

This is to certify that the
dissertation entitled

A Microcomputer Controlled Automatic Recording
Spectropolarimeter Using a Linear Diode Array Detector
presented by

Peter Joseph Aiello

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Chemistry

A handwritten signature in cursive script, reading "C. G. Enke".

Major professor

Date 8/25/83



RETURNING MATERIALS:
Place in book drop to
remove this checkout from
your record. FINES will
be charged if book is
returned after the date
stamped below.

DO NOT CIRCULATE

ROOM USE ONLY

**A MICROCOMPUTER CONTROLLED
AUTOMATIC RECORDING SPECTROPOLARIMETER
USING A LINEAR DIODE ARRAY DETECTOR**

By

Peter Joseph Aiello

A DISSERTATION

**Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of**

DOCTOR OF PHILOSOPHY

Department of Chemistry

1983

ABSTRACT

A MICROCOMPUTER CONTROLLED AUTOMATIC RECORDING SPECTROPOLARIMETER USING A LINEAR DIODE ARRAY DETECTOR

By

Peter Joseph Aiello

A microcomputer controlled system for the simultaneous measurement of absorbance and optical rotatory dispersion has been developed. The system also incorporates an automatic optimization of integration time method to improve the dynamic range of the detector. The instrument uses a linear diode array detector for simultaneous multiwavelength detection. The array used has 512 individual light sensitive diodes each of which is sensitive to electromagnetic radiation with wavelengths from 200-1000 nm.

Photodiode detectors have a limited dynamic range relative to a photomultiplier tube. Sensitivity (in terms of intensity) can be adjusted by changing the integration time, but the integration time for all diodes in the array detector is the same for any given scan, and all portions of a spectrum are acquired within the same limited dynamic range. However, the results of scans taken at varying integration times can be combined to produce a data set with improved dynamic range. A microcomputer controlled system

for the automatic sequencing of detector integration times and the storage of only the optimum readings is demonstrated. An improvement in dynamic range of 215 times that for a single integration time is theoretically possible, but in most systems, stray light and dark current will limit the practical dynamic range attainable.

The inlet optics allow the simultaneous measurement of absorbance and optical rotatory dispersion (ORD) and consist of a light source, 2 polarizers (one of which is rotated under computer control), a sample cell and a focusing lens. By collecting successive spectra at different angular positions of the polarizers the absorbance and ORD can be calculated. The systems studied include the inversion of sucrose and Vitamin B₁₂.

ACKNOWLEDGMENTS

My sincere appreciation goes to Professor Chris Enke for the help and encouragement that he has given me during the course of my research here at Michigan State University. I would also like to thank Professor Stan Crouch for many invaluable discussions

I would like to thank Michigan State University, Abbott Laboratories and the Office of Naval Research for the financial support they provided.

I would like to thank the members of Professor Enke's research group for their help and friendship, especially Hugh Gregg, Bruce Newcome and Phil Hoffman.

Finally, I offer my thanks and appreciation to my wife, Pat, for her continuous love and support.

-

TABLE OF CONTENTS

Chapter

LIST OF TABLES.	v
LIST OF FIGURES	vi
CHAPTER I. INTRODUCTION.	1
A. The History of Imaging Type Detectors.	1
B. Optical Rotatory Dispersion and Circular Dichroism	9
C. Thesis Overview.	12
CHAPTER II. AN OVERVIEW OF THE INSTRUMENT.	14
A. Introduction	14
B. The Optical Design	14
C. The Data Acquisition System.	20
D. Data Acquisition, Reduction and Analysis	25
CHAPTER III. CHARACTERIZATION OF THE LDA SPECTROMETER, CONTINUED.	28
A. Introduction	28
B. The Effect of Cooling the Linear Diode Array	28
C. Saturation Effects	31
D. The Effect of Arc Lamp Stabilization	35
CHAPTER IV. AUTOMATIC OPTIMIZATION OF INTEGRATION TIME	37
A. Introduction	37
B. Why Vary the Integration Time?	38
C. Programmed Sequence Control of Integration Time.	40
D. Total Dynamic Range for the Instrument	43
E. Data Storage	44

Chapter	Page
F. Performance	47
CHAPTER V. THE DATA ACQUISITION SYSTEM	58
A. Introduction	58
B. Hardware	58
C. Software	67
CHAPTER VI. THE SIMULTANEOUS MEASUREMENT OF ABSORBANCE AND ORD.	78
A. Introduction	78
B. The Experimental Technique	78
C. The Hydrolysis of Sucrose.	82
D. Vitamin B12	88
E. Conclusion	93
APPENDIX A: INTERFACE SCHEMATICS AND TIMING DIAGRAMS . .	95
APPENDIX B: SELECTED PROGRAM LISTINGS.	101
A. FORTH Documentation.	101
B. PDP 11/40 Software	114
REFERENCES.	139

—

LIST OF TABLES

Figure		Page
TABLE 3.1	Percent Lag for 4 Consecutive Scans after Saturation Condition.	34
TABLE 5.1	Commonly used FORTH Words for Instrument Operation.	71
TABLE 5.2	Data Reduction and Analysis Routines. . . .	75
TABLE 6.1	Observed Precision of the Phase Information produced by the Curve Fitting Procedure	81

LIST OF FIGURES

Figure		Page
FIG. 2.1	A Block Diagram of the Optical System. . . .	15
FIG. 2.2	An Approximate drawing of the Spectrograph	18
FIG. 2.3	A Block Diagram of the Microcomputer	21
FIG. 2.4	A Block Diagram of the Data Acquisition Circuitry.	23
FIG. 3.1	Dark Current versus Temperature at a 1 s Integration Time	30
FIG. 3.2	Dark Current versus Integration Time at Temperatures of 25 and -2 degrees C.	32
FIG. 4.1	Q.I. Lamp Spectra at varying integration times.	39
FIG. 4.2	Integration Time Control Sequence.	41
FIG. 4.3	Data and Integration Time Code Storage. In this illustration, a minimum integration time of 3 ms is used	46
FIG. 4.4	Hg Lamp Spectrum at a 25 ms integration time	48
FIG. 4.5	Enlargement of the Hg Lamp Spectrum at a 25 ms integration time	49
FIG. 4.6	Hg Lamp Spectrum at a 1 s integration time	51
FIG. 4.7	Hg Lamp Spectrum acquired with an initial integration time of 15 ms with 6 integration time doublings (7 readings).	52
FIG. 4.8	Enlargement of FIG. 4.7	53
FIG. 4.9	Dynamic Range Measurement using Neutral Density Filters.	55
FIG. 4.10	Dynamic Range Measurement using known KMnO4 solutions.	56
FIG. 5.1	A Block Diagram of the Microcomputer System and Linear Diode Array Interface.	59

Figure		Page
FIG. 5.2	A Block Diagram of the Integration Time Control Circuit.	61
FIG. 5.3	A Block Diagram of the ADC Board	65
FIG. 6.1	The Absorbance and Rotational Spectra of Sucrose.	83
FIG. 6.2	The Hydrolysis of Sucrose, acid concentration 3.0 M.	85
FIG. 6.3	The Hydrolysis of Sucrose, acid concentration 1.5 M.	86
FIG. 6.4	The Rate of Hydrolysis of Sucrose at acid concentrations of 3.0 and 1.5 M versus wavelength.	87
FIG. 6.5	The Absorbance Spectra of Vitamin B12 at concentrations of 16.4, 11.5, 8.2, and 4.9 mg/ml.	90
FIG. 6.6	The Rotational Spectra of Vitamin B12 at concentrations of 16.4, 11.5, 8.2, and 4.9 mg/ml.	91
FIG. 6.7	Absorbance and Rotational Working Curves of Vitamin B12 at 355 nm	92
FIG. A.1	ADC Board Schematic Diagram.	95
FIG. A.2	Schematic Diagram of the Address Control Circuit.	96
FIG. A.3	Schematic Diagram of the Integration Time Control Circuit	97
FIG. A.4	Schematic Diagram of the Data Acquisition Control Circuit.	98
FIG. A.5	Single Scan Timing Diagram	99
FIG. A.6	Multiple Scan Timing Diagram	100

CHAPTER I

INTRODUCTION

A. The History of Imaging Type Detectors

Over the past decade, there has been considerable development in imaging type detectors for the measurement of ultra-violet (UV) and visible light. These new detectors have attracted the interest of a number of analytical spectroscopists. Traditionally, the analytical chemist has used such instruments as the photometer, which uses a narrow-band light source (for example, the 254 nm emission line from a low pressure Hg lamp or a continuous source and a selective filter), a sample cell and a photomultiplier tube (PMT) detector. While useful for many specific applications, the photometer cannot determine multiple sample components simultaneously or provide general absorbance characterization of the system. When information at multiple wavelengths is desired, a continuous source (such as a tungsten lamp) and a dispersive element (a prism or grating) is used. The dispersive element is mechanically rotated to vary the wavelength of light passed through a fixed exit slit. This selected monochromatic light beam then passes through the sample cell and is detected by a PMT. Spectrometers of this type maintain the single photometer's characteristics of wide dynamic range of

absorbance, good sensitivity and rapid, linear response. In addition, they provide a continuous variation in the wavelength sampled with a relatively high degree of resolution in wavelength selection. Despite the great convenience and analytical power of the scanning spectrometer, it has two characteristics which are limiting in a number of areas of application. First, by sampling over only a narrow range of the dispersed light at any given time, it makes inefficient use of the optical information available and thus it prolongs the necessary measurement time. Second, because the wavelength must be physically scanned to provide measurements at multiple wavelengths, this data is acquired at different times. This limits its applicability in situations where the sample's optical properties are changing, particularly if the rate of such changes is an object of the measurement.

An obvious improvement in both these limitations could be achieved with simultaneous multiple wavelength detection. Of course, one would like to have such capability without sacrificing high resolution of wavelength selection, wide dynamic range and good sensitivity at all wavelengths, linear response, good geometric stability, no stray light, rapid response and rapid electronic readout. Unfortunately, no currently available multi-wavelength spectrometer has all these desirable properties.

Several spectrometers have been developed which use

multiple detectors arranged in the focal plane of the dispersing element, thus achieving multi-wavelength detection. Multiple detectors are available in a variety of types and spatial geometries. These types of detectors include: the photographic plate, multiple PMT's (direct reader), the image dissector, the silicon vidicon, the charge-coupled device, the charge-injection device, and the photodiode array. The number of photosensitive elements in these detectors can vary from just a few to many thousands, and these elements can be arranged linearly or in a two-dimensional array. As the number of elements increases, greater wavelength resolution is possible. When a linear array is placed in the focal plane, each element detects a different wavelength region. This is also true for two-dimensional arrays as the spectrometer entrance slit image is focused on a separate row of detectors for each wavelength region, and the signal from all elements in each row are averaged together. The two-dimensional array, however, has another advantage in that it can be used for two-dimensional imaging applications such as Eschelle grating and streak camera images. Both linear and two-dimensional arrays are available with photosensitive elements of various dimensions. In general, the larger the element, the greater the dynamic range; the smaller the element, the greater the resolution. The ideal element dimensions would match those of the spectrometer slit, i.e.

narrow and tall for the best resolution and dynamic range. The following is a discussion of a few types of multi-channel detectors which includes a summary of their advantages and disadvantages.

The first multi-wavelength detector used was the photographic plate. It has several advantages: relatively easy channel identification, simple operation, low-cost, photometric integration and physical dimensions to suit a wide variety of spectrometer geometries. Its disadvantages include limited dynamic range, non-linear response, difficult calibration and a very slow data retrieval procedure.

Another type of multichannel detector in use is called the direct reader. This instrument uses multiple PMT's arranged across the focal plane of a polychromator (1). Many desirable characteristics are maintained through the use of the PMT detectors; however the number of channels for a reasonably sized instrument is severely limited (usually < 10) and the PMT's must be arranged at wavelengths appropriate for predetermined specific applications.

An early electronic scanning spectrometer utilized an image dissector as the detector (2). An image dissector is nothing more than an electron multiplier which is sensitive to a small, electronically selectable region of a relatively large photocathode. The image dissector tube offers advantages such as high resolution, the absence of

broadening effects of intense spectral lines and the capability of single photon counting (3). However, even though electronic wavelength scanning can be much faster than mechanical scanning, the image dissector does not provide simultaneous wavelength detection and light impinging on the unsampled area of the photocathode is lost. The single largest application of image dissector tubes is the measurement of low-light levels in astronomy (4).

The silicon vidicon (SV) has been widely used as a spectrometric detector. The development and characterization of vidicon spectrometers has been described in many recent papers (5-11). The light-sensitive surface of a vidicon tube is a two dimensional array of up to 1000 by 1000 pixels (picture elements). Absorbed photons reduce the charge stored in the pixel. The charge is restored by an electron beam focused on and scanned across the reverse side of the diode surface. The measured charging current at each pixel is proportional to the charge lost which is in turn related to the light intensity integrated over the interval between scans. Advantages of the SV include high scanning speed and large number of pixels. Disadvantages are many, however. Dynamic range is limited (<1000), blooming (adjacent pixel cross-talk) causes spectral line broadening, UV sensitivity is low and incomplete readout or lag is nearly 10% of the charge. For low light level spectroscopy, a silicon intensified target tube (SIT) is

used. In the visible region of the spectrum, spectral response and sensitivity are comparable to those of a PMT; however, its sensitivity is 1 to 2 orders of magnitude lower in the UV region (12). The SIT has been compared to the PMT for the measurement of transient fluorescent signals and has been proven to provide comparable signal to noise ratios (S/N) (13,14).

A charge-coupled device (CCD) is a solid state imaging device. CCD's contain up to 2000 pixels in a two-dimensional array which provides relatively high resolution. CCD's are also less expensive than other solid state imaging devices (15,16). High scan speeds are possible, the response is linear, and the dynamic range is about 1000. One significant feature of this detector is that all elements of the array integrate intensity during exactly the same time frame. The integration period is terminated by simultaneously transferring the integrated intensity signals to an analog shift register from which they are subsequently read out serially. Unfortunately, blooming and lag do occur if saturation is attained. Sensitivity is below that of a typical silicon device in the UV region of the spectrum.

The charge-injection device (CID), also a solid state imaging device, has several unique features (17). One of these is its capability of nondestructive readout. With this capability, blooming (still a problem if saturation is

attained) can be avoided by periodically scanning the array in a nondestructive readout mode. This mode can determine which pixels are near saturation. These pixels can then be selectively sampled in the normal destructive mode so that saturation is avoided while adjacent pixels with low incident light intensity can continue integrating. Another unique capability is that the integrated circuit is fabricated to allow random addressing of pixels. Therefore, even faster readout speeds can be achieved when only a subset of the pixel information is required for a particular application.

The characteristics of silicon photodiode arrays (SPD) have been discussed in many recent papers (18-25). Over the last few years, self-scanned linear SPD arrays have been used for a number of analytical spectrochemical measurements (26-34). These arrays are currently available with up to 4096 photodiodes. It has been shown that SPD arrays have superior blooming and lag performance when compared to most other imaging type detectors (20,23). This allows the signal integrating capability of the array to become a very powerful asset. Since blooming does not occur, one can allow several photodiodes to saturate while adjacent photodiodes can integrate low light level signals.

E.G. and G. Reticon (35) has made available photodiode arrays specifically designed for spectroscopy. These arrays are self-scanned and provide real-time

electronic readout with up to 1024 diodes. Each photodiode is a slit-shaped 25 μm wide by 2.5 mm high. This relatively large active area gives a dynamic range of up to 10,000 and the narrow width allows for excellent wavelength resolution. Although the arrays are much less sensitive than the PMT, useful measurements can be made over a region from 200 to 1000 nm. When improved sensitivity is needed, an electron multiplier type image intensifier can be used. These image intensifiers use an array of electron multipliers in a structure called a microchannel plate (MCP). Each channel in the MCP provides a gain of up to six orders of magnitude (36) and the density of these channels often determines the spatial resolution of the detector. Complete MCP's can be stacked to provide even higher gains.

For response in the vacuum ultra-violet spectral region (50-200 nm) a SSANACON, self-scanned anode array with microchannel plate electron multiplier, has been used (37). This involves photoelectron multiplication through two MCP's, collection of the electrons directly on aluminum anodes and readout with standard diode array circuitry. In cases where analyte concentrations are well above conventional detection limits, multi-element analysis with multi-channel detectors by atomic emission has been demonstrated to be quite feasible (12,38). Spectral source profiling has also been done with photodiode arrays (28,30,32). In molecular spectrometry, imaging type

detectors have been used in spectrophotometry, spectrofluorometry and chemiluminescence (24,25,27,34). These detectors are often employed to monitor the output from an HPLC or GC (14,39,40,41).

B. Optical Rotatory Dispersion and Circular Dichroism

Many chemical substances rotate the plane of polarized light and are therefore described as being optically active. Optical rotatory power has its origin in any structural asymmetry in the substance. This may be inherent in the structure of the molecule (for example, the presence of a chiral carbon atom) or it may be a property of the crystalline form of the substance (as in quartz).

Normally, light waves are oriented in all possible angles normal to the light path. Linearly polarized light is produced by passing light through a polarizer (such as a prism or film) which passes only a single orientation of the light waves. This plane-polarized light can be thought of as the sum of left and right circularly polarized light. Left circularly polarized light has its electric vector rotating counter-clockwise along the direction of the wave (right circularly polarized light rotates clockwise). Optically active substances affect left and right circularly polarized light in two distinct ways.

First, it is known that these substances have different

refractive indices for left and right circularly polarized light. When this occurs, the speed of the left circularly polarized light through the material is different than that of the right circularly polarized light. The result is the rotation of the plane of the incident linearly polarized light. Since refractive index is a function of wavelength, the rotation of the linear polarized light is also a function of wavelength. The curve which results from plotting the number of degrees of rotation versus wavelength is called the optical rotatory dispersion (ORD) spectrum. This characteristic curve gives both qualitative and quantitative information about the substance.

Second, optically active substances have different molar absorptivity coefficients for left and right circularly polarized light. This differential absorbance causes the light to become elliptically polarized. A Circular Dichroism (CD) spectrum is a plot of the degree of ellipticity versus wavelength.

Theoretical relationships exist between ORD and CD which permit calculating one curve from the other. These relationships are called the Kronig-Kramers transformations (52,53). This means that in principle no additional information can be obtained about a molecule by measuring its ORD spectrum if its CD spectrum is known or vice versa. However, there are practical considerations which sometimes make it advantageous to measure one or the other. For

example, the ORD of a molecule can be measured at wavelengths quite different from the optically active absorption band causing it. Such measurements in the visible region can be used to determine concentrations, purity, rates of reaction, etc., but basically what is being studied is the CD of an absorption band at a shorter wavelength. CD has obvious advantages over ORD in that relatively weak transitions can be identified by CD, but may often be masked by ORD. The higher resolution of CD bands makes it superior to ORD, particularly in the structural studies of complex molecules. These can possess many asymmetric elements which give rise to many optically active electronic transitions (58-61).

The photoelectric polarimeter was the first instrument developed for the measurement of optical rotation (54,55). This instrument uses a sodium lamp and a polarizer to produce a linearly polarized monochromatic light beam. This beam then passes through the sample cell and a second polarizing plate called the analyzer. The analyzer is rotated until the transmitted light intensity is at a minimum. The number of degrees of rotation is then read from a calibrated scale. In order to measure ORD, the sodium lamp is replaced with a continuous source and a monochromator. This is called a spectropolarimeter. Several commercial instruments are available and some of the more recent of these have been put under microcomputer

control (62,63).

If the first polarizer is replaced by a device which alternately generates left and right circularly polarized light (such as a Pockels cell), CD can be measured. By measuring the differential absorbance of the left and right circularly polarized light, the molar ellipticity or CD can be calculated. Several commercial circular dichrometers are currently available and are capable of high-precision measurements over a wide region of the spectrum. However, there is still uncertainty concerning the calibration of these instruments because of the lack of a universally accepted standard. Recently compounds such as 10-camphorsulfonic acid and tris(ethylenediamine)-cobalt(III)iodate have been suggested (64-72).

C. Thesis Overview

An instrument was developed that uses a linear photodiode array detector (LDA) for the simultaneous measurement of absorbance and optical rotatory dispersion (ORD). Through the use of the array detector, both spectra can be acquired at high speed, which allows the monitoring of kinetic systems.

Chapter II presents a general overview of the instrument. Several important features of the spectrometer are described including the inlet optics, polychromator,

array detector, microcomputer and software. Chapter III is a continuation of the diode array characterization begun in my M.S. thesis (22). The effect of temperature on the detector dark current and of the linearity of the dark current versus integration time is studied as the detector can now be cooled to a minimum temperature of -5° C. Chapter IV discusses a means by which the detector integration time is used to maximize dynamic range. When using any type of array detector, dynamic range is a problem. Since no single integration time is optimum for every wavelength, a programmed sequence of increasing integration times is implemented to provide intensity information for each photodiode at its optimum integration time. This greatly increases the dynamic range and only doubles the total data acquisition time for a given sensitivity. Chapter V presents a detailed description of the microcomputer system, LDA interface, implementation of integration time optimization, data acquisition software and data analysis software. This chapter also includes normal instrument operating procedures. Chapter VI presents the experimental procedure for the simultaneous acquisition of absorbance and ORD spectra. Also included are the chemical systems studied and the sensitivity limits achieved.

—

CHAPTER II

AN OVERVIEW OF THE INSTRUMENT

A. Introduction

In this chapter, a general overview of the instrument which was developed is presented. The goal of the instrument is to provide simultaneous absorbance and optical rotatory dispersion measurements in the ultra-violet (UV) and visible spectral regions and to repeat this at very high speeds. An instrument capable of measuring simultaneously more than one physical property of a chemical system has great potential utility. Also such an instrument can assure that the properties measured actually refer to the same material under the same conditions. All the major parts of the spectrometer are discussed: the optical design, the microcomputer, data acquisition system and software, and the minicomputer data processing/analysis which produces the absorbance and ORD curves.

B. The Optical Design

Figure 2.1 is a block diagram of the optical setup. Descriptions of each block follow. The light source used is a high intensity xenon arc lamp (150 watt). The Osram HBO-150W lamp powered by a PRA Model 303X power supply

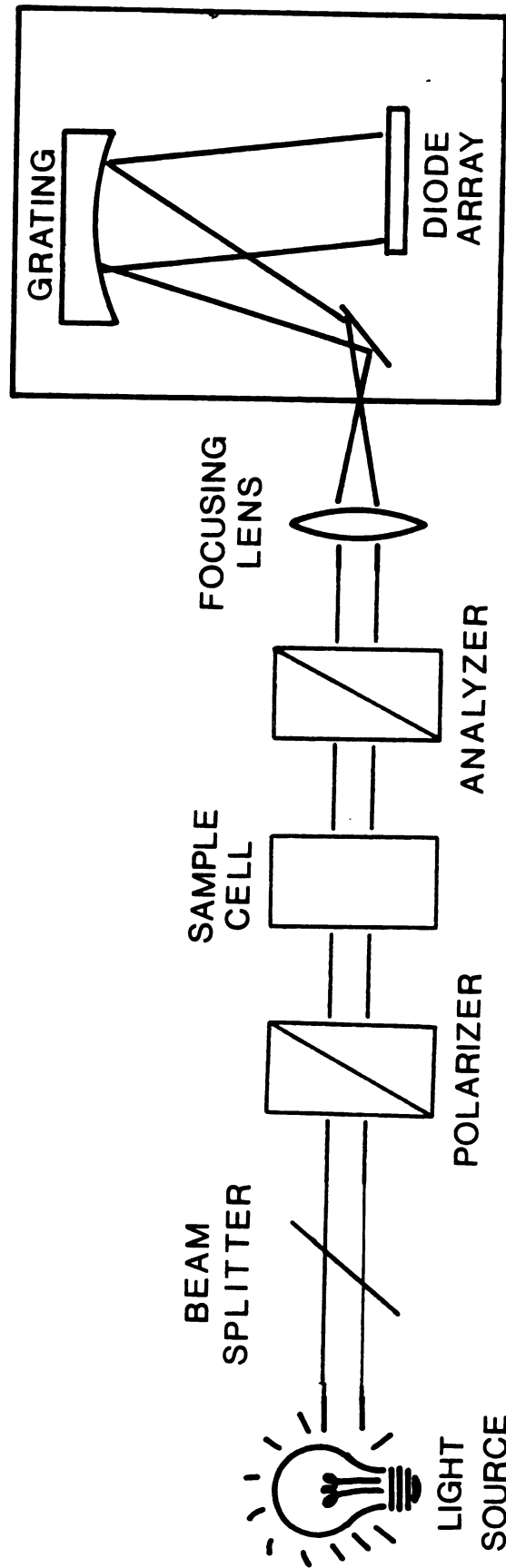


FIG. 2.1 A Block Diagram of the Optical System

generates a high light intensity which is necessary when there is interest in the ultra-violet region of the spectrum because of the relatively low detector sensitivity at shorter wavelengths. The source is mounted on a linear optical rail. The next optical component on the optical rail is a beam splitter. At this point, a small fraction of the light is split off and used to help maintain constant source intensity. A photodiode monitors the sampled fraction of the source and its output is used to modulate the output current of the arc lamp power supply as required. The arc lamp power supply has an external input designed for this type of optical feedback control. The light then passes through a polarizer of the Glan-Thompson prism type. The linearly polarized light then passes through a sample cell which is 1 decimeter in length. If the sample is optically active, the plane of polarization is rotated an amount which is a function of the wavelength. The beam then passes through the analyzer. The analyzing polarizer is rotated under computer control in 0.01 degree increments. This is accomplished through the use of a stepper motor (200 steps/revolution) and a worm screw gear set (reduction of 180:1). Both the amount and direction of rotation are user programmable via the microcomputer and interface board (49).

The light is then collected and focused by a lens onto the entrance slit of a spectrograph. The major components of the spectrograph are an entrance slit assembly, a

reflecting front surface mirror and a concave holographic grating. All of these were purchased for the J-Y Division of Instruments S. A. Inc. An approximate drawing of the spectrograph is shown in Figure 2.2. The entrance slit of the spectrograph is interchangeable (available widths are 25, 50, 100, and 1000 μm) which allows the operator to make the best compromise between resolution and signal strength. A reflecting mirror simply folds the light path such that the inlet optics are physically separated from the spectral image produced. The light is then dispersed by a concave holographic grating. This grating (F/2.0, focal length of 100 mm) provides a flat-field image on the light sensitive region of the linear diode array detector. The grating has a reciprocal linear dispersion of 32 nm/mm resulting in a 400 nm wide spectral window to be viewed by the detector. The window which is viewed can be changed by physically sliding the detector back and forth in the focal plane. The low F number of the grating helps to maximize the light throughput of the spectrograph. Holographic gratings serve to reduce stray light as the random error in groove position is virtually non-existent.

The feature that distinguishes this spectrometer from others is the linear photodiode array (LDA) detector. This detector, the RL512S made by Reticon (35), has 512 photodiodes each 2.5 mm high and 25 μm wide. This 100:1 aspect ratio gives a slit-like geometry for each photodiode.

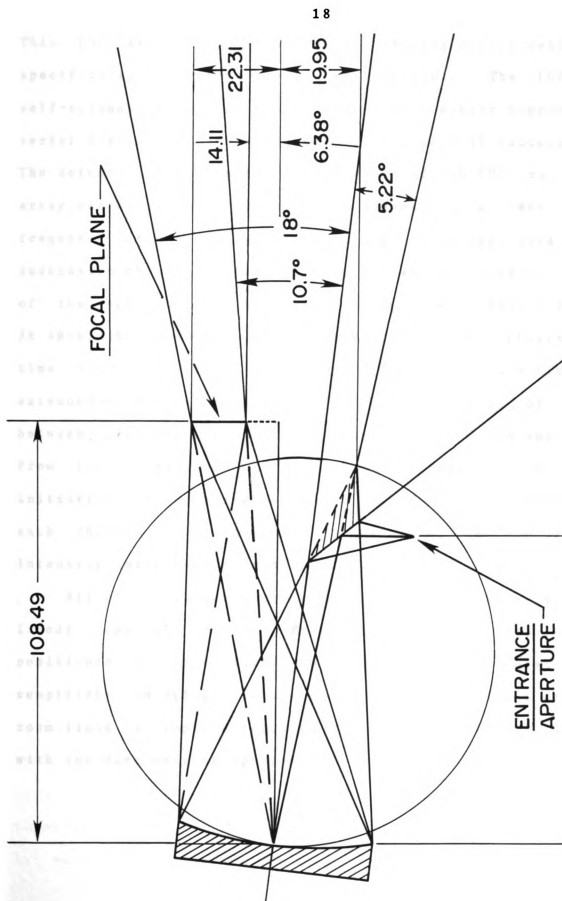


FIG. 2.2 An Approximate drawing of the Spectrograph

This LDA was the first solid-state imaging device designed specifically for spectroscopic applications. The LDA is self-scanned which provides real-time computer compatible serial electronic output for all 512 channels in succession. The detector is responsive over a range of 200-1000 nm. The array can be read out at very high speeds. At a 1 MHz clock frequency (the array can be clocked up to 10 MHz) data from successive channels appear every 4 μ s, and the readout time of the complete LDA is only slightly greater than 2.0 ms. It should be noted that this is also the minimum integration time for this clock frequency. The integration time is extended by simply pausing for the desired length of time between readouts. The total integration time is the time from the initiation of the previous readout to the initiation of the current readout. The charge collected at each photodiode is directly proportional to the light intensity integrated over this time.

All of the optical components are mounted on a single linear optical rail. Each component can easily be positioned anywhere along the rail as the design is completely unenclosed to the room. This is possible as any room light is simply integrated and subtracted out along with the dark current spectrum.

