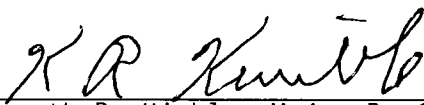
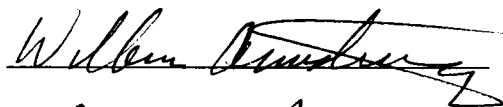
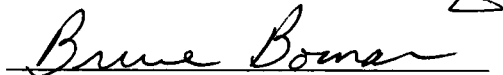


To the Graduate Council:

I am submitting herewith a thesis written by Fred G. Sherrell entitled "Data Acquisition and Processing for a Fast Scanning Spectroradiometer." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.


Kenneth R. Kimble, Major Professor

We have read this thesis
and recommend its acceptance:

Accepted for the Council:


Vice Provost
and Dean of The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the materials is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature Fred G. Sherrill

Date July 11, 1988

DATA ACQUISITION AND PROCESSING
FOR A FAST SCANNING
SPECTRORADIOMETER

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Approved for public release; distribution is unlimited.

Fred G. Sherrell

August 1988

**STATEMENT OF TERMS AND CONDITIONS RELEASE OF AIR FORCE-OWNED OR DEVELOPED
COMPUTER SOFTWARE PACKAGES**Date 6-27-88

1. In accordance with the provisions of AFR 300-6, release of the following US Air Force software package (computer programs, systems descriptions, and documentation) is requested:

"Data Acquisition and Processing for a Fast Scanning Spectroradiometer" (MS Thesis)

2. The requested software package will be used for the following purpose:

Satisfy academic requirements for a MS degree in Computer Science.

Such use is projected to accrue benefit to the government as follows:

Personnel training; improved productivity; improved test data quality; encourage the utilization of computer technology in AEDC testing programs.

3. I/we will be responsible for assuring that the software package received will not be used for any purpose other than shown in paragraph 2 above; also, it will not be released to anyone without prior approval of the Air Force. Further, the release of the requested software package will not result in competition with other software packages offered by commercial firms.

4. I/we guarantee that the provided software package, or any modified version thereof, will not be published for profit or in any manner offered for sale to the government; it will not be sold or given to any other activity or firm, without the prior written approval of the Air Force. If this software is modified or enhanced using government funds, the government owns the results, whether the software is the basis of, or incidental to a contract. The government may not pay a second time for this software or the enhanced or modified version thereof. The package may be used in contract with the government but no charge may be made for its use.

5. The US Air Force is neither liable or responsible for maintenance, updating or correcting any errors in the software provided.

6. I/we understand that no material subject to national defense security classification or proprietary rights was intended to be released to us. I/we will report promptly the discovery of any material with such restrictions to the Air Force approving authority. I/we will follow all instructions concerning the use or return of such material in accordance with regulations applying to classified material, and will make no further study, use, or copy such material subject to security or proprietary rights marking.

7. I/we understand that the software package received is intended for domestic use only. It will not be made available to foreign governments nor used in any contract with a foreign government.

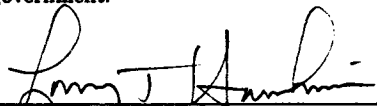


Signature of Requestor
F. G. Sherrell

Name of Requestor
Sverdrup Technology, Inc./AEDC

Organization/Address
ETF/EL3/MS900

City, State and Zip Code
Arnold Air Force Base, TN 37389-9998



Signature of Air Force
Approving Authority LARRY J. HAWKINS, Lt Col, US
Deputy for Comm-Computer Sys

Name/Title of Air Force
Approving Authority AEDC/SC
Arnold AFB, TN 37389-5000

Organization/Location

ACKNOWLEDGMENTS

The author is very appreciative of the advice and assistance of Dr. Kenneth R. Kimble in completing the Computer Science program at The University of Tennessee Space Institute (UTSI). Also, the efforts of Dr. Bruce W. Bomar and Professor Wilbur C. Armstrong in reviewing the thesis manuscript are greatly appreciated.

The author extends thanks to the United States Air Force and the Arnold Engineering Development Center (AEDC) for the opportunity to attend UTSI. The author is also very appreciative of the permission extended by the managers of Sverdrup Technology, Inc., AEDC Group, to attend UTSI.

The excellent support of Schneider Services International, Inc., in preparing the figures and providing printing services, is also appreciated. The author also expresses thanks to his daughter, Elise, for typing the original thesis draft, and to Linda W. Hershman of Lin-Hersh Enterprises for the final manuscript preparation.

Finally, the author extends a special thanks to Mr. Ron K. Wood of Sverdrup Technology, Inc., for his outstanding technical support on the system assembly and checkout.

References to named commercial products in this thesis are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

The research reported herein was performed by the Arnold Engineering Development Center (AEDC), Air Force Systems Command. Work and analysis for this research were done by personnel of Sverdrup

Technology, Inc., AEDC Group, operating contractor for the AEDC propulsion test facilities. Further reproduction is authorized to satisfy needs of the U. S. Government.

ABSTRACT

A customized digital data acquisition system and a comprehensive set of applications programs were developed to provide automatic, high-speed operation of a modified grating spectroradiometer. A 1/2-meter grating spectrograph was rendered a fast scanning device by using a wavelength scanning mirror behind the entrance slit, along with a customized detector assembly on the exit focal plane. The detector assembly consists of eight Indium Antimonide detectors, with each detector having a separate order-sorting filter and exit slit. The concatenated ranges of the eight detectors cover the range from 1.1 to 5.1 microns. Spectral scan times are on the order of 1 second per scan.

