM II, IN THE RIGHT POSITION FOR THE POINT OF VIEW OF THE N.L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.A
10 DATA CHA$ (4)
15 N = PEEK (-16255):N = PEEK (-16255)
20 FOR Y = 0 TO 7
30 REM $2340=8000
35 PRINT D$"LOAD AVE.""A"".""AV",Y"",Di$"
40 REM NOW, SUPERROM II CONTAINS 8 AVERAGES FOR CH.A, FROM $8000-1F1FF, 10400-105FF
45 X$=D$"LOAD CLEAR.D$30,Di$": CALL 768: PRINT D$"LOAD AVE.,CH.1",Di$"
50 HOME: VTAS 15: INVERSE: PRINT "THE FINAL AVERAGE OF 8 SPECIF.NR FOR CHAN. A WILL BE DON"
60 E NOW AND WILL BE SAVED ON THE DISK"
: I NORMAL
65 PRINT D$"LOAD AVE.,CH.1",Di$"
70 CALL 768: TEXT
75 REM NOW, THE AVERAGE OF 8 AVERAGES (EACH OF 8 SPEC. 1 OF CH.A) IS LOCATED AT $4000
80 PRINT D$"SAVE AVE.AVE.A",AM4000,L#200,Di$"
85 PRINT D$"RUN SEE FIN.AVE.A,Di$"

LOAD FINAL-AVE.B

JLIST

5 REM THIS PROGRAM LOADS CH.B AVERAGES IN SUPERROM II, IN THE RIGHT POSITION FOR THE POINT OF VIEW OF THE N.L AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.B
10 DATA CHB$ (4)
15 N = PEEK (-16255):N = PEEK (-16255)
20 FOR Y = 0 TO 7
25 REM $2360=10200
30 K = PEEK(+1024)
35 PRINT D$"LOAD AVE.""B"".""AV",Y"",Di$"
40 REM NOW, SUPERROM II CONTAINS 8 AVERAGES FOR CH.B, FROM $10200-1F1FF, 10400-105FF
45 X$=D$"LOAD CLEAR.D$30,Di$": CALL 768: PRINT D$"LOAD AVE.,CH.1",Di$"
50 HOME: VTAS 15: INVERSE: PRINT "THE FINAL AVERAGE OF 8 SPECIF.NR FOR CHAN. B WILL BE DON"
60 E NOW AND WILL BE SAVED ON THE DISK"
: I NORMAL
65 PRINT D$"LOAD AVE.,CH.1",Di$"
70 CALL 768: TEXT
75 REM IN LINE 46, THE HOME STATEMENT IS USED TO ERASE THE N.L THAT APPEARS ON THE SCREEN BY LOADING THE BINARY FILE "AVEL0700.B"
80 PRINT D$"LOAD AVE.,CH.1",Di$"
85 CALL 768: TEXT
90 REM NOW, THE AVERAGE OF 8 AVERAGES (EACH OF 8 SPEC. 1 OF CH.B) IS LOCATED AT $4000
95 PRINT D$"SAVE AVE.AVE.B",AM4000,L#200,Di$"
100 PRINT D$"RUN SEE FIN.AVE.B,Di$"
3LOAD SEE FIN.AVE.A

LIST
5 REM THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.A ON THE SCREEN
10 D$ = CHR$(4): PRINT DS"LOAD MA,D$"; PRINT DS"LOAD MA,D$"; PRINT DS"LOAD M4,A11; PRINT DS"LOAD C5,D$"
20 MAC.D$1
30 PRINT DS"LOAD BFS,R1; PRINT DS"LOAD CT,D$; PRINT DS"LOAD CS,D$"
40 PRINT DS"LOAD AVE.AVE.A,10000,D$"
50 POKE 37272.1; TEXT : HOME : PRINT "PRESS A TO NOT SEE AVERAGE"
60 PRINT "PRESS OTHER KEY TO SEE AVERAGE"
70 GET A$:
80 IF A$ = "A" GOTO 80
90 PRINT "TITLE FOR AVERAGE OF 3 AVERAGES,CH.A": INPUT TA$: PRINT DS"PR$1": PRINT T CHR$(14): TA$: CHR$(20): PRINT DS"PR$1"
50 POKE 15212.001: POKE 15211.192: POKE 15214.100
60 POKE 15315.001: POKE 15314.192: POKE 15317.100
70 POKE 15216.144: POKE 15215.144: POKE 15220.155
80 POKE 15212.100: POKE 15211.100: POKE 15212.200
90 CALL 3044: PRINT DS: POKE 74,0: TEXT : HOME
100 POKE 15215.001: POKE 15216.144: POKE 15217.144: POKE 15220.155
110 GET A$:
20 PRINT "TITLE FOR AVERAGE OF 8 AVERAGES,CH.B": INPUT TA$: PRINT DS"PR$1": PRINT T CHR$(14): TA$: CHR$(20): PRINT DS"PR$1"
50 POKE 15212.001: POKE 15211.192: POKE 15214.100
60 POKE 15315.001: POKE 15314.192: POKE 15317.100
70 POKE 15216.144: POKE 15215.144: POKE 15220.155
80 POKE 15212.100: POKE 15211.100: POKE 15212.200
90 CALL 3044: PRINT DS: POKE 74,0: TEXT : HOME
100 POKE 15215.001: POKE 15216.144: POKE 15217.144: POKE 15220.155
110 PRINT DS"PRINT PLOT-A1,D$"

3LOAD SEE FIN.AVE.B

LIST
5 REM THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.B ON THE SCREEN
10 D$ = CHR$(4): PRINT DS"LOAD MA,D$"; PRINT DS"LOAD MA,D$"; PRINT DS"LOAD M4,A11; PRINT DS"LOAD C5,D$"
20 MAC.D$1
30 PRINT DS"LOAD BFS,R1; PRINT DS"LOAD CT,D$; PRINT DS"LOAD CS,D$"
40 PRINT DS"LOAD AVE.AVE.A,10000,D$"
50 POKE 37272.1; TEXT : HOME : PRINT "PRESS A TO NOT SEE AVERAGE"
60 PRINT "PRESS OTHER KEY TO SEE AVERAGE"
70 GET A$:
80 IF A$ = "A" GOTO 80
90 PRINT "TITLE FOR AVERAGE OF 8 AVERAGES,CH.B": INPUT TA$: PRINT DS"PR$1": PRINT T CHR$(14): TA$: CHR$(20): PRINT DS"PR$1"
50 POKE 15212.001: POKE 15211.192: POKE 15214.100
60 POKE 15315.001: POKE 15314.192: POKE 15317.100
70 POKE 15216.144: POKE 15215.144: POKE 15220.155
80 POKE 15212.100: POKE 15211.100: POKE 15212.200
90 CALL 3044: PRINT DS: POKE 74,0: TEXT : HOME
100 POKE 15215.001: POKE 15216.144: POKE 15217.144: POKE 15220.155
110 PRINT DS"PRINT PLOT-B1,D$"

3LOAD CHOICE

LIST
5 D$ = CHR$(4): HOME
10 PRINT "WHICH CHANNEL DO YOU WANT TO PROCESS(AVERAGE,HISTOGRAM)?:
15 PRINT "************************************************************
20 VTAB 12: PRINT "PRESS 1 FOR CHANNEL A"
30 VTAB 13: PRINT "PRESS 2 FOR CHANNEL B"
40 VTAB 14: PRINT "PRESS 3 FOR BOTH CHANNEL, A AND B"
50 VTAB 17: GET A$: IF VAL (A$) < 1 THEN GOTO 55
60 GOTO 55
70 IF VAL (A$) < 2 THEN GOTO 45
80 GOTO 55
90 IF VAL (A$) < 3 THEN GOTO 45
100 GOTO 95
110 PRINT DS"PRINT A-AVERAGE.D$"
120 PRINT DS"PRINT B-AVERAGE.D$"
130 END
140 END
150 POE LED9,1
160 POE LED9,1
LOAD AXIS

LIST

50 G = CHR$(4)
10 REM DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BICOMAP
15 MOV
20 MODEL 3
25 HPLT 15.144 TO 15,0
30 HPLT 14.1 TO 16,11: HPLT 13,2 TO 17,2
35 HPLT 15,144 TO 279,144
40 HPLT 277,142 TO 277,146: HPLT 278,143 TO 278,145
45 PRINT DEF"LOAD MAGICSPACER,01"
50 POKE 222,0: POKE 223,96
55 SCALE = 1
60 RGT = 0
65 REM ANY CENTER UP TO CENTER DOWN
70 REM THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75 REM DRAW UP NUMBERS
80 A = 14418 = 184
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 = 112
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 = 160
210 IF N < 8 GOTO 220
215 IF N = 8 THEN X = 239
220 NEXT N
225 DATA 14,12,10,14,15,10,16,17,14,10,17
230 REM DRAW DOWN NUMBERS
235 X = 16
240 FOR N = 1 TO 4
245 READ S
250 DRAW S AT X,0
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,0
290 X = X + 4
295 IF N < 4 GOTO 315
300 IF N = 4 THEN Y = 15
305 IF N < 8 GOTO 315
310 IF N = 8 THEN X = 93
315 NEXT N
320 DATA 49,10,10,10,12,13,13,13,13,13,13,13
325 X = 151
330 FOR N = 1 TO 12
335 READ S
340 DRAW S AT X,0
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 = 256
370 NEXT N
375 DATA 14,20,10,10,10,10,10,10,10,10,10,10
380 REM THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE A
385 IS 5
380 REM DRAW=% ON THE HORIZONTAL AXIS
390 X = 250
395 FOR N = 1 TO 2
400 READ S
405 DRAW S AT X,126
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 I(10)
445 FOR I = 1 TO 5
450 READ S
455 Y(I) = 144 - 9 * C - 2: REM 2 IS BECAUSE THE X/L WHICH DRAW THE CHAR. COVE
460 IDENT THE C/A
465 R. PROD THE TOP ONE CHAR. HAS 5 VERT. POINTS:
466 Z(I) = Y(I) + (S * C) / (2 * I)
470 PRINT "=",Y(I),S,C,2
475 PRINT "=",Z(I),"=",I
480 NEXT I
485 DATA 40,80,100,40,200
490 FOR I = 1 TO 5
495 DRAW 9 AT X,Y(I)
500 NEXT I
505 NEXT I
510 FOR I = 1 TO 5
515 X = 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 S49,15,49,19,12,13,13,17,17,27 AND U=49,15,49,17,12,13,27,12,15,12,19
560 DATA 49,49,15,49,49,19,17,12,13,13,17,12,13,15,12,19
565 REM DRAW"%" ON THE VERTICAL AXIS
570   X = 221. 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 39, 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
ST PART, "HM2.."
    "WHERE IS S
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, 25, 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 26, 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, 10, 12, 10, 12, 49, 50, 49, 9, 25, 30, 10, 25
855   PRINT $"RUN PLOT-A,01"
LIST

5 DI = CHR$(4)
10 REM DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15 MSR
20 HCOLOR = 3
25 HPLT 15,14 TO 15,0
30 HPLT 14,1 TO 16,1: HPLT 13,2 TO 17,2
35 HPLT 15,14 TO 277,144
40 HPLT 277,142 TO 277,145: HPLT 278,143 TO 278,145
45 PRINT DI"BLOAD BASICSPLAS,01"
50 POKE 222,01: POKE 223,96
55 SCALE = 1
60 ROT= 0
65 REM ANY CENTER UP IS X CENTER DOWN
70 REM THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS A
75 REM DRAW UP NUMBERS
80 A = 145: B = 154
85 X = 0
90 FOR N = 1 TO 6
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 = 25
120 NEXT N
125 DATA 49,37,49,49,14,10,19
130 X = 51
135 FOR N = 1 TO B
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 = 190
210 IF N > 8 GOTO 220
215 IF N = 8 THEN X = 239
220 NEXT N
225 DATA 12,10,14,12,18,10,16,17,14,10,17
230 REM DRAW DOWN NUMBERS
235 X = 16
240 FOR N = 1 TO 4
245 READ S
250 DRAW S AT X,A
255 X = X + 4
260 NEXT N
265 DATA 49,12,10,20
270 X = 37
174

275 FOR N = 1 TO 12
280 READ S
285 DRAW S AT X,Y
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 49,18,10,18,12,16,10,16,12,16
325 X = 131
330 FOR N = 1 TO 12
335 READ S
340 DRAW S AT X,Y
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 14,20,10,15,10,16,10,16,12,10,12
380 REM THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
385 IS 6
390 REM DRAW "X1" ON THE HORIZONTAL AXIS
395 X = 260
400 FOR N = 1 TO 2
405 READ S
410 DRAW S AT X,Y
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. CONV
460 PRINT "H" OR "I" FROM THE TOP (ONE CHAR., HAS 5 VERT. POINTS)
465 PRINT "V(I) = Y(I) + (S * C) / (2 * I)"
470 PRINT "Z(I) = Y(I)"
475 PRINT "Z(I) = Z(I)"
480 NEXT I
485 DATA 49,50,60,120,160,200
490 FOR I = 1 TO 5
495 DRAW S AT X,Y(I)
500 DRAW S 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
530 DRAW S AT X,Y(I)
535 DRAW U AT X,Z(I)
540 X = X + 4
545 NEXT I
550 REM DRAW "X", "V", "H", "Y", "Z" ON THE VERTICAL AXIS
555 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 27, 29
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, 57, 42, 10
650 X = X + 5
655 FOR N = 1 TO 5
660 READ S
665 DRAW S AT X, Y
670 X = X + 5
675 IF N = 5 THEN X = X + 1
680 NEXT N
685 DATA 36, 27, 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 \+1\ EXCEPTION IN THE LAST PART, "H".
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 + 5
790 FOR N = 1 TO 3
795 READ S
800 DRAW S AT X, Y
805 X = X + 7
810 NEXT N
815 DATA 20, 27, 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, 50, 48, 9, 25, 20, 10, 24
855 PRINT "RUN: PLOT-9.91"
JLOAD PLOT-A