C. The Data Acquisition System

It immediately becomes apparent that the linear diode array detector can generate data at extremely high rates and that the data acquisition would best be done by computer. Rather than overburden a single minicomputer by interfacing several instruments to it, a single board computer was designed to give each instrument its own control and data acquisition capabilities. One of these single board computers (SBC), designed by Bruce Newcome (42), was used in the LDA spectrometer. The Intel 8085A microprocessor chip used in this microcomputer is powerful enough to fully control all instrument functions as well as to perform the data acquisition process. The system has been mechanically and electrically designed for easy interfacing. A block diagram of the microcomputer system excluding the LDA interface is shown in Figure 2.3. Ten Kbytes ($K=1024$) of programmable read-only memory (PROM) on the CPU bus contain the FORTH programming language. There are 22 Kbytes of random access memory (RAM) available for user program space. An interrupt controller is used for input/output (I/O) operations as well as data acquisition. Two universal synchronous/asynchronous receiver transmitters (USART) are used, the first for direct terminal I/O, the other for communication to a PDP 11/40 minicomputer. Also interfaced to the CPU bus is a Persci dual floppy disk system (43).

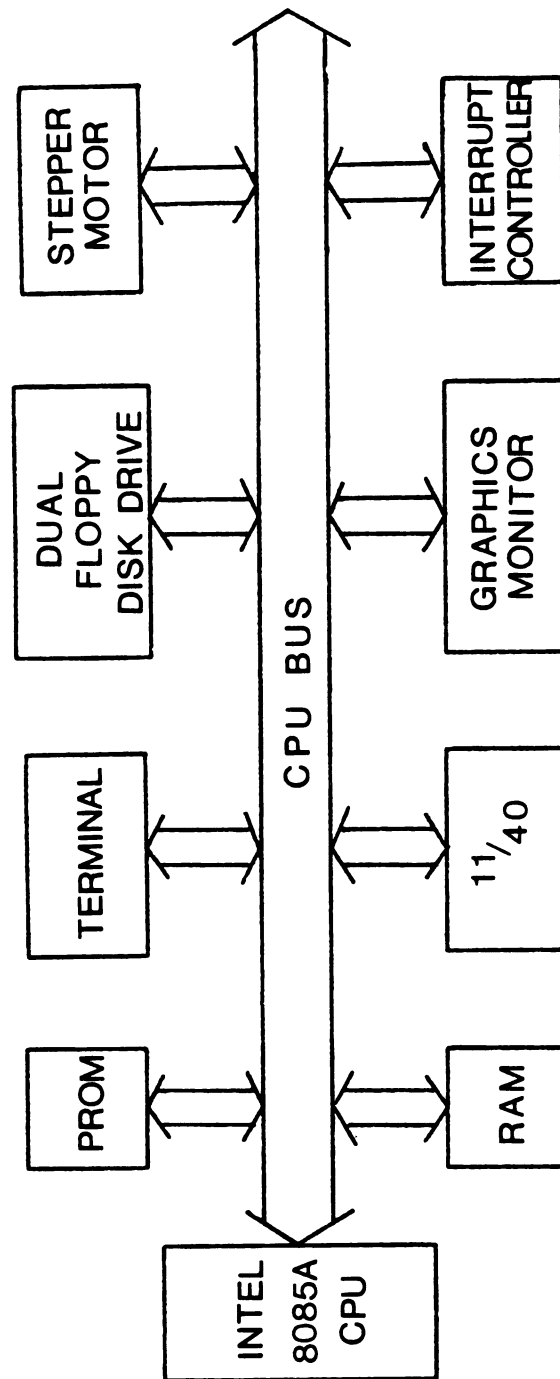


FIG. 2.3 A Block Diagram of the Microcomputer

This provides storage space for both data and software. A local graphics monitor is used (256 by 256 pixel resolution) for initial data display. The CPU is also interfaced to the stepper motor which rotates the analyzer in the optical path.

Figure 2.4 is a block diagram showing the data acquisition hardware. Since the LDA is self-scanned, the only control the CPU can have on the LDA is to initiate the scan (readout) process. By controlling the timing of the initiation signal, the CPU controls the integration time of the detector. The integration time is the time between the readout of a specific photodiode from one scan to the next. The data acquisition is done completely in hardware when enabled by the CPU. Normally, when data are acquired at these high speeds, it is done by taking control of the CPU bus while putting the CPU on hold and storing the data directly in RAM. This is called direct memory access (DMA). However, because the CPU is needed to program the integration time for the scan following the current readout, this straightforward approach could not be used. Instead, data are stored directly in RAM as in DMA, but the memory is not connected to the CPU bus while this occurs. During data storage, the RAM is on the 'data acquisition' bus. Certain sections of RAM can be switched between the data acquisition bus and the CPU bus to allow the CPU access to the data after collection. There are actually two 4 Kbyte RAM banks

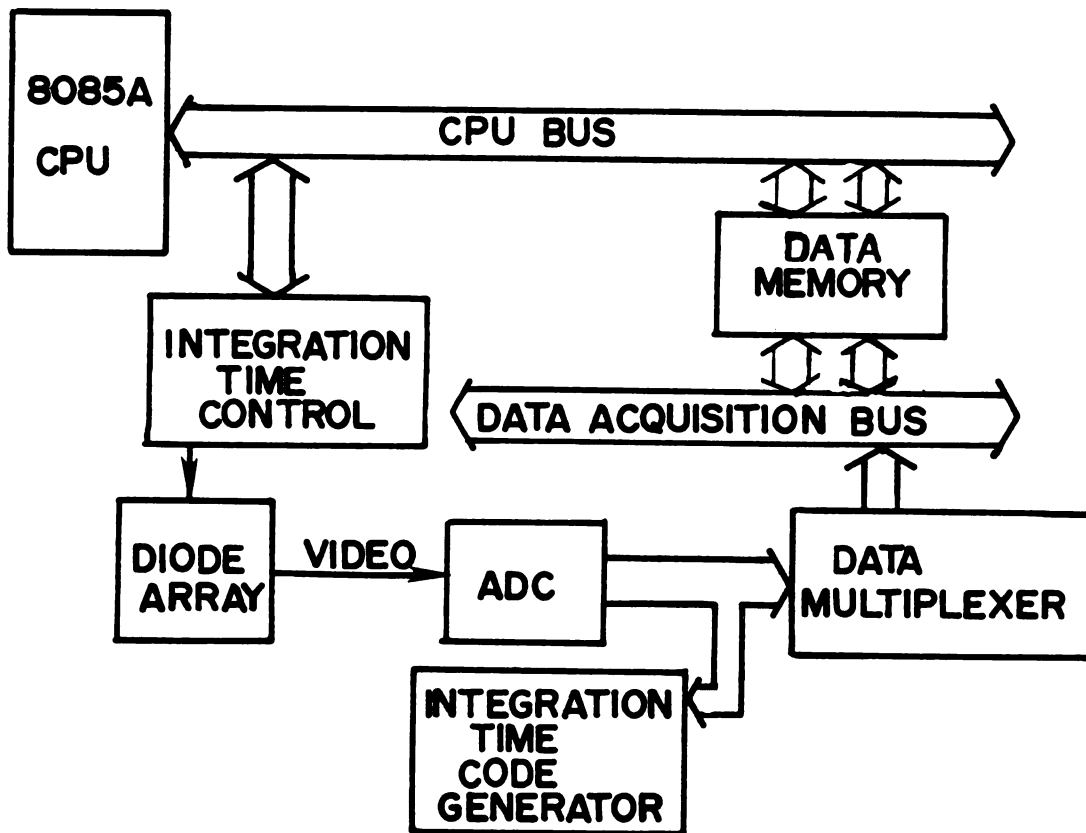


FIG. 2.4 A Block Diagram of the Data Acquisition Circuitry

which, under CPU control, are exchanged between the CPU bus and the data acquisition bus. Since the CPU does not need to be inactive during acquisition, it can unload one RAM bank of data as well as prepare for the next acquisition while the other RAM bank is being filled with new data on the data acquisition bus.

Another feature of the LDA interface is its ability to collect up to 16 successive spectra, each one at twice the integration time of the previous one. The automatic ranging of integration time and the data collection from the optimum range are explained in Chapter IV.

In a research environment, a software system must allow ease of instrument operation, ease of data storage and the ability to reprogram quickly for new experiments. The program polyFORTH, sold by FORTH Inc. (44), has all these capabilities. FORTH is a language which contains a list of 'dictionary' entries called 'words'. Each word is a small program, or subroutine, that performs a particular function. A word can be at as low a level as a single assembly language instruction or a small program written in assembly language. However, a word can also use any other word currently in the dictionary as an instruction in its program. Therefore, the building of a vocabulary of increasingly higher function words can result in powerful set of high level commands which are, at the same time, specific to the tasks at hand. Over 100 words have been

written to control the LDA spectrometer, collect the data, store both data and software on local floppy disk or the PDP 11/40 and plot the data on the local graphics monitor. Use of the software can be found in Chapter V. Documentation of all the current dictionary words for this system can be found in Appendix B.

D. Data Acquisition, Reduction and Analysis

In order to acquire data from the LDA spectrometer, the use of the microcomputer system is essential. Upon power up, the FORTH software present in the microcomputer PROM's becomes active. This then allows all the applications programming to be loaded from the local disk or the PDP 11/40. There are many FORTH words which perform the following functions: setting up scan parameters, collecting data, storing data and plotting data on the local graphics monitor.

The scan parameters consist of a sequence of detector scans. The parameters for each scan in the sequence can have several options. These include data acquisition [Y,N], autoranging [Y,N], number of doublings [0,15], and RAM bank number [0,1]. The sequence of scans is then executed using a single word, GO. The data can then be stored either on the local floppy disk or the PDP 11/40 minicomputer. The data is stored in a coded, unformatted binary form in order

to minimize storage space requirements. The data can be plotted in many different ways on the local graphics monitor very easily with the highest level FORTH graphics words. The microcomputer can also be made to appear transparent thereby allow direct access to the PDP 11/40 as a remote terminal. There are many polyFORTH words all of which are described in Appendix B part 1. The normal commands for instrument operation are described in more detail in Chapter V.

All the data analysis software (which runs on the PDP 11/40 minicomputer) consists of FORTRAN and MACRO-11 (PDP 11 assembly code) routines. All this software acts directly on the coded, unformatted, binary data which were acquired by the microcomputer system, and transferred to the 11/40 via direct serial line transmission or floppy disk.

A single core routine, CRUNCH, is used to call all other subroutines and provides data transmission from one subroutine to another. This allows any combination of the following data manipulation subroutines to be performed upon any data set: ENTER decodes and enters a data file, LOG takes the base 10 logarithm, ABSORB calculates the absorbance using reference, sample and dark current data files, SUBTR subtracts any 2 data files, NRED smooths the data using a modified Savitsky-Golay algorithm (56,57), FIT curvefits the data to a cosine squared wave, fitting the phase, amplitude, and offset parameters knowing only the

frequency (i. e. x-axis spacing) and STORE stores the data in an ASCII file. At this point, MULPLT, a multiple data set plotting routine written by Dr. T. V. Atkinson (45), is used for data display.

-

CHAPTER III

CHARACTERIZATION OF THE LDA SPECTROMETER, CONTINUED

A. Introduction

In this chapter, several characteristics of the linear diode array spectrometer are discussed. Previous work done in this area was presented in my M.S. thesis (22). Improvements made to the spectrometer such as detector cooling and a new computer interface provided the basis for further experimentation and discussion on several detector characteristics.

B. The Effect of Cooling the Linear Diode Array

One of the major operational characteristics of the linear photodiode array is its electronic background noise or dark current. Since the dark current is very reproducible, it can be removed from any spectrum by subtraction. However, the presence of dark current can severely limit the use of the integrating capability of the array. This is because the charge from the dark current increases as the integration time increases and can, at integration times of longer than 2 s, completely saturate the array. At this extreme there is no dynamic range left for signal measurement. Fortunately, the dark current

(which is due to charge leakage) is temperature dependent. By cooling the array, the dark current level can be dramatically reduced (20). In this instrument, the photodiode array is cooled using two single stage thermoelectric coolers (Melcor CP-1.4-3-06L). These coolers are placed between the detector integrated circuit and a copper heat sink. Each cooler is beneath one half of the detector. By passing a large current ($>6A$) through these thermoelectric devices, heat is pumped from one side to the other. With these coolers, the detector can be cooled to a minimum temperature of $-5^{\circ}C$. If lower temperatures were desired, multistage thermoelectric coolers could be used. In order to prevent frost from building up on the face of the detector, the entire spectrograph housing is kept under a dry nitrogen atmosphere at a slightly higher than ambient pressure. This simple antifrosting technique is very effective at these temperatures.

Figure 3.1 shows the dependence of the detector dark current on temperature. The integration time was held constant at 1 s. Over a 30 degree range, the dark current varies by an order of magnitude. Thus cooling makes significantly longer integration times available for the measurement of low light levels. At a temperature of $-2^{\circ}C$, the detector dark current reaches saturation at an integration time of 17 s or more. It has been shown that at room temperature the dark current response versus the

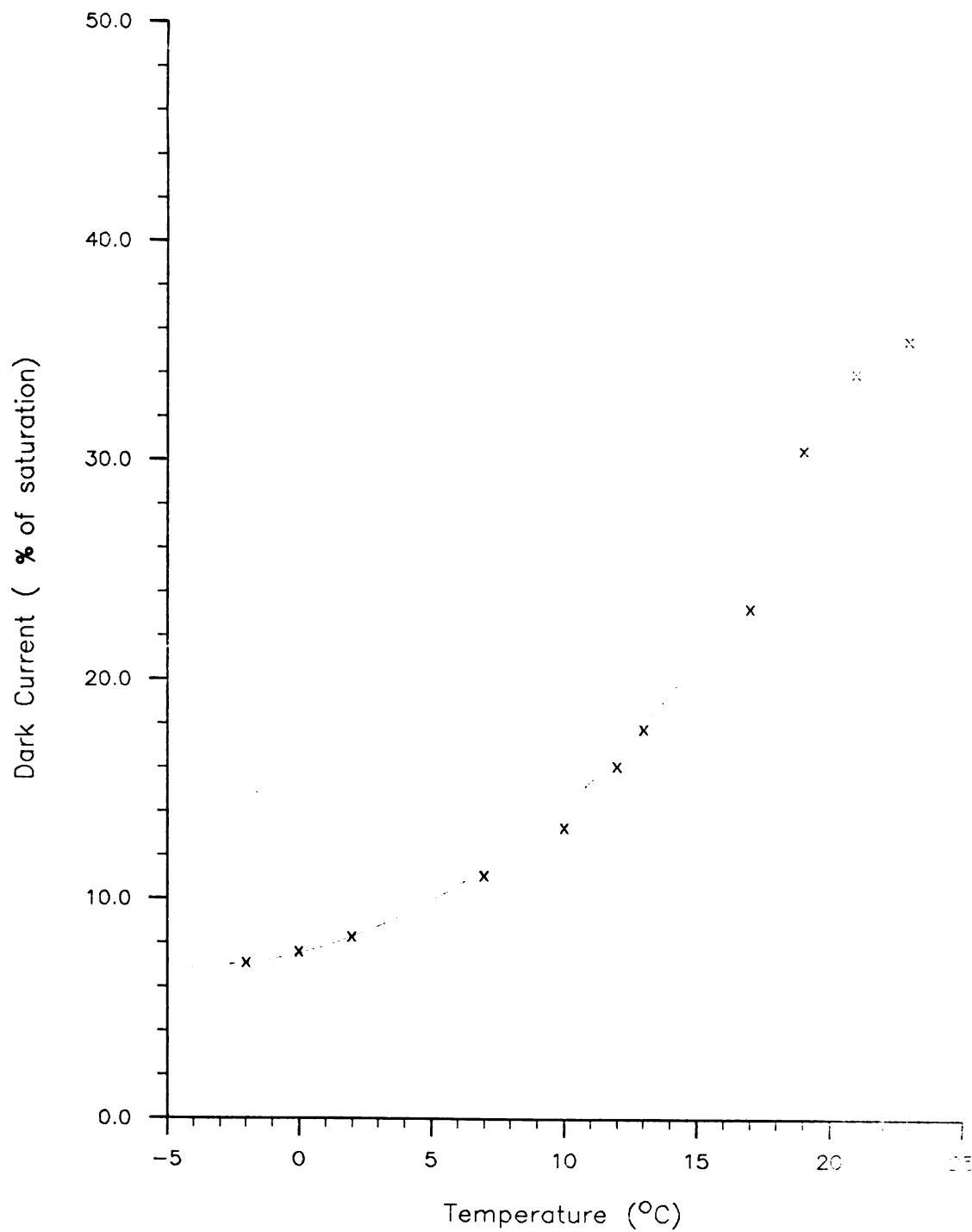


FIG. 3.1 Dark Current versus Temperature at a 1 s Integration Time

integration time is non-linear (22). It was determined that this response was due to a charge leakage from the video recharge line on the LDA integrated circuit and that cooling should alleviate this problem as well. It can be seen in Figure 3.2 that at a temperature of -2°C , the detector dark current responds linearly over a wide range of integration times.

C. Saturation Effects

Blooming, an important characteristic of electron imaging sensors, refers to the situation where a strong signal induced in one region spreads to adjacent sensor elements. In the case of the silicon vidicon (SV), a tube device, blooming is a serious problem. With CCD's and CID's, both of which are solid state devices, blooming is not a problem as long as saturation is not attained. Blooming is observed with these devices, however, if saturation occurs. With SPD arrays, it has been reported that blooming is negligible (22,23), even under saturation conditions.

Lag, or the incomplete readout of the signal stored, is also a problem with electronic imaging devices. In the case of the SV, lag is as high as 10% (46). Lag is undesirable if the detector is being used for time resolution studies. The lag of the LDA was measured in the following manner:

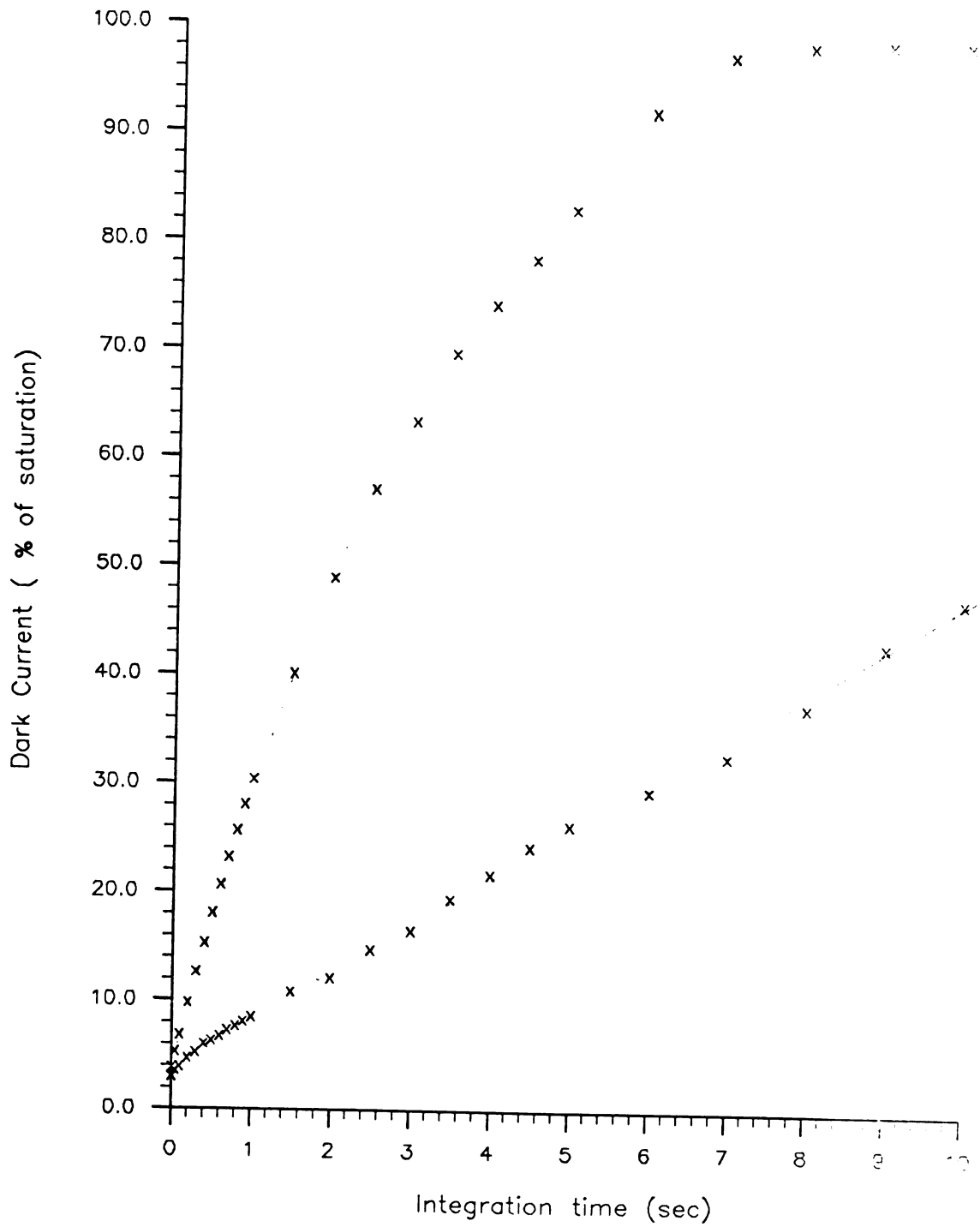


FIG. 3.2 Dark Current versus Integration Time at Temperatures of 25 and -2 degrees C

The detector dark current was allowed to saturate the array. At this time, five sequential readout scans were recorded, each at a 50 ms integration time. Any residual charge not read in the first scan would show in the following four scans. This experiment was done at room temperature and at -2° C. The % residual charge remaining in each scan is given in Table 3.1. The % lag is calculated as shown below:

$$\%LAG = [(S_1 - S_d) / (S_i - S_d)] \times 100 \quad (1)$$

where S_1 is the initial scan which reads out most of the charge. S_i is the next readout and S_d is the dark current at the 50 ms integration time. In the LDA, lag barely exceeds 1% at -2° C. This compares well with a previous reported value of 1% (47). This phenomenon, however, has been attributed to the 1 ms time constant of the preamplifier feedback circuit by Y. Talmi (23) as it was not observed with an E.G. and G. PARC voltage mode preamplifier. With this preamplifier, lag was determined to be less than 0.1%. However, the temperature dependence of the lag indicates that the problem is not with the preamplifier, but with the LDA integrated circuit itself.

SCAN #	% LAG AT FOLLOWING TEMPERATURES	
	25 °C	-2 °C
1	3.6%	1.3%
2	0.9%	0.2%
3	0.2%	0.0%
4	0.0%	0.0%

**TABLE 3.1 Percent Lag for 4 Consecutive Scans
after Saturation Condition**

D. The Effect of Arc Lamp Stabilization

In applications of the optical system, it quickly became apparent that the measurement precision decreased as the length of the experiment increased. This was attributed to an intensity variance of the xenon arc lamp. It was determined that over a 1 minute period, the intensity at 315 nm varied as much as 20%. In order to determine the frequency components of the intensity variation, 2000 successive intensity values were acquired at 315 nm using an integration time of 3 ms with 4 doublings. A Fourier transform was performed on this data. This determined conclusively that most of the drift occurs at frequencies lower than 0.7 Hz. The length of the experiment is approximately 1.5 minutes when 50 spectra are collected and transmitted to the PDP 11/40. Because of this time limit, two options were considered. The first option involved the modulation of reference and sample cells at a rate of 1 Hz. This proved to be ineffective as the cell position was not reproducible. The second option involved the stabilization of the lamp drift. This was done using the previously mentioned optical feedback circuit. This circuit monitors the lamp intensity (using a beam-splitter and a photodiode) and modulates the output current of the arc lamp power supply thereby keeping the light intensity constant. This circuit is damped so that it only responds to low frequency

(<50 Hz) intensity fluctuations. To test its effectiveness, 2000 successive intensity values were again acquired in the same manner as before and a Fourier transform performed. With the corrective feedback, the lamp intensity now varies less than 2% at frequencies below 0.016 Hz. All simultaneously acquired spectra (absorbance and ORD) presented in this work were obtained with this optical feedback circuit in operation.

CHAPTER IV

AUTOMATIC OPTIMIZATION OF INTEGRATION TIME

A. Introduction

Analytical chemists have long sought a multichannel imaging device with excellent resolution and linear response over a wide dynamic range. One of the most severe limitations of current imaging detectors is dynamic range. The Reticon RL512S photodiode array has been shown to have a dynamic range of about 10,000 (23,35) for a single integration time. The conventional single channel detector, the photomultiplier tube (PMT) exceeds this by more than 2 orders of magnitude. In order to extend the dynamic range, the integration time (time between successive scans) of the LDA can be varied. In most analytical spectroscopy applications, however, no single integration time is optimum for every element (wavelength) in the array. This chapter describes a solution to this problem which involves acquiring successive spectra at increasing integration times so that data taken at the optimum integration time for each photodiode is available.

B. Why Vary the Integration Time ?

The integrating capability of the linear diode array is achieved by charge integration, a technique which enhances the signal and averages the noise. This is very useful as the signal-to-noise ratio (S/N) under many conditions increases linearly with the integration time, whereas S/N enhancement by averaging replicate measurements increases only with the square root of the number of scans averaged (46,48). Therefore, the maximum resolution and S/N are obtained if each photodiode is allowed to integrate charge until it nears saturation. Figure 4.1 shows multiple spectra of a quartz halogen lamp at integration times of 5, 10, 20, 50, 200 and 500 ms. The detector dark current has been subtracted out in each of these spectra. Because of this, the saturation level appears to decrease. Actually, the total dynamic range is decreasing because of the increase in the integrated dark current. It should also be noted that the profile of these spectra is a product of the polychromator transfer coefficient, the lamp spectral output and the detector sensitivity at each wavelength. It can be immediately observed that at the shorter integration times, the photodiodes detecting the wavelengths above 500 nm are nearing saturation. At longer integration times the wavelengths near 300 nm are optimized. However, this is achieved at the expense of the information at the longer

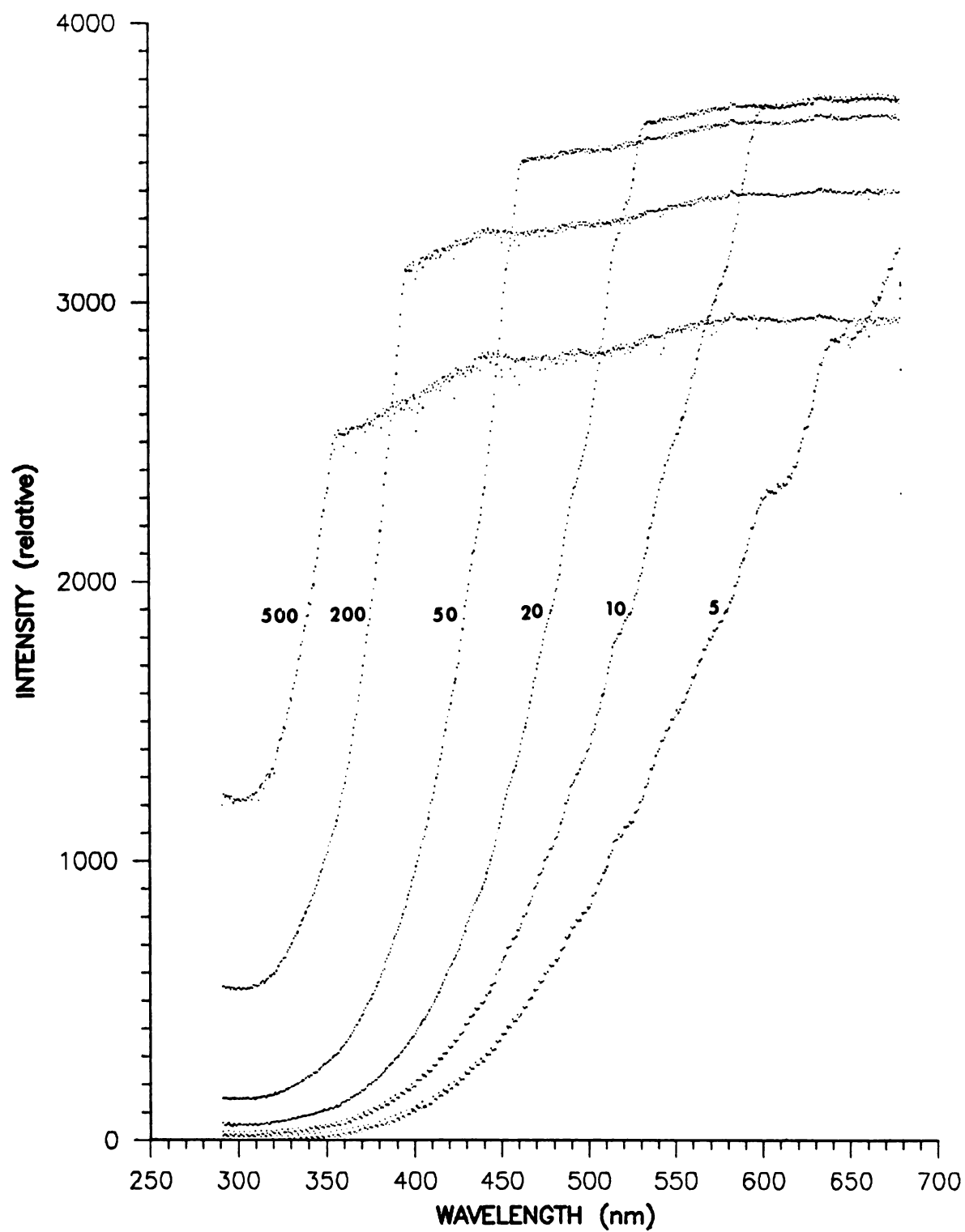
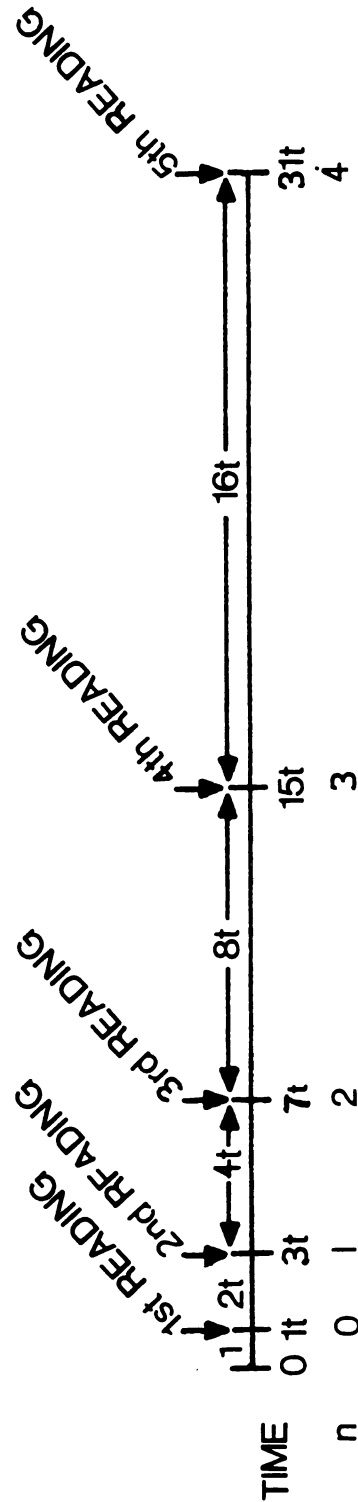


FIG. 4.1 Q.I. Lamp Spectra at varying integration times

wavelengths which is now lost due to saturation of those photodiodes. Normally, spectra are acquired at various trial integration times and then data are taken at a single integration time which is determined to be the best compromise. This usually results in a severe degradation of the photometric accuracy at the wavelengths for which the signal is weakest. The next section describes a multiple integration time technique which alleviates this problem.