Approximately 2100 lines of C programming code were written to acquire raw data, organize and store data, perform wavelength calibrations, perform radiance calibrations, and reduce raw data to absolute radiometric units. The code is executed on a Zenith Z-248 microcomputer. Very high data densities on the wavelength scale were realized. Up to 8,000 data points can be stored per spectral scan, and data run times of 10 minutes or longer are possible. All data are digitized and stored on a Bernoulli disk recorder during the data run. The raw data can be reduced to engineering units within minutes of the completion of the data run. A complete experimental demonstration of the applications software is presented. The performance of the data acquisition and processing system was excellent in all respects. However, several operating deficiencies of the spectroradiometer were discovered, and it was concluded that in a

qualitative sense, the performance of the data acquisition and processing system exceeds the performance of the spectroradiometer. Improvements in the spectroradiometer performance are suggested. The most important possible improvement is the use of an evacuable spectrometer to eliminate calibration errors near the atmosphere absorption bands.

TABLE OF CONTENTS

CHAPTER	PAGE
I. INTRODUCTION	1
II. SPECTRORADIOMETER DESCRIPTION	11
III. DIGITAL DATA ACQUISITION SYSTEM DESIGN	24
1. Data System Design Concept	24
2. Data Interrupt Generation	34
3. Data Flow Control	39
4. Data Input/Output Implementation	50
IV. DATA ACQUISITION SOFTWARE DESCRIPTION	55
1. Design Concept Overview	55
2. Description of RECORDAT.C	58
3. Description of RECOVDAT.C	71
4. Description of COADD.C	73
V. DATA REDUCTION SOFTWARE DESCRIPTION	88
VI. EXPERIMENTAL RESULTS	100
VII. SUMMARY AND CONCLUSIONS	109
LIST OF REFERENCES	111
APPENDICES	114
APPENDIX A. INVERSE RESPONSIVITY DATA	115
APPENDIX B. RADIANCE MEASUREMENT DEMONSTRATION	132
VITA	141

LIST OF FIGURES

FIGURE	PAGE
1.1. Block Diagram of a Typical Spectroradiometer	3
1.2. Spectral Radiance of a 1000°C Blackbody	5
1.3. Illustration of a Possible Output Function of a Spectroradiometer	7
2.1. Spectrometer Concept Design Based on the Czerny-Turner System	12
2.2. Spectroradiometer Optical Diagram	17
2.3. Spectroradiometer Detector Assembly	18
2.4. Detector Locations and Nominal Spectral Range	20
2.5. Autoranging Amplifier/Demodulator Circuit Design	22
3.1. Major Components of the Data Acquisition System	26
3.2. Basic Concept Design of One Detector Data Acquisition Channel	28
3.3. Bit Pattern Typical of All Eight Detector Channels	30
3.4. NIR/IR Spectroradiometer Data System Concept Design	32
3.5. Data Interrupt Generator Concept Design	33
3.6. Data Control Board Design	40
3.7. Data Flow Control and Handshake Functions	43
3.8. Operational Configuration of I/O Board No. 1	51
3.9. Operational Configuration of I/O Board No. 2	52
4.1. Data Acquisition Programs and Data Flow Paths	56
4.2. Data Structure of DATARUN.DAT	59
4.3. Header Data Format in DATARUN.DAT	61

FIGURE	PAGE
4.4. One Data Point in DATARUN.DAT	62
4.5. Flow Diagram of Program RECORDAT.C	63
4.6. Code Segment for the Primary Data Acquisition Function	70
4.7. Data Content of All Files Produced by RECOVDAT.C . . .	72
4.8. Flow Diagram of Program RECOVDAT.C	74
4.9. Pseudo Code for Program COADD.C	77
4.10. Processes of Pass No. 1	79
4.11. Processes of Pass No. 2	80
4.12. Flow Diagram for Program COADD.C	82
5.1. Data Reduction Programs and Data Flow Paths	89
5.2. Primary Functions of Program XMISS.C	91
5.3. Structure of the Calibration Data in RESP.DAT	94
5.4. Primary Functions of Program RESPCALC.C	96
5.5. Primary Functions of REDUCDAT.C	98
6.1. Transmission of Plexiglass in the Range of CH 2	102
6.2. Transmission of Polystyrene in the Range of CH 6 . . .	103
A.1. Inverse Responsivity for Detector Channel 1	116
A.2. Inverse Responsivity for Detector Channel 2	118
A.3. Inverse Responsivity for Detector Channel 3	120
A.4. Inverse Responsivity for Detector Channel 4	122
A.5. Inverse Responsivity for Detector Channel 5	124
A.6. Inverse Responsivity for Detector Channel 6	126
A.7. Inverse Responsivity for Detector Channel 7	128

FIGURE	PAGE
A.8. Inverse Responsivity for Detector Channel 8	130
B.1. Source Radiance for Channel 1	133
B.2. Source Radiance for Channel 2	134
B.3. Source Radiance for Channel 3	135
B.4. Source Radiance for Channel 4	136
B.5. Source Radiance for Channel 5	137
B.6. Source Radiance for Channel 6	138
B.7. Source Radiance for Channel 7	139
B.8. Source Radiance for Channel 8	140

CHAPTER I

INTRODUCTION

Spectroradiometry is the process of characterizing radiant power in terms of its distribution in the electromagnetic spectrum. Most applications of spectroradiometry deal with electromagnetic radiation that is generated by virtue of an elevated source temperature. This type of radiation is referred to as thermal radiation and is generally characterized in the spectral range from the vacuum ultraviolet (VUV) region, through the visible region and to 15 microns in the infrared (IR) range. Spectral power measurements are essential to a broad range of scientific and engineering endeavors such as optical design, chemical analysis, radiation heat transfer engineering, and the development of combustion systems. The present project is concerned with the development of a spectroradiometer to measure the infrared radiation from rocket and jet engine plumes. The analysis of plume radiation provides information about the combustion processes in the engine. The information is useful in evaluating engine performance parameters and in helping to detect anomalous engine operations.