JLIST

50 X = CHR$(4)
10 HFOR 10: VTAB 25: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15 REM CACL: TOTAL POWER
20 G = 0:01 = 1001:02 = 101: REM 01 IS FOR THE X AND 02 IS FOR HIST.AMPLT.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 "TOTAL POWER ="G
55 REM J,K,T,U ARE NO. OF POINTS FOR THE鲱 INTERVALS OF THE HISTOGRAM
60 REM J,K,T,U
70 DATA 7,14,29,27
75 REM CACL: % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90 FOR Y1 = 0 TO 2 * J
95 X1 = PEEK(16384 + Y1)
100 N1 = X1 ^ 2
105 S1 = S1 + P1
110 IF N = 7 THEN GOTO 130
120 IF N = 7 THEN A1 = (S1 / S) * 01
125 IF N = 8 THEN S1 = 0
130 NEXT Y1
135 A2 = (S1 / S1) * 01
140 PRINT "% OF FIRST BAND ="A1
145 PRINT "% OF SECOND BAND ="A2
150 REM CACL: % OF NEXT FIVE BANDS
155 U = 0
160 S2 = 0
165 FOR Y2 = 0 TO 2P
170 X2 = PEEK(16384 + 2 * J + Y2)
175 P2 = X2 ^ 2
180 S2 = S2 + P2
185 N = N + 1
190 IF N = 16 THEN GOTO 230
195 IF N = 16 THEN A2 = (S2 / S) * 01
200 IF N = 16 THEN S2 = 0
205 IF N = 26 THEN GOTO 220
210 IF N = 26 THEN A2 = (S2 / S) * 01
215 IF N = 29 THEN S2 = 0
220 IF N = 8 THEN GOTO 250
225 IF N = 42 THEN A2 = (S2 / S) * 01
230 IF N = 42 THEN S2 = 0
235 IF N = 56 THEN GOTO 200
240 IF N = 96 THEN A2 = (S2 / S) * 01
245 IF N = 96 THEN S2 = 0
250 NEXT Y2
255 A7 = (S2 / S) * 01
260 PRINT "% OF THIRD BAND ="A7
265 PRINT "% OF FOURTH BAND ="A2
270 PRINT "% OF FIFTH BAND ="A3
275 PRINT "% OF SIXTH BAND="A6
280 PRINT "% OF SEVENTH BAND="A7
285 REM CALC % POWER OF NEXT FIVE BANDS
290 N = 0
295 J5 = 0
300 FOR Y3 = 0 TO 143
305 IF = PEEX (16284 + 2 * J + 5 * K + Y3)
310 IF = Y3 + 2
315 S3 = S3 + P2
320 N = N + 1
325 IF N < 29 THEN GOTO 325
330 IF N = 29 THEN A9 = (S2 / S1) * 01
335 IF N = 30 THEN B3 = 0
340 IF N < 55 THEN GOTO 345
345 IF N = 55 THEN A9 = (S2 / S1) * 01
350 IF N = 57 THEN B2 = 0
355 IF N < 87 THEN GOTO 365
360 IF N = 87 THEN B1 = (S2 / S1) * 01
365 IF N = 87 THEN B2 = 0
370 IF N < 116 THEN GOTO 375
375 IF N = 116 THEN B2 = (S2 / S1) * 01
380 IF N = 117 THEN B3 = 0
385 NEXT Y3
390 S3 = (S3 / S1) * 01
395 PRINT "% OF EIGHTH BAND="S6
400 PRINT "% OF NINTH BAND="A9
405 PRINT "% OF TENTH BAND="B1
410 PRINT "% OF ELEVENTH BAND="S2
415 PRINT "% OF TWELTH BAND="B3
420 REM CALC % POWER OF LAST BAND
425 B4 = 0
430 FOR Y4 = 0 TO 24
435 X4 = PEEX (16284 + 2 * J + 5 * K + 5 * Y4)
440 F4 = X4 / 2
445 B4 = B4 + P4
450 NEXT Y4
455 B4 = (S4 / S3) * 125
460 PRINT "% OF THIRTEENTH BAND="B4
465 PRINT "TOTAL % BEFORE ROUND OFF IS "A1 = A2 + A3 = A4 + A5 + A6 + B1 + B2 + B3 + B4
470 A1 = INT (A1 * 01) + 0.05 / 02
475 A2 = INT (A2 * 01) + 0.05 / 02
480 A3 = INT (A3 * 01) + 0.05 / 02
485 A4 = INT (A4 * 01) + 0.05 / 02
490 A5 = INT (A5 * 01) + 0.05 / 02
495 A6 = INT (A6 * 01) + 0.05 / 02
500 B1 = INT (B1 * 01) + 0.05 / 02
505 B2 = INT (B2 * 01) + 0.05 / 02
510 B3 = INT (B3 * 01) + 0.05 / 02
515 B4 = INT (B4 * 01) + 0.05 / 02
520 B1 = INT (B1 * 01) + 0.05 / 02
525 B2 = INT (B2 * 01) + 0.05 / 02
530 B3 = INT (B3 * 01) + 0.05 / 02
535 B4 = INT (B4 * 01) + 0.05 / 02
540 PRINT "% OF THE THIRTEENTH BAND ARE:
550 PRINT "A5="A5 PRINT "A6="A6 PRINT "B1="B1 PRINT "B2="B2 PRINT "B3="B3 PRINT "B4="B4
555 B1 = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
'570 PRINT "TOTAL % POWER IS IX+TX"
572 PRINT D*"FREQ"
580 DIM O(20): DIM A(20)
585 DIM N(20): DIM M(20)
590 DIM F(20)
595 REM CALC. NO. OF POINTS/UNIT %,"F"
600 S = .7112 / 256
605 PRINT "S=
610 REM CALC. NO. OF POINTS/UNIT %,"F"
615 P = 144 / 240: P = INT (1P + Q2) = .5) / 2
620 REM CALC. NO. OF POINTS/UNIT %,"F"
625 REM F(I) ARE THE FREQUENCIES OF THE END OF EACH FREQUENCY INTERVAL
630 F(0) = 0
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 S = 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 FREQUENCY INTERVALS
690 D(0) = F(0)
695 S = D(0)
700 FOR I = 1 TO 12
705 D(I) = F(I) - D(I) = 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 = O(I) = A2 = 02
745 A(2) = A3 = O(2) = A4 = 02
750 A(4) = A5 = O(2) = A6 = 02
755 A(6) = A7 = O(2) = A8 = 02
760 A(8) = A9 = O(2) = A10 = 11 = 02
765 A(10) = A11 = 11 = 02
770 A(12) = 02
775 FOR I = 0 TO 12
780 D(I) = INT ((D(I) + Q2) + .5) / 2
785 PRINT "D(I)=D(I): NEXT I"
790 REM CALC. NO. OF POINTS/INTERVAL FOR FREQUENCY SCALE,"N(I)"
795 N(I) = D(I) / S:
800 N(I) = INT (N(I) * .2) + .5)
805 PRINT "N(I)=N(I): NEXT I"
810 REM CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
815 M(I) = P * A(I)
820 PRINT "M(I)=M(I): NEXT I"
825 REM DRAW THE HISTOGRAM
830 G = 0: R = 15
835 FOR I = 0 TO 12
840 R = R + N(I): PRINT "R(I)=R"
845 FOR Y = 0 TO M(I)
850 M(LPLOT 0,144 - Y TO 0,144
855 M(LPLOT R,144 - Y TO R,144
860 NEXT Y
865 FOR X = 0 TO N(I)
870 M(LPLOT 0,144 - M(I) TO X + 0,144 = M(I)
875 NEXT X
875 G = G + N(I)
880 NEXT I
885 PRINT D*"RUN MICROBUF,01"
890 NEXT I
LIST

5 DI = CHR$(4)
10 HTRB 101 VTAB 22: INVERSE I PRINT "PLEASE WAIT!": NORMAL
15 REM CALL TOTAL POWER
20 B = 0101 + 1091; S2 = 101 REM 01 IS FOR THE % AND S2 IS FOR HIST AMPLITUDE 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 "TOTAL POWER="S
60 REM J,K,T,U ARE NO. OF POINTS FOR THE FREQUENCY INTERVALS OF THE HISTOGRAM
65 READ J,K,T,U
70 DATA 7,14,29,27
75 REM CALL % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90 FOR Y1 = 0 TO 2 * J
95 XI = PEESK(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) * 01
125 IF N = 8 THEN S1 = 0
130 NEXT Y1
135 A2 = (S1 / S) * 01
140 PRINT "% OF FIRST BAND="A1
145 PRINT "% OF SECOND BAND="A2
150 REM CALL % OF NEXT FIVE BANDS
155 N = 0
160 S2 = 0
165 FOR Y2 = 0 TO 69
170 X2 = PEESK(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) * 01
200 IF N = 15 THEN S2 = 0
205 IF N < 28 THEN GOTO 250
210 IF N = 28 THEN A4 = (S2 / S) * 01
215 IF N = 29 THEN S2 = 0
225 IF N = 30 THEN GOTO 250
230 IF N = 31 THEN A5 = (S2 / S) * 01
235 IF N = 32 THEN S2 = 0
240 IF N = 33 THEN GOTO 250
245 IF N = 36 THEN A6 = (S2 / S) * 01
250 IF N = 35 THEN S2 = 0
255 NEXT Y2
260 A7 = (S2 / S) * 01
265 PRINT "% OF THIRD BAND="A2
270 PRINT "% OF FOURTH BAND="A3
275 PRINT "% OF FIFTH BAND="A5
275 PRINT "% OF SIXTH BAND=\%\\n280 PRINT "% OF SEVENTH BAND=\%\\n285 REM CALC. % POWER OF NEXT FIVE BANDS\\n290 N = 0\\n295 B3 = 0\\n300 FOR Y3 = 0 TO 144\\n305 F3 = F3 + (18432 - 2 * J + S + K + Y3)\\n310 P3 = Y3 - 2\\n315 G3 = G3 + P3\\n320 N = N + 1\\n325 IF N < 29 THEN GOTO 385\\n330 IF N = 29 THEN A8 = (G3 / S) * 01\\n335 IF N = 30 THEN S1 = 0\\n340 IF N = 35 THEN G02 = 395\\n345 IF N = 35 THEN A9 = (G3 / S) * 01\\n350 IF N = 47 THEN S2 = 0\\n355 IF N = 87 THEN GOTO 385\\n360 IF N = 97 THEN BI = (G3 / S) * 01\\n365 IF N = 97 THEN S2 = 0\\n370 IF N = 111 THEN GOTO 385\\n375 IF N = 116 THEN S2 = (G3 / S) * 01\\n380 IF N = 117 THEN S2 = 0\\n385 NEXT Y3\\n390 G3 = (G3 / S) * 01\\n395 PRINT "% OF EIGHTH BAND=\%\\n400 PRINT "% OF NINTH BAND=\%\\n405 PRINT "% OF TENTH BAND=\%\\n410 PRINT "% OF ELEVENTH BAND=\%\\n415 PRINT "% OF TWELFTH BAND=\%\\n420 REM CALC. % POWER OF LAST BAND\\n425 S4 = 0\\n430 FOR Y4 = 0 TO 24\\n435 X4 = X4 + (18432 - 2 * J + S + K + T + Y4)\\n440 P4 = X4 - 2\\n445 G4 = S4 + P4\\n450 NEXT Y4\\n455 S4 = (S4 / S) * 125\\n460 PRINT "% OF THIRTEENTH BAND=\%\\n465 PRINT "TOTAL % BEFORE ROUND OFF IS\%\\n470 A1 = INT (((A1 * G3) + 0.05) / 02\\n475 A2 = INT (((A2 * G3) + 0.05) / 02\\n480 A3 = INT (((A3 * G3) + 0.05) / 02\\n485 A4 = INT (((A4 * G3) + 0.05) / 02\\n490 A5 = INT (((A5 * G3) + 0.05) / 02\\n495 A6 = INT (((A6 * G3) + 0.05) / 02\\n500 A7 = INT (((A7 * G3) + 0.05) / 02\\n505 A8 = INT (((A8 * G3) + 0.05) / 02\\n510 A9 = INT (((A9 * G3) + 0.05) / 02\\n515 A10 = INT (((A10 * G3) + 0.05) / 02\\n520 A11 = INT (((A11 * G3) + 0.05) / 02\\n525 A12 = INT (((A12 * G3) + 0.05) / 02\\n530 A13 = INT (((A13 * G3) + 0.05) / 02\\n535 A14 = INT (((A14 * G3) + 0.05) / 02\\n540 PRINT "% POWER OF THE THIRTEEN BANDS ARE:"\\n545 PRINT "A1=A1; PRINT "A2=A2; PRINT "A3=A3; PRINT "A4=A4; PRINT "A5=A5; PRINT "A6=A6; PRINT "A7=A7; PRINT "A8=A8; PRINT "A9=A9; PRINT "A10=A10; PRINT "A11=A11; PRINT "A12=A12; PRINT "A13=A13; PRINT "A14=A14; R4\\n545 PRINT "TA=TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+TA+T
570 PRINT "TOTAL % POWER IS 1000%"
580 DIM D(20); DIM A(20)
590 DIM X(20); DIM Y(20)
600 REM CALC. NO. OF HZ/POINT,"B"
605 S = 71.12 / 266
610 PRINT "S^T=S"
615 P = 144 / 2401P = INT (P + 02) + .5 / 02
620 REM CALC. FREQU. INTERVALS,"D(1)"
625 REM F(I) ARE THE FREQU. OF THE END OF EACH FREQU. INTERVAL.
630 F(0) = 2 * S
635 F(I) = 2 * S(I)
640 G = F(1)
645 FOR I = 2 TO 6
650 F(I) = F(I - 1) + 6I + 1F = F(I)
655 NEXT I
660 C = F(6)
665 FOR I = 7 TO 11
670 F(I) = E + T + SIE = F(I)
675 NEXT I
680 F(12) = F(11) + U + S
685 REM D(I) ARE FREQU. INTERVALS
690 D(0) = F(0)
695 D(I) = D(0)
700 FOR I = 1 TO 12
705 D(I) = D(I) + D(I) = F(I)
710 NEXT I
715 FOR I = 0 TO 12
720 PRINT "F(1)=F(I)"
725 PRINT "***************"
730 PRINT D(1)="C(1)" NEXT I
735 REM CALC. AMPLITUDES FOR HISTOGRAM
740 A(0) = A1 + 02I(A1) = A2 + 02
745 A(2) = 02I + 02I = 02I = 02
750 A(4) = 02I + 02I = 02I = 02
755 A(6) = 02I + 02I = 02I = 02
760 A(8) = 02I + 02I = 02I = 02
765 A(10) = 02I + 02I = 02I = 02
770 A(12) = 02I = 02
775 FOR I = 0 TO 12
780 D(I) = INT (D(1) + 02) + 0.5 / 02
785 PRINT "D(1)=D(I)"n"
790 REM CALC. NO. OF POINTS/INTERVAL FOR FREQU. SCALE,"N(I)"
795 N(I) = D(I) / 02I = INT (D(I) + 02) + .5 / 02
800 PRINT "N(I)=N(A)"PRINT"
805 REM CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK,"M(I)"
810 M(I) = P + 02I + 02I = INT (P + 02I + 02) +.5 / 02
815 NEXT I
820 REM DRAW THE HISTOGRAM
825 Q = 0 + 15R = 15
830 FOR I = 0 TO 12
835 R = R + H(I)
840 FOR Y = 0 TO M(I)
845 HPLCST 0,144 - Y TO 0,144
850 HPLCST R,144 - Y TO R,144
855 NEXT Y
860 FOR X = 0 TO N(I)
865 HPLCST 0,144 - M(I) TO X + 0,144 - M(I)
870 NEXT X
875 G = 0 + N(I)
880 NEXT I
885 PRINT D="RUN MICROCPU,D1"
JLOAD PLOT-A1:

JLIST

5 D$ = CHR$(4) 10 HOME : INVERSE : PRINT "WAIT FOR THE CALCULATIONS" : NORMAL
15 REM CALC. TOTAL POWER 20 S$ = "0000" : 10: REM 01 IS FOR THE X AND 02 IS FOR ROUND-OFF
25 FOR Y = 0 TO 255 30 X = PEEK (16254 + Y)
35 P = X ^ 2 40 S = S + P
45 NEXT Y 50 HOME
55 PRINT D$"PWR="," 60 PRINT "TOTAL POWER=S"
65 REM J,K,L,U ARE NO OF POINTS FOR THE FREQUENCY INTERVALS OF THE HISTOGRAM
70 REM J,K,L,U
75 DATA 7,14,29,27
80 REM CALC. % OF FIRST TWO BANDS 85 N = 0
90 REM S1 = 0 95 FOR Y1 = 0 TO 2 * J
100 X1 = PEEK (16254 + Y1)
105 P1 = X1 ^ 2 110 S1 = S1 + P1
115 N = N + 1
120 IF N < 7 THEN GOTO 125 125 IF N = 7 THEN A1 = (S1 / 0) * 01
130 IF N = 8 THEN S1 = 0 135 NEXT Y1
140 REM S2 = 0 145 PRINT "% OF FIRST BAND=",A1
150 PRINT "% OF SECOND BAND=",S1
155 REM CALC. % OF NEXT FIVE BANDS 160 N = 0
165 S2 = 0
170 FOR Y2 = 0 TO 60
175 X2 = PEEK (16254 + 2 * J + Y2)
180 P2 = X2 ^ 2 185 S2 = S2 + P2
190 N = N + 1
195 IF N < 14 THEN GOTO 195
200 IF N = 14 THEN A2 = (S2 / 0) * 01
205 IF N = 15 THEN S2 = 0
210 IF N = 16 THEN S2 = 0
215 IF N = 18 THEN S2 = 0
220 IF N = 20 THEN S2 = 0
225 IF N = 22 THEN S2 = 0
230 IF N = 24 THEN S2 = 0
235 IF N = 26 THEN S2 = 0
240 IF N = 28 THEN S2 = 0
245 IF N = 30 THEN A4 = (S2 / 0) * 01
250 IF N = 32 THEN S2 = 0
255 NEXT Y2
260 A7 = (S2 / 0) * 01
265 PRINT "% OF THIRD BAND=",A2
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 CALCULATE % POWER OF NEXT FIVE BANDS
295 N = 0
300 S2 = 0
305 FOR Y3 = 0 TO 144
310 X3 = PREV (16384 + 2 * J + S * 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 AS = (S3 / S) * O1
340 IF N = 30 THEN B5 = 0
345 IF N < 56 THEN B5 = (S3 / S) * O1
350 IF N = 57 THEN B5 = 0
355 IF N < 87 THEN GOTO 390
360 IF N = 87 THEN B5 = (S3 / S) * O1
365 IF N = 97 THEN B1 = (S3 / S) * O1
370 IF N = 97 THEN B5 = 0
375 IF N < 116 THEN GOTO 390
380 IF N = 116 THEN B5 = (S3 / S) * O1
385 IF N = 117 THEN B5 = 0
390 NEXT Y3
395 S2 = (S2 / S) * O1
400 PRINT "% OF EIGHTH BAND=A8"
405 PRINT "% OF NINTH BAND=A9"
410 PRINT "% OF TENTH BAND=A1"
415 PRINT "% OF ELEVENTH BAND=B2"
420 PRINT "% OF TWELTH BAND=B3"
425 REM CALCULATE % POWER OF LAST BAND
430 S4 = 0
435 FOR Y4 = 0 TO 26
440 X4 = PREV (16384 + 2 * J + S * K + T + Y4)
445 P4 = X4 ^ 2
450 S4 = S4 + P4
455 NEXT Y4
460 S4 = (S4 / S) * O1
465 PRINT "% OF THIRTEENTH BAND=B4"
470 PRINT "TOTAL % BEFORE ROUND OFF IS TX=A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + S7 + S4"
475 A1 = INT ((A1 + G2) * O0) / G2
480 A2 = INT ((A2 + G2) * O0) / G2
485 A3 = INT ((A3 + G2) * O0) / G2
490 A4 = INT ((A4 + G2) * O0) / G2
495 A5 = INT ((A5 + G3) * O0) / G2
500 A6 = INT ((A6 + G2) * O0) / G2
505 A7 = INT ((A7 + G2) * O0) / G2
510 A8 = INT ((A8 + G2) * O0) / G2
515 A9 = INT ((A9 + G2) * O0) / G2
520 B1 = INT ((B1 + G2) * O0) / G2
525 B2 = INT ((B2 + G2) * O0) / G2
530 S2 = INT ((S2 + G2) * O0) / G2
535 S7 = INT ((S7 + G2) * O0) / G2
540 S4 = INT ((S4 + G3) * O0) / G2
545 PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
570  T1 = A1 + A2 = A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B4
572  PRINT "TOTAL % POWER IS T1"; T1
580  PRINT "";
585  DIM S(0:0): DIM N(0:0): DIM F(0:0)
600  FOR I = 0 TO 12:IN(I) = INT (B(I))
605  F(I) = (B(I) - INT (B(I))) * 0.05 * 0.02
610  PXE (16999 = 1),N(I): POKE (16912 + I),F(I)
615  NEXT I
620  PRINT DS:"RUN PIECHART.D1"

3LOAD MICROBUF

3LIST

5 PRINT CHR$(44)" PR1"
10 PRINT CHR$(96)"PR1"; PRINT CHR$(69):"G66";
15 PRINT CHR$(40);" PR0"
20 TEXT
25 HOME
30 E = PEEK (16866);
35 IF B < > 1 THEN 45
40 POKE 16894,0: PRINT CHR$(40)"RUN B-averabe.D1"
45 END
JLIST

5 D4 = CDR3 (4)
10 HOME : INVERSE : PRINT "WAIT FOR THE CALCULATIONS!": NORMAL
15 REM CALC. TOTAL POWER
20 S = 0: G1 = 100: G2 = 10: REM G1 IS FOR THE % AND G2 IS FOR ROUND OFF
25 FOR Y = 0 TO 255
30 X = 18432 - Y
35 P = X ^ 2
40 S = S + P
45 NEXT Y
50 HOME
55 PRINT D4:"FRE#1"
60 PRINT "TOTAL POWER="S
65 REM J,K,L,U ARE NO. OF POINTS FOR THE FREQ INTERVALS OF THE HISTORI
70 READ J,K,L,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 XI = PEEK (18432 + Y1)
105 P1 = XI ^ 2
110 S1 = S1 + P1
115 N = N + 1
120 IF N < 7 THEN GOTO 135
125 IF N = 7 THEN A1 = (S1 / N) * G1
130 IF N = 6 THEN B1 = 0
135 NEXT Y1
140 A2 = (S1 / N) * G1
145 PRINT "% OF FIRST BAND="A1
150 PRINT "% OF SECOND BAND="A2
155 REM CALC. % OF NEXT FIVE BANDS
160 N = 0
165 S2 = 0
170 FOR Y2 = 0 TO 69
175 XI = PEEK (18432 + 2 * J + Y2)
180 P2 = XI ^ 2
185 S2 = S2 + P2
190 N = N + 1
195 IF N < 14 THEN GOTO 225
200 IF N = 14 THEN A3 = (S2 / N) * G1
205 IF N = 12 THEN B2 = 0
210 IF N < 18 THEN GOTO 225
215 IF N = 18 THEN AS = (S2 / N) * G1
220 IF N = 19 THEN B2 = 0
225 IF N < 42 THEN GOTO 235
230 IF N = 42 THEN A4 = (S2 / N) * G1
235 IF N = 42 THEN B2 = 0
240 IF N < 56 THEN GOTO 255
245 IF N = 56 THEN A5 = (S2 / N) * G1
250 IF N = 57 THEN B2 = 0
255 NEXT Y2
260 J7 = (S2 / N) * G1
265 PRINT "% OF THIRD BAND="A3
270 PRINT "% OF FOURTH BAND="A4
186

276 PRINT "% OF FIFTH BAND=43
280 PRINT "% OF SIXTH BAND=42
285 PRINT "% OF SEVENTH BAND=47
290 REM CALC: % POWER OF NEXT FIVE BANDS
295 N = 0
300 ST = 0
305 FOR Y = 0 TO 144
310 X5 = FEED (15932 + 2 * J + 5 * K + Y5)
315 X5 = X5 * 2
320 ST = ST + P3
325 N = N + 1
330 IF N < 29 THEN GOTO 390
335 IF N = 29 THEN A9 = (E3 / B) * 01
340 IF N = 30 THEN B3 = 0
345 IF N = 56 THEN GOTO 390
350 IF N = 57 THEN A9 = (E3 / B) * 01
355 IF N = 58 THEN B3 = 0
360 IF N = 67 THEN GOTO 390
365 IF N = 68 THEN B3 = (E3 / B) * 01
370 IF N = 69 THEN B3 = 0
375 IF N = 116 THEN GOTO 390
380 IF N = 117 THEN B3 = (E3 / B) * 01
385 IF N = 118 THEN B3 = 0
390 NEXT Y3
395 B3 = (E3 / B) * 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 SA = 0
435 FOR YA = 0 TO 28
440 X4 = FEED (15932 + 2 * J + 5 * K + 5 * T + Y4)
445 P4 = X4 * 2
450 SA = SA + P4
455 NEXT YA
460 SA = (E1 / B) * 120
465 PRINT "% OF THIRTEENTH BAND=B1
470 PRINT "TOTAL % BEFORE ROUND OFF BW T%=A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8
475 B + A9 = B1
480 B2 = B3 + B4
477 A1 = INT (A1 + C2) + 0.05) / 02
480 A2 = INT (A2 + C2) + 0.05) / 02
485 A3 = INT (A3 + C2) + 0.05) / 02
490 A4 = INT (A4 + C2) + 0.05) / 02
495 A5 = INT (A5 + C2) + 0.05) / 02
500 A6 = INT (A6 + C2) + 0.05) / 02
505 A7 = INT (A7 + C2) + 0.05) / 02
510 A8 = INT (A8 + C2) + 0.05) / 02
515 A9 = INT (A9 + C2) + 0.05) / 02
520 BI = INT (B1 + C2) + 0.05) / 02
525 B2 = INT (B2 + C2) + 0.05) / 02
530 B3 = INT (B3 + C2) + 0.05) / 02
535 B4 = INT (B4 + C2) + 0.05) / 02
540 PRINT "% POWER OF THE THIRTEENTH BAND="
565 PRINT "B4=",B4
570 T%= A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + B1 + B2 + B3 + B4
575 PRINT "TOTAL % POWER OF BW T%=T%"
580 PRINT "END OF WORM"
LOAD PIECHART