C. Programmed Sequence Control of Integration Time

If different integration times are used for successive scans, the longest integration time scan will provide the best reading for the wavelengths where the signal is weakest, while the shorter integration time scans will give the best reading for those diodes that were saturated during the longer times. Figure 4.2 shows that in this system, the integration time is successively doubled. The initial integration time, t , should be chosen such that no photodiode has neared saturation. The number of doublings should be chosen to provide the maximum output for the lowest intensity wavelengths that is possible within the time limits of the experiment. If the integration time is doubled n times (in Fig. 4.2, $n=4$), then the total experiment time, T , will be:



t = INTEGRATION TIME FOR MINIMUM SENSITIVITY

n = NUMBER OF INTEGRATION TIME DOUBLINGS = # Readings - 1

$2(2^n t) - t$ = TOTAL TIME FOR n DOUBLINGS

DYNAMIC RANGE IMPROVEMENT = $2^{(n-1)}$

EACH INTEGRATION TIME USES ALL BITS IN ADC

FIG. 4.2 Integration Time Control Sequence

$$T = 2(2^n t) - t \quad (1)$$

This means that data are acquired at the optimum integration time for each photodiode in a total time of approximately twice the longest integration time in the sequence. For each element, the optimum integration time is that which results in a readout between 50% and 95% of saturation. The dynamic range improvement, D , over that of a single integration time experiment, will be:

$$D = 2^{(n-1)} \quad (2)$$

assuming the dark current is negligible at the longest integration time used.

The number of integration time doublings, n , can be increased until either a time limitation imposed by the experiment is reached, or the integrated dark current exceeds 50% of saturation. The dark current can be reduced by cooling the LDA. The thermoelectric coolers used in this instrument cool the detector to a temperature of about -5° C. Further cooling continues to diminish the dark current thereby allowing integration times as long as 24 hours at liquid nitrogen temperatures (20).

It should be noted that as the integration time is increased, many of the photodiodes may reach saturation while adjacent photodiodes have yet to attain their optimum

integration time. This would be a significant problem in other imaging detectors such as CCD's, CID's and vidicons. In the case of these detectors, when saturation is reached, charge will leak from the saturated pixels (picture elements) to adjacent pixels. This 'blooming' effect has been shown to be minimal in linear photodiode arrays (22,23). Thus integration times that cause many photodiodes to saturate can be used while useful information is still obtained from the non-saturated photodiodes. This information will contain only the integrated light intensity for each photodiode, free of bleed-through from adjacent saturated photodiodes.

D. Total Dynamic Range for the Instrument

The total dynamic range is a function of the detector dynamic range and the range over which the integration time is varied. A 12-bit analog to digital converter (ADC) is used which provides nearly enough resolution to use all the dynamic range information available for each photodiode (about 10^4). When a 4-bit code is used to identify the doubling number (this code will be described in Chapter V), up to 16 doublings can be performed. With the LDA at a temperature of 0°C , the dark current nearly saturates the array in about 17 s. In this case, the maximum useful integration time is about 9 s. Again, the maximum

integration time could be increased by further cooling. The minimum integration time is 2 ms with a 512 element array clocked at 1 MHz. By doubling this 13 times the final integration time is about 16.4 s, longer than the maximum usable integration time at 0° C. For the general case, disregarding the dark current, the total dynamic range D is :

$$D = 2^{(\text{number of ADC bits})} \times 2^{(n-1)} \quad (3)$$

For a 12-bit ADC and a limit of 16 for n, the maximum dynamic range allowed by the electronics is 1.34×10^8 , neglecting the loss of ADC resolution due to dark current. If the LDA is cooled so that dark current is negligible, the total dynamic range available using this technique is thus greater than that of a conventional PMT detector with an autoranging amplifier or converter.

E. Data Storage

In an experiment in which spectra are to be recorded frequently and dynamic range enhancement by the variable integration time method is desired, the storage of every reading can require a huge amount of memory. This memory requirement can be reduced, however, by only storing the measured intensity for each photodiode at its optimum

integration time. This is advantageous for two reasons: first, the memory requirement is reduced by a factor of n ; and second, the data are in a readily retrievable form. That is, it would not be necessary to search several data sets for the optimum integration time of each photodiode. In order to do this, an indication of the integration time must be part of the intensity value word stored in memory for each photodiode. An integration time code, which is simply the number of readings or 1 more than the number of integration time doublings, is generated and combined with the ADC conversion value. Since the computer has a record of the initial integration time, the actual integration time for each photodiode can be calculated from the code. Figure 4.3 shows how the complete sequence of up to 16 integration time doublings is stored in 1 Kbyte of memory. Two bytes of space are allocated to each photodiode, 12 bits of which are the ADC output. The remaining 4 bits comprise the integration time code. The ADC conversions from the initial scan along with an integration time code of 1 (first reading) are written in the 1 Kbyte block of memory. On the next scan, for which the integration time has been doubled, the ADC conversions and the new integration time code (1 greater than the previous code) are written over the same block of memory with the exception of those channels for which the photodiodes which are within 95% of saturation. In the case of λ_i in Figure 4.3, the data at each of the

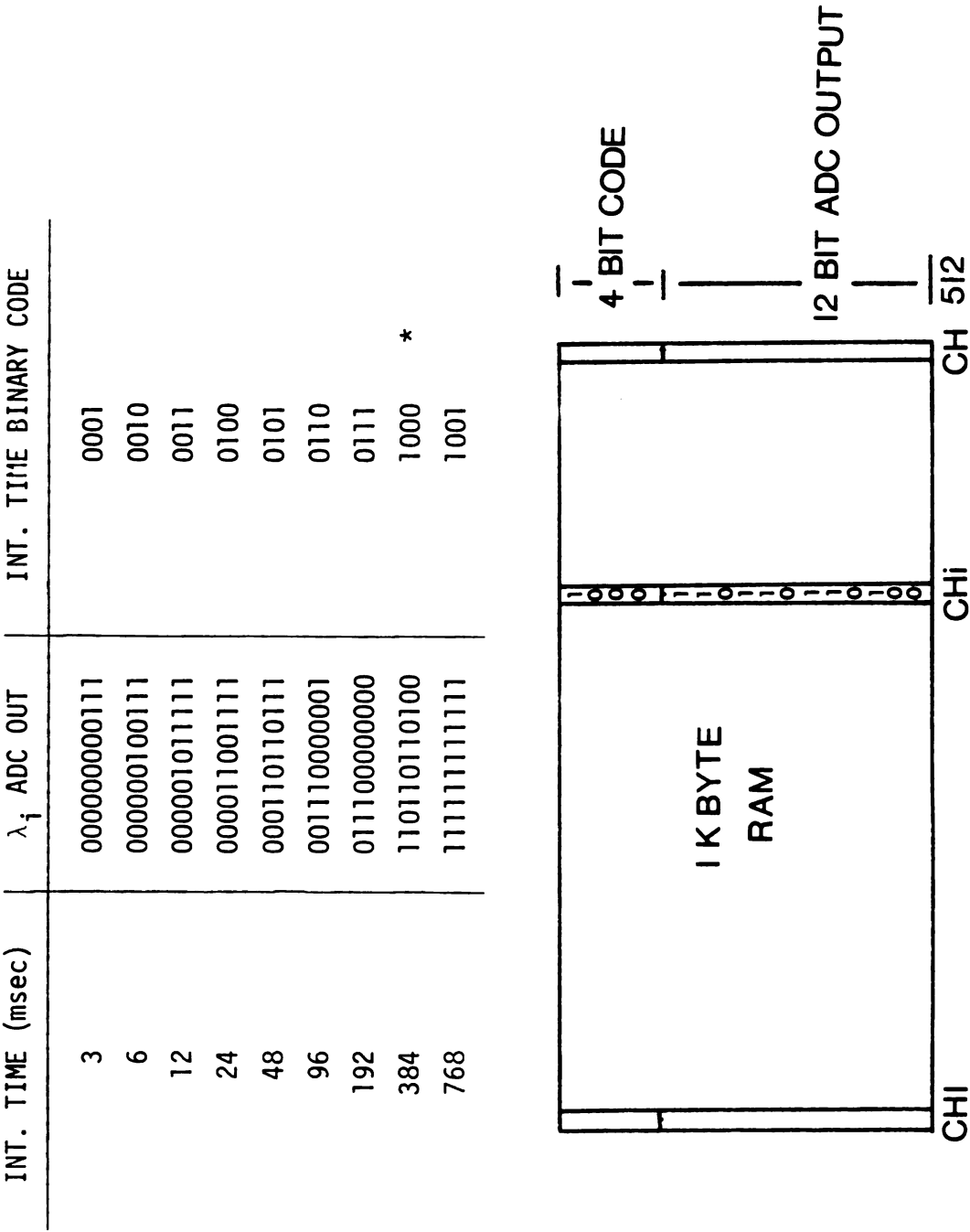


FIG. 4.3 Data and Integration Time Code Storage. In this illustration, a minimum integration time of 3 ms is used.

integration times from 3 to 384 ms will be written over the previous shorter time result (384 ms data is written most recently), but the 768 ms integration time and all further scans are not, as they have attained at least 95% of saturation (ADC output is all 1's). Thus, the ADC conversion and integration time code recorded are the optimum result for this photodiode. This same process occurs for each photodiode individually so that by the end of the longest integration time, the 1 Kbyte block of memory is filled with the ADC conversions and integration time codes made at the optimum integration time. In this instrument, this process occurs for all 512 photodiodes in the LDA.

F. Performance

Figures 4.4-4.8 demonstrate the effectiveness of the procedure for the automatic optimization of integration time. Each spectrum of a Hg pen lamp was acquired at the indicated integration time (or sequence of integration times), and the background was subtracted. At an integration time of 25 ms (Figure 4.4), the relative intensities of the strong emission lines are easily observed; however, low intensity lines cannot be seen even after enlarging the baseline (Figure 4.5). If a longer integration time is selected, such as the 1 s integration

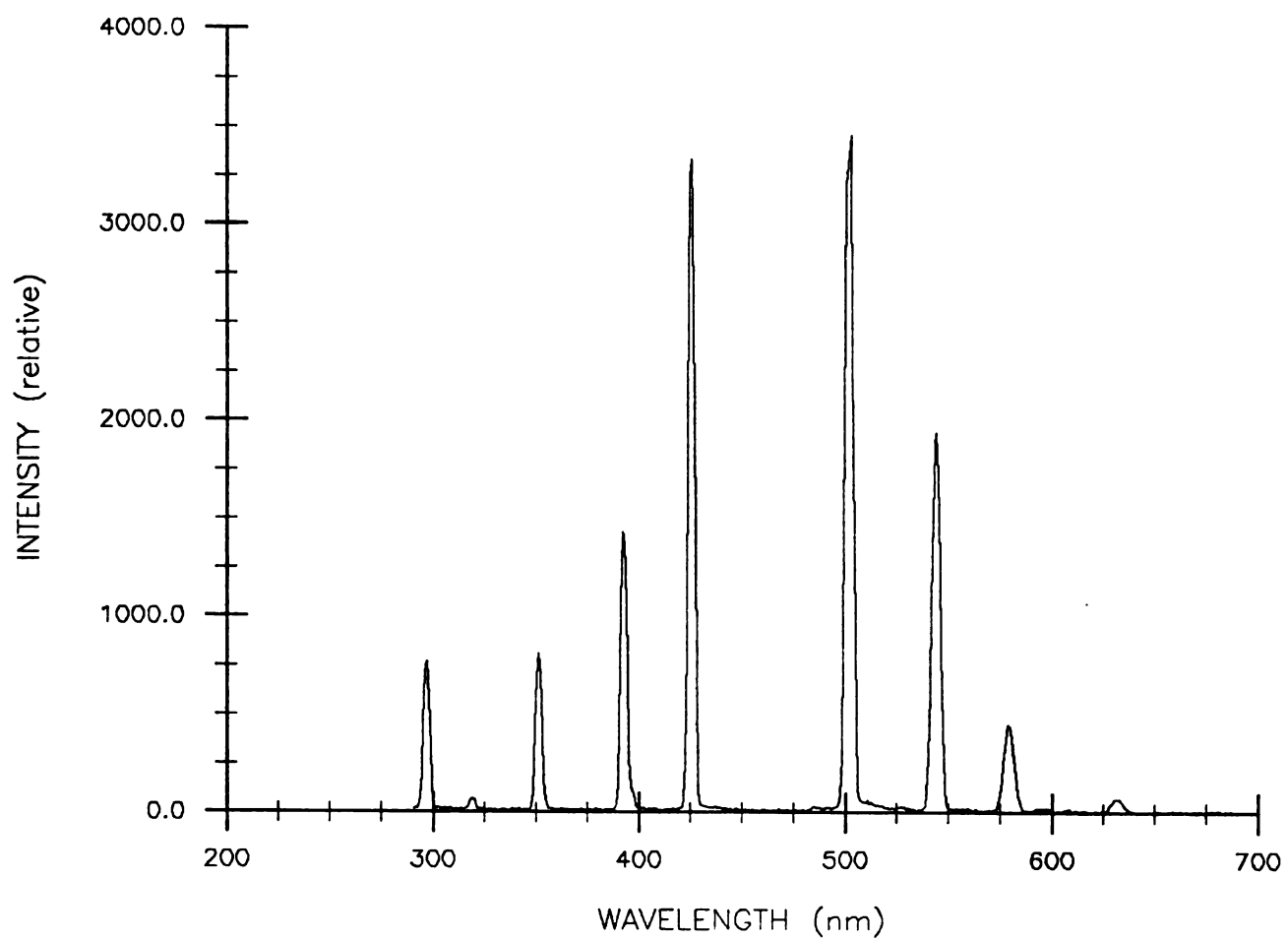


FIG. 4.4 Hg Lamp Spectrum at a 25 ms integration time

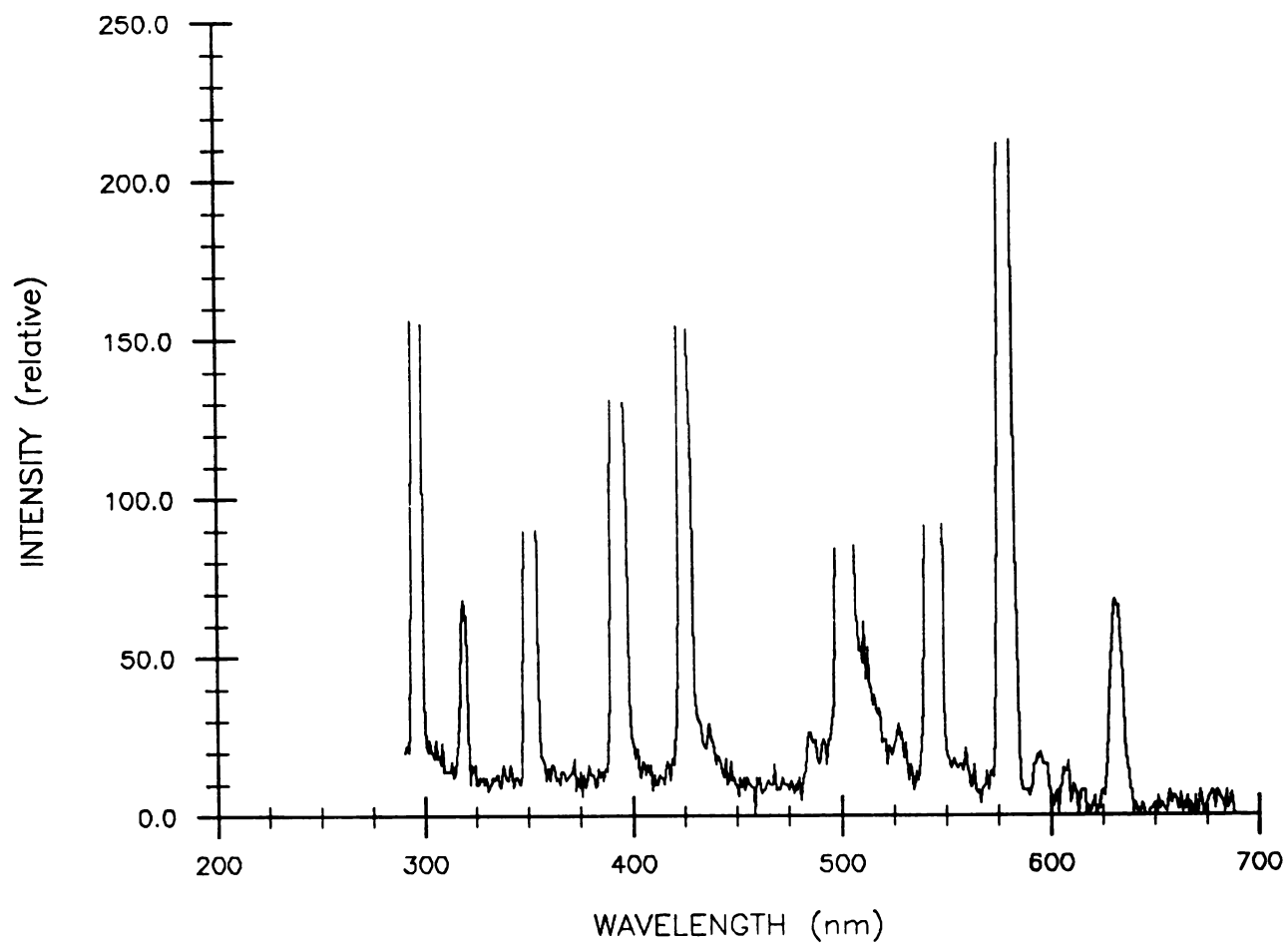


FIG. 4.5 Enlargement of the Hg Lamp Spectrum at a 25 ms integration time

time used in Figure 4.6, the intensities of the weak spectral lines can be measured (380 nm and 610 nm) because of the improved S/N attained through charge-integration. However, the relative intensities of the strong lines are lost as they have all reached saturation. The saturation level was observed in this case by lowering the LDA amplifier board output voltage (with a voltage divider) in order to remain within the range of the ADC. Also note that the line width of these strong lines has not increased which indicates that there is no degradation of resolution. Peak-broadening would have been observed with imaging type detectors which bloom. Figure 4.7 is the spectrum obtained using an initial integration time of 15 ms with 6 integration time doublings. The relative intensities of the strong lines are present, and when the baseline is expanded (Figure 4.8), the relative intensities of the weak lines are also available. The total time for this experiment was 1.92 s, about twice as long as the single 960 ms integration time scan. However, 7 spectra were actually acquired, which allowed data at each photodiode to be taken at its optimum integration time and the complete optimally acquired 512 element spectrum was automatically packed into only 1 Kbyte of memory.

In order to measure the practical dynamic range, the absorbance of a series of neutral density filters was measured. The source used was a Xe arc lamp (150 watt), and

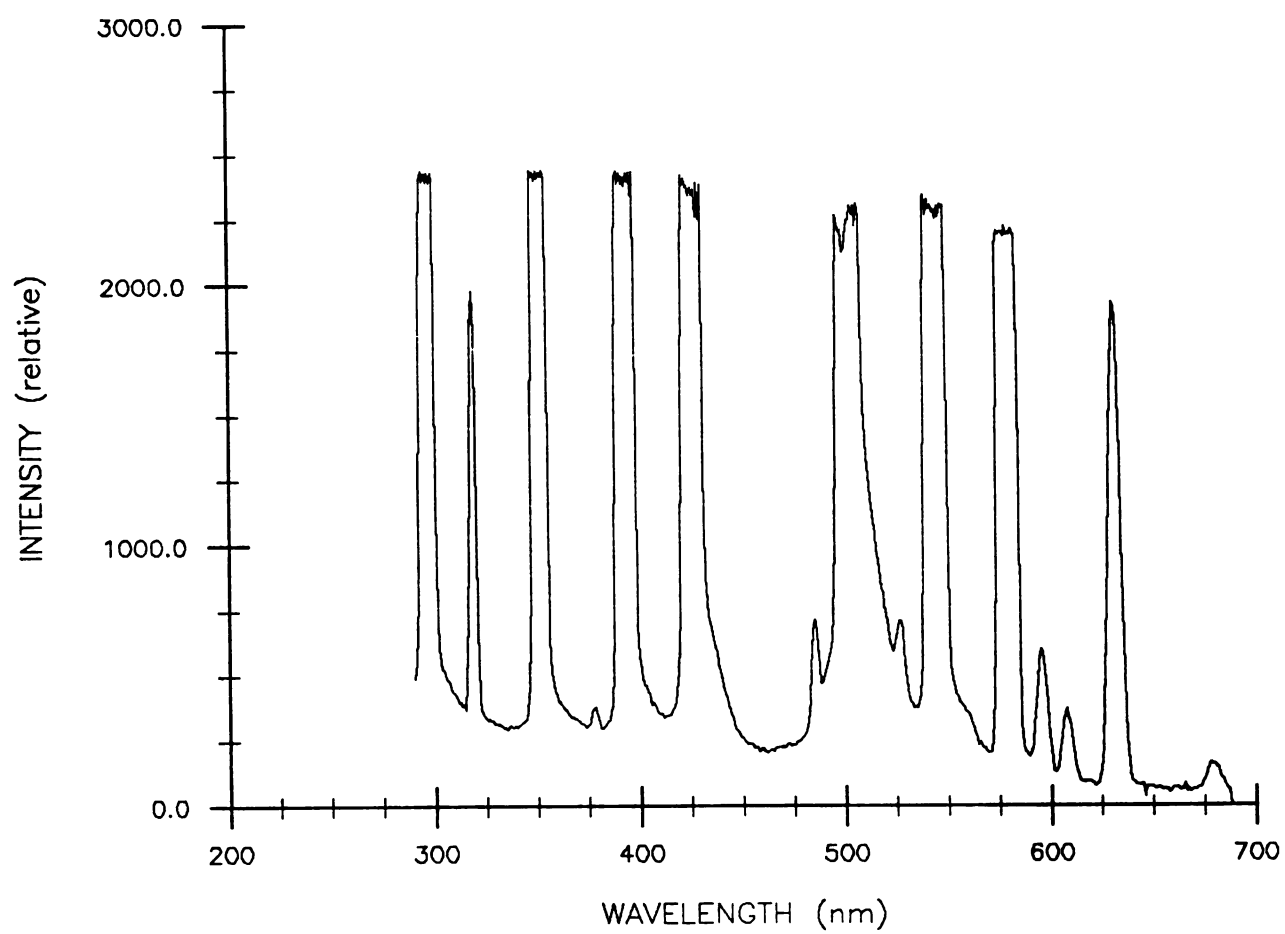


FIG. 4.6 Hg Lamp Spectrum at a 1 s integration time

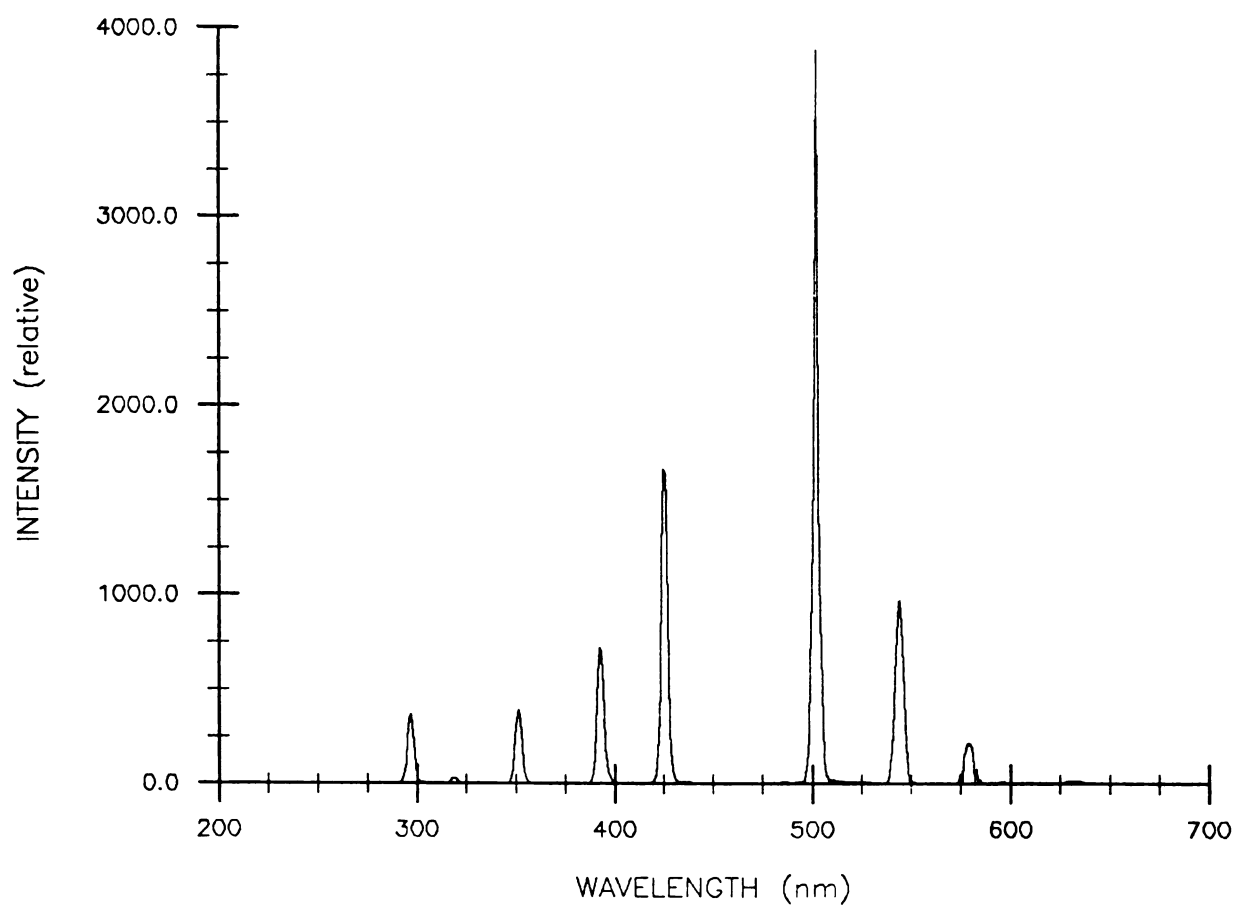


FIG. 4.7 Hg Lamp Spectrum acquired with an initial integration time of 15 ms with 6 integration time doublings (7 readings)

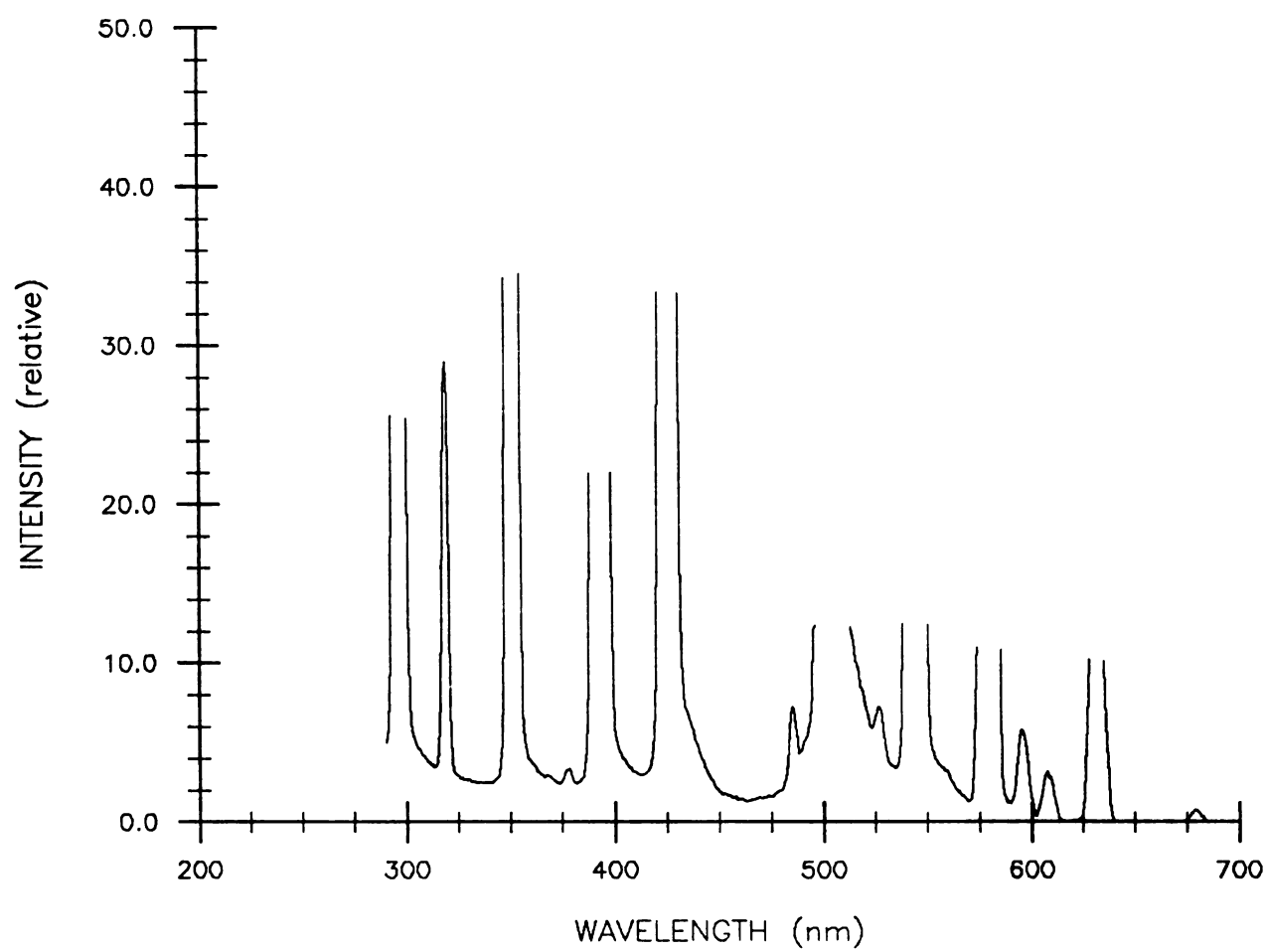


FIG. 4.8 Enlargement of FIG. 4.7

the measurements were all taken at 600 nm. The results are plotted in Figure 4.9. The data were acquired using an initial integration time of 3 ms followed by 13 integration time doublings (14 readings) with the detector at -2° C. The longest integration time of 24.6 s was only slightly longer than the time in which the dark current saturates the array. It is easily seen that the response is linear from about 0.0-4.2 absorbance units. For a nominal absorbance of 3.96, an absorbance of 4.03 was measured. This corresponds to a 2% error at an intensity 4 orders of magnitude below the maximum. The dynamic range is thus about 5.0×10^5 . This could be improved considerably by using a 14-bit ADC to take advantage of the full dynamic range of the LDA (10^4), or by further cooling of the LDA to allow longer integration times.