A spectroradiometer for plume studies should be designed to cover as much as possible of the thermal radiation wavelength range from 200 nanometers to 15 microns. However, because of the characteristics of available detectors and optical components, this range cannot be covered by a single instrument. The instrument developed in this project was designed to cover the 1- to 5-micron region, which

is a primary spectral region of interest in plume radiometry [1]*. The coverage of this range is aided by the availability of a high-performance infrared detector. Specifically, the indium antimonide (InSb) detector is a fast, highly sensitive photovoltaic detector that covers the 1- to 5-micron range [2].

The major components of a spectroradiometer, as reviewed by Walker, et al [3], are depicted in the block diagram of Figure 1.1. The input radiation is collected by a receptor (e.g., an objective lens or a Cassegrain telescope) and passed into the wavelength disperser. Prism monochromators, grating monochromators, and circular variable interference filters are examples of wavelength dispersers. They isolate and transmit the radiation in a narrow spectral band. The power in the transmitted spectral band is measured by the radiation detector. The wavelength disperser continuously scans the band position across the spectral range of operation of the spectroradiometer. During the scanning process, the wavelength disperser continuously outputs a voltage representing the instantaneous value of the center wavelength of the band of transmitted energy. The wavelength disperser employed in this project is a 1/2-meter, grating spectrograph (see Chapter II). The data acquisition system in Figure 1.1 must process and record the two analog voltage signals it receives, and pass the data on to the data reduction system. The function of the data reduction system is to convert the two signals

* Numbers in brackets refer to similarly numbered references in the List of References.

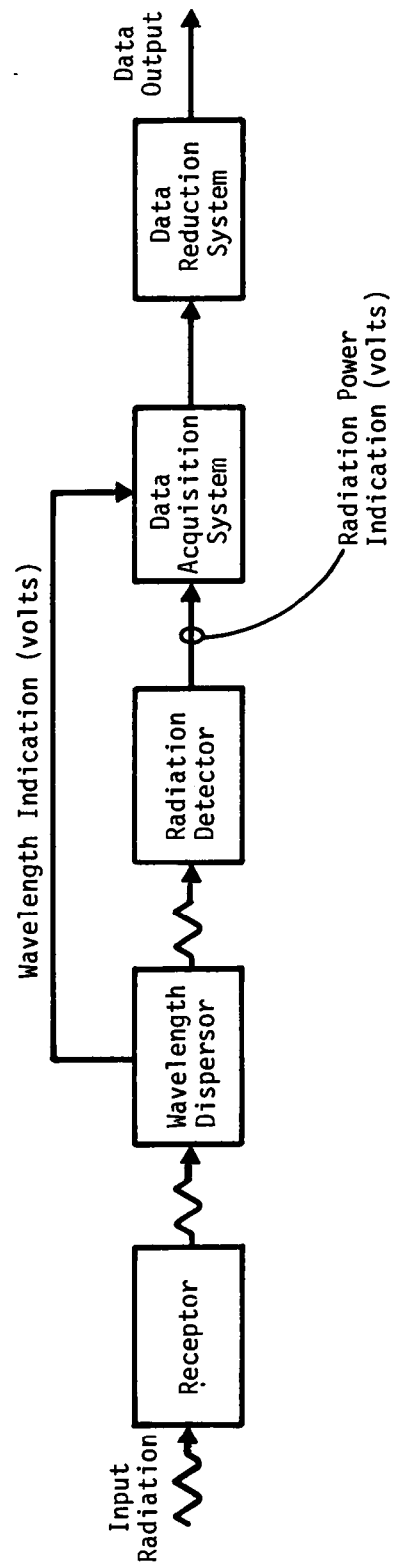


Figure 1.1. Block Diagram of a Typical Spectroradiometer.

to the characteristic spectral power distribution function of the input radiation.

The scope of the work presented in this thesis is the design and development of the data acquisition and data reduction systems depicted in Figure 1.1. To enhance the operational characteristics of the spectroradiometer, digital data acquisition techniques are employed. Data acquisition and data reduction are accomplished with customized digitizing circuitry, a dedicated microcomputer, and customized applications software. The designs and implementations of the data acquisition system and the applications software will be presented in detail.

To understand the basic design requirements, suppose the input radiation is blackbody radiation as described by Planck's law [4]. With the radiation receptor in Figure 1.1 configured for source radiance measurement (i.e., field of view uniformly filled), the source radiance as a function of absolute temperature, T , and wavelength, λ , is as follows:

$$L_b(\lambda, T) = \frac{C_1}{\pi \lambda^5 [\exp(C_2/\lambda T) - 1]} \quad (1.1)$$

The radiance unit is power per unit source area per unit wavelength per unit solid angle ($\text{watts/cm}^2 \cdot \text{micron} \cdot \text{sr}$), and the parameters C_1 and C_2 are constants [4]. A plot of the spectral radiance of a 1000°C blackbody, in the 1- to 5-micron range, is shown in Figure 1.2. In general, the analog data function (detector signal versus wavelength signal) presented to the data acquisition system will