LIST

5 REM **************
10 REM *PIECHART BY CHING*
15 REM **************
20 REM A=1ST, B=2NOX..., ETC.
25 DIM S(201):DIM N(201):DIM R(20)
30 FOR I = 0 TO 12
35 N(I) = PEEK(16897 + I)
40 R(I) = PEEK(16912 + I)
45 S(I) = N(I) + R(I) / 10
50 NEXT I
65 REM PORTIONS
70 P = 13
75 P1 = P1 + P2 = P
80 PRINT CHR$(45)"LOAD HABISPACER"
85 POKE 222,1:POKE 222,96
90 DIM A(201),B(201),C(201),D(201)
95 DEF FN Y(Z) = RAD * COS (Z) + XCTR
100 DEF FN Y(Z) = RAD * SIN (Z) / .85 + YCTR
105 MGR = MCOLOR = 7:IST = .1
110 XCTR = 70: YCTR = 90: RADC = .85
115 HPLT XCTR, YCTR
120 FOR I = 0 TO 6.3 STEP ST
125 X = RAD * COS (Z) + XCTR
130 Y = RAE * SIN (Z) / .85 + YCTR
135 HPLT 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 + F: GOTO 220
175 IF P = 7 THEN T = A + B + C + D + E + F + G: GOTO 220
180 IF P = 8 THEN T = A + B + C + D + E + F + G + H: GOTO 220
185 IF P = 9 THEN T = A + B + C + D + E + F + G + H + I: GOTO 220
190 IF P = 10 THEN T = A + B + C + D + E + F + G + H + I + J: GOTO 220
195 IF P = 11 THEN T = A + B + C + D + E + F + G + H + I + J + K: GOTO 220
200 IF P = 12 THEN T = A + B + C + D + E + F + G + H + I + J + K + L: GOTO 220
205 IF P = 13 THEN T = A + B + C + D + E + F + G + H + I + J + K + L: GOTO 220
210 Z = (471 + 1) * A(P)) / 2
215 RETURN
220 R = A + T / 1000:Z = Z + 102 = INT (Z):Z = Z / 10
225 HPLT 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 I = (A.5 + A(I)) / 2: GOTO 310
250 IF P1 = 12 THEN I = (A.5 + A(I)) / 2: GOTO 310
255 IF P1 = 11 THEN I = (A.5 + A(I)) / 2: GOTO 310
260 IF P1 = 10 THEN I = (A.5 + A(I)) / 2: GOTO 310
265 IF P1 = 9 THEN I = (A.5 + A(I)) / 2: GOTO 310
270 IF P1 = 8 THEN I = (A.5 + A(I)) / 2: GOTO 310

275 IF P1 = 7 THEN Z = (A.3 + A(1)) / 2; GOTO 310
280 IF P1 = 6 THEN Z = (A.3 + A(6)) / 2; GOTO 310
285 IF P1 = 5 THEN Z = (A.3 + A(5)) / 2; GOTO 310
290 IF P1 = 4 THEN Z = (A.3 + A(4)) / 2; GOTO 310
295 IF P1 = 3 THEN Z = (A.3 + A(3)) / 2; GOTO 310
300 IF P1 = 2 THEN Z = (A.3 + A(2)) / 2; GOTO 310
305 GOTO 350
310 RAD = RAD / 2 + 10
315 SCALE= 1
320 P1 = 0
325 BL = 25 + P1
330 DRAW BL AT FN X(21), FN Y(21)
335 P1 = P1 + 1
340 IF P1 = 0 THEN GOTO 360
350 GOSUB 210
355 GOTO 325
360 REM EEO PIECHART
365 Y = 15 * X = 153
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, 36, 31, 27, 25, 35, 35, 40, 42
435 IF P2 = 12 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(12) = INT (M / 101); C(12) = INT (M - B(12) * 101); D(12) = (M - B(12) * 10 - C(12)) / 10
505 B(12) = INT (M / 101); C(12) = INT (M - B(12) * 101); D(12) = (M - B(12) * 10 - C(12)) / 10
510 B(11) = INT (K / 101); C(11) = INT (K - B(11) * 101); D(11) = (K - B(11) * 10 - C(11)) / 10
515 B(10) = INT (J / 101); C(10) = INT (J - B(10) * 101); D(10) = (J - B(10) * 10 - C(10)) / 10
520 B(9) = INT (I / 101); C(9) = INT (I - B(9) * 101); D(9) = (I - B(9) * 10 - C(9)) / 10
525 B(8) = INT (H / 101); C(8) = INT (H - B(8) * 101); D(8) = (H - B(8) * 10 - C(8)) / 10
530 B(7) = INT (G / 101); C(7) = INT (G - B(7) * 101); D(7) = (G - B(7) * 10 - C(7)) / 10
LIST 500,

500 B(12) = INT (L / 1011C(12)) = INT (L - B(12) * 1011C(12)) = (L - B(12) * 10 - C(12)) * 10

505 B(12) = INT (L / 1011C(12)) = INT (L - B(12) * 1011C(12)) = (L - B(12) * 10 - C(12)) * 10

510 B(11) = INT (K / 1011C(11)) = INT (K - B(11) * 1011C(11)) = (K - B(11) * 10 - C(11)) * 10

515 B(10) = INT (J / 1011C(10)) = INT (J - B(10) * 1011C(10)) = (J - B(10) * 10 - C(10)) * 10

520 J(9) = INT (I / 1011C(9)) = INT (I - B(9) * 1011C(9)) = (I - B(9) * 10 - C(9)) * 10

525 B(8) = INT (H / 1011C(8)) = INT (H - B(8) * 1011C(8)) = (H - B(8) * 10 - C(8)) * 10

530 B(7) = INT (G / 1011C(7)) = INT (G - B(7) * 1011C(7)) = (G - B(7) * 10 - C(7)) * 10

535 B(6) = INT (F / 1011C(6)) = INT (F - B(6) * 1011C(6)) = (F - B(6) * 10 - C(6)) * 10

540 B(5) = INT (E / 1011C(5)) = INT (E - B(5) * 1011C(5)) = (E - B(5) * 10 - C(5)) * 10

545 B(4) = INT (D / 1011C(4)) = INT (D - B(4) * 1011C(4)) = (D - B(4) * 10 - C(4)) * 10

550 B(3) = INT (C / 1011C(3)) = INT (C - B(3) * 1011C(3)) = (C - B(3) * 10 - C(3)) * 10

555 B(2) = INT (B / 1011C(2)) = INT (B - B(2) * 1011C(2)) = (B - B(2) * 10 - C(2)) * 10

560 B(1) = INT (A / 1011C(1)) = INT (A - B(1) * 1011C(1)) = (A - B(1) * 10 - C(1)) * 10

565 PRINT "COMPENSATION FOR ROUND OFF NUMBER"

570 FOR N = 1 TO P2

575 D(0) = INT (2(N) / 0.5

580 NEXT N

585 PRINT D"PRINT"; PRINT B(8),C(8),D(8)

590 FOR J = 1 TO P2

595 IF B(J) = 0 THEN B(J) = 36

600 IF C(J) = 0 THEN C(J) = 26

605 IF D(J) = 0 THEN D(J) = 26

610 NEXT J

615 PRINT D(10),C,(8),D,(8): PRINT D"PRINT"

620 Y = 36

625 REM THE FOR-NEXT STATEMENT IS FROM P2 TO 1 AND NOT FROM P2 TO 10. NEXT J: TO 1011C(1)

630 END

635 END

640 REM THE THREE VARIABLES B(1), C(1), D(1), ARE USED TO DISPLAY THE THREE DIGITS OF THE VALUE

650 REM OF THE RESULT AND NOT A SINGLE DIGIT.
190

LIST 665.
665 X = X + 5
670 DRAW C:11 + 11 AT X,Y
675 X = X + 5
700 DRAW 10 AT X,Y
705 X = X + 5
710 DRAW D:11 + 11 AT X,Y
715 X = X + 5
720 DRAW 3 AT X,Y
725 X = X + 10
750 REM DRAW THE NUMBERS TO EXPRESS THE FREQ. INTERVAL AND THE UNITS
755 FOR U = 1 TO 11
760 RED S
765 DRAW S AT X,Y
770 X = X + 5
775 NEXT U
780 Y = Y + 5
790 NEXT T
795 REM THERE ARE 13 LINES OF DATA, ONE FOR EACH PORTION OF THE PIECHART
800 DATA 215,59,49,49,37,9,49,12,10,20,20,48
805 DATA 24,11,49,12,10,20,9,47,14,10,19,20,48
810 DATA 26,2,49,14,10,9,19,49,18,10,18,30,48
815 DATA 26,5,49,13,10,13,9,12,12,17,20,48
820 DATA 27,2,12,12,12,10,17,9,12,16,10,18,30,48
825 DATA 28,2,12,14,10,16,9,15,20,10,15,30,48
830 DATA 29,2,12,20,10,15,9,13,14,10,14,20,48
835 DATA 30,2,14,12,10,14,9,14,20,10,15,20,48
840 DATA 31,2,14,12,10,14,9,14,20,10,15,20,48
845 PRINT CHR$(97): "BNEUEL"
850 PRINT CHR$(94): "PRAG"
855 TEXT
860 MOVE
865 S = FED: (16596)
870 IF S < > 1 THEN 590
875 FOR Z = 1 TO 5 PRINT CHR$(1): "RUN Z-AVERAGE.D1"
LOAD PIECHART

LIST 5-500

5 REM ***************
10 REM *PIECHART BY J.H. P. TANG*
15 REM *****************
20 REM AW,*L,5,END,...,ETC.
25 DIM S(20); DIM N(20); DIM X(20)
30 FOR I = 0 TO 10
35 N(I) = PEER (16925 - I)
40 S(I) = PEER (16975 + I)
45 R(I) = 1.0 / 10
50 NEXT
55 X = S(0)I = S(1)X = S(2)I = S(3)I = S(4)I = S(5)
65 REM PLOT:JMP
70 P = 12
75 P1 = P; P2 = P
80 PRINT CMS (4) "$BLOOD MAGICSPACER" $B
85 POKE 223, POKE 219, 9a
90 DIM A(120, B(120, C(120), D(120)
95 DEF FN X(Z) = RAD * COS (Z) + XCTR
100 DEF FN Y(Z) = RAD * SIN (Z) / .85 + YCTR
105 MGR = MGR+1: JST = 1
110 XCTR + 70; YCTR = 30; XNO = 45
115 HPLET XCTR,YCTR
120 FOR Z = 0 TO 3 STEP .75
125 X = RAD * COS (Z) + XCTR
130 Y = RAD * SIN (Z) / .85 + YCTR
135 HPLET X,Y
140 NEXT
145 IF P = 1 THEN GOTO 245
150 IF P = 1 THEN T = A: GOTO 220
155 IF P = 2 THEN T = A + B: GOTO 220
160 IF P = 3 THEN T = A + B + C: GOTO 220
165 IF P = 4 THEN T = A + B + C + D: GOTO 220
170 IF P = 5 THEN T = A + B + C + D + E: GOTO 220
175 IF P = 6 THEN T = A + B + C + D + E + F: GOTO 220
180 IF P = 7 THEN T = A + B + C + D + E + F + G: GOTO 220
185 IF P = 8 THEN T = A + B + C + D + E + F + G + H: GOTO 220
190 IF P = 9 THEN T = A + B + C + D + E + F + G + H + I: GOTO 220
195 IF P = 10 THEN T = A + B + C + D + E + F + G + H + I + J: GOTO 220
200 IF P = 11 THEN T = A + B + C + D + E + F + G + H + I + J + K: GOTO 220
205 IF P = 12 THEN T = A + B + C + D + E + F + G + H + I + J + K + L: GOTO 220
210 Z = (A(Z) = 1 + A(Z)): / 2
215 RETURN
220 I = 0 : J = T / 100; I = I + 10; I = INT (I): I = I / 10
225 HPLET FN X(Z), FN Y(Z) TO XCTR,YCTR
230 @P1 = I
235 P = P + 1
240 GOTO 145
245 IF P = 13 THEN Z = (A(Z) + A(17)): / 2: GOTO 210
250 IF P = 13 THEN Z = (A(Z) + A(11)): / 2: GOTO 210
255 IF P = 14 THEN Z = (A(Z) + A(11)): / 2: GOTO 210
260 IF P = 14 THEN Z = (A(Z) + A(11)): / 2: GOTO 210
265 IF P = 15 THEN Z = (A(Z) + A(9)): / 2: GOTO 210
270 IF P = 16 THEN Z = (A(Z) + A(8)): / 2: GOTO 210
275 IF P1 = 7 THEN Z = (A1 + A2) / 2: GOTO 510
280 IF P1 = 0 THEN Z = (A1 + A2) / 2: GOTO 510
285 IF P1 = 5 THEN Z = (A1 + A2) / 2: GOTO 510
290 IF P1 = 4 THEN Z = (A1 + A2) / 2: GOTO 510
295 IF P1 = 9 THEN Z = (A1 + A2) / 2: GOTO 510
300 IF P1 = 2 THEN Z = (A1 + A2) / 2: GOTO 510
305 GOTO 330
310 RAD = RAD / 2 + 10
315 SCALE = 1
320 P1 = 0
325 SC = SC + P1
330 DRAW BL AT PN X(2), PN Y(2)
335 P1 = P1 - 1
340 P1 = P1 - 1
345 IF P1 = 0 THEN GOTO 340
350 GOTO 310
355 GOTO 330
360 REN E09 PIECHART
365 Y = 151 X = 151
370 FOR W = 1 TO 10
375 RENO 0
380 DRAW 0 AT X,Y
385 X = X + 5
390 NEXT W
395 X = X + 10
400 FOR Y = 1 TO 10
405 RENO 1
410 DRAW 1 AT X,Y
415 X = X + 5
420 NEXT W
425 REN DATA FOR TITLE
430 DATA 27,27,27,49,49,36,31,27,25,23,23,40,42
435 IF F2 = 13 THEN 500
440 IF F2 = 12 THEN 500
445 IF F2 = 11 THEN 510
450 IF F2 = 10 THEN 515
455 IF F2 = 9 THEN 520
460 IF F2 = 8 THEN 525
465 IF F2 = 7 THEN 530
470 IF F2 = 6 THEN 535
475 IF F2 = 5 THEN 540
480 IF F2 = 4 THEN 545
485 IF F2 = 3 THEN 550
490 IF F2 = 2 THEN 555
495 IF F2 = 1 THEN 560
500 B(13) = INT (M / 100) * I(13) = INT (M - B(13) * 10) / 13 = (M - B(13)) * 10
505 B(13) = INT (M / 100) * I(13) = INT (M - B(13) * 10) / 13 = (M - B(13)) * 10
LIST 500-660