Figure 4.10 shows, however, that with the current optics system, measurements of highly absorbing systems are not practical because of an extreme deviation from Beer's Law. The curve was acquired by measuring the absorbance of KMnO_4 solutions of known concentration using the LDA channel at $\lambda = 545$ nm. The Xe arc lamp was used and an IR cutoff filter was placed in the light path. Using a 12 ms integration time and 10 doublings (11 readings), absorbances of greater than 2 show a large deviation from Beer's Law which is attributed to the presence of stray light in the polychromator.

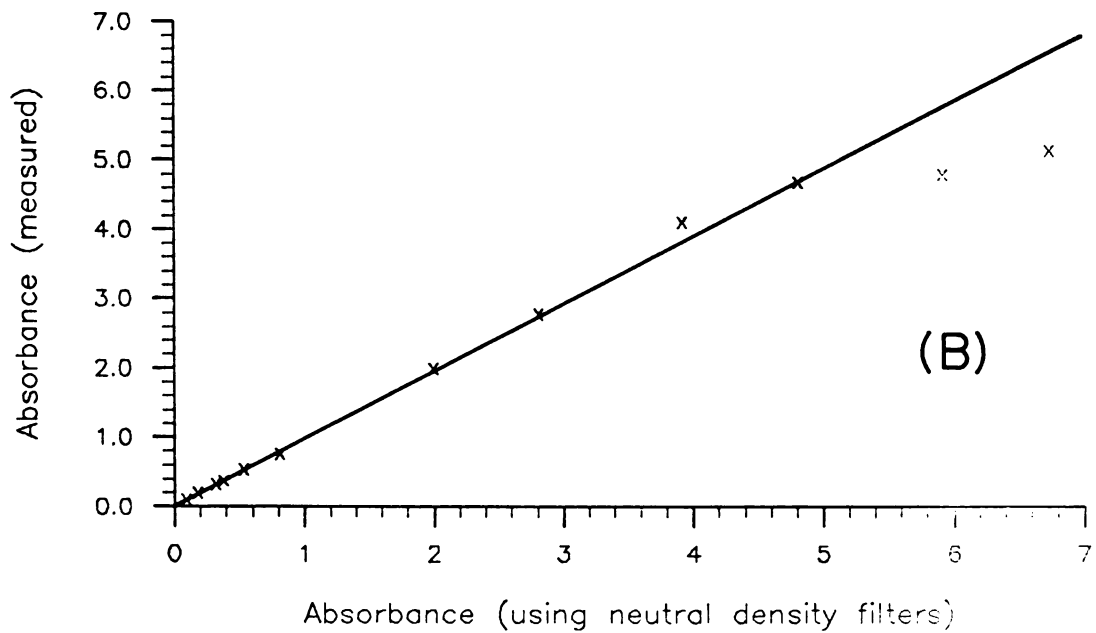
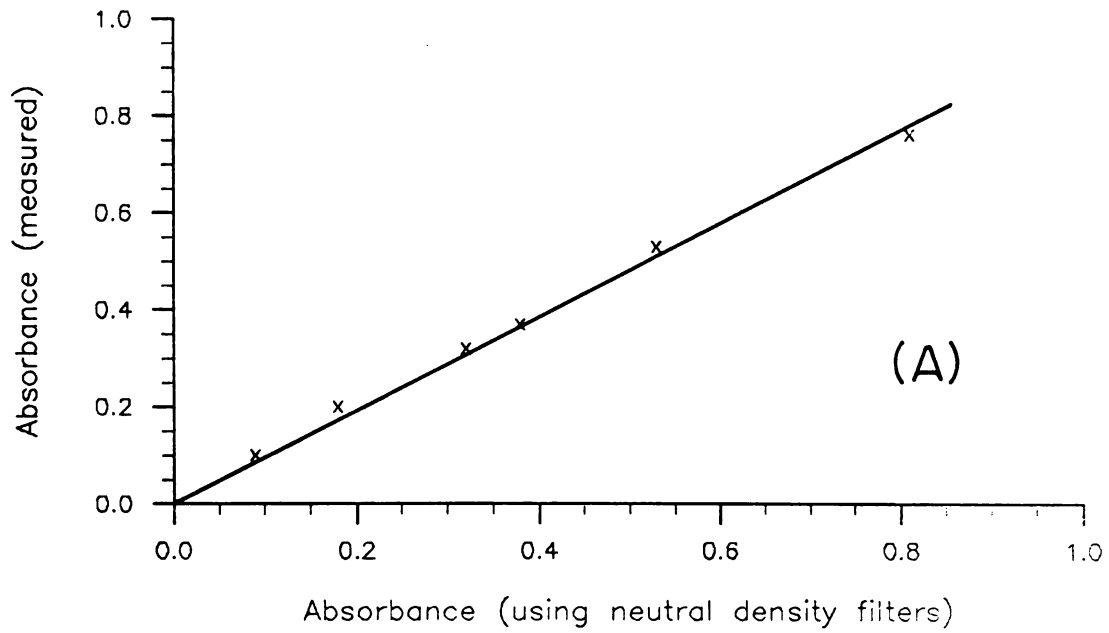


FIG. 4.9 Dynamic Range Measurement using Neutral Density Filters

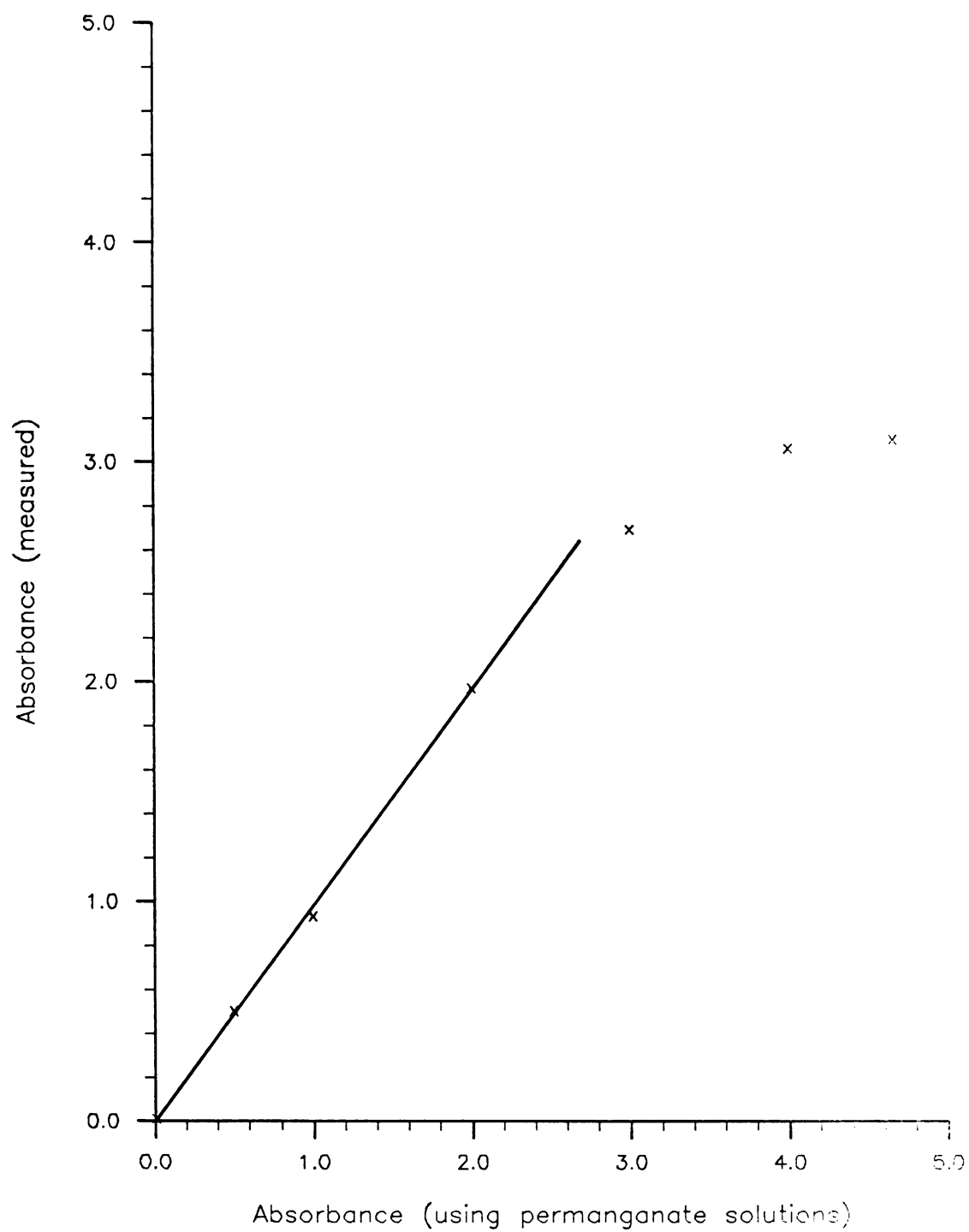


FIG. 4.10 Dynamic Range Measurement using known KMnO₄ solutions

Stray light for this instrument was previously determined to be $< 0.004\%$ (22). However, in an instrument of this type, order overlap and all internal reflections inside the polychromator cannot be eliminated through the use of filters as in a normal single wavelength spectrometer. If a narrow bandpass filter is used, absorbances up to 4.0 can be measured as demonstrated in Figure 4.9 where stray light, as well as the signal intensity, is reduced by the neutral density filters. The stray light problem is much worse in our system for short wavelength measurements than for long because both the source intensity and detector sensitivity decrease greatly toward the UV region of the spectrum. Array detector spectroscopy would be greatly aided by the development of two new devices. One is a filter that would have an absorption spectrum that complements the source spectrum so that the combination would have more uniform spectral intensity. The other is a filter that would pass only the blue part of the spectrum in one region, only the red part in another and only the mid part in between. If such a filter were designed to match the dimensions of an array detector, and the dispersion of the polychromator, each detection element would only be sensitive to its appropriate wavelength. The resulting system would be like a dual monochromator, but would not entail the large losses involved in tandem dispersion systems.

CHAPTER V

THE DATA ACQUISITION SYSTEM

A. Introduction

Through the use of a microcomputer, data acquisition is no longer a tedious process. Also, the high speed capabilities of computer-controlled instrumentation make new analytical procedures possible. In this chapter, a detailed description of the microcomputer system is presented. Hardware, software and normal operating procedures are described.

B. Hardware

Figure 5.1 is a detailed block diagram of the microcomputer system and the LDA interface. As mentioned in Chapter II, the single board computer (SBC), designed by Bruce Newcome (42), is based on the Intel 8085A 8-bit microprocessor. The same board holds 16 Kbytes of RAM or PROM, two USARTS and an interrupt controller. The system is readily expandable through a mother board and backplane arrangement (42). This extension of the CPU bus allows user modules (interface circuits) access to the CPU bus and also provides for connector space to remote devices. The microcomputer and LDA interface populate 5 mother boards.

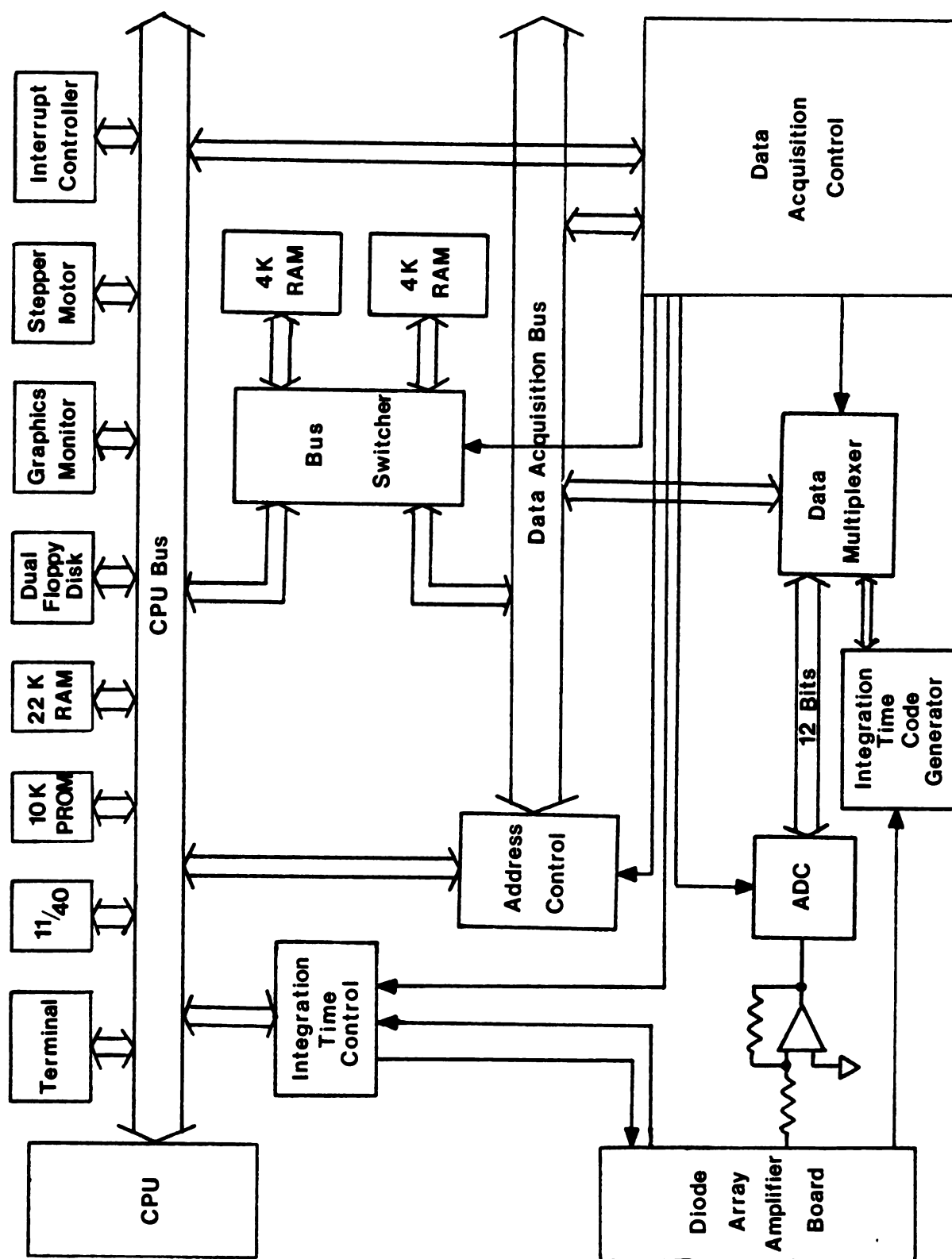


FIG. 5.1 A Block Diagram of the Microcomputer System and Linear Diode Array Interface

The first mother board contains the modules comprising the SBC as described above. The second mother board, an extension of the CPU bus only, contains an additional 8 Kbytes of RAM for program space, a transceiver module and a stepper motor driver module. The transceiver module is used to transmit a subset of the CPU bus to a remote box which houses a similar transceiver module (reconstructing the CPU bus subset) and interface modules to floppy disk and graphics systems. The floppy disk system consists of a dual 8-inch single-sided drive and an intelligent controller, Persci model numbers 760 and 1070 respectively. This provides 500 Kbytes of mass storage for program and data space. The graphics system includes a 9-inch Ball monitor and a Matrox MTX-256**2 graphics display board. This system provides a 256 by 256 dot matrix display, each individually addressed. The stepper motor driver module (49) is used to rotate a polarizer in the optical setup. The stepper motor is a RAPID-SYN Model 23D-6102, which has 200 steps/revolution. This is geared down using a worm screw gear set so that each complete revolution results in a 2 degree rotation of the polarizer. Therefore, the CPU can control rotation in 0.01 degree increments as well as determine the direction of rotation.

The third mother board, also an extension of the CPU bus, contains another 8 Kbyte RAM module for program space, and the integration time control circuitry. Figure 5.2 is a

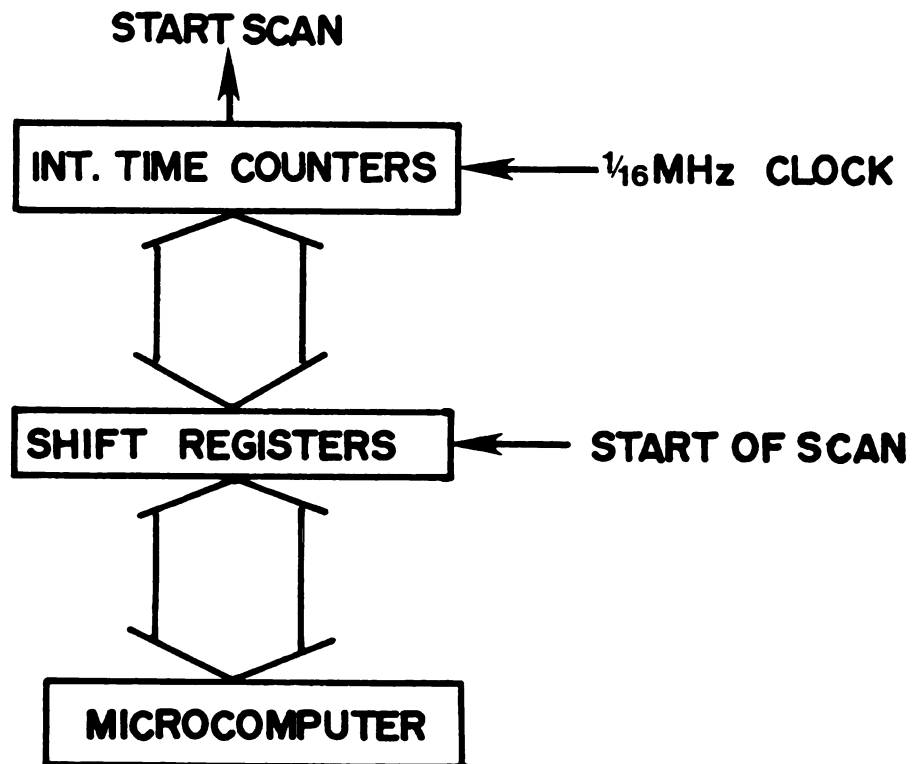


FIG. 5.2 A Block Diagram of the Integration Time Control Circuit

block diagram of the integration time control circuitry. A schematic diagram can be found in Appendix A. The integration time can be controlled directly by the microcomputer or it can be automatically doubled by the data acquisition control circuitry. When direct computer control is desired, the integration time is written into a set of shift registers and transferred to the integration time counters. A 1 MHz clock is used for the detector scanning circuit and the integration time controller. The integration time counter is 28 bits long, which results in a possible range of integration times from 2 ms to 268.4 s ($2^{28} \mu s$). When automatic integration time doubling is desired, the CPU only programs the initial integration time. Upon the start of each successive scan, the integration time shift registers then shift left, thereby exactly doubling the integration time for the next scan.

The fourth mother board is divided into upper and lower halves. The lower half, an extension of the CPU bus, contains the data acquisition control circuitry. The upper half, the data acquisition bus, contains hardware which generates the address, data and control lines for the direct memory access circuit.

The data acquisition control circuitry is programmed by the CPU through an 8-bit control register. At the start of each scan of the detector, this register, as well as the integration time and address counters, must be loaded by the

CPU for control of the next scan (except when automatic doubling of the integration time is in operation). The low bit (D0) of the control register enables the data acquisition cycle. Another bit (D1) enables autoranging of the integration time. If enabled, the top 4 bits (D4-D7) are loaded with the number of integration time doublings desired (0-15). Another bit (D3) determines which memory bank the data is to be written into. The last bit (D2), currently unused, generates a pulse synchronous with the start of the scan pulse. This could be useful if some other equipment (for example, a shutter or stopped-flow) needs to be synchronized with the self-scanning detector.

The data acquisition control circuitry uses the information at the control register, the LDA clock, start of scan (generated by the integration time control circuitry) and the EOL (end of line) signal from the LDA to generate the following control signals necessary in the data acquisition cycle: Start Conversion, which signals the ADC to begin conversion of the current photodiode; Latch Enable, which latches the 12-bit conversion and the 4-bit integration time code; \overline{OE} LSB, which drives the low 8 bits of the conversion onto the data bus; \overline{OE} MSB, which drives the high 4 bits of the conversion and the 4-bit integration time code onto the data bus; ADDRESS COUNTER CLK, which increments the address for the storage of the next data byte; \overline{WR} , which writes the data byte into memory and LOAD,

which loads the next set of commands from the control register for the next scan. Schematics of the data acquisition control circuitry as well as a timing diagram showing the relationship of these signals to the detector output during a single data acquisition cycle can be found in Appendix A.

Another set of counters, the address control circuit, is simply loaded by the CPU with the initial address for data storage. These counters then drive the 16-bit address onto the data acquisition bus. The address is incremented after each byte of the 1024 bytes/scan is written into RAM by the data acquisition control circuitry. The starting address for data collected from the next readout is written by the CPU during the current readout. When the autosequencing of the integration time is enabled, the address counters are automatically reloaded with the same initial address for each successive scan. This is required for the efficient data packing described in Chapter IV. Schematics of the address control circuitry can be found in Appendix A.

The data lines on the data acquisition bus are generated by the ADC module. A block diagram of this circuitry can be found in Figure 5.3 (Schematic in Appendix A). The video signal (0.0 to 2.5 volts) output from the Reticon RC1024SA amplifier board is amplified by an op-amp to give the 0 to 10 volt input signal required by the

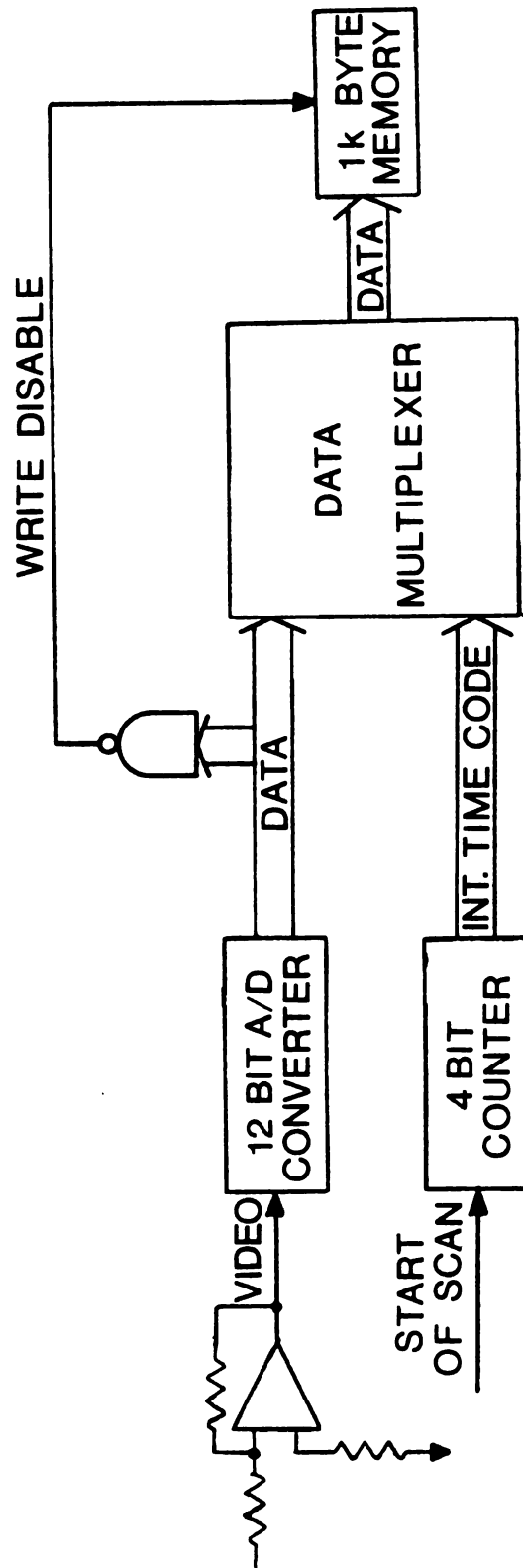


FIG. 5.3 A Block Diagram of the ADC Board

Computer Labs HAS-1202 2.2 μ s 12 bit analog-to-digital converter (ADC). This high speed ADC allows detector readout rates of 4 μ s per diode. The resulting conversion is latched along with the 4-bit integration time code. This code is simply a 4-bit counter which, when automatic integration time doubling occurs, is incremented each time the integration time is doubled. The two bytes of data are then written into RAM by the data acquisition control circuitry. Also, when autosequencing of integration time is used, a 12-input NAND gate is used to detect 95% of photodiode saturation or more (all ADC bits are 1); its output inhibits the write pulse (generated by the data acquisition control circuitry) which results in data storage for only those photodiodes that have not yet reached 95% of saturation.

The fifth mother board contains two half-populated 8 Kbyte RAM modules for data storage and also a bus switcher module (designed by Bruce Newcome) which can exchange these two memory banks between the data acquisition bus and the CPU bus. This arrangement allows the CPU to actively program the next data acquisition cycle and to unload data from one RAM bank while the ADC circuitry on the data acquisition bus fills the other with new data.

C. Software

The software for the collection and subsequent analysis of data is divided into two distinct categories. The microcomputer acquires and transmits data using programs written in the language polyFORTH (44). Data analysis and reduction occur in the PDP 11/40 by programs written in FORTRAN IV and MACRO-11 (PDP-11 assembly) which run under the RSX-11M operating system.

The multi-level language and operating system polyFORTH is used exclusively on the microcomputer system (44,50,51). FORTH is characterized by four major elements: the dictionary, stack, interpreter and assembler. The dictionary occupies most of the memory space. It is a list of 'words', each of which is a program which can include any other dictionary entry. New words can easily be added to the dictionary which makes FORTH an extensible language and one which is versatile, almost to a fault. For example, already defined entries such as '+' can be redefined and thus drastically change the language. Also, several dictionaries can be maintained simultaneously allowing the user to create identically named entries with a different function in each dictionary. Two push-down stacks are maintained. The first, the parameter stack, provides an efficient mechanism for transferring numbers from one word to another. The second, the return stack, is used for

internal flow from word to word. FORTH is fundamentally an interpretive system. There are two interpreters, one which works in the conventional manner, simply passing text strings and looking up each word in the dictionary; the other interprets strings of absolute memory addresses by executing words which contain the addresses of the next word. This threading provides for a faster execution speed as each word does not have to be located as in a conventional interpreter. FORTH also includes an assembler, which allows the user to define words on the assemble language level. Such words are often necessary to perform I/O operations and highly time-critical functions.

FORTH is used for many reasons. It is a very compact language using only about 8 Kbytes of memory. FORTH is an extensible language, a great advantage when software modifications are required due to a change in the experimental procedure. FORTH is also highly portable. FORTH is most useful in applications where speed is of the utmost importance. Since portions (or all) of a FORTH program can be written in assembly language, FORTH programs can be written that compare favorably in speed with machine-language programs (about 90%). Since FORTH can grow to a high-level language, programming productivity is much higher than using machine-language, and yet speed is still much faster than other high-level languages such as FORTRAN.

In addition to the words supplied in the basic FORTH

dictionary, over 100 words have been created which control the LDA spectrometer. As each word can call all others previously defined, a high level set of instructions has evolved to perform various experiments. As an example, a listing of six FORTH words which control rotation of the polarizer follows:

C1C0 CONSTANT STPMTR

C1C8 CONSTANT DIRMTR

: STEP 10 0 DO LOOP STPMTR C@ DROP ;

: CDIR 50 0 DO LOOP DIRMTR C@ DROP ;

: ROTATE 0 DO STEP LOOP ;

: DEGREES 2DUP 0. D< IF

CDIR DABS DROP 500 + ROTATE CDIR 500 ROTATE

ELSE DROP ROTATE THEN ;

The first two words, STPMTR and DIRMTR, are CONSTANT definitions. When a CONSTANT definition is executed, the constant is placed on top of the stack. Therefore, either directly entering STPMTR from the terminal or calling STPMTR from another word will place the value C1C0H on top of the parameter stack. The stepper motor interface hardware is designed so that reading from the addresses C1C0H and C1C8H causes the stepper motor to step once or change direction respectively. The third entry, STEP, executes a single read from STPMTR using the FORTH definition C@ and places the value read on top of the stack. The 10 0 DO LOOP is simply a wait loop inserted so that the maximum stepping frequency

of the motor is not exceeded. This allows STEP to be called several times sequentially without loss of motor steps. The FORTH word DROP simply discards the top value from the parameter stack. Since a value was placed on top of the stack by C@ , it must be discarded when it is no longer useful. The fourth word, CDIR, operates in a similar fashion to STEP and when called, changes the direction of motor rotation. Calling the fifth word, ROTATE, steps the motor the number of times found on top of the parameter stack. The sixth word, DEGREES, is the only entry normally called by the operator. Entering the command -18.75 DEGREES rotates the polarizer appropriately. DEGREES accomplishes this by first determining the direction of rotation by the sign of the number on top of the stack. It then calculates the number of steps required for the desired rotation, and rotates the polarizer accordingly. If a negative direction is entered, it rotates the polarizer 5 degrees past the final position and then rotates 5 degrees positively to eliminate any gear lash. This results in positive positioning of the polarizer. This set of words shows how easily and quickly FORTH can become a high level language for the needed experimental functions. Table 5.1 is a listing of the most often used and highest level words written for the operation of the instrument along with their functions.