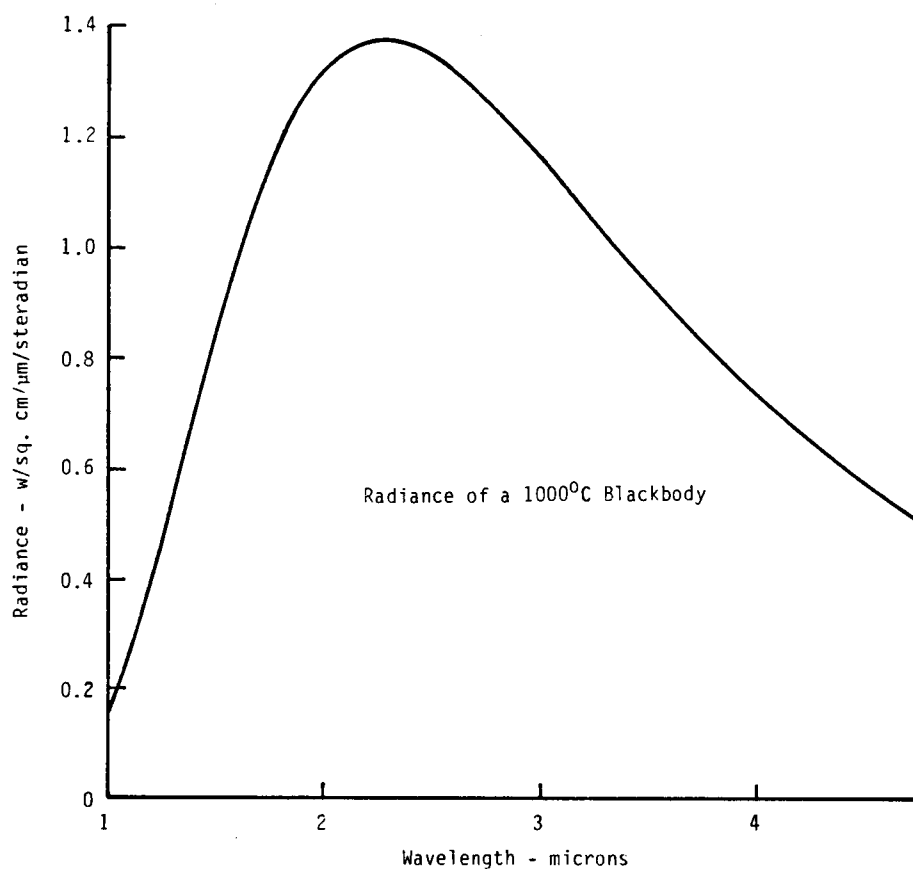


Figure 1.2. Spectral Radiance of a 1000°C Blackbody.

bear little resemblance to the input distribution of Figure 1.2. To illustrate, with a grating monochromator as the wavelength disperser, the analog data function could resemble the plot of Figure 1.3. The differences in the shapes of the two distribution functions originate from the fact that the performance characteristics of the receptor, the wavelength disperser, and the detector are wavelength dependent.

At this point, the design requirements can be outlined. The data acquisition system must acquire, digitize, and store a set of data points, where each data point consists of the instantaneous wavelength indication, W_i , and the corresponding instantaneous detector signal, S_i :

$$D_i \equiv (W_i, S_i), \quad 1 \leq i \leq N. \quad (1.2)$$

The set of N data points constitutes one scan of the spectroradiometer through its spectral range. Typical scan times for fast scanning spectroradiometers are on the order of one second.

The data reduction system must transform each of the raw data points back to its corresponding wavelength, λ_i , and radiance, L_i , values:

$$P_i \equiv (\lambda_i, L_i), \quad 1 \leq i \leq N. \quad (1.3)$$

It is evident that this transformation requires two different calibration functions. A wavelength calibration function is required to transform W_i to λ_i :

$$\lambda_i = f(W_i). \quad (1.4)$$

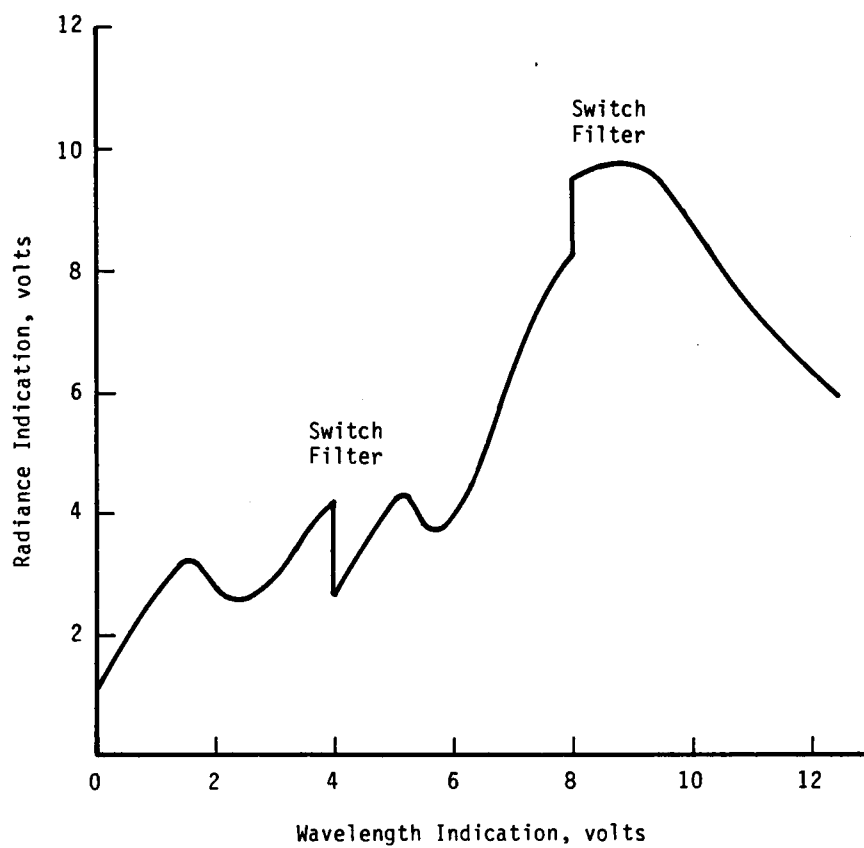


Figure 1.3. Illustration of a Possible Output Function of a Spectroradiometer.