500 B(I) = INT (I / 101) + C(I) = INT (I - B(I) * 10) / 101 + C(I) + 10
505 B(I2) = INT (L / 101) + C(I2) = INT (L - B(I2) * 10) / 101 + C(I2) + 10
510 B(I1) = INT (K / 101) + C(I1) = INT (K - B(I1) * 10) / 101 + C(I1) + 10
515 B(I0) = INT (J / 101) + C(I0) = INT (J - B(I0) * 10) / 101 + C(I0) + 10

520 B(I9) = INT (I / 101) + C(I9) = INT (I - B(I9) * 10) / 101 + C(I9) + 10
525 B(I8) = INT (H / 101) + C(I8) = INT (H - B(I8) * 10) / 101 + C(I8) + 10
530 B(I7) = INT (G / 101) + C(I7) = INT (G - B(I7) * 10) / 101 + C(I7) + 10
535 B(I6) = INT (F / 101) + C(I6) = INT (F - B(I6) * 10) / 101 + C(I6) + 10
540 B(I5) = INT (E / 101) + C(I5) = INT (E - B(I5) * 10) / 101 + C(I5) + 10
545 B(I4) = INT (D / 101) + C(I4) = INT (D - B(I4) * 10) / 101 + C(I4) + 10
550 B(I3) = INT (C / 101) + C(I3) = INT (C - B(I3) * 10) / 101 + C(I3) + 10
555 B(I2) = INT (B / 101) + C(I2) = INT (B - B(I2) * 10) / 101 + C(I2) + 10
560 B(I1) = INT (A / 101) + C(I1) = INT (A - B(I1) * 10) / 101 + C(I1) + 10

562 REM COMPENSATION FOR ROUND-OFF NUMBER

570 FOR N = 1 TO P2
575 D(I) = INT (D(I)) + 0.5
580 NEXT N

585 FOR J = 1 TO P2
590 IF B(J) = 0 THEN B(J) = 28
595 IF C(J) = 0 THEN C(J) = 28
600 IF D(J) = 0 THEN D(J) = 28
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
620 B(I2) = 41(1)
625 C(I1) = 50, ..., AND, IN THE DISPLAY OF THE RESULT WE WANT "M" FIRST
630 NEXT I

623 FOR T = P2 TO 1 STEP -1
625 X = 145
630 REM DRAW "A", "E", "C", ..., "M"
640 FOR D = 1 TO 2
645 NEXT D
650 NEXT T

652 REM THE THREE VARIABLES, D(T), C(T), B(T), ARE USED TO DISPLAY THE THREE DIJ
660 OF THE UV,
665 USE OF THE FOR-THE AREA IN THE PICTURE
670 DRAW J + 11 AT X, Y
675 X = X + 6
680 NEXT T

680 DRAW C(T) + 11 AT X, Y
Variant 2 - Processing

Before Running

After Running

JCATALOG

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 MAA
*B 018 MAB
*B 023 MAB
*B 023 MAA
*B 022 MA
*B 022 CT
*B 002 C5
*B 002 RETTAB1
*B 002 CLEAR.OBJO
*B 002 AVELOTAB
*B 002 AVEH1.OBJO
*B 004 AVE.8
*B 002 CLEARD.OBJO
*B 005 AVELOTAB
*B 002 AVEH1B.OBJO
*B 004 AVE.3
*B 004 MAGICSPACE
*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 AVEB.AVE.9
LOCAL CHOICE:

JIST

5 HOME
10 FOR D = 0 TO 120
15 VTAB 3: VTAB 10: INVERSE: PRINT "E E B PROCESSING"
25 VTAB 2: VTAB 15: PRINT "V A R I A N T 2"
30 NEXT D: HOME
35 DEF = CHR$(4)
40 PRINT "WHICH CHANNEL DO YOU WANT TO PROCESS (AVERAGE, HISTOGRAM)?"
45 PRINT
50 PRINT "***************************************************************************"
55 VTAB 12: PRINT "PRESS "L" FOR CHANNEL A"
60 VTAB 13: PRINT "PRESS "2" FOR CHANNEL B"
65 VTAB 14: PRINT "PRESS "J" FOR BOTH CHANNEL A AND B"
70 PCE 16894, 0
75 VTAB 15: GET A6: IF VAL (A6) < > 1 THEN GOTO 93
80 GOTO 119
85 IF VAL (A6) < > 2 THEN GOTO 95
90 GOTO 105
95 IF VAL (A6) < > 3 THEN GOTO 75
100 GOTO 120
105 PRINT D$: PRINT D$: "RUN B-AVERAGE,11"
110 END
115 PRINT D$: PRINT D$: "RUN A-AVERAGE,01"
120 END
125 PCE 16894, 1
130 PRINT D$: PRINT D$: "RUN A-AVERAGE,01"
Variant 3 - Data Acquisition

`ICATALOG
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`
LLOAD CALIBRATE

LLOAD AUTOCALL

LLOAD AUTOCALL1
Variant 3 - Data Processing

ICATALCB

DISK VOLUME 254

+A 005 SPC
+A 004 SFC1
+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 CB
+B 002 RETTAB
+B 002 OPEN.OBJO
+B 002 SAVESC.OBJO
+B 002 SWEEP.OBJO
+B 002 SUTAB
+B 002 RETTAB1
+B 002 CLEAR.OBJO
+B 002 AVELDTAB
+B 002 AVEHI.OBJO
+B 002 CLEARB.OBJO
+B 003 AVELDTABB
+B 002 AVEHIB.OBJO
+B 004 MAGICSSPACE
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
JLOAD SPC

ILIST

5 D4 = CHR$(4)
10 HOME: VTAG 10: INVERSE: PRINT "E E E DATA PROCESSING"
20 HTAB 4: VTAG 15: FLASH: PRINT "LOADING BINARY FILES": NORMAL
25 PRINT D$:"LOAD $CHR$(10),D1": PRINT D$:"LOAD M4A,D1": PRINT D$:"LOAD M4B,D1": PRINT D$:"LOAD H4,C,D1"
30 PRINT D$:"LOAD CT,D1": PRINT D$:"LOAD CS,D1"
35 PRINT D$:"RUN SPCI,D1"

JLOAD SPCI

ILIST

5 D4 = CHR$(4)
10 HOME: VTAG 10: INVERSE: PRINT "D OR SPECTRUMS, 44 FOR EACH CHANNEL, WILL BE 6"
15 ONE NOW AND W"
20 PRINT D$:"LOAD RETTAB,D1": POKE 823, Y = 2: POKE 624, 0: CALL 768
25 PRINT D$:"LOAD OPEX,OBJ1,D1": PRINT D$:"LOAD SAVETO,OBJ1,D1"
30 N = PEEX (- 16225): IN = PEEX (- 16225): PRINT D$:"LOADaw1,OBJ1,D1"
35 FOR X = 0 TO 15: CALL 012
40 POKE 15313, 20$: POKE 15312, 19$: POKE 15314, 16$: POKE 15315, 167
45 POKE 15316, 18$: POKE 15317, 16$: POKE 15318, 14$: POKE 15319, 211
50 POKE 15320, 15$: POKE 15321, 13$: POKE 37372, 4: POKE 37372, 1
55 CALL 045: PRINT D$: POKE 37272, 0: CALL 768
60 POKE 37272, 11: NEXT X
65 PRINT D$:"LOAD SVTAB,B1"
70 POKE 823, Y = 2 + 16: POKE 824, 0: CALL 768: NEXT Y
75 HOME: POKE 49, 0: POKE 4923, 0
80 PRINT D$: PRINT D$:"RUN A-AVERAGE,D1"

JLOAD A-AVERAGE

ILIST

5 D4 = CHR$(4)
10 HOME: VTAG 15: INVERSE: PRINT "B AVERAGES, OF 8 SPECTRUMS EACH, FOR CHAN. A M"
15 WILL BE DONE N"
20 PRINT D$:"LOAD RETTAB,D1"
25 PRINT D$:"LOAD OPEX,OBJ1,D1": CALL 768: PRINT D$:"LOAD AVERAGE,OBJ1,D1"
30 N = PEEX (- 16225): PEEX (- 16225): CALL 768: NEXT Y
35 PRINT D$:"RUN AVERAGE,A,D1"
40 CALL 768: TEXT
45 PRINT D$:"SAVE AVE.", A$ = 300, L4200, D1"
50 NEXT Y
55 PRINT D$:"RUN FINAL-AVE,A,D1"
ILOAD FINAL-AVE,A

LIST

5 REM THIS PROGRAM LOADS CH.A AVERAGES IN SUPPER II, IN THE RIGHT POSITION

10 REM THE POINT

OF USE OF THE H/M AND DOES THE AVERAGE OF B AVERAGES FOR CH.A

15 N = PEEK (-16250):N = PEEK (-16250)

20 FOR Y = 0 TO 7

25 REM 52326=4000

30 I = 53248 = Y * 1024

35 PRINT DB$"LOAD AVE.","V","A","I",DI"

40 REM NOW, SUPPER II CONTAINS B AVERAGES FOR CH.A, FROM 100K-110FF, 1200-123FF

45 PRINT DB$"LOAD CLEAR,0,BIO,0,";CALL 768:PRINT DB$"LOAD AVE,0,BI"

50 PEEK : V#3 IS: INVERSE: PRINT "THE FINAL AVERAGE, OF 44 SPECTRUMS, FOR CH.A

A WILL BE NOW:"

"NORMAL"

55 PRINT DB$"LOAD AVE","1",DB$0,BI"

60 CALL 768:TEXT

65 REM NOW, THE AVERAGE OF B AVERAGES, EACH OF 8 SPEC., OF CH.A, IS LOADED AT B400:

70 PRINT DB$"SAVE AVE.AVE.A,A4000,L#200,DI"

75 PRINT DB$"RUN SEE FIN.AVE.A"

ILOAD SEE FIN.AVE.A

LIST

5 REM THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.A ON THE SCREEN

10 DI = CH. A(43): PRINT DB$"LOAD MARK,D1" PRINT DB$"LOAD MARK,D1" PRINT DB$"LOAD"

15 PRINT DB$"LOAD MARK,D1": PRINT DB$"LOAD CT,D1": PRINT DB$"LOAD MB,D1"

20 PRINT DB$"LOAD AVE.A",A,"A1000,001"

25 POKE 37722;: TEXT 1: HOME

30 POKE 15312,208: POKE 15315,197: POKE 15314,160: POKE 15316,207

35 POKE 15318,181: POKE 15317,180: POKE 15319,164

40 REM POKE TITLE CHARACTER

45 POKE 15321,197: POKE 15320,214: POKE 15321,197: POKE 15322,210

50 POKE 15323,197: POKE 15324,199: POKE 15325,197: POKE 15326,199: POKE 15327,1

55 POKE 15328,182: POKE 15329,180: POKE 15330,141: POKE 15321,150: POKE 15322,1

60 CALL 3024

65 PRINT DB$: POKE 34,0: TEXT : HOME : POKE 168,9,1: REM LOC.16896 IS USED TO R

ECO."