Since the CPU is actively programming control of the

FORTH word

FUNCTION

5CD	Collects 50 sets of spectra, each differentiated by 3.6 degrees, and transmits this data to the 11/40 via direct serial line onto the FORTH pseudo disk LDA.FTH.
GO	Acquires a single set of spectra by executing the parameter table.
DEGREES	Rotates the polarizer XXX.XX degrees.
MODIFY	Modifies entries in the parameter table for the scan number on top of the parameter stack.
PARAM	Lists the parameter table on the terminal.
RESPTABLE	Restores the parameter table from the FORTH disk block on top of the stack.
SAVEPTABLE	Saves the current parameter table on the FORTH disk block.
IVW	Plots intensity vs. wavelength spectra on the local graphics screen.
AVW	Plots absorbance vs. wavelength spectra using 3 data files stored on disk.
IVR	Plots intensity vs. rotation angle of the polarizer on the graphics monitor.
IVT	Plots intensity vs. time on the graphics monitor.
CLEAR	Clears the graphics monitor.
DRES	Returns data to memory from FORTH disk block on top of the stack.
DSAVE	Saves data currently in memory in the FORTH disk block on top of the stack.
11DISK	Redirects mass storage device to the pseudo-disk on the 11/40.
FLOPPY	Redirects mass storage device to the local floppy disk.

TABLE 5.1 Commonly used FORTH words for Instrument Operation

next detector readout during a current data acquisition cycle, it must already have access to the desired control parameters. In order to do this, a 1 Kbyte parameter table is maintained in memory. This parameter table is a list of detector control options for up to 146 consecutive scans. For each scan, these options include: data collection [Y,N], doubling [Y,N], number of doublings [0-15], external flag set [Y,N], memory bank [0,1], address of data storage, initial integration time, and a last scan flag in the parameter table [Y,N]. The command PARAM displays this parameter table on the terminal. Parameter table options are modified by entering the scan number to be modified followed by the word MODIFY. MODIFY prompts the operator for each possible option and enters any changes for the appropriate scan in the parameter table. Because of the potentially large size of a particular set of scan parameters, it is convenient to save this parameter table if similar experiments are likely to be performed. The words RESPTABLE and SAVEPTABLE restore and save the parameter table on the local floppy disk. Once the parameter table is established as desired for an experiment, the command GO executes the entire sequence of scans (stopping at the scan marked with the last scan flag). After the data are acquired, they can be stored on the floppy disk using the word DSAVE. If another look at any particular stored data set is desired, it can be returned to memory from the floppy

di

of

u

a

p

p

t

o

n

f

d

l

e

e

e

s

v

v

v

v

v

v

v

v

v

v

v

v

disk using the word DRES.

Since local mass storage is limited, it is necessary to off-load the data quickly. Communications software utilizing a serial line connection between the microcomputer and our PDP 11/40 minicomputer is currently used. The program FORTHPIP was written by Phil Hoffman (73) for this purpose. FORTHPIP runs on the 11/40 and is capable of transferring data to or from the serial line, pseudo disks or RSX-11M unformatted data files. Pseudo disks are files maintained on any 11/40 disk which emulate a local FORTH floppy disk. By entering 11DISK, all FORTH words requiring disk I/O now communicate with the pseudo disk (named LDA.FTH) instead of the local floppy disk and remain completely transparent to the operator. To return communication to the local floppy, the word FLOPPY is entered. Several graphics commands are available such as IVW which plots an intensity vs. wavelength spectrum on the local graphics monitor of any data set currently in memory. The most powerful FORTH word written so far for this system is 5CD which collects 50 spectra, each taken after rotating the polarizer 3.6° and transmits all the data to the 11/40.

Once the data have been transmitted to the minicomputer via direct serial line transmission or floppy disk, the data analysis and reduction procedure can begin. These routines are written in FORTRAN and MACRO-11 and run on a PDP 11/40 minicomputer. A single core routine CRUNCH, makes available

several functions for the analysis of data from various types of experiments. Table 5.2 lists these subroutines and thier functions.

With the exception of the EA subroutine, all the available subroutines read in the data from RSX data files that were generated by FORTHPIP. ENTER simply reads in a single data file. This data file normally contains the ADC conversions and integration time codes for 512 points. These points do not necessarily correspond to the 512 photodiodes. ENTER is used when direct intensity values are desired as output by the LDA. EA has the same function as ENTER with the exception that an ASCII formatted data file is entered. EA is useful when a data file output by the STORE subroutine requires further analysis. AVG simply averages the data files entered- a useful function for signal averaging. The subroutine LOG reads in a data file and returns the log of each point. SUBTR subtracts any 2 data files, point by point.

ABS calculates the absorbance of the sample at each wavelength using the following equation:

$$A_{\lambda} = -\log [(S_{\lambda}-D_{\lambda}) / (R_{\lambda}-D_{\lambda})] \quad (1)$$

where A_{λ} is the absorbance at wavelength λ , S_{λ} is the signal intensity, D_{λ} is the dark current and R_{λ} is the reference intensity. Note that ABS calls both the LOG and SUBTR

SUBROUTINE	FUNCTION
ABS	Calculates the absorbance at all wavelengths using Reference, Sample and Dark current files.
AVG	Averages together multiple data sets.
DFIT	Calculates absorbance and ORD spectra from amplitude and phase information obtained by fitting 50 points at each wavelength to a \cos^2 function.
EA	Reads in any ASCII data file.
ENTER	Reads in any unformatted data file.
FIT	Calculates amplitude, phase and offset information from a single data file; basically a single wavelength DFIT.
LOG	Takes the log of any data file.
NRED	Smooths any data file using a modified Savitsky-Golay algorithm.
ORD	Calculates the observed rotation with polarizers at a fixed 45 degree angle from a reference, sample and dark current data files.
STORE	Stores any data file in an ASCII type compatible with MULPLT.
SUBTR	Subtracts any two data files.

TABLE 5.2 Data Reduction and Analysis Routines

subroutines. NRED is a data smoothing routine which uses a modified Savitsky-Golay (S-G) algorithm (56,57). Linear, quadratic and derivative smoothing functions are available with an unlimited number of passes and a 21 point window maximum. Normally, each pass through a S-G smoothing function results in the loss of a single point at each end of the data set. This limits the number of passes possible. The S-G algorithm used here has been modified to make estimates of the end points and can therefore make an unlimited number of passes. However, as the number of passes increases, errors in the end-point estimates can cause misinterpretation of the data.

The ORD subroutine calculates the observed rotation, α , in degrees from 3 data sets with the polarizers at a fixed 45° angle by the following equation:

$$\alpha_\lambda = 0.5 \cos^{-1} [((S_\lambda - D_\lambda) / (R_\lambda - D_\lambda)) - 1] - 45 \quad (2)$$

where S_λ is the signal intensity at wavelength λ , D_λ is the dark current and R_λ is the reference intensity. For the observation of large rotations ($>5^\circ$) this works well and is very fast.

DFIT is the routine which reduces the data generated by the 5CD command on the microcomputer. At each wavelength, there are 50 data points collected, each sampled at 3.6° intervals spanning a 180° sweep of the polarizer. These 50

points complete a full cycle of a \cos^2 function. By curvefitting these points to this function, the phase, amplitude and offset parameters are generated. This involves a least squares curve fitting procedure (74). By curvefitting both reference and sample spectra, absorbance and ORD spectra can be calculated using the following equations:

$$A_{\lambda} = -\log (SAMP_{\lambda} / RAMP_{\lambda}) \quad (3)$$

$$a_{\lambda} = SPHS_{\lambda} - RPHS_{\lambda} \quad (4)$$

where $SAMP_{\lambda}$ is the signal amplitude (from the curve fit) at wavelength λ , $RAMP_{\lambda}$ is the reference amplitude, $SPHS_{\lambda}$ is the signal phase and $RPHS_{\lambda}$ is the reference phase. The function of FIT is very similar to DFIT in that a curve is fit, except that the 50 points are sequential in the data file and only a single curve fit is performed. This is useful for single wavelength studies as it takes only 1/512 of the total time.

Finally, the subroutine STORE writes the resulting data file to a system device in an ASCII format compatible for input to a plotting routine MULPLT (written by Dr. T. V. Atkinson) (45). All data plots in this dissertation were generated by MULPLT.

CHAPTER VI

THE SIMULTANEOUS MEASUREMENT OF ABSORBANCE AND ORD

A. Introduction

The instrument described in this dissertation can measure both absorbance and ORD by 2 different techniques. The absorbance of a sample can easily be calculated from 3 spectra (the reference, signal and background) using equation 1 in Chapter V, or in a similar manner, the ORD can be calculated from three spectra using equation 2 in Chapter V when the angle between the polarizer and the analyzer is known to be 45 degrees. Using this procedure, however, the absorbance and ORD cannot be measured simultaneously. An alternate technique involving the rotation of the analyzer does allow for the simultaneous measurement of absorbance and ORD. This technique will be described in the next section. The detection limits achieved and the studies of the inversion of sucrose and of Vitamin B₁₂ are also described.

B. The Experimental Technique

It has long been known that the light intensity transmitted through two linear polarizers is a function of the incident light intensity and the \cos^2 of the angle at

which the polarizers are set. If the intensity is measured while the analyzer is rotated continuously, the signal detected will vary sinusoidally between 0 (when the polarizers are crossed) and some maximum value, I_{\max} , with a frequency twice that of the rotation of the analyzer. If the sample shows absorption but no optical activity, the new maximum intensity value observed, I , will be lower than the value of I_{\max} and the absorbance can be calculated using equation 3 in Chapter V. If the sample is optically active but does not absorb, the amplitude of the signal will not change ($I=I_{\max}$), but a phase difference will be observed between the sample and the reference signals. The observed rotation, α , is exactly equal to this phase difference as shown in equation 4 in Chapter V. By recording the signal intensity at several analyzer positions, both the intensity and phase information are obtained. If this signal intensity is measured at all wavelengths simultaneously, both the absorption spectrum and the ORD spectrum are measured.

To obtain these \cos^2 waves with this instrument, several spectra are acquired at different angular positions of the analyzer. The data at each wavelength are fit to a \cos^2 function (with the subroutine DFIT) by a non-linear least squares algorithm (74). The resulting amplitude, phase and offset parameters are used to calculate the absorbance and ORD spectra. The only parameter not fit by

the routine DFIT is the spacing between positions of the analyzer as this is known accurately.

Several experiments were conducted in order to determine the relationship between the number of data points taken along a 25 to 360 degree range of rotation of the analyzer and the precision of the phase information produced by the curve fitting procedure. The results are shown in Table 6.1. It is easily seen that as the total number of points acquired increases, the error decreases. However, if 500 points are to be collected at every wavelength, 500 scans need to be acquired and transmitted directly to the 11/40. At the minimum integration time, this process takes about 15 minutes and 500 Kbytes of mass storage are required. If only 25 points are collected, then a very much larger error results. As a compromise, 50 points are normally acquired. This results in a relatively low error and the data acquisition time and disk space requirements are reasonable (about 1.5 minutes at the minimum integration time and 50 Kbytes respectively). It was also observed that if the points are acquired over only a fraction of a complete cycle of the \cos^2 function, relatively large errors result from the curve fit. For these reasons, the FORTH routine 5CD collects 50 spectra over a 180° range of analyzer rotation (each point is 3.6° apart).

		Number of Points Acquired Evenly Spaced over the Degree Range Covered			
		25	50	100	500
Degree Range Covered by Analyzer	90	+1.67	+0.23		
	180	+0.33	+0.11	+0.08	+0.05
	360	+0.34	+0.10		

TABLE 6.1 Observed Precision of the Phase
Information produced by the Curve
Fitting Procedure

C. The Hydrolysis of Sucrose

Figure 6.1 shows the simultaneously obtained absorption and ORD spectra of a 0.831% aqueous sucrose solution. These spectra were obtained with a 100 μm slit width and a 15 ms integration time followed by 4 doublings. The detector and sample were at 23 degrees C and the Xe arc lamp was used. The path length used was 1 dm. As explained previously, 50 sets of spectra are acquired with each following a 3.6° rotation of the analyzer. It is easily seen that the absorbance of the sample is minimal, however, the observed rotation, α , displays the expected behavior. When the observed rotations are compared to a value calculated from a 1 term Drude equation (75), the maximum absolute error over the entire wavelength region covered by the instrument is within ± 0.03 degrees.

The hydrolysis of sucrose to give equimolar amounts of glucose and fructose is catalyzed by hydrogen ion. By monitoring the observed angle or rotation, the rate at which this reaction occurs is measured at various hydrogen ion concentrations. For this reaction, the integrated rate equation can be expressed in the following form:

$$\log(\alpha - \alpha_\infty) = -kt/2.303 + \log(\alpha_0 - \alpha_\infty) \quad (2)$$

where α is the angle of rotation at time t , α_0 is the angle

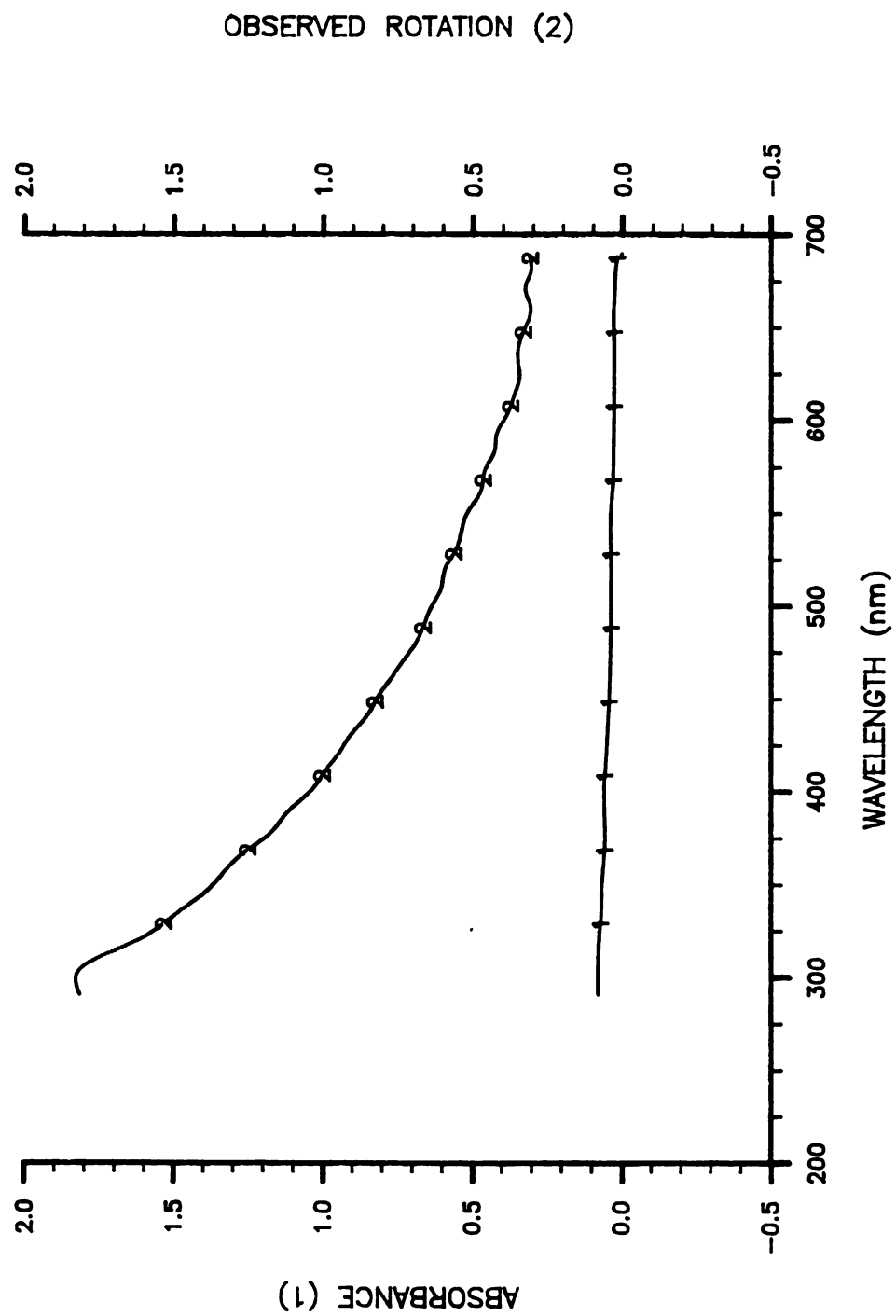


FIG. 6.1 The Absorbance and Rotational Spectra of Sucrose

of rotation at the first reading ($t=0$), and α_{∞} is the rotation when the reaction has gone to completion. Since it is difficult to obtain a reliable reading for α_{∞} experimentally, and the characteristics of the reaction are most sensitively determined by the data for the initial phases of the reaction, this quantity need not be determined.

Figures 6.2 and 6.3 show the observed rotation spectra over a time span of 4 to 32 minutes after the addition of acid (each 4 minutes apart) at the HCl concentrations of 3.0M and 1.5M respectively. Again all spectra were obtained with a 15 ms integration time followed by 4 doublings, the sample and detector were at room temperature, the slit width was 100 μm and the sucrose concentration was 25.0g/100ml. The path length used was 1 dm. All the spectra were smoothed with the modified Savitsky-Golay algorithm (5 point window and two passes).

Figure 6.4 shows the rate constant, k , at both acid concentrations versus wavelength. The rate constant was calculated with a non-linear least squares procedure. The program assumed a trial value for α_{∞} and then determined the best values of the slope, intercept, and α_{∞} to fit the above rate equation. At all points, the correlation coefficient was greater than 0.997.

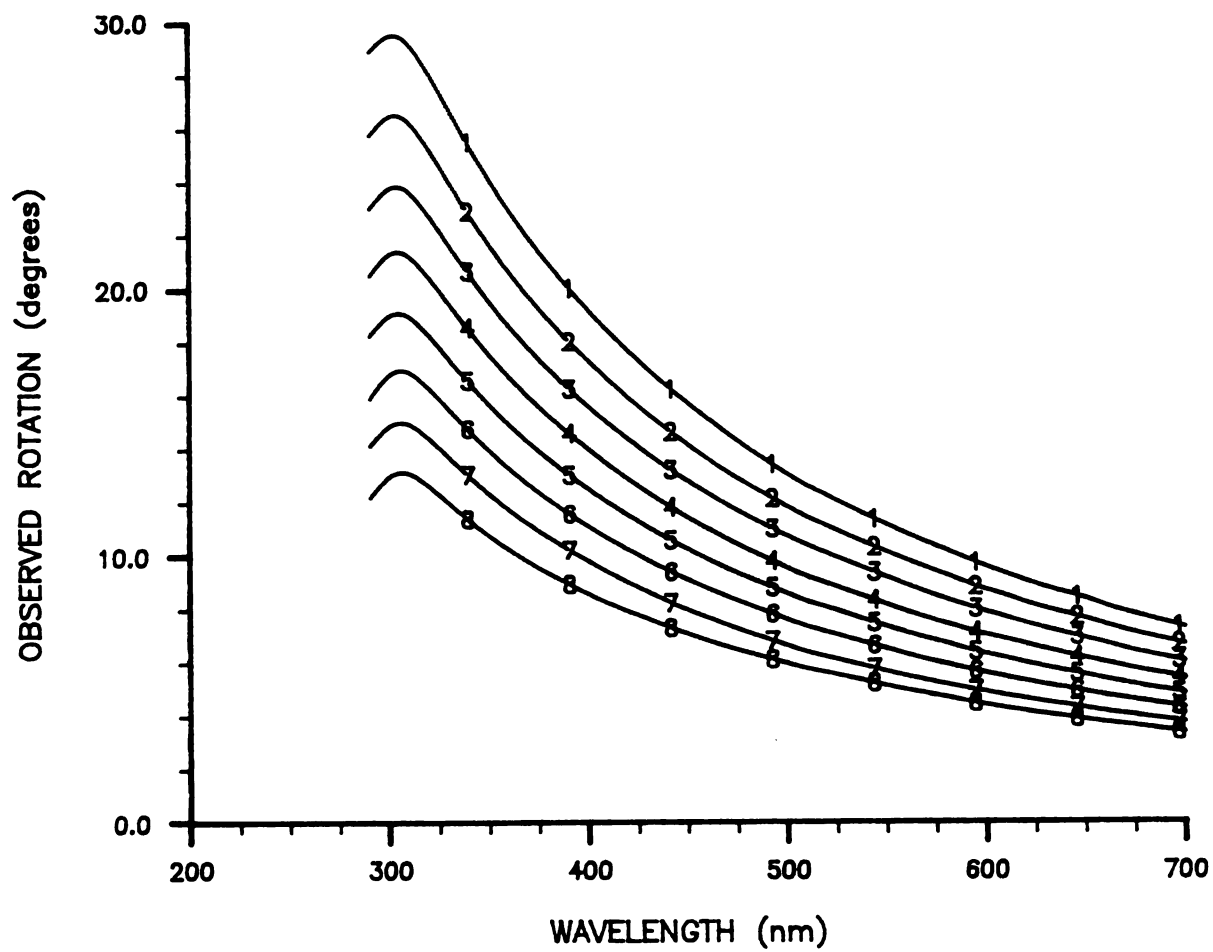


FIG. 6.2 The Hydrolysis of Sucrose, acid concentration 3.0 M

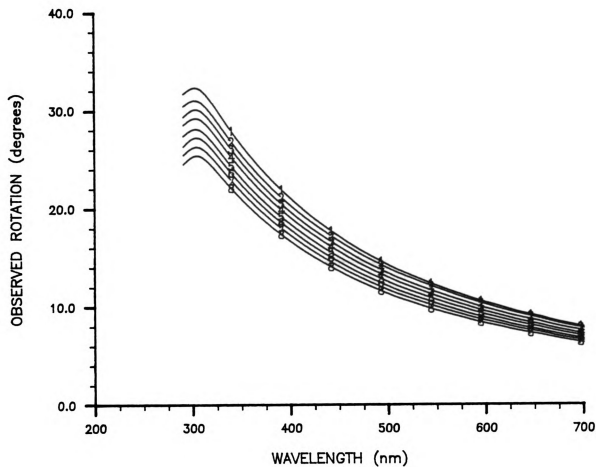


FIG. 6.3 The Hydrolysis of Sucrose, acid concentration 1.5 M

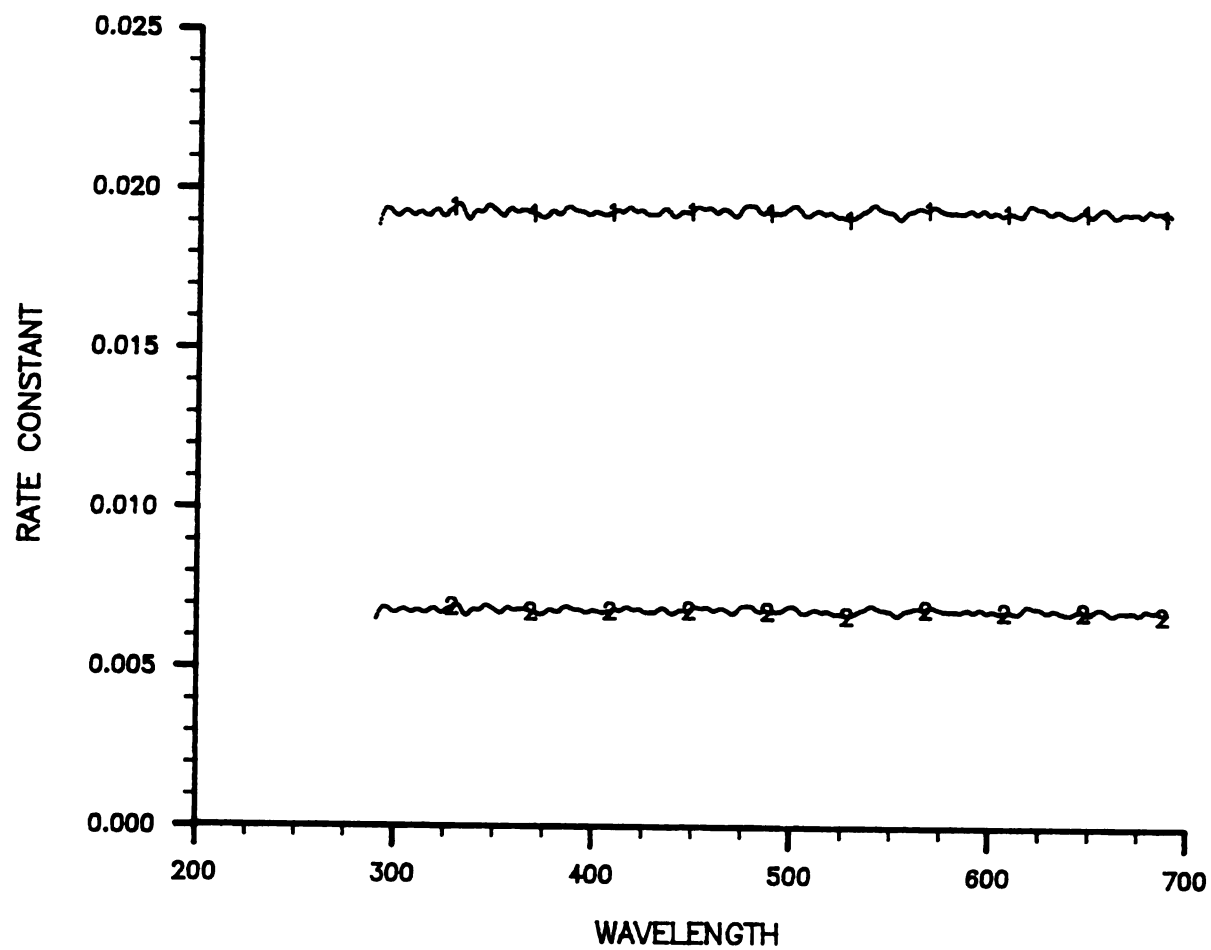


FIG. 6.4 The Rate of Hydrolysis of Sucrose at acid concentrations of 3.0 and 1.5 M versus wavelength

D. Vitamin B₁₂

The discovery of vitamin B₁₂ in 1947 culminated a worldwide scientific search that had lasted for over two decades. In 1926, Minot and Murphy had announced the successful use of whole liver in the diet for the treatment of pernicious anemia, which previously had almost always been fatal (76). This lifesaving discovery resulted in many studies all directed toward finding the anti-pernicious anemia factor in pure form. In early 1948, Merck Research Laboratories announced the discovery of pure, crystalline vitamin B₁₂ (77).

Since the discovery of vitamin B₁₂, numerous laboratory and clinical studies have provided extensive information on its chemistry, its biological functions, its clinical uses, and the technology of its production on a large scale.

The first communication on the chemical nature of vitamin B₁₂ did little more than report that the molecule contained nitrogen and phosphorus, and that it appeared to be a coordination complex with the metal cobalt (78). Today, as the result of research by many, the complete structure of vitamin B₁₂ is well known.

In 1948, vitamin B₁₂ was known only to possess high activity as a microbial growth factor and to be active clinically in producing a hematologic response in pernicious anemia. Since then, it has been shown to be essential for

normal blood formation and neural function. Evidence indicates that it is also involved in other metabolic processes and that it is required for human and animal growth (79).

Vitamin B₁₂, also known as cyanocobalamin, may be identified and assayed by its light absorption spectrum and its optical rotatory spectrum in the visible and ultraviolet regions. In an aqueous solution, absorption maxima are observed at 278, 355, and 550 nm. The solution also shows a strong Cotton effect at 355 nm (80).