Similarly, a spectral power (inverse) responsivity function, $R(\lambda_i)$, is required to convert the detector voltage signal, S_i , to L_i :

$$L_i = [R(\lambda_i)] S_i. \quad (1.5)$$

The unit of $R(\lambda_i)$ is radiance per volt.

In summary, the required data output representing the input radiation in Figure 1.1 is a set of N discrete data points that specify the source spectral radiance as

$$L(\lambda, T) \big|_{\lambda = \lambda_i} = L_i, \quad 1 \leq i \leq N. \quad (1.6)$$

There are two side issues which will be addressed at this time in order to help clarify the material that follows. First, it is desirable that the set of points be taken at uniform wavelength intervals, because it simplifies the wavelength calibration function, Equation (1.4). That is, it is desirable for $\Delta\lambda$ to be a constant, where

$$\Delta\lambda = (\lambda_{i+1} - \lambda_i). \quad (1.7)$$

This was accomplished in the design of the data acquisition system by digitizing the data at uniform intervals, ΔW , of wavelength indication W . By relying on the fact that the transmitted wavelength varies linearly with W (see Chapter II), the transmitted wavelength expressed in Equation (1.4) becomes

$$\lambda_i = \lambda_0 + \Delta\lambda \left[\frac{W_i - W_0}{\Delta W} \right], \quad 0 \leq i \leq (N - 1). \quad (1.8)$$

The wavelength calibration procedure becomes that of measuring λ_0 and W_0 corresponding to the first (zeroth) data point, and measuring

the $\Delta\lambda$ and ΔW intervals between data points. All parameters on the right side of the equation are constants except W_i . The operations indicated in Equation (1.8) are implemented in the data reduction software (see Chapter V), and the calibration procedure (Chapter VI).

The second side issue is the question of how many discrete data points are adequate for one spectral analysis. The answer depends on the characteristics of the input radiation, as well as the bandwidth of the energy reaching the detector. The slit function for a grating monochromator with equal widths of the entrance and exit slits is triangular [5]. That is, for monochromatic input energy, the detector output signal representing transmitted energy versus wavelength is triangular.

Similarly, for continuous spectral input energy, the spectral distribution of the transmitted band is also triangular. The triangle's full width at half maximum (FWHM) bandwidth is given by the product of the exit slit width and the monochromator's inverse linear dispersion ($\Delta\lambda/\Delta x$) at the exit focal plane. The FWHM is not strictly constant over a broad spectral range, but dividing the monochromator's spectral range by the FWHM gives an approximation of the number of "slices" or "spectral band elements" in the spectral analysis.

If the input energy is Planckian, only one data point per spectral band element could suffice for a spectral analysis. However, if the input radiation contains monochromatic lines or absorption bands (the usual case), then several data points are required to define the triangular spectral bands. In this project, the number of data points per band element was limited to a maximum of approximately

eight. This means that there are approximately eight data points available to describe the slit function of a monochromatic line. The exact number depends on the slit widths as explained above.

The data acquisition system was designed to provide a selection of eight, four, two, or one data points per spectral band element (see Chapter III). The operator can trade spectral resolution for faster scan time and faster data reduction.

Chapter II presents a brief review of spectroradiometers and describes the analog data generation. Chapter III presents the design of the data acquisition system. Chapter IV covers the data acquisition software, and Chapter V covers the data reduction software. Chapter VI presents the experimental results of the software checkout phase. The results of a preliminary spectroradiometer application are also presented in Chapter VI. Finally, a concluding discussion is presented in Chapter VII.

CHAPTER II

SPECTRORADIOMETER DESCRIPTION

The modern era of grating spectroscopy began when the Bausch and Lomb Company started producing high-quality, replicated, plane diffraction gratings [6]. This occurred around 1950. Simultaneously, Fastie and others were improving mechanical grating mounts and slit designs for plane grating monochromators [7, 8, 9]. These developments were based on the earlier designs of Ebert (1889) who described the original plane grating spectrometer and Czerny and Turner (1930) who worked to improve the spectral resolution of monochromators.

A spectrometer based on the Czerny-Turner system is shown in Figure 2.1. In operation, the entrance slit is uniformly filled with the radiation of an extended source image. The radiation passing the entrance slit is collected and collimated by spherical mirror M1 and directed onto grating G. The f/number of the objective lens is sized to uniformly fill M1 and the grating with light. The incident angle α is expressed relative to the grating normal $\vec{N}$, and the grating disperses the radiation through angles dependent on wavelength. The wavelength dispersed at angle β is collected by M2 and focused at the exit slit. The direction of the grating rulings is perpendicular to the plane of Figure 2.1, as are the directions of both slits. The power in the wavelength (or more accurately, the narrow band of wavelengths) passing the exit slit is intercepted and measured by the detector.