67 REM LOC.16896 IS USED TO R

ECO. WHEN

TO STOP

70 PRINT DB$"RUN AXIS,DI";
LOAD AXIS

LIST

5 DS = CHR$ (4)
10 REM DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BIOGRAPH
15 HLR
20 HCOLOR = 3
25 HPL 15,144 TO 15,0
30 HPL 14.1 TO 16,1: HPL 13,2 TO 17,2
35 HPL 15,144 TO 279,144
40 HPL 277,142 TO 277,144: HPL 278,143 TO 278,145
45 PRINT DI"LOAD MAGICSPACE,DI"
50 POKE 232,0: POKE 233,96
55 SCALE = 1
60 ROT = 0
65 REM ANY CENTER UP=HEX CENTER DOWN
70 REM THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXIS IS 4
75 REM DRAW UP NUMBERS
80 A = 14/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 = 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 160
160 IF N = 4 THEN X = 79
165 NEXT N
170 DATA 12,12,16,17,12,20,10,15
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 205
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 14,12,10,14,15,19,16,16,17,14,10,17
229 REM DRAW DOWN NUMBERS
234 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 Y = 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 49,19,10;18,12,14,10,16,13,14,10,14
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 = 264
370 NEXT N
375 DATA 14,27,10,15,14,10,16,13,14,12,16
390 REM THE DISTANCE BETWEEN THE LETTERS GIVE THE EXPLANATION FOR THE AX
395 IS 18.3
400 REM DRAW "Z" ON THE HORIZONTAL AXIS
405 Y = 260
410 FOR N = 1 TO 2
415 NEXT N
420 DATA 30,48
425 REM MARK THE DIVISIONS ON THE VERTICAL AXIS
425 C = 144 / 240
430 X = 15
440 DIM V(I): DIM I(10)
445 FOR I = 1 TO 5
450 READ S
455 Y(I) = 144 - S * C - 2: REM 2 IS BECAUSE THE M/V WHICH DRAWS THE CHAR. CONSID.
460 R. FROM THE TOP (ONE CHAR HAS 5 VERT. POINTS)
465 Z(I) = Y(I) + 15 + C / (2 * 1)
470 PRINT "S" = S + C
475 PRINT "V" = V(I)
480 PRINT "Z" = Z(I)
485 NEXT I
490 DATA 40,80,120,160,200
495 FOR I = 1 TO 5
490 DRAW S AT X,Y(I)
500 DRAW S AT X,Z(I)
505 NEXT I
510 FOR I = 1 TO 5
515 X = S
520 FOR N = 1 TO 2
525 READ S,L
530 DRAW S AT X,Y(I)
535 DRAW L AT X,Z(I)
540 X = X + 4
545 NEXT I
550 NEXT I
555 REM DATA SHOWN FOR THE POSTER."Z" ON THE VERTICAL AXIS
204

570 X = 20: Y = 2
575 DRAW S AT X, Y
580 X = X + 3
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 = 2 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 LAT
710 REM "HEL-
715 SCALE = 1
720 X = 90: Y = 10
725 FOR N = 1 TO 3
730 READ S
735 DRAW S AT X, Y
740 X = X + 7
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 2
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 57, 9, 18, 12, 10, 12, 49, 30, 29, 9, 25, 30, 10, 23
855 PRINT $"RUN PLOT"-A.51"
LIST

5 D=CHR$ (4)
10 HD TAB 10: VTAB 20: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15 REM CALC. TOTAL POWER
20 S = 0: G1 = 1000: G2 = 101: REM G1 IS FOR THE X AND G2 IS FOR HIST. AMPL. CALC.
25 FOR V = 0 TO 255
30 X = PEEK (16284 + Y)
35 P = X * 2
40 S = S + P
45 NEXT V
50 PRINT "SUM=" + STR$(S)
55 PRINT: PRINT: PRINT "TOTAL POWER=" + STR$(S)
60 REM J.K.T.U ARE NO. OF POINTS FOR THE FREQ. INTERVALS OF THE HISTOGRAM
65 REM 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 N1 = PEEK (16284 + 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) * G1
125 IF N = 8 THEN S1 = 0
130 NEXT Y1
135 A2 = (S1 / S) * G1
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 (16324 + 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) * G1
200 IF N = 15 THEN S2 = 0
205 IF N < 28 THEN GOTO 250
210 IF N = 29 THEN A4 = (S2 / S) * G1
215 IF N = 29 THEN S2 = 0
220 IF N < 42 THEN GOTO 250
225 IF N = 42 THEN A5 = (S2 / S) * G1
230 IF N = 43 THEN S2 = 0
235 IF N < 56 THEN GOTO 250
240 IF N = 56 THEN A6 = (S2 / S) * G1
245 IF N = 57 THEN S2 = 0
250 NEXT Y2
255 A7 = (S2 / S) * G1
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 J3 = A
300 FOR Y = 0 TO 144
305 X3 = PEEK (16384 + 2 * J + S * K + Y3)
310 P2 = X2 + 2
315 S3 = S2 + S2
320 N = N + 1
325 IF N < 29 THEN GOTO 265
330 IF N = 29 THEN A8 = (S3 / S) * 01
335 IF N = 30 THEN S3 = 0
340 IF N < 56 THEN GOTO 265
345 IF N = 56 THEN A9 = (S3 / S) * 01
350 IF N = 57 THEN S3 = 0
355 IF N < 87 THEN GOTO 265
360 IF N = 87 THEN A10 = (S3 / S) * 01
365 IF N = 88 THEN S3 = 0
370 IF N < 116 THEN GOTO 265
375 IF N = 116 THEN S3 = (S3 / S) * 01
380 IF N = 117 THEN S3 = 0
385 NEXT Y2
390 S3 = (S2 / S) * 01
395 PRINT "% OF EIGHTH BAND=",A9
400 PRINT "% OF NINTH BAND=",A9
405 PRINT "% OF TENTH BAND=",01
410 PRINT "% OF ELEVENTH BAND=A2"
415 PRINT "% OF TWELTH BAND=A3"
420 REM CALC: % POWER OF LAST BAND
425 S4 = 0
430 FOR Y = 0 TO 255
435 X4 = PEEK (16384 + 2 * J + S * K + 5 * T + Y4)
440 P4 = X4 + 2
445 S4 = S4 + P4
450 NEXT Y4
455 S4 = (S4 / S) * 125
460 PRINT "% OF THIRTEENTH BAND=A4"
465 PRINT "TOTAL % BEFORE ROUND OFF IS TI=A1+A2+A3+A4+A5+A6+A7+..."
470 B = A9 + B1 + B2 + B3 + B4
475 A1 = INT (A1 * 02) / 02
480 A2 = INT (A2 * 02) / 02
485 A3 = INT (A3 * 02) / 02
490 A4 = INT (A4 * 02) / 02
495 A5 = INT (A5 * 02) / 02
500 A6 = INT (A6 * 02) / 02
505 A7 = INT (A7 * 02) / 02
510 A8 = INT (A8 * 02) / 02
515 A9 = INT (A9 * 02) / 02
520 B1 = INT (B1 * 02) / 02
525 B2 = INT (B2 * 02) / 02
530 B3 = INT (B3 * 02) / 02
535 B4 = INT (B4 * 02) / 02
540 PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
565 PRINT "B4=",B4
570 T = A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4
570 PRINT "TOTAL % POWER IS %":"%"  575 PRINT"%"  580 DIM D(20): DIM A(20)  585 DIM M(20): DIM N(20)  590 DIM F(20)  595 REM CALC. NO. OF Hz/POINT, "%"  600 B = 71.12 / 256  605 PRINT "SL = %"  610 REM CALC. NO. OF POINTS/UNIT %,*"%"  615 P = 144 / 240: P = INT ((P * 0.5) / 0.5) / 0.5  620 REM CALC. PREC. INTERVALS, "%"  625 REM F(1) ARE THE FREQ. OF THE END OF EACH PREC. INTERVAL  630 F(0) = 1 * 5  635 F(1) = 2 * F(0)  640 = F(1)  645 FOR I = 2 TO 5  650 F(I) = F(I) * K * S(I) = F(I)  655 NEXT I  660 E = F(I)  665 FOR I = 7 TO 11  670 F(I) = E + T * S(E) = F(I)  675 NEXT I  680 F(12) = F(I) + U * 5  685 REM D(I) ARE PREC. INTERVALS  690 D(0) = F(I)  695 D = D(I)  700 FOR I = 1 TO 12  705 D(I) = F(I) - D(I) = 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) = A * D2A(1) = A + 3  745 A(I) = A + D2A(3) = A + 4  750 A(I) = A + D2A(6) = A + 6  755 A(I) = A + D2A(7) = A + 8  760 A(I) = A + D2A(9) = A + 10  765 A(I) = A + D2A(11) = A + 12  770 S(I) = S(I) = B  775 FOR I = 0 TO 12  780 D(I) = INT ((D(I) * 0.5) / 0.5) / 0.5  785 PRINT "D(I)=D(I)\"%\"  790 REM CALC. NO. OF POINTS/INTERVAL FOR FREQ. SCALE, "%"  795 D(I) = D(I) / SIN(I) = INT ((D(I) * 0.5) / 0.5) / 0.5  800 PRINT "F(I)=F(I) NEXT I  805 REM CALC. NO. OF POINTS/AMPLITUDE OF HISTOGRAM BLOCK, "%"  810 J(I) = P * H(I) = A(I) = INT ((M(I) * 0.5) / 0.5) / 0.5  815 NEXT J  820 FEED THE HISTOGRAM  825 S = S + 1500 = 0  830 FOR I = 0 TO 12  835 R = R + S(I): PRINT "R(I)="  840 FOR V = 0 TO M(I)  845 HPLT Q(1,44) = Y TO G,144  850 HPLT Q(1,44) = Y TO R,144  855 NEXT Y  860 FOR J = 0 TO M(I)  865 HPLT Q(1,44) = M(I) TO I = 0,444 = M(I)  870 NEXT X  875 C = C + M(I)  880 NEXT I  885 PRINT"%"  890 NEXT I
ILoad MicroBASF
ILIST
5 PRINT CHR$(43);"PRNT"
10 PRINT CHR$(09);"GE"; PRINT CHR$(09);"EDER"
15 PRINT CHR$(41);"PRAO"
20 TEXT
25 HOME
30 S = PEEK (16783)
35 IF S < > 1 GOTO 15
40 FOR E = ASC(41); PRINT CHR$(41);"RUN B-AVERAGE,D1"
45 E = CHR$(71); FOR X = 1 TO 31: PRINT ""S6"
50 FOR Y1 = 1 TO 31: PRINT ""NEXT X1:" FOR Y = 1 TO 4: PRINT ""S8:" NEXT Y1
55 FOR Y2 = 1 TO 31: PRINT ""NEXT Y2:" NEXT X
60 HOME: MTAB 6; VTAB 7; PRINT "INSERT ANOTHER DATA DISK (DRIVE 2)"
65 MTAB 9; VTAB 5; INVERSE: PRINT ""PROCESSING DONE:" 1 NORMAL
70 MTAB 6; VTAB 0; PRINT "RUN AGAIN THE PROGRAM (DRIVE 1) BY" 3
75 MTAB 9; VTAB 11: PRINT "PRESSING"
80 NTAB 15; VTAB 12; INVERSE: PRINT ""PRNT:" 1 NORMAL
85 END

ILoad B-Average
ILIST
5 DI = CHR$(4)
10 HOME: VTAB 15: INVERSE: PRINT "B AVERAGES,OF E SPECTRUMS EACH,FOR CHAn.B W
15 ILL BE DONE N
20 FOR Y = 0 TO 31: PRINT DI""ILoad RetTabl,D1"
25 MOVE 267, Y = 3 = (1) CALL 766
30 PRINT DI""ILoad CLEAR3,GE0,DI:" CALL 766: PRINT DI""ILoad AvelTab3,DI"
35 HOME: VTAB 15: INVERSE: PRINT "B AVERAGES,OF E SPECTRUMS EACH,FOR CHAn.B W
40 ILL BE DONE N
45 ON AND WILL BE SAVED ON THE DISK ""1 NORMAL
50 N = PEEK (-15233): N = PEEK (-15235)
55 PRINT DI""ILoad Avel,D00,DI:"
60 CALL 766: TEXT
65 PRINT DI""ILoad Aver,""Y",A#4830,L#200,DI"
70 NEXT Y
75 PRINT DI""ILoad Final-Ave,D,DI"
JLOAD  F FINAL-AVE.B

JLIST

5 REM THIS PROGRAM LOADS CH.B AVERAGES IN SUPERRAM II, IN THE RIGHT POSITION FOR
ON THE POINT
OF VIEW OF THE M/H AND DOES THE AVERAGE OF 8 AVERAGES FOR CH.B
10 D$ = CHR$(4) 15 N = PEEK(=16295):N = PEEK(=16295)
20 FOR Y = 0 TO 7 25 REM $3744=#2000
30 X = $3760 - Y * $1024 35 PRINT D$:LOAD AVEEL "Y",",A",11"
40 REM NOW, SUPERRAM II CONTAINS 8 AVERAGES FOR CH.B, FROM $2000-$3EFF,$4000-$57
FF,$6000-$7FFFF
F,$8000-$9FFFF,$A000-$BFFFF,$C000-$EFFF
45 PRINT D$:LOAD CLEARB,330,01: CALL 768: PRINT D$:"LOAD AVELOTAGE, D1"
50 HOME: U TAB 15: INVERSE: PRINT "THE FINAL AVERAGE, IF 84 SPEEDLOG, FOR CHN.
B WILL BE DON"
55 REM IN LINE 44 THE HOMESTatement IS USED TO ERASE THE M/H THAT APPEARS ON
THE SCREEN BY
LOADING THE BINARY FILE "AVELOTAGE"
60 PRINT D$:"LOAD AVE1B,330,01"
65 CALL 768: TEXT
70 REM NOW THE AVERAGE OF 8 AVERAGES (EACH OF 8 SPC.) OF CH.B IS LOCATED AT $460
0
75 PRINT D$:"SAVE AVE.AVE.3,44800,L#200,01"
80 PRINT D$:"RUN SEE FIN.AVE.B,01"

JLOAD SEE FIN.AVE.B

JLIST

5 REM THIS PROGRAM OFFERS THE FINAL AVERAGE FOR CH.B ON THE SCREEN
10 D$ = CHR$(141):PRINT D$:"LOAD MAC, D1": PRINT D$:"LOAD MAC, D1": PRINT D$:"SLGA
0 MAC.D1"
15 PRINT D$:"LOAD BFR, D1": PRINT D$:"LOAD CT, D1": PRINT D$:"LOAD CT, D1"
20 PRINT D$:"LOAD AVE.AVE.5,842000,01"
25 POKE 17727,11: TEXT = HOME
30 POKE 15214,192: POKE 15714,192: POKE 15715,267
35 POKE 15716,181: POKE 15217,160: POKE 15216,164
40 REM POKE TITLE CHARACTERS
45 POKE 15219,193: POKE 16220,214: POKE 15721,197: POKE 15722,216
50 POKE 15723,192: POKE 15724,199: POKE 15725,197: POKE 15726,194: POKE 15727,1
55 POKE 15728,182: POKE 15729,180: POKE 15730,141: POKE 15731,155: POKE 15732,1
60 CALL 7064
65 PRINT D$:"RUN AISS, D1"
70 PRINT D$:"RUN AISS, D1"