Figures 6.5 and 6.6 show the absorption and rotational spectra of vitamin B₁₂ at the concentrations of 16.4, 11.5, 8.2, and 4.9 mg/ml. Each pair of spectra was acquired simultaneously using an integration time of 20 ms followed by 3 doublings. The sample cell used was 1 dm long. Each spectrum was averaged 15 times. The Xe arc lamp was used and an infrared cutoff filter was inserted into the light path in order to reduce stray light. The rotational spectra were smoothed by the modified Savitsky-Golay algorithm with a window size of 5 points. Two smoothing passes were made through each data set. Figure 6.7 shows the linear response of both the absorption and rotational spectra versus concentration at 355 nm. The deviation from Beer's law in the case of the absorbance working curve is due to the presence of stray light. This deviation occurs at an absorbance value which is lower than the value obtained from

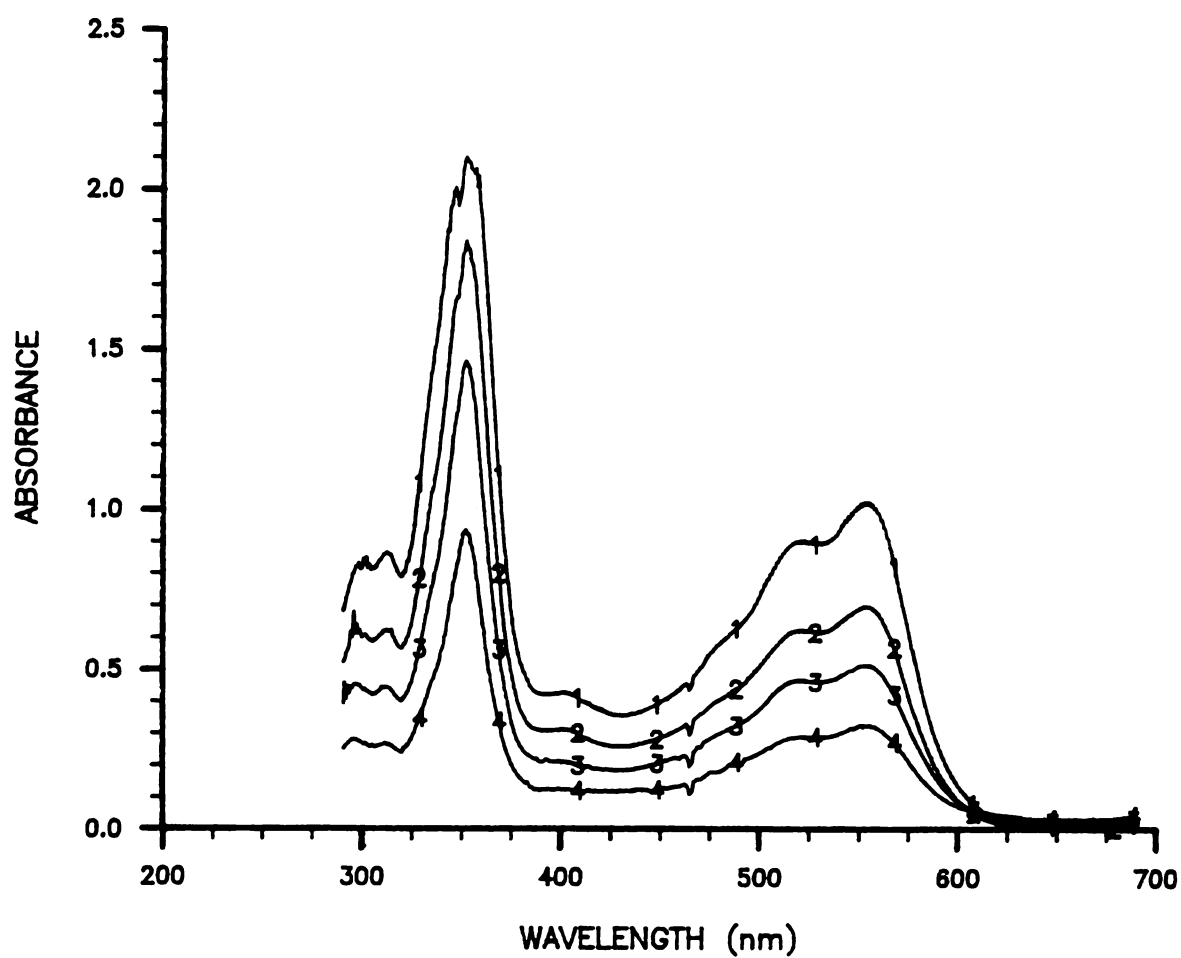


FIG. 6.5 The Absorbance Spectra of Vitamin B12 at concentrations of 16.4, 11.5, 8.2, and 4.9 mg/ml

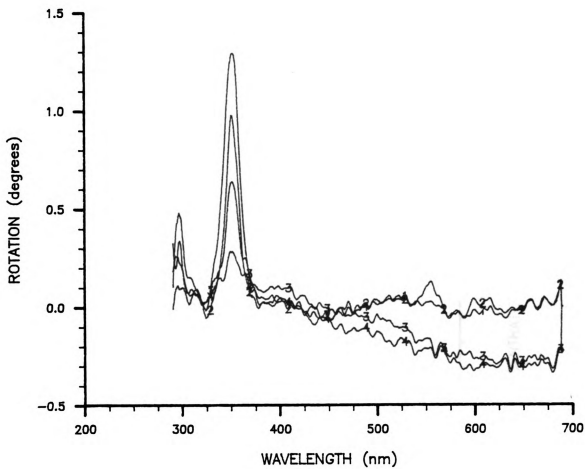


FIG. 6.6 The Rotational Spectra of Vitamin B12 at concentrations of 16.4, 11.5, 8.2, and 4.9 mg/ml

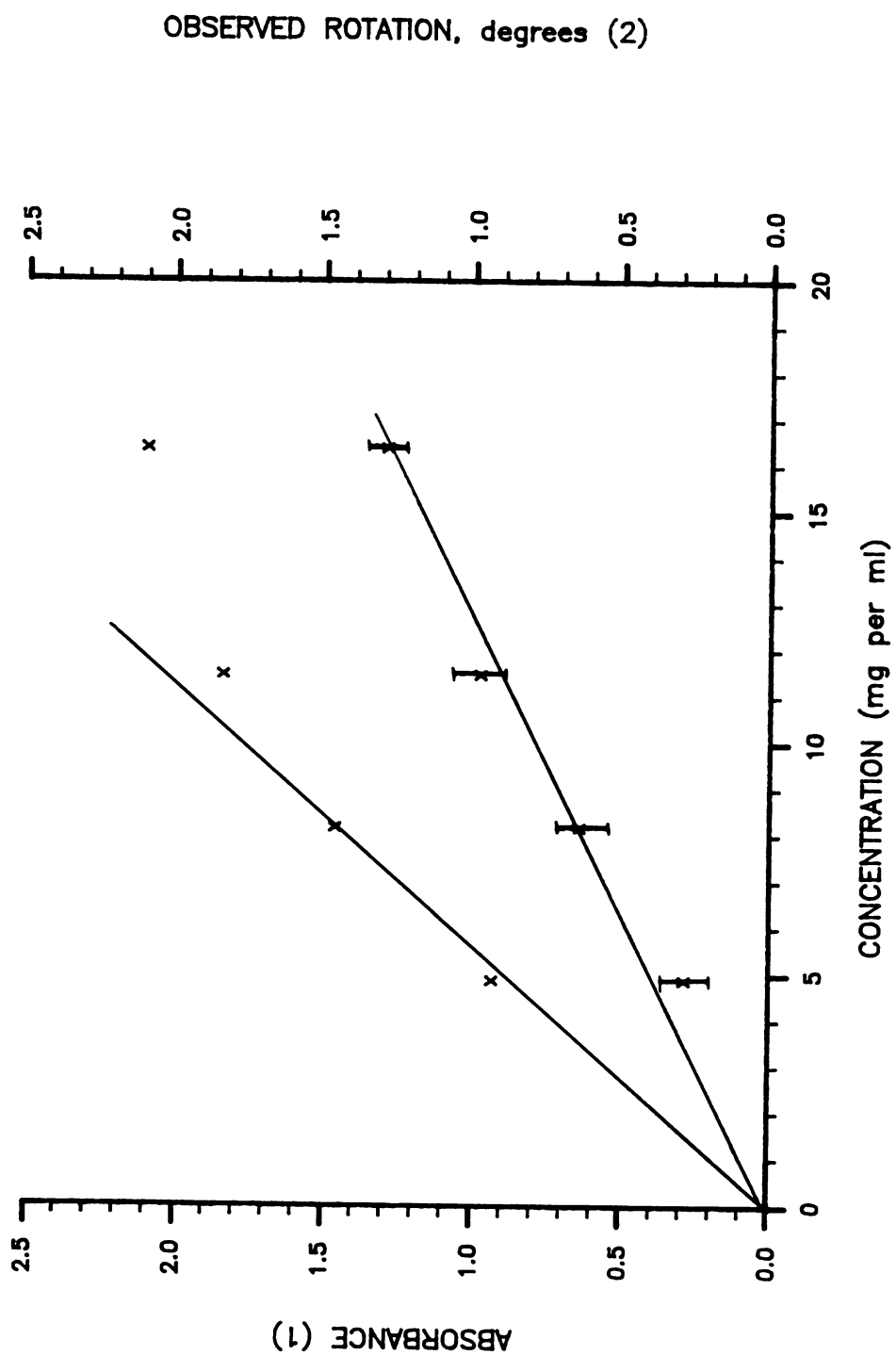


FIG. 6.7 Absorbance and Rotational Working Curves of Vitamin B12 at 355 nm

the KMnO_4 experiments. This occurs because the percent stray light at the 355 nm photodiode is greater than that at the 545 nm photodiode. Even when the absorbance exceeds 2.0, the rotational spectrum obtained is still very accurate and shows a linear response. This demonstrates that valid rotational data can still be acquired even when the sample is highly absorbing.

E. Conclusion

Array detector spectrometry offers many advantages over scanning spectrometers, especially where near simultaneous information at multiple wavelengths is required. With suitable optics and electronics, the resolution and dynamic range of modern array detector spectrometers can match that of a medium resolution scanning monochromator and PMT. It has been shown that wide dynamic range can be achieved using silicon photodiode arrays with a variable integration time technique. In a total time of only twice the longest integration time, the optimum signal measurement is made at each photodiode while memory requirements remain at a minimum.

An instrument capable of simultaneously measuring more than one physical property of a chemical system has great potential utility. Since the measurements are made simultaneously, it can be assured that the properties

measured actually refer to the same material under the same conditions. Also an instrument of this type can become cost-effective as the duplication of major pieces of equipment can be avoided. This LDA spectrometer has shown the ability to measure both absorbance and ORD spectra over the near UV and visible regions of the spectrum simultaneously. Even when the absorbance of the sample is high, valid rotational data can still be acquired.

Array detector spectrometers are somewhat more susceptible to stray light interferences than scanning monochromators because the baffling cannot be so exclusive and the use of switched in filters for various wavelength regions is precluded. Because of the many and proven advantages of this new technology, the years ahead should see great strides in both the technology and applications of array detector spectrometry.

—

APPENDIX A

INTERFACE SCHEMATICS AND TIMING DIAGRAMS

—

From Control Circuitry

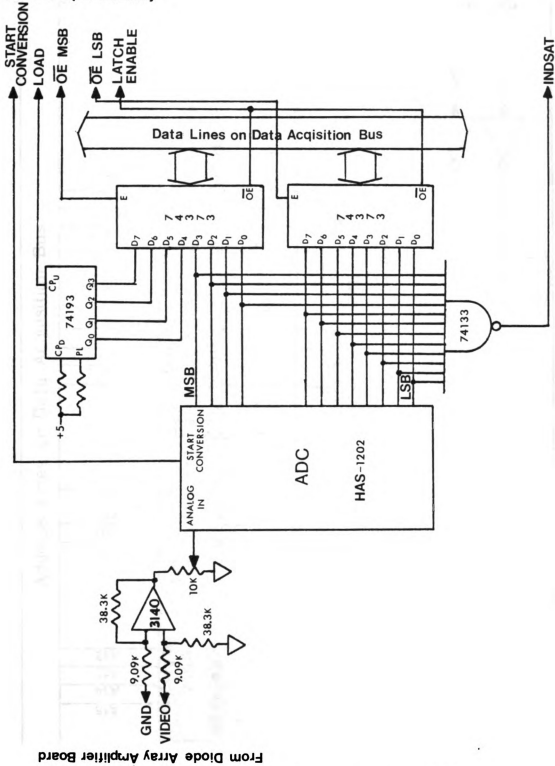


FIG. A.1 ADC Board Schematic Diagram

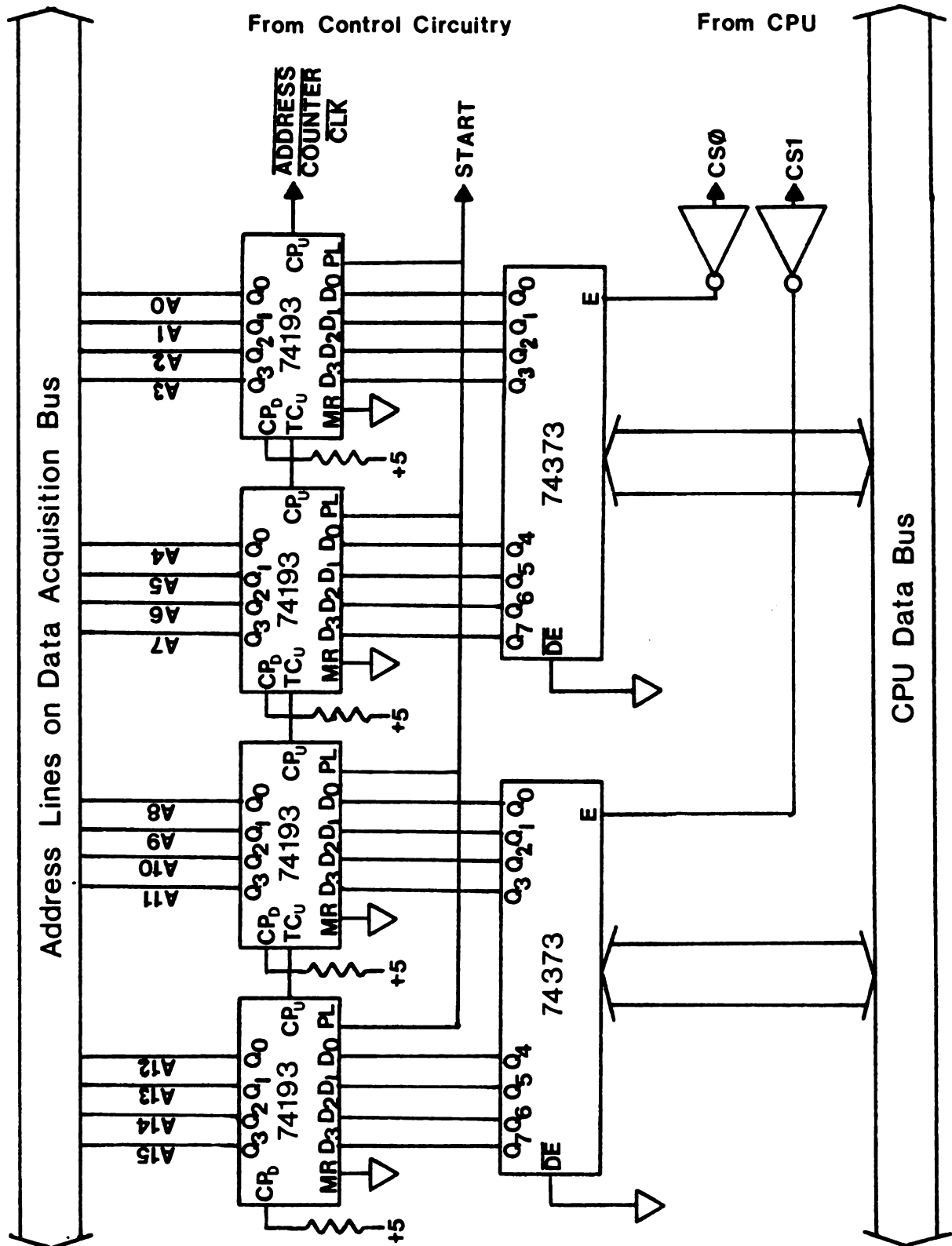


FIG. A.2 Schematic Diagram of the Address Control Circuit

FIG. A.3 Schematic Diagram of the Integration Time Control Circuit

FIG. A.4 Schematic Diagram of the Data Acquisition Control Circuit

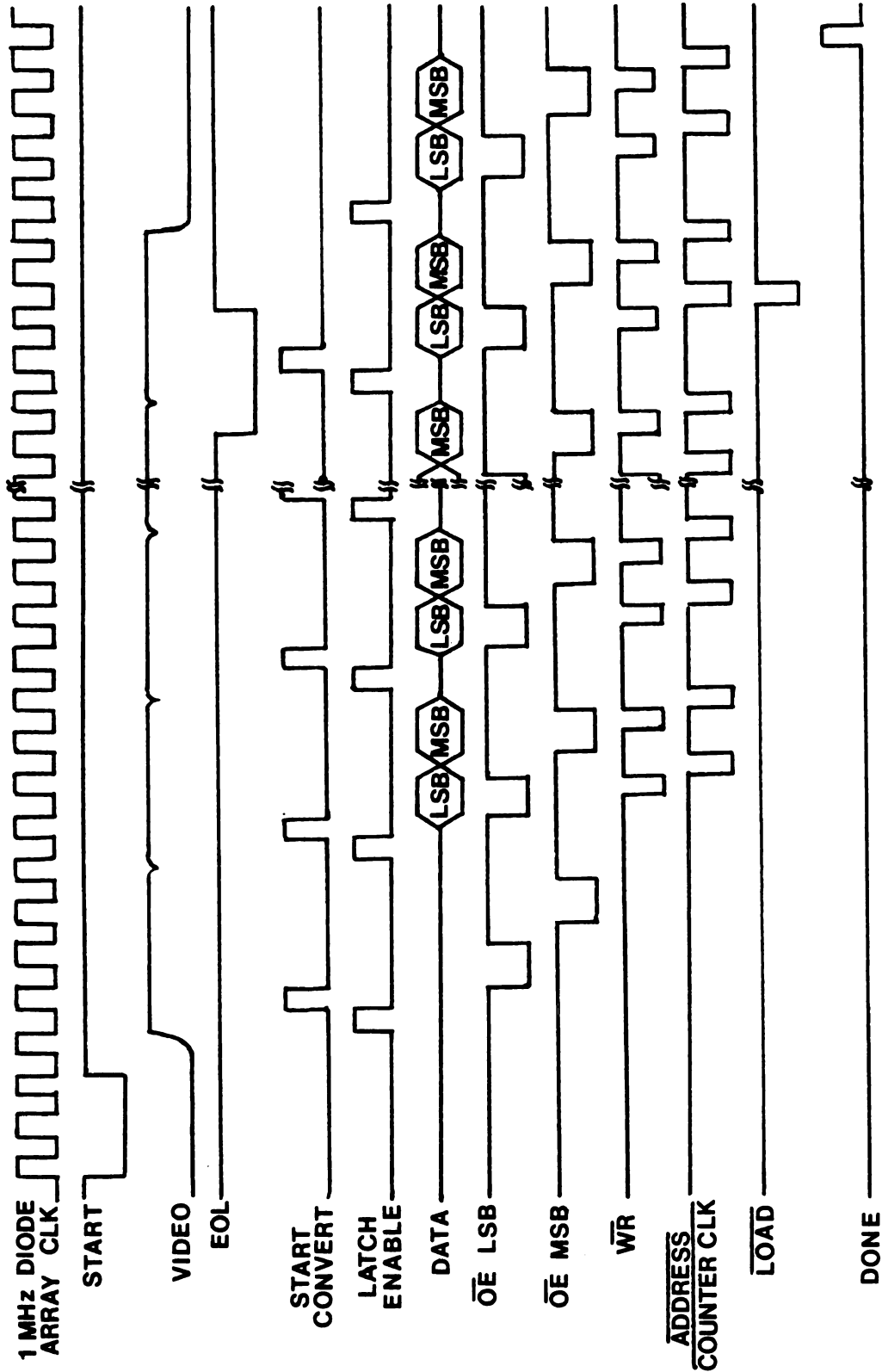


FIG. A.5 Single Scan Timing Diagram

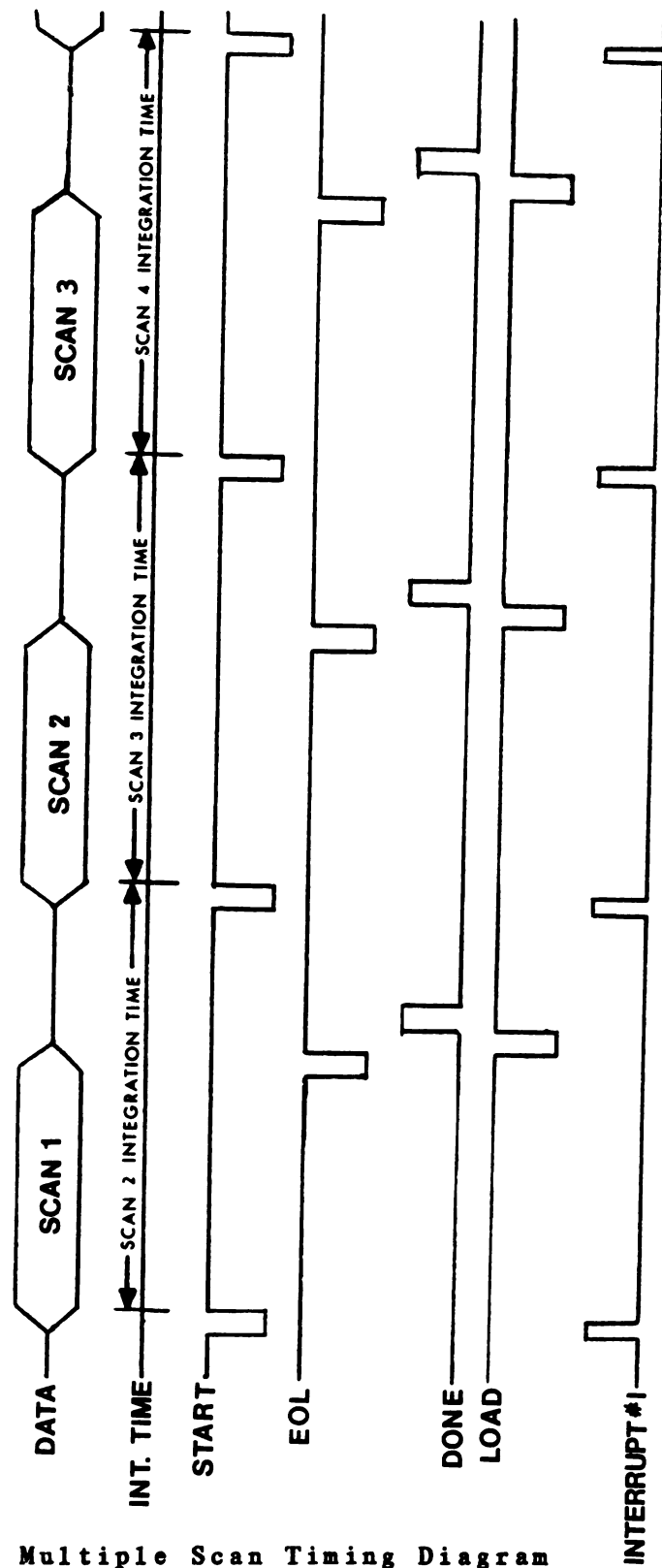


FIG. A.6 Multiple Scan Timing Diagram

APPENDIX B

SELECTED PROGRAM LISTINGS

APPENDIX B Part I - polyForth Documentation

The following tables consist of the Forth words and their function. The words are divided into four categories:

1. LDA - general system routines
2. LDA-MOTOR - polarizer rotation routines
3. LDA-PTABLE - parameter table routines for scan
setup
4. LDA-GRAPH - graphics routines

polyFORTH GLOSSARY

WORD	VOCABULARY	BLOCK	STACK	ENTERED
CMDSTR	LDA Prints out a command string for FORTHPIP to copy 1 block from the micro into the emulator file named LDA.FTH.	160	0-0	27-OCT-82
5CD	LDA Begins data acquisition sequence of 50 scans. Each scan is 3.6 degrees apart in polarizer rotation. All data is directly transmitted to the 11/40 into the LDA.FTH emulator file to be accessed by CRUNCH.	160	0-0	27-OCT-82
BANK	LDA Number on stack (0 or 1 only) is used and selects which RAM bank number the CPU is connected with.	123	1-0	25-JAN-82
CLRSCANINT	LDA Constant. Returns the address BFC0 hex. This address when selected clears a flip-flop on the interrupt controller board which is set by start of scan.	80	0-1	19-JAN-82
CMDSTR	LDA Prepare FORTHPIP command string to send to the 11/40.	160	0-0	27-OCT-82
CTRLREG	LDA Constant. Returns the address C5D0 hex. This address points to a latch in the LDA interface to which is written the control byte.	80	0-1	19-JAN-82
DACQ	LDA Checks the current scan in the parameter table and writes out the parameters for the next scan as needed. When the last scan parameters are written out, LSCANFLG is set to 1.	121	0-0	25-JAN-82
DBLOCK	LDA Variable. 2 Kbytes long used as a storage area by SGNVAVG.	145	0-1	26-JAN-82
DIRMTR	LDA Constant. Returns the address C1C8 hex. This address when selected changes the direction of rotation of the stepper motor.	80	0-1	19-JAN-82

WORD	VOCABULARY	BLOCK	STACK	ENTERED
DO-IT	LDA When called, activates the detector background task. This task allows GO to run in background mode.	122	0-0	25-JAN-82
ENDFLG	LDA Variable. Returns address of the end of the parameter table flag. Contains a 1 if end, 0 if not.	122	0-1	25-JAN-82
GETDAT	LDA Executes GO 512 times, recording data at a single wavelength only and storing consecutively at 8000 hex. Only functions on data set acquired beginning at 8000.	161	0-0	27-OCT-82
GO	LDA Enables background task, and calls DACQ, loops and waits till the entire parameter table worth of data is acquired, disables the background task, and returns control to the terminal.	123	0-0	25-JAN-82
HIADD	LDA Constant. Returns the address C5C8 hex. This address points to a latch in the LDA interface to which is written the high byte of the 2 address bytes.	80	0-1	19-JAN-82
HIGHINT	LDA Constant. Returns the address C3D0 hex. This address points to a latch in the LDA interface to which is written the high byte of the 3 int. time bytes.	80	0-1	19-JAN-82
LOADD	LDA Constant. Returns the address C5C0 hex. This address points to a latch in the LDA interface to which is written the low byte of the 2 address bytes.	80	0-1	19-JAN-82
LOWINT	LDA Constant. Returns the address C3C0 hex. This address points to a latch in the LDA interface to which is written the low byte of the 3 int. time bytes.	80	0-1	19-JAN-82
LSCANFLG	LDA Variable. Returns the addresss of the last scan flag variable. Contains a 1 if it is the last scan, 0 if not.	121	0-1	25-JAN-82

WORD	VOCABULARY	BLOCK	STACK	ENTERED
MIDINT	LDA	80	0-1	19-JAN-82
Constant. Returns the address C3C8 hex. This address points to a latch in the LDA interface to which is written the middle byte of the 3 int. time bytes.				
RECDAT	LDA	161	1-0	27-OCT-82
Records data value of single wavelength (315 nm) into the address on top of the stack.				
SCANCOUNT	LDA	80	0-1	19-JAN-82
Constant. Returns the address C5D8 hex. This address points to a latch in the LDA interface from which is read the current autoranging count (D7-D4).				
SGNAVG	LDA	145	1-0	26-JAN-82
Takes as many scans of the parameter table as are on top of the stack. These scan results are averaged from the address at DATAADDRESS and the result is stored beginning at DATAADDRESS.				
STPMTR	LDA	80	0-1	19-JAN-82
Constant. Returns the address C1C0 hex. This address when selected steps the stepper motor.				
THRU	LDA	9	2-0	19-JAN-82
Requires 2 FORTH disk block numbers on the stack. Loads all disk blocks between these inclusive and lists the loaded blocks on the terminal.				
UPCMD	LDA	160	0-0	27-OCT-82
Sends the FORTHPIP command string to the 11/40 for data transmission.				
W-ADDRESS	LDA	120	0-0	25-JAN-82
When called writes out the 2 address bytes to the hardware latches. Both bytes in the parameter table are pointed to by TPOINTER.				
W-CONTROL	LDA	120	0-0	25-JAN-82
Writes out the control byte to the hardware latch. This byte is pointed to in param table by TPOINTER.				
W-INTTIME	LDA	120	0-0	25-JAN-82
When called, writes out the integration time to the hardware latches. The 3 bytes written out are pointed to in the parameter table by the value at TPOINTER.				

polyFORTH GLOSSARY

WORD	VOCABULARY	BLOCK	STACK	ENTERED
CDIR	LDA-MOTOR Changes the direction of polarizer rotation when called.	85	0-0	19-JAN-82
DEGREES	LDA-MOTOR Requires a signed 32 bit number representing the number of degrees to rotate the polarizer and direction. Input number format must be XXX.XX.	85	2-0	19-JAN-82
ROTATE	LDA-MOTOR Calls STEP number of times on top of stack for motor rotation.	85	1-0	19-JAN-82
SPIN	LDA-MOTOR Rotates polarizer indefinitely. Must stop using break key. Does not remember current position.	85	0-0	19-JAN-82
STEP	LDA-MOTOR Steps the stepper motor once. Includes a time delay to not exceed maximum motor speed. The polarizer is rotated 1/100 degrees per step.	85	0-0	19-JAN-83

polyFORTH GLOSSARY

WORD	VOCABULARY	BLOCK	STACK	ENTERED
#RNGS	LDA-PTABLE	93	0-0	20-JAN-82
	For the current scan, the number of autoranges is calculated (T4BITS) and printed out right justified in the param table.			
'	LDA-PTABLE	94	0-0	20-JAN-82
	Inserts a character into numeric string using FORTH word HOLD.			
?ADDRNG	LDA-PTABLE	114	1-2	22-JAN-82
	Checks number on stack for valid data storage address (8000-9FFF H). If -1 is on stack a -1 and 1 are returned. If invalid address is on stack, a 0 is returned. If valid address the address and a 1 are returned.			
?INTRNG	LDA-PTABLE	112	2-3	22-JAN-82
	Requires double precision number on stack. If less than LINT returns a 0, if greater than HINT, returns a 0, if <0 then a 1 is returned, if between LINT and HINT, the number returns with a 1 on top of stack.			
ADDRESS	LDA-PTABLE	94	0-0	20-JAN-82
	Prints out the hex address of the current scan pointed to by TPOINTER. Uses bytes 4 and 5 of the 7 byte parameter list for each scan.			
ADDSET	LDA-PTABLE	114	0-0	22-JAN-82
	Prompts the user for the address of data storage. If a CR is entered, no changes made, otherwise the new address is entered into the param table.			
ARNG?	LDA-PTABLE	92	0-0	20-JAN-82
	Checks bit 1 of scan pointed at by TPOINTER. If set prints a Y, otherwise N. This bit controls whether data is autoranged or not this scan.			
AUTORNGSET	LDA-PTABLE	110	0-0	22-JAN-82
	Prompts user for autoranging bit set. If yes, prompts for number of autoranges. CR causes no change.			
BANK#	LDA-PTABLE	92	0-0	20-JAN-82
	Checks bit 3 of current scan and if set prints a 1, otherwise 0. This bit determines RAM bank for data storage.			