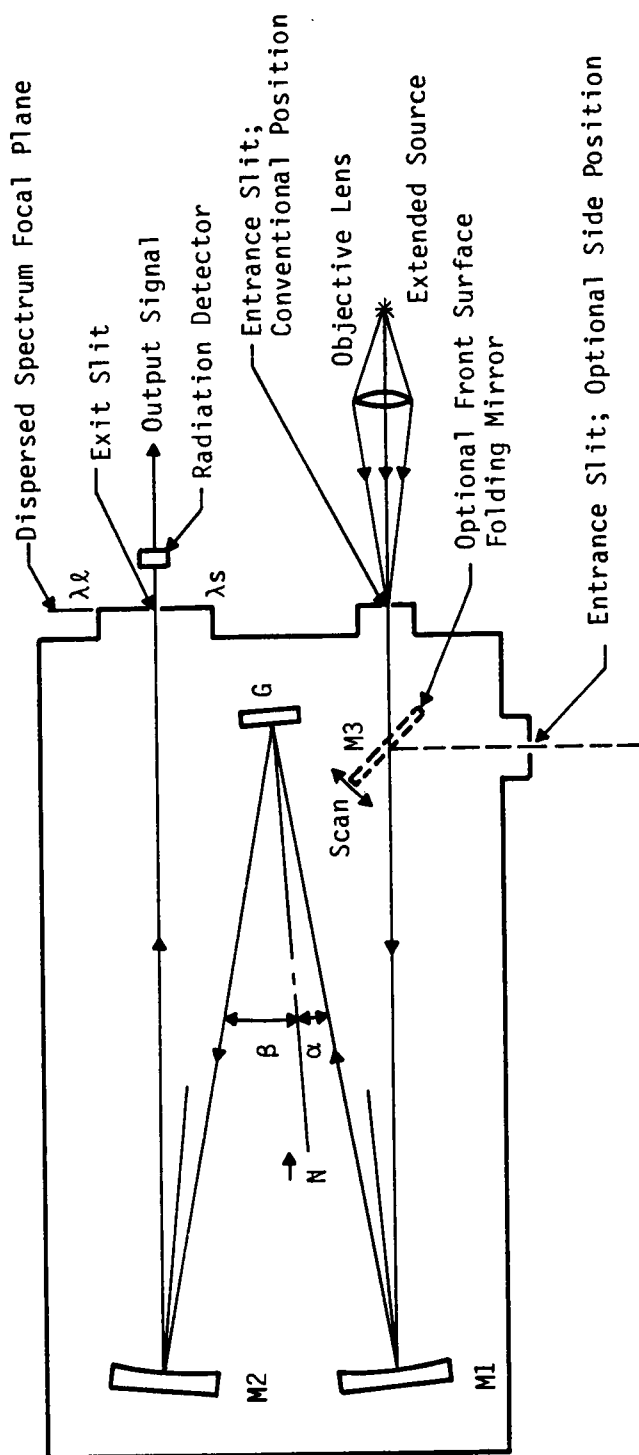


Figure 2.1. Spectrometer Concept Design Based on the Czerny-Turner System.

The wavelength passed by the exit slit is given by the grating equation as [6]:

$$m\lambda = a (\sin \alpha \pm \sin \beta), \quad (2.1)$$

where α and β are the incident and diffraction angles, a is the grating groove spacing, and m is the diffraction order. The minus sign applies when α and β are on opposite sides of the grating normal. A spectral range is scanned by rotating the grating about an axis perpendicular to the plane of Figure 2.1. Typical scan times are on the order of minutes.

The equation indicates that at constant α and β , the detector will receive energy at wavelength λ in the first order ($m = 1$), at $\lambda/2$ in the second order ($m = 2$), and at higher orders. The usual procedure for making unambiguous measurements is to work in the first order and eliminate higher orders by using a long wavelength pass interference filter at the entrance or exit slit.

A variation of the Czerny-Turner system is the spectrograph, which utilizes an exit focal plane instead of an exit slit. Referring to Figure 2.1, wavelengths diffracted at angles slightly away from β will be focused at positions adjacent to the exit slit. A spectrograph is constructed by enlarging M2 to collect energy over a range of diffraction angles and replacing the exit slit with a film strip, or a linear detector array, to intercept a wide segment of the dispersed spectrum. Each wavelength component of the input radiation at the entrance slit will be focused with unit magnification at a unique position on the exit focal plane. The shorter wavelengths will be imaged near the λ_s position indicated in Figure 2.1, while

the longer wavelengths are imaged on the λ_g end. The first order dispersion of wavelengths along the focal plane can be estimated by differentiating Equation (2.1) to get the angular dispersion

$$\frac{d\beta}{d\lambda} = \frac{1}{a \cos \beta}. \quad (2.2)$$

Since an incremental change of β relative to the grating normal generates the same incremental change in the direction of the radiation reflected off M2, the linear dispersion on the focal plane is found by multiplying by the focal length (f) of M2:

$$f \left(\frac{d\beta}{d\lambda} \right) = \frac{f}{a \cos \beta}. \quad (2.3)$$

The product $fd\beta$ will be recognized as an incremental distance, $d\ell$, along the focal plane. Therefore, the inverse linear dispersion is given as

$$\frac{d\lambda}{d\ell} = \frac{a \cos \beta}{f}. \quad (2.4)$$

Typical units can be Angstroms per millimeter or microns per millimeter. Since $\cos \beta$ is generally greater than 0.9, the inverse linear dispersion can be estimated from the ratio (a/f). The product of $(d\lambda/d\ell)$ and the exit slit width gives the FWHM passband of the energy reaching the detector as discussed earlier.