LOAD AXIS9

LIST
5 DS = CHR$ (4)
10 REM DRAW THE HORIZONTAL AND VERTICAL AXIS OF THE BICREP
15 HDR
20 HCOLOR = 3
25 HPLOT 15,144 TO 15,0
30 HPLOT 14,1 TO 16,11 HPLOT 15,2 TO 17,2
35 HPLOT 12,144 TO 277,144
40 HPLOT 277,145 TO 277,146 HPLOT 278,145 TO 279,145
45 PRINT "DISPLAY MATHMATICIAN,21"
50 POKE 222,0: POKE 233,96
55 SCALES = 1
60 ROT= 0
65 REM AX:V CENTER UP:EY CENTER DOWN
70 REM THE DISTANCE BETWEEN THE NUMBERS WHICH MARK THE AXES 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 = 21
120 NEXT N
125 DATA 49,57,49,49,49,14,10,19
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 12,12,10,17,13,20,10,15
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 X = 4 THEN Y = 180
210 IF N < 4 GOTO 220
215 IF N = 4 THEN X = 239
220 NEXT N
225 DATA 14,12,10,14,15,10,16,17,14,10,17
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 = 27
275 FCS N = 1 TO 12
280 READ S
285 DRAW S AT X, Y
290 X = X + 4
295 IF N < 4 GOTO 315
300 IF N = 4 THEN X = 45
305 IF N < 8 GOTO 315
310 IF N = 8 THEN X = 95
315 NEXT N
320 DATA 49,18,10,19,12,16,10,16,13,14,10,14
325 Y = 151
330 FCS N = 1 TO 12
335 READ S
340 DRAW S AT X, Y
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 14,20,10,15,15,16,10,16,16,10,12,10,12
380 REM THE DISTANCE BETWEEN THE LETTERS WHICH GIVE THE EXPLANATION FOR THE AX
385 IS 18 S
385 REM DRAW H/z ON THE HORIZONTAL AXIS
390 X = 260
395 FCS N = 1 TO 2
400 READ S
405 DRAW S AT X, Y
410 X = X + 5
415 NEXT N
420 DATA 30,40
425 REM MARK THE DIVISIONS ON THE VERTICAL AXIS
430 C = 144 / 240
435 X = X + 15
440 DIM Y(10); DIM Z(10)
445 FOR I = 1 TO Y
450 READ S
455 Y(I) = 144 - S + C - 2; REM 2 IS BECAUSE THE M/L WHICH DRAWS THE CHAR. CONSIDERS THE CHAR.
460 FOR I = 1 TO (ONE CHAR. HAS S VERT. POINTS)
465 PRINT "X=X+S+C"; C
470 PRINT "Y=Y+I(I)"
475 PRINT "Z(Z(I))"=Z(I)
480 NEXT I
485 DATA 40,80,120,160,200
490 FOR I = 1 TO 5
495 DRAW P AT 1,Y(I)
500 DRAW P 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
535 DRAW S AT X,Y(I)
550 DRAW U AT X,Z(I)
560 X = X + 4
565 NEXT N
575 REM DRAW 2 OF DOT. POWER ON THE VERTICAL AXIS
212

270 X = 21 Y = 2
275 DRAW 3 AT X,Y
280 X = X + 5
285 FOR N = 1 TO 2
290 READ S
295 DRAW S AT X,Y
300 X = X + 5
305 NEXT N
310 DATA 37,28
315 X = X + 5
320 FOR N = 1 TO 4
325 READ S
330 DRAW S AT X,Y
335 X = X + 5
340 NEXT N
345 DATA 42,37,42,10
350 X = X + 5
355 FOR N = 1 TO 5
360 READ S
365 DRAW S AT X,Y
370 X = X + 5
375 IF N = 3 THEN X = X + 1
380 NEXT N
385 DATA 39,37,45,27,40
390 REM DRAW TITLE OF THE GRAPH
395 REM THE DISTANCE BETWEEN THE NUMBERS IN THE TITLE IS 5
400 REM THE DISTANCE BETWEEN THE WORDS IN THE TITLE IS 6
405 REM THE DISTANCE BETWEEN THE LETTERS OF THE TITLE IS 7 EXCEPTION IN THE LAST PART OF "HEL..."
410 X = X + 5
415 SCALE = 1
420 I = 207 Y = 10
425 FOR N = 1 TO 3
430 REM DRAW AT X,Y
435 NEXT N
440 DATA 27,27,29
445 X = X + 6
450 FOR N = 1 TO 9
455 READ S
460 DRAW S AT X,Y
465 X = X + 7
470 NEXT N
475 DATA 30,31,41,42,37,29,40,23,35
480 X = X + 6
485 FOR N = 1 TO 5
490 READ S
495 DRAW S AT X,Y
500 X = X + 7
505 NEXT N
510 DATA 28,37,40
515 X = X + 6
520 FOR N = 1 TO 14
525 READ S
530 DRAW S AT X,Y
535 X = X + 5
540 NEXT N
545 DATA 37,9,18,12,10,12,49,30,48,9,25,30,10,24
550 PRINT "RUN PLOT=8,01"
LOAD PLOT-B

LIST

5 DX = CHRI (4)
10 MTRB 101 VYAB 232: INVERSE : PRINT "PLEASE WAIT!": NORMAL
15 REM: CALC TOTAL POWER
20 S = M31 = 100402 = 101: REM O1 IS FOR THE % AND O2 IS FOR HIST.AMPLITUDE:CALC.
25 FOR Y = 0 TO 255
30 X = PFX: (18432 + Y)
35 S = X = 2
40 X = S + P
45 NEXT Y
50 PRINT $"PFX":
55 PRINT : PRINT : PRINT "TOTAL POWER":S
60 FOR J,K:T,U ARE NO. OF POINTS FOR THE FREQUENCY INTERVALS OF THE HISTOGRAM
65 REAC J,K,T,U
70 DATA 7,14,29,27
75 REM: CALC % OF FIRST TWO BANDS
80 N = 0
85 S1 = 0
90 FOR Y1 = 0 TO 2 * J
95 X1 = PFX: (18432 + Y1)
100 P1 = X1 = 2
105 S1 = S1 + P1
110 N = N + 1
115 IF N < 7 THEN GOTO 120
120 IF N = 7 THEN A1 = (S1 / N) * D1
125 IF N = 8 THEN S1 = 0
130 NEXT Y1
135 A2 = (S1 / N) * D1
140 PRINT PRINT "% OF FIRST BAND=A1"
145 PRINT PRINT "% OF SECOND BAND=A2"
150 REM: CALC % OF NEXT 5 BANDS
155 N = 0
160 S2 = 0
165 FOR Y2 = 0 TO 49
170 X2 = PFX: (18432 + 2 * J + Y2)
175 P2 = X2 = 2
180 S2 = S2 + P2
185 N = N + 1
190 IF N < 14 THEN GOTO 250
195 IF N = 14 THEN A5 = (S2 / N) * D1
200 IF N = 15 THEN S2 = 0
205 IF N < 28 THEN GOTO 250
210 IF N = 28 THEN A4 = (S2 / N) * D1
215 IF N = 29 THEN S2 = 0
220 IF N < 42 THEN GOTO 250
225 IF N = 42 THEN A3 = (S2 / N) * D1
230 IF N = 43 THEN S2 = 0
235 IF N < 56 THEN GOTO 250
240 IF N = 56 THEN A2 = (S2 / N) * D1
245 IF N = 57 THEN S2 = 0
250 NEXT Y2
255 A7 = (S2 / N) * D1
260 PRINT PRINT "% OF THIRD BAND=A2"
265 PRINT PRINT "% OF FOURTH BAND=A4"
270 PRINT PRINT "% OF FIFTH BAND=A5"
275 PRINT "% OF SIXTH BAND=";STR$(A6)
280 PRINT "% OF SEVENTH BAND=";STR$(A7)
285 REM CALC. % POWER OF NEXT FIVE BANDS
290 N = 0
295 S3 = 0
300 FOR Y3 = 0 TO 144
305 T3 = 1 + 0.004222 * J + 0.002 * K + Y3
310 T3 = T3 * 2
315 S3 = S3 + T3
320 N = N + 1
325 IF N < 29 THEN GOTO 285
330 IF N = 29 THEN A9 = (S3 / B) + 0.01
335 IF N = 30 THEN S3 = 0
340 IF N < 56 THEN GOTO 285
345 IF N = 56 THEN A9 = (S3 / B) + 0.01
350 IF N = 57 THEN S3 = 0
355 IF N < 87 THEN GOTO 285
360 IF N = 87 THEN B1 = (S3 / B) + 0.01
365 IF N = 87 THEN S3 = 0
370 IF N < 116 THEN GOTO 285
375 IF N = 116 THEN S3 = (S2 / B) + 0.01
380 IF N = 117 THEN S3 = 0
385 NEXT Y3
390 R7 = (S2 / B) + C1
395 PRINT "% OF EIGHTH BAND=";A8
400 PRINT "% OF NINTH BAND=97
405 PRINT "% OF TENTH BAND=91
410 PRINT "% OF ELEVENTH BAND=92
415 PRINT "% OF TWELFTH BAND=93
420 REM CALC. % POWER OF LAST BAND
425 S4 = 0
430 FOR Y4 = 0 TO 26
435 T4 = 66.028522 * 2 + 0.002 * T + Y4
440 T4 = T4 * 2
445 S4 = S4 + T4
450 NEXT Y4
455 T4 = (S4 / B) + 125
460 PRINT "% OF THIRTEENTH BAND=94
465 PRINT "TOTAL % BEFORE ROUND OFF IS";STR$(A1 + A2 + A3 + A4 + A5 + A6 + A7 + A8 + A9 + B1 + B2 + B3 + B4 + B5 + B6 + B7 + B8 + B9)
470 A1 = INT (A1 * 0.05) / 0.05
475 A2 = INT (A2 * 0.05) / 0.05
480 A3 = INT (A3 * 0.05) / 0.05
485 A4 = INT (A4 * 0.05) / 0.05
490 A5 = INT (A5 * 0.05) / 0.05
495 A6 = INT (A6 * 0.05) / 0.05
500 A7 = INT (A7 * 0.05) / 0.05
505 A8 = INT (A8 * 0.05) / 0.05
510 A9 = INT (A9 * 0.05) / 0.05
515 B1 = INT (B1 * 0.05) / 0.05
520 B2 = INT (B2 * 0.05) / 0.05
525 B3 = INT (B3 * 0.05) / 0.05
530 B4 = INT (B4 * 0.05) / 0.05
535 PRINT "% POWER OF THE THIRTEEN BANDS ARE:"
215

570 PRINT "TOTAL % POWER"; IE 7 = "%"
575 PRINT DE"SUM";
580 DIM D1(20); DIM A(20)
585 DIM N(20); DIM M(20)
590 DIM F(20)
595 REM CALC. NO. OF POINTS/UNIT % "P"
600 S = 71.12 / 256
605 PRINT S "=$m"
610 REM CALC. NO. OF POINTS/UNIT % "F"
615 P = 144 / 2401P = INT (P + 0.5) / 22
620 REM CALC. FREQUENCY INTERVALS,"D(1)"
625 REM F(1) ARE THE FREQUENCY INTERVALS, END IN E 7 = 0
630 F(0) = 3 "= B"
635 F(1) = 3 + F(0)
640 F = F(1)
645 FOR I = 2 TO 6
650 E = K * B IF F(I) = F(I)
660 NEXT I
665 REM NEXT I
670 F(I) = E + T * B IF F(I) = F(I)
680 NEXT I
685 REM NEXT I
690 "(1) = U = B"
695 REM NEXT I
700 FOR I = 1 TO 12
705 B(I) = F(I) - D1 = F(I)
710 NEXT I
715 P = 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 * 02A(1) = A2 * 02
745 A(I) = A2 = 02A(0) = A4 + 02
750 A(4) = A5 = 02A(5) = A6 + 02
755 A(6) = A7 = 02A(7) = A8 + 02
760 A(8) = A9 = 02A(9) = A10 + 02
765 A(10) = A2 = 02A(11) = A3 + 02
770 A(12) = B4 = 02
775 FOR I = 0 TO 12
780 D(I) = INT (D(I) + 02) / 02
785 PRINT "D(I)=D(I)"
790 REM CALC. NO. OF POINTS/INTERVAL FOR FREQUENCY SCALE,"(1)"
795 N(I) = D(I) / 222(I) = INT (N(I) + 02) + 0.5 / 22
800 PRINT "N(I)=N(I)"
805 REM NEXT I
810 "(1) = P = A(I)N(I) = INT (P(I) + 02) + 0.5 / 02
815 NEXT I
820 REM DRAW THE HISTOGRAM
825 D = 014 = G
830 FOR I = 0 TO 12
835 R = R + N(I)
840 FOR Y = 0 TO M(I)
845 IF PLOT 0.144 = Y TO 0.144
850 HPLT 0.144 = Y TO R, 0.144
855 NEXT Y
860 FOR X = 9 TO M(I)
865 HPLT 0.144 = X TO 0.144 = 0
APPENDIX G

Machine Language Routines Listing
LOAD AVELD   LOAD AVELOTAB,A$300
            JCCALL-151