WORD	VOCABULARY	BLOCK	STACK	ENTERED
BANK#SET	LDA-PTABLE	111	0-0	22-JAN-82
	Prompts user for RAM bank number. If any other character than 0 or 1 are entered, current entry remains unchanged.			
CDATA?	LDA-PTABLE	92	0-0	20-JAN-82
	Checks bit 0 of current scan and prints Y if set. This bit controls data acquisition enable.			
CDATASET	LDA-PTABLE	110	0-0	22-JAN-82
	Prompts the user if the collect data bit should be set. Recognizes Y or N. Entry left unchanged with another other value.			
EFLG?	LDA-PTABLE	92	0-0	20-JAN-82
	Checks bit 2 of the current scan, if set prints a Y, otherwise N. This bit controls the external flag signal.			
EFLGSET	LDA-PTABLE	111	0-0	22-JAN-82
	Prompts the user for external flag bit set. Recognizes Y or N. Leaves entry unchanged with any other character.			
ENDSCAN	LDA-PTABLE	115	0-0	22-JAN-82
	Clears the CDATA, AUTORNG, and #RNGS bits for the last scan of the param table. Requests and sets the final integration time.			
F3.3	LDA-PTABLE	94	2-0	20-JAN-82
	Prints out the double precision number on top of stack in FORTRAN format F3.3. The number is printed out right justified in a 7 space field.			
HEADER	LDA-PTABLE	91	0-0	19-JAN-82
	Prints out a header for the param table listing.			
HINT	LDA-PTABLE	112	0-2	22-JAN-82
	2Constant. Value of 268.435 is returned on the stack. This number is used for max. int. time check.			
INPUT	LDA-PTABLE	100	0-2	22-JAN-82
	Inputs numerics from terminal and places on stack (10 chars) using NUMBER. If CR is entered, a -1 is put on the stack.			

WORD	VOCABULARY	BLOCK	STACK	ENTERED
INTSET	LDA-PTABLE	113	0-0	22-JAN-82 Prompts user for int time of current scan. Number must be entered in format XXX.XXX. If CR entered, no change is made. The coded integration time is entered into the param table.
INTTIME	LDA-PTABLE	94	0-0	20-JAN-82 Prints out the integration time of current scan. Uses bytes 1-3 of the 7 byte parameter list for each scan.
LINT	LDA-PTABLE	112	0-2	22-JAN-82 Constant. Value of 0.003 is returned on stack. This is used as a minimum integration time check.
LSCAN?	LDA-PTABLE	95	0-0	20-JAN-82 Prints out a Y if bit 7 of byte 7 of current scan is set. N otherwise.
LSCANSET	LDA-PTABLE	111	0-0	22-JAN-82 Prompts the user for last scan in param table. Recognizes Y or N. No change occurs with any other character.
MODIFY	LDA-PTABLE	115	1-0	22-JAN-82 Modifies the scan number on top of the stack. Will alert user to an out of range scan number. Calls all param table set routines.
PARAM	LDA-PTABLE	96	0-0	22-JAN-82 Prints out the entire scan parameter table on the terminal. Includes all scans until the last scan as well as final integration time.
PKEY	LDA-PTABLE	100	0-1	22-JAN-82 Expects 1 character input, returns 0 if char was a CR and the ASCII value of any other character.
PTABLE	LDA-PTABLE	90	0-1	19-JAN-82 Variable. Returns the address of a 1 Kbyte variable used to store scan parameters. Number of scans possible is equal to the size of this array divided by 7.
RESPTABLE	LDA-PTABLE	101	1-0	22-JAN-82 Restore the parameter table from the Forth disk block.
SAVEPTABLE	LDA-PTABLE	101	1-0	22-JAN-82 Save the param table on the Forth disk block.

WORD	VOCABULARY	BLOCK	STACK	ENTERED
SCAN#	LDA-PTABLE	92	0-0	20-JAN-83
Prints out the current scan number right justified in a 4 wide field. The current scan number is calculated by dividing the TPOINTER address by the 7 byte scan param set.				
T4BITS	LDA-PTABLE	93	1-1	20-JAN-82
Takes the number on the stack, and returns the value of the top 4 bits.				
TPOINTER	LDA-PTABLE	90	0-1	19-JAN-83
Variable. Contains address which points into the parameter table array.				

polyFORTH GLOSSARY

WORD	VOCABULARY	BLOCK	STACK	ENTERED
#DEGREES	LDA-GRAPH	131	0-2	26-JAN-83
	Variable. Returns an address of a 4 byte variable used to store the number of degrees of polarizer rotation (XXX.XX) for an intensity vs. rotation plot.			
#SCANS	LDA-GRAPH	131	0-1	26-JAN-83
	Variable. Returns an address used to store the number of scans taken at a single wavelength being displayed in a plot (one such as W vs T time=#scans).			
3DRES	LDA-GRAPH	132	3-0	26-JAN-83
	Restores 3 Kbytes of data from the 3 disk blocks. The data is returned to memory beginning at DATAADDRESS.			
?POINTPLOT	LDA-GRAPH	131	0-1	26-JAN-83
	Variable. Returns an address, if contents are a 1, the point is a point plot, if 0, a histogram.			
ABSORBANCE	LDA-GRAPH	133	0-0	26-JAN-83
	Prints the word absorbance on the terminal. Used by the axis labeling routines.			
AVW	LDA-GRAPH	142	3-0	26-JAN-83
	Plots the abs vs. wavelength of 3 data files on the graphics monitor. The first number is the block of the dark current scan, the second is the reference, and the third is the sample. Data is restored beginning at DATAADDRESS.			
AYLABEL	LDA-GRAPH	134	0-0	26-JAN-83
	Labels the y axis on the graphics screen ABSORBANCE.			
CALCDAT	LDA-GRAPH	140	1-1	26-JAN-83
	Calculates the value of the data point by getting the number of ranges and dividing the low 12 bits by 2 to the power of the number of ranges -1. The 16 bit coded data is on top of the stack, the result actual data value is returned.			

WORD	VOCABULARY	BLOCK	STACK	ENTERED
CALCTICS	LDA-GRAPH	136	1-2	26-JAN-83
Requires the difference between the minimum and maximum tic mark labels. Returns the distance between large tic marks and small tic marks (Small on top of stack).				
CX-AXISLABEL	LDA-GRAPH	133	0-0	26-JAN-83
Determines the center for the x-axis label and sets the cursor using <XYCUR> to that position on the monitor.				
CY-AXISLABEL	LDA-GRAPH	133	0-0	26-JAN-83
Determines the center for the y-axis label and sets the cursor using <XYCUR> to that position on the monitor.				
DATAADDRESS	LDA-GRAPH	131	0-1	26-JAN-83
Variable. Returns an address whose contents contain the address of the scan of interest (RAM) from which the plot is made.				
DATAPLOT	LDA-GRAPH	140	2-0	26-JAN-83
Plots a single data point (x,y on stack).				
DRES	LDA-GRAPH	132	1-0	26-JAN-83
Restores 1 Kbytes of data from the disk block on top of stack. Data is returned to memory whose starting address is stored in DATAADDRESS.				
DSAVE	LDA-GRAPH	132	1-0	26-JAN-83
Saves 1 Kbytes of data pointed to by the address stored at DATAADDRESS in the disk block on top of the stack.				
EXPONENT	LDA-GRAPH	130	0-1	26-JAN-83
Variable. Returns an address of a variable which stores an exponent. Used by POWER.				
GRNG	LDA-GRAPH	140	1-1	26-JAN-83
Gets the number of autoranges used at that wavelength. Data point is on top of stack, number of ranges are returned.				
INTENSITY	LDA-GRAPH	133	0-0	26-JAN-83
Prints the word INTENSITY on the terminal. Used by the axis labeling routines.				

WORD	VOCABULARY	BLOCK	STACK	ENTERED
IVR	LDA-GRAPH	141	0-0	26-JAN-83
	Plots the intensity versus rotation as the polarizer is rotated DEGREES between each scan. The wavelength plotted is stored at WLENGTH and the number of scans taken is stored at #SCANS.			
IVT	LDA-GRAPH	141	0-0	26-JAN-83
	Plots the intensity vs. time as each scan is taken. The wavelength plotted is stored at WLENGTH and the number of scans is stored at #SCANS.			
IVW	LDA-GRAPH	141	0-0	26-JAN-83
	Plots the intensity versus wavelength of the data at the address at DATAADDRESS on the graphics monitor.			
IYLABEL	LDA-GRAPH	134	0-0	26-JAN-83
	Labels the y-axis on graphics monitor INTENSITY.			
LXTICS	LDA-GRAPH	135	0-0	26-JAN-83
	Labels the x tic marks on the monitor.			
LYTICS	LDA-GRAPH	135	0-0	26-JAN-83
	Labels the y tic marks on the monitor.			
ORD	LDA-GRAPH	133	0-0	26-JAN-83
	Prints the word ORD on the terminal. Used by the axis labeling routines.			
OVW	LDA-GRAPH	142	3-0	26-JAN-83
	Plots the light intensity vs. wavelength of 3 data files on the graphics screen. The first num is the dis block of the dark current, the second the reference and the third is the sample. Data is restored beginning at DATAADDRESS.			
OYLABEL	LDA-GRAPH	134	0-0	26-JAN-83
	Labels the y-axis on the graphics screen ORD.			
POWER	LDA-GRAPH	130	2-1	26-JAN-83
	Returns the value x to the power of y. Y is on top of the stack, x below it. The result is only 16 bits.			
ROTATION	LDA-GRAPH	133	0-0	26-JAN-83
	Prints the word ROTATION on the terminal. Used by the axis labeling routines.			

WORD	VOCABULARY	BLOCK	STACK	ENTERED
RXLABEL	LDA-GRAPH Labels the x-axis monitor.	134	0-0 ROTATION	26-JAN-83 on the graphics
SETUP	LDA-GRAPH Draws axes, tic marks and labels for the current field.	140	0-0	26-JAN-83 on the monitor
TIME	LDA-GRAPH Prints the word TIME on the terminal. axis labeling routines.	133	0-0	26-JAN-83 Used by the
TITLE	LDA-GRAPH The string following the command displayed centered on the top of the monitor.	131	0-0 TITLE	26-JAN-83 is
TRANSMITTANCE	LDA-GRAPH Prints the word TRANSMITTANCE on the Used by the axis labeling routines.	133	0-0	26-JAN-83 terminal.
TX-AXIS	LDA-GRAPH Draws the tic marks for the x-axis on the screen. Uses CALCTICS, axis limits and XTICS.	136	0-0	26-JAN-83 on the graphics
TXLABEL	LDA-GRAPH Labels the x-axis on the graphics screen TIME.	134	0-0	26-JAN-83
TY-AXIS	LDA-GRAPH Draws the tic marks for the y-axis on the Uses CALCTICS, axis limits and YTICS.	136	0-0	26-JAN-83 monitor.
TYLABEL	LDA-GRAPH Labels the y-axis on the graphics TRANSMITTANCE.	134	0-0	26-JAN-83 screen
WAVELENGTH	LDA-GRAPH Prints the word WAVELENGTH on the by the axis labeling routines.	133	0-0	26-JAN-83 terminal. Used
WLENGTH	LDA-GRAPH Variable. Returns address which stores the wavelength in nm used in plots versus time or rotation.	131	0-1	26-JAN-83
WXLABEL	LDA-GRAPH Labels the x-axis on the graphics WAVELENGTH.	134	0-0	26-JAN-83 screen

APPENDIX B Part II - PDP 11/40 Software

The following programs perform the data reduction and analysis required on the PDP 11/40 minicomputer.

-

```

C      NRED.FTN
C
C      Subroutine used to smooth a data file.
C      Calls SMOOTH (written by P. Hoffman)
C      which uses a Savitsky-Golay smoothing algorithm.
C      REALS(510,1)=smoothed (REALS(510,1)
C
C      P. Aiello      6/9/82
C
C      SUBROUTINE NRED(N)
C
C      Variable declarations
C
C      COMMON /PSTR/ STR
C      COMMON /RDATA/ REALS
C      REAL REALS(512,3), COEFFS(21,21), TMP(21), POINTS(510)
C      INTEGER NUM, NCOEFF, IWIN, IFUNC, ITER, IERR, N, LEN
C      BYTE STR(82)
C
C      Initialize number of coefficients and number of dat
C
C      NUM=510
C      NCOEFF=21
C
C      Determine if the filespec is to be requested
C
C      GO TO (10,20)N
C
C      Obtain input file and put in array REALS(510,1)
C
C      10  CALL INPUT(7,1,IERR)
C          IF (IERR .EQ. 0) GO TO 20
C          GO TO 10
C
C      Now request for the smoothing function parameters
C
C      GO TO 20
C      20  CALL PROMPT(12,STR,LEN)
C          DECODE(LEN,900,STR)IWIN,IFUNC,ITER
C      900  FORMAT(4I7)
C
C      Copy REALS(510,1) to POINTS(510)
C
C      DO 30 I=1,510
C          POINTS(I)=REALS(I,1)
C      30  CONTINUE
C
C      And call SMOOTH
C
C      CALL SMOOTH(POINTS,NUM,COEFFS,TMP,NCOEFF,IWIN,IFUNC
C      5      ITER,IERR)
C      IF(IERR .EQ. 0) GO TO 40
C      WRITE(5,910)IERR

```



```

910      FORMAT(' Subroutine SMOOTH error = ',I7)
      GO TO 60
C
C      Now copy smoothed data back the array REALS(I,1)
C
40      DO 50 I=1,510
      REALS(I,1)=POINTS(I)
50      CONTINUE
C
C      And go home
C
60      RETURN
      END
C      LOG.FTN
C
C      Subroutine which takes the log(10)
C      of input file which is requested or of
C      data already in array REALS(512,1) if parameter
C      N = 2.  If N = 1, asks for input filespec.
C
C      P. Aiello          5/14/82
C
      SUBROUTINE LOG(N)
C
C      Variable declarations
C
      COMMON /RDATA/ REALS
      REAL REALS(512,3)
      INTEGER IERR,N
C
C      Determine if filespec is to be requested
C
      GO TO (10,20)N
C
C      Request input filespec and put into array REALS
C
10      CALL INPUT(3,1,IERR)
      IF (IERR .EQ. 0)GO TO 20
      GO TO 10
C
C      Calculate the log and store in array REALS(512,1)
C
20      DO 30 I=1,512
      REALS(I,1)=FULLOG(REALS(I,1))
30      CONTINUE
C
C      And go home
C
      RETURN
      END
C
C
C      FULLOG

```

```

C
C      Function which takes the base 10 log of
C      parameter X.  Returns 0 if X<=0.
C
      FUNCTION FULLOG(X)
      IF(X .LE. 0.0)GO TO 10
      FULLOG=ALOG10(X)
      RETURN
10    FULLOG=0.0
      RETURN
      END
C      GSM.FTN
C
C      Subroutine which questions user if the data in
C      the array REALS(512,1) is to be stored, smoothed
C      stored again.  STORE is called, then
C      NRED is called and also STORE so that the smoothed
C      data can be stored immediately
C
      SUBROUTINE GSM
C
C      Variable declarations
C
      COMMON /PSTR/ STR
      COMMON /RDATA/ REALS
      REAL REALS(512,3)
      BYTE ANS(2),STR(82)
      INTEGER LEN
C
C      Ask if data is to be stored
C
      CALL PROMPT(15,STR,LEN)
      DECODE(LEN,900,STR)ANS
900   FORMAT(2A1)
      IF(LEN .EQ. 0)GO TO 10
      IF(ANS(1) .EQ. 'Y')GO TO 20
C
C      Ask if data is to be smoothed and then stored
C
      CALL PROMPT(16,STR,LEN)
      DECODE(LEN,900,STR)ANS
      IF(LEN .EQ. 0)GO TO 40
      IF(ANS(1) .EQ. 'Y')GO TO 30
      RETURN
20    CALL STORE
      GO TO 10
30    CALL NRED(2)
      CALL STORE
40    RETURN
      END
C      INPUT.FTN
C
C      Subroutine which requests for the file to be input,

```

```

C      reads in the data into the array REALS(SET)
C      and closes the input file.
C
C      P. Aiello      5/14/82
C
C      SUBROUTINE INPUT(PMPT,SET,IERR)
C
C      Variable declarations
C
C      COMMON /PSTR/ STR
C      COMMON /RDATA/ REALS
C      REAL REALS(512,3)
C      INTEGER INTS(32),PMPT,IERR,SET,LEN
C      BYTE FILE(32),STR(82)
C      IERR=0
C
C      Request the filespec
C
C      CALL PROMPT(PMPT,STR,LEN)
C
C      Read in the filespec
C
C      DECODE(LEN,900,STR)LENGTH,FILE
900    FORMAT(G,32A1)
C      FILE(LENGTH+1)=0
C
C      If a carriage return is entered, simply return.
C      This is useful for the arrays to remain unchanged
C      and other subroutines to be executed on this data
C
C      IF(LENGTH .EQ. 0)GO TO 70
C
C      Open the file with OPEN statement
C
C      OPEN(UNIT=3,NAME=FILE,TYPE='OLD',
5        CARRIAGECONTROL='LIST',READONLY,ERR=40)
C
C      Read in the input file, calculate the bottom 12 and
C      and top 4 bits, calculate the data point value
C      and store in the array REALS(SET)
C
C      DO 20 I=1,16
C      J=(I-1)*32
C      READ(3,910,ERR=50)(INTS(K), K=1,32)
910    FORMAT(32A2)
C      DO 10 L=1,32
C      REALS(J+L,SET)=FLOAT(IBOT12(INTS(L)))
5        /2.**FLOAT(ITOP4(INTS(L)))
C      CONTINUE
10    CONTINUE
20
C      Close the input file
C

```

```

30      CLOSE(UNIT=3,ERR=60)
C
C      And go home
C
      RETURN
C
C      Send out any error messages
C
40      WRITE(5,920)
      IERR=-1
920     FORMAT(' ERROR ON OPEN')
      GO TO 5
50      WRITE(5,930)
      IERR=-1
930     FORMAT(' ERROR DURING READ')
      GOTO 30
60      WRITE(5,940)
      IERR=-1
940     FORMAT(' ERROR ON CLOSE')
70      RETURN
      END
C      ABSORB.FTN
C
C      Subroutine used to calculate the absorbance
C      of a data file using and requesting for
C      3 files: Dark, Ref, and Sample
C
C      P. Aiello    5/14/82
C
      SUBROUTINE ABSORB
C
C      Variable declarations
C
      COMMON /RDATA/ REALS
      REAL REALS(512,3)
      INTEGER IERR
C
C      Request for and read in the reference data file
C      into array REALS(512,1)
C
10     CALL INPUT(4,1,IERR)
      IF(IERR .EQ. 0)GO TO 20
      GO TO 10
C
C      Request for and read in the dark current data file
C      into array REALS(512,2)
C
20     CALL INPUT(5,2,IERR)
      IF (IERR .EQ. 0) GO TO 30
      GO TO 20
C
C      Subtract the dark current from the reference file
C      and the result is returned in the array REALS(512,1

```

```

C
30      CALL SUBTR(2)
C
C      Copy REF-DARK to array REALS(512,3)
C
      DO 40 I=1,512
      REALS(I,3)=REALS(I,1)
40      CONTINUE
C
C      Request for and read in the sample file into the
C      array REALS(512,1)
C
50      CALL INPUT(6,1,IERR)
      IF(IERR .EQ. 0)GO TO 60
      GO TO 50
C
C      Subtract SAM-DARK with result into REALS(512,1)
C
60      CALL SUBTR(2)
C
C      Now take the -log(SAM-DARK/REF-DARK)
C      In other words -ALOG10(REALS(512,1)/REALS(512,3))
C
      DO 70 I=1,512
      REALS(I,1)=REALS(I,1)/REALS(I,3)
      IF(REALS(I,1) .LE. 0.)REALS(I,1)=1.
70      CONTINUE
      CALL LOG(2)
      DO 80 I=1,512
      REALS(I,1)=-REALS(I,1)
80      CONTINUE
C
C      And go home
C
      RETURN
      END
C
      SUBTR.FTN
C
C      Subroutine which subtracts two files
C      Parameter N = 1 then filespecs are requested
C      If N = 2 filespecs are not requested and the
C      subtraction is executed.
C      REALS(512,1)=REALS(512,1)-REALS(512,2)
C
C      P. Aiello      5/14/82
C
      SUBROUTINE SUBTR(N)
C
C      Variable declarations
C
      COMMON /RDATA/ REALS
      REAL REALS(512,3)
      INTEGER IERR,N

```

```

C
C      Determine if filespecs are needed
C
C      GO TO (10,30)N
C
C      Request for first filespec
C      Data is returned in array REALS(512,1)
C
10    CALL INPUT(10,1,IERR)
      IF (IERR .EQ. 0)GO TO 20
      GO TO 10
C
C      Request for second filespec
C      Data is returned in array REALS(512,2)
C
20    CALL INPUT(11,2,IERR)
      IF (IERR .EQ. 0)GO TO 30
      GO TO 20
C
C      Do the subtraction REALS(512,1)=1-2
C
C
30    DO 40 I=1,512
      REALS(I,1)=REALS(I,1)-REALS(I,2)
40    CONTINUE
C
C      And go home
C
      RETURN
      END
C      ORD.FTN
C
C      Subroutine which calculates the observed
C      rotation in degrees of a sample while the
C      polarizers are set at a known 45 degree angle.
C
C      P. Aiello      5/14/82
C
      SUBROUTINE ORD
C
C      Variable declarations
C
      COMMON /RDATA/ REALS
      REAL REALS(512,3)
      INTEGER IERR
C
C      Request for REF filespec and data in REALS(512,1)
C
10    CALL INPUT(4,1,IERR)
      IF (IERR .EQ. 0)GO TO 20
      GO TO 10
C
C      Request for DARK filespec, put in array REALS(512,2)
C

```

```

20      CALL INPUT(5,2,IERR)
      IF (IERR .EQ. 0)GO TO 30
      GO TO 20

C
C      Calculate REF - DARK and store in REALS(512,3)
C
30      CALL SUBTR(2)
      DO 40 I=1,512
      REALS(I,3)=REALS(I,1)
40      CONTINUE

C
C      ORD filespec and read in to array REALS(512,1)
C
50      CALL INPUT(8,1,IERR)
      IF(IERR .EQ. 0)GO TO 60
      GO TO 50

C
C      Calculate ORD-DARK and store in array REALS(512,1)
C
60      CALL SUBTR(2)

C
C      Calculate observed rotation and store in REALS
C

      DO 80 I=1,512
      IF(REALS(I,3) .LE. 0.0)GO TO 70
      IF(REALS(I,1) .LT. 0.0)REALS(I,1)=0.0
      REALS(I,1)=(REALS(I,1)/REALS(I,3))-1.0
      IF(REALS(I,1) .GT. 1.0 .OR. REALS(I,1) .LT. -1.0)
      5          GO TO 70
      REALS(I,1)=ATAN(REALS(I,1)/SQRT(1-(REALS(I,1)**2)))
      REALS(I,1)=(REALS(I,1)*360)/(2*3.141593)
      REALS(I,1)=((90-REALS(I,1))/2)-45.0
      GO TO 80
70      REALS(I,1)=0.0
80      CONTINUE

C
C      And go home
C
      RETURN
      END

C
C      STORE.FTN
C
C      Subroutine which outputs array REALS(512,1)
C      ready for MULPLT.
C
C      P. Aiello      5/17/82
C
      SUBROUTINE STORE

C
C      Variable declarations
C
      COMMON /PSTR/ STR
      COMMON /RDATA/ REALS

```

```

REAL REALS(512,3)
BYTE OTFILE(32),STR(82)
INTEGER LEN

C
C      Obtain output filespec
C
      CALL PROMPT(9,STR,LEN)
      DECODE(LEN,900,STR)LENGTH,OTFILE
900    FORMAT(Q,32A1)
      OTFILE(LENGTH+1)=0

C
C      And open the output file
C
      OPEN(UNIT=4,NAME=OTFILE,TYPE='NEW',
           5      CARRIAGECONTROL='LIST',ERR=20)

C
C      And write out the output file
C
      DO 10 I=1,512
      A=FLOAT(I)
      B=REALS(I,1)
      WRITE(4,930)A,B
930    FORMAT('RD',2G15.7)
10     CONTINUE

C
C      And close the file
C
      CLOSE(UNIT=4,ERR=30)

C
C      And go home
C
      RETURN

C
C      Send error messages
C
20     WRITE(5,950)
950    FORMAT(' ERROR ON OPEN ')
      RETURN

30     WRITE(5,960)
960    FORMAT(' ERROR ON CLOSE ')
      RETURN
      END
      ENTER.FTN

C
C      Subroutine used to enter any data file into
C      the array REALS(512,1)
C
C      P. Aiello      5/14/82
      SUBROUTINE ENTER

C
C      Variable declarations
C
      COMMON /RDATA/ REALS

```



```

REAL REALS(512,3)
INTEGER IERR

C
C   Request for the input file specification
C   and return data in array REALS(512,1)
C
10  CALL INPUT(2,1,IERR)
    IF(IERR .EQ. 0)GO TO 20
    GO TO 10

C
C   and go home
C
20  RETURN
    END
FIT.FTN

C
C   Subroutine which curve fits to a cos**2 wave a
C   data file which is asked for. This routine calls
C   SFIT.
C
C   P. Aiello    9/21/82
C
SUBROUTINE FIT

C
C   Declare variables
C
COMMON /RDATA/ REALS
REAL REALS(512,3)
BYTE STR(82)
INTEGER IERR,LEN

C
C   Read in input file and put into array REALS(512,2)
C
CALL INPUT(13,2,IERR)

C
C   Read in initial estimates for curve fit
C
CALL PROMPT(14,STR,LEN)
DECODE(LEN,900,STR)AMP,PHS,OFF,XDIM,NPTS
900  FORMAT(4G15.7,1I4)

C
C   Generate XDIM array and store in REALS(512,1)
C
REALS(1,1)=0.
DO 10 I=2,NPTS
REALS(I,1)=REALS(I-1,1)+XDIM
10  CONTINUE

C
C   Time to do the actual fit by calling SFIT
C
CALL SFIT(AMP,PHS,OFF,NPTS)

C
C   and write the result to the terminal

```

```

C
WRITE(5,910)AMP,PHS,OFF
910  FORMAT('AMP,PHS,OFF= ',3G15.7)
C
C    and go home
C
RETURN
END
SFIT.FTN
C
C *****LEAST SQUARES*****
C    Program for fitting parameters A,B,C,... to observe
C    values of variables X,Y,Z,..., given a function
C          FO(X,Y,Z,...,A,B,C,...)=0
C    Partial derivatives of function FO with respect to
C    A,B,C,... must be defined as FA,FB,FC,...
C    Initial estimates of parameters A and B must be sup
C    These are corrected by factors AA and BB. If initi
C    are good, first correction will give good results.
C    iterations may be necessary.
C    If AA and BB are of same order of magnitude
C    as A and B, convergence may not occur.
C    Method used is described by W. E. Deming,
C    "Statistical Adjustment
C    of Data", Wiley, N.Y., 1943, Chapter 9.
C
SUBROUTINE SFIT(AMP,PHS,OFF,NPTS)
C
C          AMP=      initial estimate of amplitude
C          PHS=      initial estimate of phase angle
C          OFF=      initial estimate of offset
C          NPTS=     number of points in data set to fit
C
C    Delcare variables
C
COMMON /RDATA/ REALS
REAL REALS(512,3),FF(3,3),AMP,PHS,OFF
INTEGER NPTS
DRCONV=3.14159/180.
NITER=0
C
C    Now it is time to enter the fitting routine
C
10  DO 30 I=1,3
DO 20 J=1,3
20  FF(I,J)=0.0
30  CONTINUE
FX1=0.0
FX2=0.0
FX3=0.0
DO 40 I=1,NPTS
C
C    This section is modified to suit function to be fit

```

```

C      FO=(AMP*(COS((PHS+REALS(I,1))*DRCONV)**2)+OFF)
5      -REALS(I,2)
      FA=(COS((PHS+REALS(I,1))*DRCONV)**2)
      FB=-2*AMP*DRCONV*COS((PHS+REALS(I,1))*DRCONV)
5      *SIN((PHS+REALS(I,1))*DRCONV)
      FC=1.0

C
C      End of function section
C

      FF(1,1)=FF(1,1)+FA*FA
      FF(1,2)=FF(1,2)+FA*FB
      FF(2,2)=FF(2,2)+FB*FB
      FF(1,3)=FF(1,3)+FA*FC
      FF(2,3)=FF(2,3)+FB*FC
      FF(3,3)=FF(3,3)+FC*FC
      FX1=FX1+F0*FA
      FX2=FX2+F0*FB
      FX3=FX3+F0*FC
40     CONTINUE
      DENOM=(FF(1,1)*FF(2,2)*FF(3,3))+(FF(1,2)*FF(2,3)
5      *FF(1,3))
5      +(FF(1,3)*FF(1,2)*FF(2,3))-(FF(1,3)*FF(2,2)
5      *FF(1,3))
5      -(FF(1,2)*FF(1,2)*FF(3,3))-(FF(1,1)*FF(2,3)
5      *FF(2,3))
      AA=((FF(1,2)*FF(2,3)*FX3)+(FF(1,3)*FX2*FF(2,3))+
5      (FX1*FF(2,2)*FF(3,3))-(FX1*FF(2,3)*FF(2,3))
5      -(FF(1,3)*FF(2,2)*FX3)-(FF(1,2)*FX2*FF(3,3)
5      ))/DENOM
      BB=-((FF(1,1)*FF(2,3)*FX3)+(FF(1,3)*FX2*FF(1,3))+
5      (FX1*FF(1,2)*FF(3,3))-(FX1*FF(2,3)*FF(1,3))
5      -(FF(1,3)*FF(1,2)*FX3)-(FF(1,1)*FX2*FF(3,3)
5      ))/DENOM
      CC=((FF(1,1)*FF(2,2)*FX3)+(FF(1,2)*FX2*FF(1,3))+
5      (FX1*FF(1,2)*FF(2,3))-(FX1*FF(2,2)*FF(1,3))
5      -(FF(1,2)*FF(1,2)*FX3)-(FF(1,1)*FX2*FF(2,3)
5      ))/DENOM
      ATEST=AMP
      BTEST=PHS
      CTEST=OFF
      AMP=AMP-AA
      PHS=PHS-BB
      OFF=OFF-CC
      NITER=NITER+1
      IF((ABS(ATEST-AMP))/ATEST .LT. 0.0001
5      .AND. (ABS(BTEST-PHS))/BTEST .LE. 0.0001
5      .AND. (ABS(CTEST-OFF))/CTEST .LE. 0.0001
5      .OR. NITER .GT. 10)GO TO 50
50     GO TO 10
      RETURN
      END
C      DFIT.FTN