Another variation of the Czerny-Turner system is the installation of a plane, first-surface mirror, M3, between a side-mounted entrance slit and M1, as shown in Figure 2.1. Usually, M3 serves no function other than to make the entrance slit accessible from the side of the instrument. However, SPEX Industries, Inc. provides a design where M3 is mounted on the armature of a fast scanning

galvanometer. The design is used for fast, repetitive scanning over a small spectral range. The mirror is scanned back and forth, which has the effect of varying α and β without moving the grating. The mirror position is scanned in a sawtooth fashion (angle vs. time) which causes the dispersed spectrum to translate back and forth on the exit focal plane. The translation is parallel to the plane of Figure 2.1.

The translation is generated by the effect that as M3 rotates, the apparent position of the entrance slit translates in the field of view of M1. If M3 rotates through an angle $\Delta\theta$, the translation in the apparent position of the entrance slit is $2L\Delta\theta$, where L is the distance between M3 and the entrance slit. Since M1 and M2 have equal focal lengths, the system magnification is unity, and the position of the exit plane image will also translate by $2L\Delta\theta$. Consequently, the wavelength of light passing the exit slit will change by the increment $\Delta\lambda = 2L\Delta\theta (d\lambda/d\lambda)$ where $d\lambda/d\lambda$ is the inverse linear dispersion from Equation (2.4). Therefore, it is concluded that $\Delta\lambda$ is linearly related to $\Delta\theta$ and a measurement of $\Delta\theta$ will provide the means for a wavelength calibration of the form required by Equation (1.8). A galvanometer with an integral position sensor was obtained from General Scanning, Inc. and installed in place of the standard SPEX model, which had no position sensor. The position sensor outputs a sawtooth waveform whose instantaneous voltage is an indication of the angular position of M3 and consequently, the wavelength of the light reaching the detector. The position voltage can be calibrated

to indicate detected wavelength. The position of the grating is maintained stationary.

Linear image translations of up to ± 12.5 mm can be generated on the exit focal plane. This corresponds to a peak-to-peak excursion of approximately 7° in the position of M3. The mirror-scanned mode of operation is much faster than the usual grating-scanned mode of operation.

The scanning mirror technique has the primary advantage of converting a normally slow scanning spectrometer to a fast scanning device. It has the disadvantage that the spectral range of the scan is too limited for general use in spectroradiometry. To counter this disadvantage, it is necessary to use multiple slit-detector pairs that are uniformly spaced on the exit focal plane.

Figure 2.2 shows a diagram of the actual spectrometer, which uses a 1/2-meter SPEX Industries grating spectrograph. The input light is chopped at the rate of 400 Hz at the entrance slit. Chopping provides a way to discriminate against detector signal noise and to improve data accuracy. The spectrograph has a 4-inch long exit focal plane originally intended for use with a film pack. For this project, a customized detector attachment was assembled for installation on the focal plane. An exploded view of the assembly is shown in Figure 2.3. The detectors are photovoltaic indium antimonide (InSb), size 0.5 by 2 mm, mounted in flat packs. Each flat pack is epoxied to a detector base which is mounted on the detector base support. Each detector is equipped with a 0.010-inch wide slit and an order-sorting filter. The detectors are spaced at regular,