END OF DATA

*300LLL

  0300- AD B1 C0   LDA #$081
  0303- AD B1 C0   LDA #$081
  0306- A2 00 00  LDX #00
  0309- B5 00 00  LDA #$00,x
  030B- E0 00 00  CPX #00
  030D- D0 F6 00  BNE #$008
  030F- B0 61 03  LDA #$0361,x
  0313- 95 00 00  STA #00,x
  0317- B0 00 00  CPX #07
  031B- D0 F6 00  BNE #$012
  031E- A2 00 00  LDX #00
  0320- AD 80 C0  LDA #$080
  0324- 20 03 0B  JSR #068
  0328- B1 05 00  LDA ($05),Y
  032B- 6A 00 00  ROR
  032F- 91 00 00  STA ($05),Y
  0333- B1 01 00  LDA ($01),Y
  0337- 64 00 00  ROR
  033B- 91 01 00  STA ($01),Y
  033F- 69 00 00  ADC #00
  0343- CB 00 00  INY
  0347- C0 00 00  CPY #00
  034B- D0 E5 00  BNE #$027
  034F- E6 02 00  INC #02
  0353- E6 06 00  INC #06
  0357- E6 00 00  INC #04
  035B- A5 00 00  LDA #00
  035F- C9 02 00  CMP #02
  0363- D0 D7 00  BNE #$025
  0367- A2 00 00  LDX #00
  036B- BD 00 FF  LDA #$FF00,x
  036F- 95 00 00  STA #00,x
  0373- 55 00 00  CPY #00
  0377- E0 00 00  CPX #00
  037B- D0 F6 00  BNE #$030
  037F- AD 82 C0  LDA #$082
  0383- AD 82 C0  LDA #$082
  0387- 60 00 00  RTS
  038B- 00 00 00  BRK
  038F- 40 00 00  RTI
  0393- 00 00 00  BRK
  0397- 00 00 00  BRK
  039B- 44 00 00  ???
  039F- 00 00 00  LDX #00
  03A3- 02 00 00  INC #$003,x
  03A7- 00 00 00  BNE #$032
  03AC- B0 B0 00  BCS #$033
220

*BLGAD AVELGATBB,A$300 00 01 EF DB 00
*300LLL

0300- AD B1 C0 LDA $COB1
0303- AD B1 C0 LDA $COB1
0306- A2 00 LDX #00
0308- B5 00 LDA $00,X
030A- 9D 00 FF STA $FF00,X
030D- EB INX
030E- E0 00 CPX #00
0310- D0 F6 BNE $0308
0312- BD 61 03 LDA $03B1,X
0315- 98 00 STA $00,X
0317- EB INX
0318- E0 07 CPX #07
031A- D0 F6 BNE $0312
031C- AD B0 C0 LDA $C0B0
031F- AD B0 C0 LDA $C0B0
0322- 20 68 03 JBR $026B
0325- A0 00 LDY #00
0327- A2 00 LDX #00
0329- 1B 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- EB INX
0336- E0 03 CPX #03
0337- D0 F0 BNE $0329
0339- 69 00 ADC #00
033B- 91 01 STA ($01),Y
033D- CB 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 LDY #00
0350- BD 00 FF LDA $FF00,X
0353- 95 00 STA #00,X
0355- EB INX
0356- E0 00 CPX #00
0358- D0 F6 BNE $0350
035A- AD B2 C0 LDA $C0B2
035D- AD B2 C0 LDA $C0B2
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 CMA ($00,X)
036F- 00 BRK
LOAD SAVESFC

END OF DATA

#L

1 BUFF EQU $D000
2 SWEEP EQU $1000
3 COUNT EQU $0343
4 BUFFA EQU $0314
5 SWEEPA EQU $0311
6 WDN EQU $C081
7 OFF EQU $C082
8 ORG EQU $0300
9 BEG LDA #$10
10 STA SWEEPA
11 LDA WDN
12 LDA WDN
13 LDX #$00
14 LDY #$00
15 START LDA SWEEP,X
16 STA BUFF,X
17 INX
18 CPX #$00
19 BNE START
20 INC SWEEPA
21 INC BUFFA
22 INY
23 CPY #$02
24 BNE START
25 LDA OFF
26 LDA OFF
27 RTS

LOAD SWEEP

END OF DATA

#L

1 BUFF EQU $D000
2 SWEEP EQU $1000
3 BUFFA EQU $F004
4 SWEEPA EQU $F009
5 RDN EQU $C080
6 ADD EQU $347
7 OFF EQU $C082
8 ORG EQU $F000
9 LDY #$00
10 LDY #$00
11 START LDA BUFF,X
12 STA SWEEP,X
13 INX
14 CPY #$00
15 BNE START
16 MID CLC
17 LDA BUFFA
18 ADC #$01
19 STA #$53
20 JSR ADD
21 INY
22 CPY #$02
23 BNE START
24 STOP RTS
LOAD CLEAR

END OF DATA

.L

1 LOW EDU #4000 0300- A0 00  LDY #4000
2 ORG #300 0301- A2 00  LDX #4000
3 LDY #00 0304- A9 00  LDA #4000
4 LDX #00 0306- 9D 00 40 STA #4000,X
5 LOOP LDA #00 0309- EB 00  INX
6 STA LDX,X 030A- E0 00  CPX #4000
7 INX 030C- D0 F6  BNE #0304
8 CPX #00 030E- CB 00  INY
9 BNE LOOP 0310- C0 02  CPY #402
10 INY 0311- F0 06  BEO #0219
11 CPY #02 0313- EE 0E 03  INC #00B6
12 BEQ END1 0316- 4C 04 03  JMP #0304
13 INC #30B 0319- C0 04  CPY #30B4
14 JMP LOOP 031B- F0 0D  BEO #032A
15 END1 CPY #04 031D- A9 04  LDA #3050
16 BEQ END 031F- BD 10 03 STA #3040
17 LDA #308 0321- A9 44  LDA #308
18 STA #310 0323- 8D 08 03 STA #30B8
19 LDA #344 0325- 4C 04 03 JMP #0304
20 STA #308 0327- 60  RTS
21 JMP LOOP 0329- 02  ???
22 END RTS 032A- AD 82 00 LDA #CDE2
23 032B- AD 82 00 LDA #CDE2
24 032C- AD 82 00 LDA #CDE2
25 032D- AD 82 00 LDA #CDE2
26 032E- AD 82 00 LDA #CDE2

---

LOAD SWEEP.OBJO,A$300 00 01 EF DB 00

.L

0300- A2 00  LDX #4000
0302- A0 00  LDY #4000
0304- BD 00 00 LDA #D000,X
0307- 9D 00 10 STA #1000,X
030A- EB 00  INX
0309- E0 00  CPX #4000
030C- D0 F6  BNE #0304
030F- 1B  CLC
0310- AD 0A 01  LDA #F000
0313- 69 01  ADC #001
0315- BD 03 03 STA #03B3
0318- 20 47 03 JSR #0347
031B- CB 00  INY
031C- CO 02  CPY #402
031E- D0 EA  BNE #0304
0320- 60  RTS
0322- C0 02  CPY #402
0323- D0 EA  BNE #030F
0325- AD 82 00 LDA #C82
032B- AD 82 00 LDA #C82
032C- 60  RTS
032E- 80  ???

---
LOAD HIALL

END OF DATA

1  ORG #F000
2  TRACK EQU #348
3  SECTOR EQU #34C
4  DTH EQU #350
5  BUFF EQU #DE00
6  SWEEP EQU #1000
7  COUNT EQU #346
8  BUFFA EQU #0314
9  SWEEPA EQU #0311
10  NON EQU #C081
11  RNK EQU #C080
12  OFF EQU #C082
13  IBD EQU #03
14  IGA EQU #47
15  STOREA EQU #F000
16  RNTS EQU #D9
17  STAR LDA #10B
18  LDB #10A
19  JSR RNTS
20  INC SECTOR
21  LDA #10
22  CMP SECTOR
23  BNE DATA
24  LDX #00
25  STX SECTOR
26  LDA #44
27  INC TRACK
28  CMP TRACK
29  BNE DATA
30  BRK
31  DATA INC DTH
32  LDA #F0
33  CMP DTH
34  BEQ STOP
35  JMP STAR
36  STOP LDA #D0
37  STA DTH
38  STA BUFFA
39  LDA #00
40  STA COUNT
41  RTS

#LOAD HIALL.6BC0,A#300 00 01 FF 9B 00

#300LL

1 LDA #03
2 LDA #47
3 LDA #03
4 LDA #03
5 LDA #03
LOAD CLEARB

END OF DATA

1 LOW EQU $4800
2 ORG #300
3 LDY #00 0300- A0 00 LDY #00
4 LDX #00 0302- A2 00 LDX #00
5 LOOP LDA #00 0304- A9 00 LDA #00
6 STA LOW,X 0306- 9D 00 4B STA #4800,X
7 INX 0309- E8 INX
8 CPX #00 030A- E0 00 CPX #00
9 BNE LOOP 030C- D0 F6 BNE #0304
10 INV 030E- CB INV
11 CPY #02 030F- C0 02 CPY #02
12 BEQ END1 0311- F0 04 BEQ #0319
13 INC $308 0313- EE 08 03 INC #0308
14 JMP LOOP 0316- 4C 04 03 JMP #0304
15 END1 CPY #04 0319- C0 04 CPY #04
16 BEQ END 031B- F0 02 BEQ #032A
17 LDA #04 031D- A9 04 LDA #04
18 STA $#10 031F- 8D 10 03 STA #0310
19 LDA #4C 0322- A9 4C LDA #4C
20 STA #308 0324- 6D 08 03 STA #0308
21 JMP LOOP 0327- 4C 04 03 JMP #0304
22 END RTS 032A- 60 RTS
ORG $300

*LOAD LOWALLTAB,A$300 00 00 01 EF DB 00

*300LL

0300- AD B1 00 LDA #C081
0303- AD B1 00 LDA #C081
0306- A9 10 LDA #$10
0309- BD 11 03 STA #0511
030C- A0 00 LDY #$00
030F- A2 00 LDY #$00
0312- BD 00 10 LDA #$1000,X
0315- 9D 00 00 STA #$D000,X
0318- E8 INX
031B- ED 00 CPX #$00
031E- FO 03 BEQ $0312
0321- 4C 0F 03 JMP #030F
0324- EE 11 03 INC #0311
0327- EE 14 03 INC #0314
032A- CB INY
032D- C0 04 CPY #$04
032E- F0 03 BEQ $0329
0331- 4C 0F 03 JMP #030F
0334- EE 46 03 INC #$346
0337- A9 08 LDA #$08
033A- C3 46 03 CMP #$346
033D- F0 01 BEQ #0336
033E- 60 RTS
0341- AD 80 CO LDA #$C050
0344- AD 80 CO LDA #$C080
0347- 20 00 FO JBR #$F000
034A- AD 82 CO LDA #$C082
034D- 60 RTS
034E- 00 BRK
0351- 01 60 QRA ($60,X)
0354- 02 ???
0357- 00 BRK
035A- 00 BRK
035D- 00 BRK
0360- 58 CLI
0363- 03 ???
0366- 00 BRK
0369- 00 BRK
036C- 00 BRK
036F- 00 BRK
0372- 00 BRK
LOAD RET2

END OF DATA

1 ORG $300
2 IDB EQU $03
3 IDA EQU $33
4 RWTS EQU $309
5 DTH EQU $33C
6 TRACK EQU $337
7 SECTOR EQU $33B
8 WGN EQU $C0B1
9 WOFF EQU $C0B2
10 LDA WGN
11 LDA WGN
12 START LDA #1DB
13 LDY #1DA
14 JBR RWTS
15 UP INC DTH
16 INC SECTOR
17 LDX #$10
18 CPX SECTOR
19 BNE NIN
20 LDX #$100
21 STX SECTOR
22 INC TRACK
23 NIN LDX #$FO
24 CPX DTH
25 BEQ DONE
26 JMP START
27 DONE LDA WOFF
28 LDA WOFF
29 RTS

$LOAD RETTAB, A$300 00 01 SF DB 00
$300LL
LOAD OPEN

END OF DATA

* L

1 WON EQU #C081
2 RON EQU #C080
3 SWEHPI EQU #F000
4 OFF EQU #C082
5 BUFA EQU #F006
6 SLEHA EQU #F009
7 ORG #003C
8 OPEN LDA RON
9 LDA RON
10 JBR SWEHPI
11 LDA WON
12 LDA WON
13 LDA #10
14 STA SLEHA
15 LDA OFF
16 LDA OFF
17 RTS
18 LDA WON
19 LDA WON
20 LDA #11
21 STA SLEHA
22 LDA #00
23 STA BUFA
24 LDA RON
25 LDA RON
26 RTS

#LOAD OPEN,OBJ0,AD300 00 01 EF 38 00

#300LL

0300- AD 80 00 LDA #C080
0303- AD 80 00 LDA #C080
0306- 20 00 F0 JBR #F000
0309- AD 81 00 LDA #C081
030C- AD 81 00 LDA #C081
030F- A9 10 LDA #F10
0311- BD 09 F0 STA #F007
0314- AD 82 00 LDA #C082
0317- AD 82 00 LDA #C082
031A- 60 RTS
0318- AD 81 00 LDA #C081
031E- AD 81 00 LDA #C081
0321- A9 11 LDA #11
0325- BD 09 F0 STA #F009
0328- A9 00 LDA #00
032B- BD 04 F0 STA #F006
032E- AD 80 00 LDA #C080
0331- 60 RTS
0332- 60 RTS
0333- 01 60 CRA ($60,X)
0336- 02 ???
0339- 00 BRK
033C- 00 BRK
033F- 44 ???
033A- 03 ???
0340- 00 BRK
0343- 00 BRK
0346- 00 BRK
0349- 00 BRK
034C- 00 BRK
034F- 00 BRK
0352- 00 BRK
0355- 00 BRK
0358- 00 BRK
035B- 00 BRK
035E- 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".


6) Applesoft II BASIC programming reference manual.


9) Apple DOS manual.

10) Apple 6502 assembler - Editor