```

```

C
SUBROUTINE DFIT
C
C   Delcare variables
C
COMMON /RDATA/ REALS
REAL REALS(512,3), FF(3,3), REFAMP(512), REFPHS(512)
5      , SAMAMP(512), SAMPHS(512)
INTEGER INTS(100,16), IERR, LEN, NINTS(16)
BYTE FILE(32), STR(82)
IERR=0
DRCONV=3.14159/180.0
C
C   Read in the number of data sets to be fit
C
CALL PROMPT(18,STR,LEN)
DECODE(LEN,900,STR)NDSETS
900  FORMAT(1I4)
C
C   Request for and read in first approximation of ampl
C   phase, offset, and XDIM.
C   (the constant spacing between points in the X dimen
C   and NPTS, which is the number of points in each dat
C   or in other words, the number of scans at each wave
C   If NPDT = 0 then intermediate result printing is in
C   if it = 1 then they are printed.
C
CALL PROMPT(19,STR,LEN)
DECODE(LEN,910,STR)AIN,BIN,CIN,XDIM,NPDT
910  FORMAT(4G15.7,1I4)
C
C   Initialize loop for the reference and sample data s
C
DO 210 NSET=1,NDSETS
C
C   Now fill in array REALS(512,1) with the
C   X dimension parameters
C   as this is clobbered later when the data is stored
C
REALS(1,1)=0.0
DO 10 I=2,512
REALS(I,1)=REALS(I-1,1)+XDIM
10  CONTINUE
C
C   Ask for data set filespec and open file
C
20  CALL PROMPT(13,STR,LEN)
DECODE(LEN,920,STR)LENGTH,FILE
920  FORMAT(G,32A1)
FILE(LENGTH+1)=0
CALL FOPEN(4,FILE,ISIZE,IERR)
IF ((IERR .AND. 128) .LT. 0) GO TO 30
GO TO 40

```

```

30      WRITE(5,930)
930     FORMAT(' OPEN ERROR ')
      GO TO 20
C
C      Calculate the number of scans in the data set from
C      the size of the file returned by FOPEN
C
40     NSCANS=(ISIZE-1)/2
C
C      Set up loop for 32 times. This is the number of 32
C      byte records there are for each scan
C
      NREC=1
      DO 120 NRS=1,32
C
C      Set up loop to read in array INTS(NSCANS,16 words)
C
      DO 80 NCNT=1,NSCANS
      NCNTS=NCNT
      CALL FREAD(NCNTS,NREC,NINTS,IERR)
      IF ((IERR .AND. 128) .LT. 0) GO TO 50
      GO TO 60
50     WRITE(5,940)
940    FORMAT(' READ ERROR, WILL TRY TO CLOSE THEN EXIT ')
      GO TO 130
60     DO 70 NCOP=1,16
      INTS(NCNT,NCOP)=NINTS(NCOP)
70     CONTINUE
80     CONTINUE
      NREC=NREC+1
C
C      Time to transfer INTS(NSCANS,16) to REALS(512,2)
C      and fit, store the data in REALS(512,3) or POINTS
C      depending on the value of NSET.
C
      DO 110 NFIT=1,16
      DO 90 I=1,NSCANS
      REALS(I,2)=FLOAT(IBOT12(INTS(I,NFIT)))/
5         2.**FLOAT(ITOP4(INTS(I,NFIT)))
90     CONTINUE
      AMP=AIN
      PHS=BIN
      OFF=CIN
C
C      THIS IS WHERE FIT IS TO BE CALLED
C
      CALL SFIT(AMP,PHS,OFF,NSCANS)
C
C      And type out result if this option is selected
C
      IPT=(NRS-1)*16+NFIT
      IF (NPDT .EQ. 1)WRITE(5,950)IPT,AMP,PHS,OFF
950    FORMAT(' IPT,AMP,PHS,OFF= ',114,3G15.7)

```

```

C
C      Check which file is being fitted and store
C
      IF(NSET .GT. 1)GO TO 100
      REFAMP(IPT)=AMP
      REFPHS(IPT)=PHS
      GO TO 110
100    SAMAMP(IPT)=AMP
      SAMPHS(IPT)=PHS
110    CONTINUE
120    CONTINUE
C
C      And close the file
C
130    CALL FCLOSE(IERR)
      IF ((IERR .AND. 128) .LT. 0) GO TO 140
      GO TO 150
140    WRITE(5,960)
960    FORMAT(' CLOSE ERROR ')
      RETURN
C
C      If we have just finished fitting the reference file
C      loop back
C
150    IF (NSET .EQ. 1) GO TO 210
C
C      Now calculate the absorbance and store the data
C
      DO 160 I=1,512
      REALS(I,1)=SAMAMP(I)/REFAMP(I)
160    CONTINUE
      CALL LOG(2)
      DO 170 I=1,512
      REALS(I,1)=-REALS(I,1)
170    CONTINUE
      WRITE(5,970)
970    FORMAT('$ABSORBANCE ')
      CALL GSM
C
C      Now calculate the ORD (phase difference) and store
C
      DO 180 I=1,512
      REALS(I,1)=REFPHS(I)-SAMPHS(I)
180    CONTINUE
      WRITE(5,980)
980    FORMAT('$ORD ')
      CALL GSM
210    CONTINUE
      RETURN
      END
C
C      CRUNCH.FTN
C
C      CRUNCH is the main program which is used to

```

```

C      call all the number crunching subroutines via
C      the first 2 letters of the subroutine name.
C
C      PROGRAM CRUNCH
C
C      Variable declarations
C
C      COMMON /PSTR/ STR
C      COMMON /RDATA/ REALS
C      REAL REALS(512,3)
C      BYTE STR(82),TEST(80)
C      INTEGER IERR,LEN
C      CALL ERRSET(72,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
C      CALL ERRSET(73,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
C      CALL ERRSET(74,.TRUE.,.FALSE.,.FALSE.,.FALSE.,15)
C
C      Print out prompt
C
C      CALL PROMPT(1,STR,LEN)
C
C      Read in input command
C
C      DECODE(LEN,900,STR)TEST
900    FORMAT(BOA1)
C
C      Call CMD, string parser to get the command and exec
C
C      CALL CMD(TEST,IERR)
C
C      If no legal command was entered send error message.
C
C      IF(IERR .NE. 0)WRITE (5,910)
910    FORMAT(' Illegal command, type HELP if you ned it')
C      GO TO 10
110    CALL EXIT
C
C      And exit the program
C
C      END
C      EA.FTN
C
C      Subroutine used to enter an ASCII data file into
C      the array REALS(512,1). This is the same as
C      ENTER.FTN but reads in ASCII data files as opposed
C      to FORTHPIP output RSX binary files.
C
C      P. Aiello      9/20/82
C
C      SUBROUTINE EA
C
C      Variable declarations
C
C      COMMON /RDATA/ REALS

```

```

REAL REALS(512,3),X
BYTE TAG(2),FILE(32),STR(82)
INTEGER IERR,LEN
IERR=0

C
C   Request for and read in the data file
C
CALL PROMPT(17,STR,LEN)
DECODE(LEN,900,STR)LENGTH,FILE
900  FORMAT(Q,32A1)
FILE(LENGTH+1)=0
OPEN(UNIT=3,NAME=FILE,TYPE='OLD',
5    CARRIAGECONTROL='LIST',READONLY)
DO 10 I=1,512
910  READ(3,910)TAG,X,REALS(I,1)
10   FORMAT(2A1,2G15.7)
C   CONTINUE
C   and close the file
C
CLOSE(UNIT=3)
RETURN
END
.PSECT  UNPACK
.TITLE  UNPACK
.IDENT  /V1.0/
.ENABL  LC

;
;   UNPACK.MAC
;
;   Contains 2 routines, IBOT12 and ITOP4, which return
;   the bottom 12 bits and top 4 bits respectively
;   as a function.
;   If the top 4 bits=0, a 0 is returned, if > 0 then
;   the number -1 is returned.
;
;   P. Aiello 6/11/82
;
;   D. ARG=2
;
IBOT12::
MOV     @D.ARG(R5),R0
BIC     170000,R0
RETURN

ITOP4::
MOV     @D.ARG(R5),R0
BIC     7777,R0
SWAB    R0
ASR     R0
ASR     R0
ASR     R0
ASR     R0
BEQ     10$

```



```

DEC      RO
10$:     RETURN
        .END
        .PSECT  FIO
        .TITLE  FIO
        .IDENT  /V1.0/
;
;       FIO.MAC
;
;       This routine does file I/O, open, read and close.
;       Three separate entry points. Written for block
;       reads from a FORTH emulator disk file.
;
;       P. Hoffman, P. Aiello      7/12/82
;
;       .MCALL  CSI$,CSI$1,CSI$2
;       .MCALL  FDBDF$,FDAT$A,FDRC$A,FDBK$A,FDOP$A
;       .MCALL  OPEN$R,READ$,WAIT$,CLOSE$
;
;       Offsets for the calling parameters
;
;       Open calls
;
;       O.LUN = 2
;       O.NAM = 4
;       O.SIZ = 6
;       O.ER1 = 8.
;
;       Read calls
;
;       O.SCN = 2
;       O.NDX = 4
;       O.DAT = 6
;       O.ER2 = 8.
;
;       Close calls
;
;       O.ER3 = 2
;
;       Define event flag for block I/O read
;
;       EFN = 1
;
;       define CSI and FDB for 1 block I/O
;
;       CSI$
CSIBLK:  .BLKB    C.SIZE
FDB:     FDBDF$
        FDAT$A   R.FIX,,512.
        FDRC$A   FD.RWM
        FDBK$A   BUF,512.,,EFN,STAT
        FDOP$A   ,CSIBLK+C.DSDS,,FO.RD,FA.DLK
;

```

```

;       Define buffer size to 512 bytes
;       STAT recieves I/O status word
;       BLKPTR contains number of virtual block to be read
;
BUF:     .BLKB    512.
STAT:    .BLKW    2
BLKPTR:  .WORD    0,0
;
;       File OPEN entry point
;       For call sequence CALL FOPEN(lun,name,size,error)
;           lun=      logical unit number
;           name=     filespec of file to be opened
;           size=     returned, number of RSX blocks + 1
;           error=    returned, low byte is negative err
;                   If high byte is 0 then FCS error
;                   If high byte is neg then DSW error
;
FOPEN::  MOV        CSIBLK,R0                ;point to CSI block
        CSI$1      R0,0.NAM(R5), 32.        ;filename syntactic
        BCC        10$                     ;if no error,branch
        JMP        CSIERR                  ;set error flag
10$:     CSI$2      R0,OUTPUT                ;filename semantic
        BCC        20$                     ;if no error,branch
        JMP        CSIERR                  ;set error flag
20$:     MOV        FDB,R0                  ;point to FDB
        OPEN$R     R0,@0.LUN(R5)            ;open the file
        MOV        F.ERR(R0),@0.ER1(R5)     ;move error code
        MOV        <F.EFBK+2>(R0),@0.SIZ(R5) ;ret block
        RETURN                                ;all done, go home
CSIERR:  MOV        IE.BAD,@0.ER1(R5)       ;set error flag
        RETURN
;
;       File READ entry point
;       Fortran calling sequence
;           CALL FREAD(scan , 32 byte block index,data
;           array(32 bytes),error)
;           scan =   LDA scan number which is to be read
;           32 byte block index=   index telling which
;                               32 bytes
;                               of record (512 bytes) are returned
;           data=    integer array 32 bytes long
;           error=   error code
;
FREAD::  MOV        FDB,R0                ;point to FDB
        MOV        BLKPTR,R1             ;point to BLKPTR
        CLR        (R1)+                 ;clear high word
        MOV        @0.SCN(R5),(R1)        ;get the scan num
        ASL        (R1)                  ;calculate block
        DEC        (R1)                  ; (2*scannum)-1
        MOV        @0.NDX(R5),R2          ;fetch 32 byte
        MOV        16.,R4                ;we need this later
;
;       The scan number is passed.  For each scan there are
;       If the read is from the first block (record number<

```

```

;      If the read is from the second block, increment the
CMP      R4,R2                      ;check if <= 16 by
BGE      10$                        ;branch if >=0
SUB      R4,R2                      ;subtract 16
INC      (R1)                      ;and increment
10$:     READ$   R0,,, BLKPTR        ;read
BCS      RDERR                      ;if error
WAIT$    R0                        ;wait for end
TSTB     F.ERR(R0)                  ;? error read
BMI      RDERR                      ;set error flag
DEC      R2                        ;calculate offset
.REPT    5                          ;to desired record
ASL      R2                        ;(INDEX-1)*32
.ENDM
ADD      BUF,R2                    ;add base address
MOV      @D.DAT(R5),R3              ;points to data
20$:     MOV      (R2)+,(R3)+        ;store record
SOB      R4,20$                    ;loop until done
RDERR:   MOV      F.ERR(R0),@D.ER2(R5) ;set error flag
RETURN                                ;and go home

```

```

;
;      CLOSE entry point
;      Fortran calling sequence CALL FCLOSE(error)
;      error= error code
;

```

```

FCLOSE::
MOV      FDB,R0                    ;point to FDB
CLOSE$   R0                        ;close the file
MOV      F.ERR(R0),@D.ER3(R5)      ;set error if need
RETURN                                ;bye

```

```

.END
.PSECT  PROMPT
.TITLE  PROMPT
.IDENT  /V2.0/
.ENABL  LC

```

```

=====
;      PROMPT.MAC
;

```

```

;      Phil Hoffman and P. Aiello
;      Department of Chemistry
;      Michigan State University
;      East Lansing, MI 48824
;      3-August-1982

```

```

=====
;      PROMPT is used by all CRUNCH subroutines to
;      fetch command input.
;      This input may be obtained from (1)The MCR
;      invocation string, (2)an
;      indirect command file, and/or (3)in response
;      to an explicit terminal
;      prompt. The calling sequence is as follows:
;

```

```

;      CALL PROMPT(IPMPT,STR,LEN)

```

```

;
; where the arguments in the call are defined as follows:
;
; IPMPT =>      an INTEGER*2 variable whose value is the
;               index of the IPMPT-th string
;               to be used as a terminal prompt.
;
; STR    =>      The address of an byte array, 82. bytes
;               in length to receive the command
;               string in response to the selected
;               prompt.
;
; LEN    =>      an INTEGER*2 variable to receive
;               the length of STR.
;
;
;                               NOTE
;
; PROMPT DOES NOT CHECK THE VALUE OF IPMPT TO BE SURE THAT
; WITHIN THE RANGE OF LEGALLY DEFINED PROMPT INDICES.
; =====
; Local Definitions, Offsets, Storage, Macros, etc.
; -----
;
;      .MCALL  GCMLB$,GCML$,EXIT$S
;      .MCALL  TEXT                                ; from (MACROS.MLB)
;
; Define the offsets to the arguments in the call
;
;      O.PMPT = 2
;      O.STR  = 4
;      O.LEN  = 6
; -----
; Specify the prompts, the prompt address table, and the
; prompt length table.
; -----
P1:      TEXT YCRUNCH> Y,P1L,1,0,0
P2:      TEXT <FILE TO ENTER : >,P2L,1,0,0
P3:      TEXT <FILE TO TAKE LOG OF : >,P3L,1,0,0
P4:      TEXT <REFERENCE FILE : >,P4L,1,0,0
P5:      TEXT <DARK CURRENT FILE : >,P5L,1,0,0
P6:      TEXT <ABSORBANCE FILE : >,P6L,1,0,0
P7:      TEXT <FILE TO BE SMOOTHED : >,P7L,1,0,0
P8:      TEXT <FILE FOR ORD CALC : >,P8L,1,0,0
P9:      TEXT <OUTPUT FILE : >,P9L,1,0,0
P10:     TEXT <FIRST FILE : >,P10L,1,0,0
P11:     TEXT <LESS SECOND FILE : >,P11L,1,0,0
P12:     TEXT <Input IWIN, IFUNC, ITER : >,P12L,1,0,0
P13:     TEXT <FILE TO BE FIT: >,P13L,1,0,0
P14:     TEXT <Input AMP, PHASE, OFF, XDIM, NPTS: >,P14L,1,
P15:     TEXT <Store the data? [Y/N]: >,P15L,1,0,0
P16:     TEXT <Smooth and store the data? [Y,N]: >,P16L,1,0,
P17:     TEXT <ASCII file to enter : >,P17L,1,0,0
P18:     TEXT <Number of data sets to smooth: >,P18L,1,0,0
P19:     TEXT <Input AMP, PHASE, OFF, XDIM, NDPT: >,P19L,1,0

```

```

;
PADD:  . WORD   P1, P2, P3, P4, P5, P6, P7, P8, P9, P10, P11, P12
        . WORD   P13, P14, P15, P16, P17, P18, P19
PLEN:  . WORD   P1L, P2L, P3L, P4L, P5L, P6L, P7L, P8L, P9L, P10L
        . WORD   P11L, P12L, P13L, P14L, P15L, P16L, P17L, P18L, P19L
; -----
; Specify the command block for the Get Command Line Macro.
; -----
GETLIN: GCMLB$ 1,,2
; -----
; Entry Point
; -----
PROMPT: :
        MOV     @O. PMPT(R5), R1          ; Get the prompt
        DEC     R1                        ; Convert to z-base
        ASL     R1                        ; word-offset addr
        MOV     GETLIN, R0                ; Point
        GCML$   RO, PADD(R1), PLEN(R1)    ; Get the command
        TSTB    G. ERR(R0)                ; Were there errors
        BGE     10$                       ; GE => No errors
        EXIT$S                                ; Errors => quit
;
10$:   MOV     G. CMLD(R0), R2              ; Get the of char
        MOV     R2, @O. LEN(R5)            ; and send back
        MOV     <G. CMLD+2>(R0), R0        ; Get starting addr
        MOV     O. STR(R5), R1             ; Get starting addr
                                           ; that string
        CALL    $CVTUC                    ; Convert to upper
        RETURN                                ; Go home
        . END
        . PSECT  CMD
        . TITLE  CMD
        . IDENT  /V1. 0/
        . ENABL  LC
;
; CMD. MAC
;
;
; This routine uses the parser to determine input
; strings and execute the proper subroutines
;
; P. Hoffman, P. Aiello 6/14/82
;
;
; Declare global references, entry to state
; tables, symbol table and keyword table
;
; . GLOBL CRNCMD, CRNSTB, CRNKTB
;
; Call some of Hoffman's macros
;
; . MCALL PUSH, POP
;
; Macros necessary for parser
;

```

```

        . MCALL ISTAT$, STATE$, TRAN$
;
; Create macro CLSUB, saves registers, points ARGLST
; necessary, calls the appropriate subroutine, and
; restores the registers
;
        . MACRO CLSUB, NAME, ARGS
        PUSH    <R0, R1, R2, R3, R4, R5>
        . IIF NB, ARGS,    MOV    ARGLST, R5
        CALL    NAME
        POP     <R5, R4, R3, R2, R1, R0>
        . ENDM CLSUB
;
; Allocate space for the argument list
;
ARGLST:  . BLKW 1
        . WORD  ARG1
ARG1:    . BLKW 1
;
; Offset to command string and error string in call
;
        O. STR=2
        O. ERR=4
;
; Entry point
;
CMD: :   MOV     O, @O. ERR(R5)           ; clear the error
        MOV     O. STR(R5), R0           ; pointer to cmd
        MOV     R0, R1                   ; put result
        MOV     80., R2                  ; 80 char string
        CALL    $CVTUC                   ; and convert to uc
;
; set up for parser
;
        MOV     1000, R1                  ; 2 char. abbrev.
        MOV     CRNKTB, R2                ; point to keyword
        MOV     80., R3                  ; 80 char string
        MOV     O. STR(R5), R4            ; string pointer
        PUSH    R5                       ; save R5
        MOV     CRNCMD, R5                ; state table entry
        CALL    . TPARS                   ; and parse it
        POP     R5                       ; restore args
        BCC     1$                       ; check if error
        MOV     -1, @O. ERR(R5)           ; set error flag
1$:      RETURN                           ; and go home
;
; Create state table
;
        ISTAT$  CRNSTB, CRNKTB           ; initialize
        STATE$  CRNCMD                   ; specify entry poin
        TRAN$    "LOGARITHM", $EXIT, LOGACT
        TRAN$    "SUBTRACT", $EXIT, SUBACT
        TRAN$    "ABSORBANCE", $EXIT, ABSACT

```

```

TRAN$      "ORD", $EXIT, ORDACT
TRAN$      "AVERAGE", $EXIT, AVGACT
TRAN$      "HELP", $EXIT, HELACT
TRAN$      "STORE", $EXIT, STOACT
TRAN$      "NOISE", $EXIT, NREACT
TRAN$      "NR", $EXIT, NREACT
TRAN$      "ENTER", $EXIT, ENTACT
TRAN$      "FIT", $EXIT, FITACT
TRAN$      "DFIT", $EXIT, DFIACT
TRAN$      "EA", $EXIT, EAACT
TRAN$      "EXIT", $EXIT, EXIACT
STATE$                                           ; closes state table
;
; Parser action routines
;
      . PSECT  $STATE
LOGACT:  MOV      1, ARGLST                      ; ARGS=1
          MOV      1, ARG1                      ; and it is =1
          CLSUB    LOG, 1                      ; call LOG
          RETURN                                     ; and go home
SUBACT:  MOV      1, ARGLST
          MOV      1, ARG1
          CLSUB    SUBTR, 1
          RETURN
ABSACT:  CLSUB    ABSORB
          RETURN
ORDACT:  CLSUB    ORD
          RETURN
AVGACT:  CLSUB    AVG
          RETURN
HELACT:  CLSUB    HELP
          RETURN
STOACT:  CLSUB    STORE
          RETURN
NREACT:  MOV      1, ARGLST
          MOV      1, ARG1
          CLSUB    NRED, 1
          RETURN
ENTACT:  CLSUB    ENTER
          RETURN
FITACT:  CLSUB    FIT
          RETURN
DFIACT:  CLSUB    DFIT
          RETURN
EAACT:   CLSUB    EA
          RETURN
EXIACT:  CLSUB    EXIT
          RETURN
      . END

```

REFERENCES

—

1

REFERENCES

1. Willard, Merritt and Dean, 'Instrumental Methods of Analysis', Fifth Ed., D. VanNostrand Co., New York, 413 (1974)
2. R. A. Harbor, G. E. Sonner, Applied Optics, 5, 1039 (1966)
3. A. Danielsson, P. Lindblom, E. Soderman, Chemica Scripta, 6, 5 (1974)
4. P. B. Boyce, Science, 198, 145 (1977)
5. M. J. Milano, H. L. Pardue, T. E. Cook, R. E. Santini, D. W. Margerum, J. Rachera, Anal Chem*, 46*, 374 (1974)
6. J. A. deHaseth, W. S. Woodward, T. L. Isenhour, Anal Chem, 48, 1513 (1976)
7. T. E. Cook, R. E. Santini, H. L. Pardue, Anal Chem, 49, 871 (1977)
8. T. A. Nieman, C. G. Enke, Anal Chem, 48, 619 (1976)
9. Y. Talmi, Anal Chem, 47, 658A (1975)
10. Y. Talmi, Anal Chem, 47, 697A (1975)
11. Y. Talmi, Am Lab, March, 79 (1978)
12. N. G. Howell, G. M. Morrison, Anal Chem, 49, 106 (1977)
13. R. P. Cooney, T. Vo-Dinh, G. Walden, J. D. Winefordner, Anal Chem, 49, 939 (1977)
14. R. P. Cooney, G. D. Boutilier, J. D. Winefordner, Anal Chem, 49, 1048 (1977)
15. K. L. Ratzlaff, S. L. Paul, Appl Spectroscopy, 33, 240 (1979)
16. K. L. Ratzlaff, Anal Chem, 52, 916 (1980)
17. H. A. Lewis, M. B. Denton, J. of Autom. Chem., Preprint August (1980)
18. G. Horlick, E. G. Coddling, Anal Chem, 45, 1490 (1973)
19. D. A. Yates, T. Kuwana, Anal Chem, 48, 510 (1976)

20. G. Horlick, Appl Spectroscopy, 30, 113 (1976)
21. R. W. Simpson, Rev. Sci. Instrum, 50, 730 (1979)
22. P. J. Aiello, M. S. Thesis, Michigan State Univ. (1980)
23. Y. Talmi, R. W. Simpson, Appl Optics, 19, 1401 (1980)
24. D. F. Marino, J. D. Ingle Jr., Anal Chem, 53, 645 (1981)
25. Y. Talmi, Appl Spectroscopy, 36, 1 (1982)
26. F. S. Chuang, D. F. S. Natusch, K. R. O'Keefe, Anal Chem, 50, 525 (1978)
27. M. J. Milano, K. Kim, Anal Chem, 49, 555 (1977)
28. M. Franklin, C. Baber, S. R. Koirttyohann, Spectrochimica Acta, 31, 589 (1976)
29. G. Horlick, E. G. Coddington, Appl Spectroscopy, 29, 167 (1975)
30. T. E. Edmunds, G. Horlick, Appl Spectroscopy, 31, 536 (1977)
31. Y. Talmi, H. P. Sieper, L-Moenke-Bankenbourg, Analytica Chimica Acta, 127, 71 (1981)
32. M. W. Blades, G. Horlick, Appl Spectroscopy, 34, 696 (1980)
33. B. R. Sandel, A. L. Broadfoot, Appl Optics, 15, 3111 (1976)
34. M. A Ryan, R. J. Miller, J. D. Ingle Jr., Anal Chem, 50, 1772 (1978)
35. E. G. and G. Reticon, S-Series Solid State Linear Diode Array, Brochure (1978)
36. E. H. Eberhardt, Appl Optics, 18, 1418 (1979)
37. A. L. Broadfoot, B. R. Sandel, Appl Optics, 16, 1533 (1977)
38. H. L. Felkel, H. L. Pardue, Anal Chem, 50, 602 (1978)
39. A. E. McDowell, H. L. Pardue, Anal Chem, 49, 1171 (1977)

40. R. E. Dessey, W. D. Reynolds, W. G. Nunn, C. A. Titus, G. F. Moler, Clin Chem, 22, 1472 (1976)
41. R. P. Cooney, T. Vo-Dinh, J. D. Winefordner, Anal Chim Acta, 89, 9 (1977)
42. B. Newcome, submitted for publication
43. Product Specifications on Model 1070 Intelligent Disk Controller, Persci, Marina del Rey, CA (1976)
44. E. Rather, L. Brodie, 'Using FORTH', Forth-79 Std. Ed., Forth Inc, Third Ed., C. Rosenberg, Editor (1981)
45. T. V. Atkinson, Department of Chemistry, Michigan State Univ. (1982)
46. E. M. Carlson, Ph. D. Thesis, Michigan State University (1978)
47. S. S. Vogt, R. G. Tull, P. Kelton, Appl Optics, 17, 574 (1978)
48. T. B. McCord, M. J. Frankston, Appl Optics, 14, 1437 (1978)
49. F. J. Holler, T. V. Atkinson, T. G. Kelly, C. G. Enke, Am Lab, September, 9 (1976)
50. C. H. Moore, Byte, August, 76 (1980)
51. J. S. James, Byte, August, 100 (1980)
52. A. Moscovitz, Advan in Chem Phys Part I, 4, 67 (1962)
53. C. Djerassi, 'Optical Rotatory Dispersion', McGraw-Hill, New York (1960)
54. T. B. Crumpler, W. H. Dyre, A. Spell, Anal Chem, 27, 1645 (1955)
55. B. Carroll, H. B. Tillem, E. S. Freeman, Anal Chem, 30, 1099 (1958)
56. A. Savitsky, M. Golay, Anal Chem, 36, 1627 (1964)
57. A. Proctor, Anal Chem, 52, 2315 (1980)
58. J. G. Foss, J. Chem Ed, 40, 592 (1963)
59. K. Wong, J. Chem Ed, 51, A573 (1974)
60. K. Wong, J. Chem Ed, 52, A9 (1975)

61. K. Wong, J. Chem Ed, 52, A83 (1975)
62. V. C. Zadnik, J. L. Scott, R. Megargle, J. Kerkay, K. H. Pearson, J. Autom Chem, 1, 206 (1979)
63. D. V. Amato, G. W. Ewing, Analytical Letters, 7, 763 (1974)
64. J. R. Krivacic, D. E. Wisnosky, D. W. Urry, Anal Biochem, 43, 547 (1971)
65. W. M. Martz, S. Akipis, Anal Biochem, 39, 327 (1971)
66. P. H. Schippers, H. Dekkers, Anal Chem, 53, 778 (1981)
67. J. Y. Cassim, J. T. Yang, Biochemistry, 8, 1947 (1969)
68. D. F. DeTar, Anal Chem, 41, 1406 (1969)
69. G. C. Chem, J. T. Yang, Analytical Letters, 10, 1195 (1977)
70. V. C. Zadnik, K. H. Pearson, Analytical Letters, 12, 1267 (1979)
71. K. H. Pearson, V. C. Zadnik, J. L. Scott, Analytical Letters, 12 1049 (1979)
72. W. C. Krueger, L. M. Pschigoda, Anal Chem, 43, 673 (1971)
73. P. A. Hoffman, C. G. Enke, Computers and Chemistry, 7, 47 (1983)
74. W. E. Deming, 'Statistical Adjustment of Data', Wiley, New York (1943)
75. T. M. Lowry, 'Optical Rotatory Power', Dover Publications, New York (1964)
76. G. R. Minot, W. P. Murphy, J. A. M. A., 87, 470 (1926)
77. E. L. Rickes, N. S. Brink, F. R. Koniuszy, T. R. Wood, K. Folkers, Science, 107, 396 (1948)
78. E. L. Rickes, N. S. Brink, F. R. Koniuszy, T. R. Wood, K. Folkers, Science, 108, 134 (1948)
79. Merck Service Bulletin, 'Vitamin B 12' (1958)

80. E. L. Smith, 'Vitamin B 12', Third Ed., Wiley, New York (1965)

-