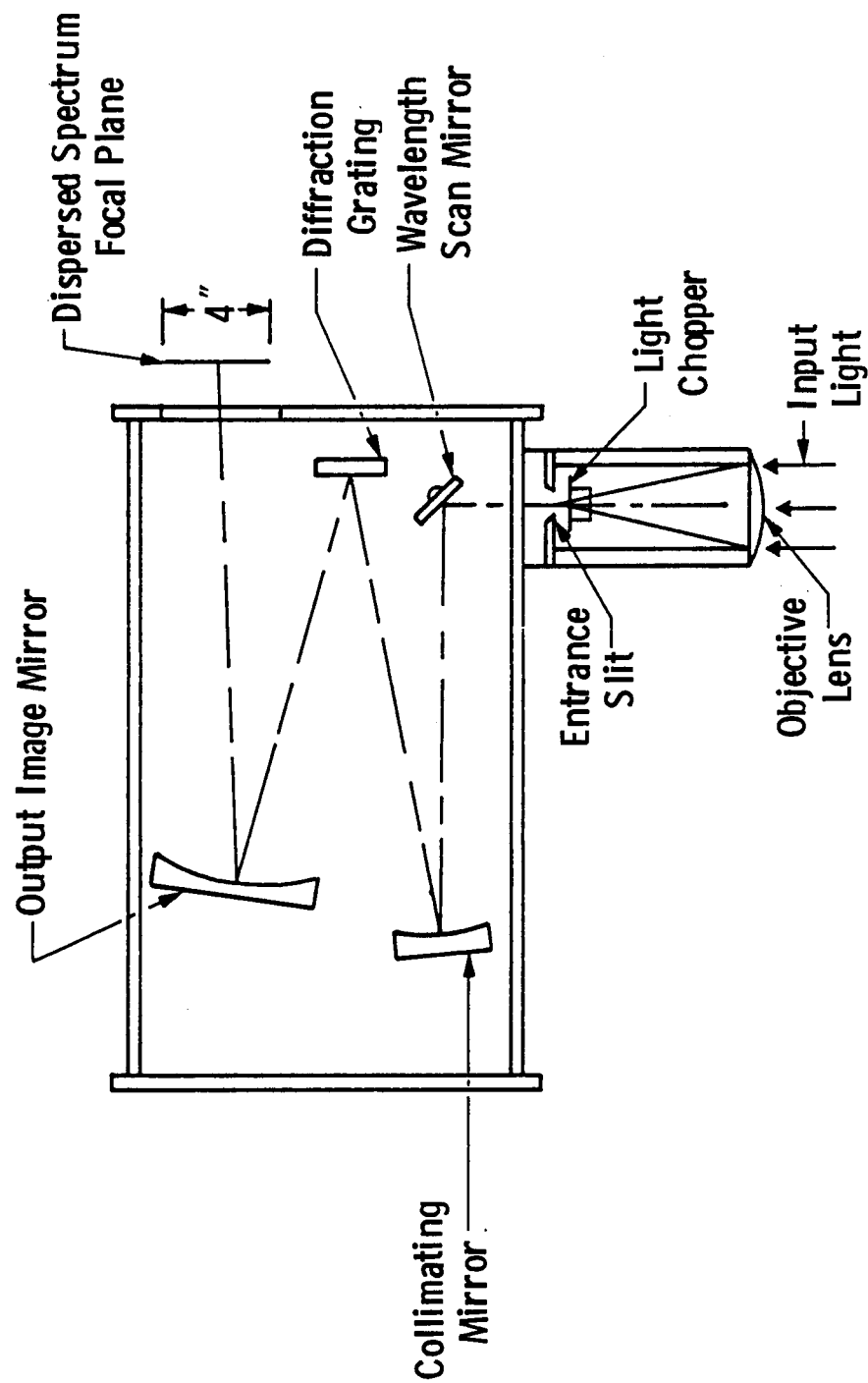


Figure 2.2. Spectroradiometer Optical Diagram.

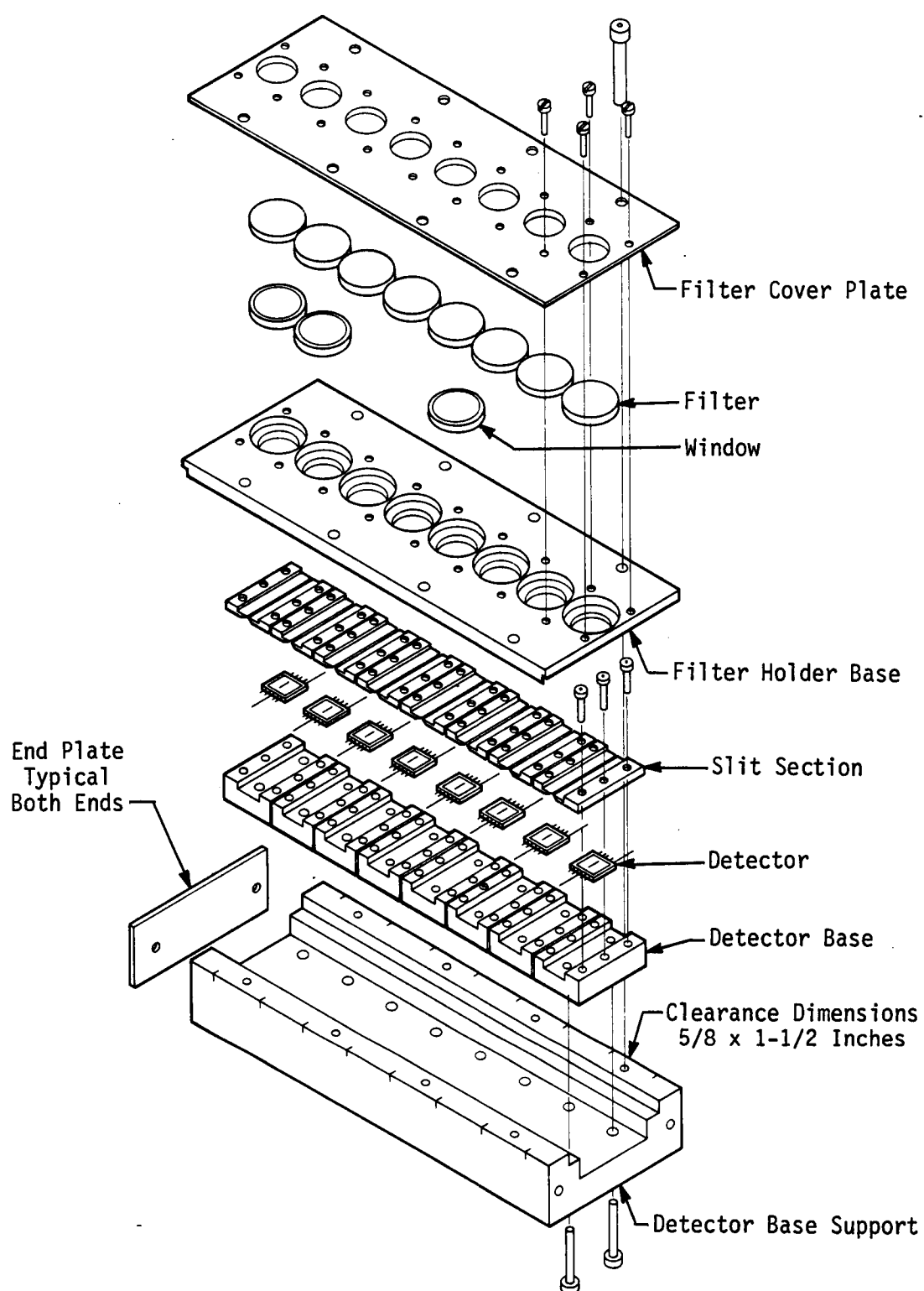


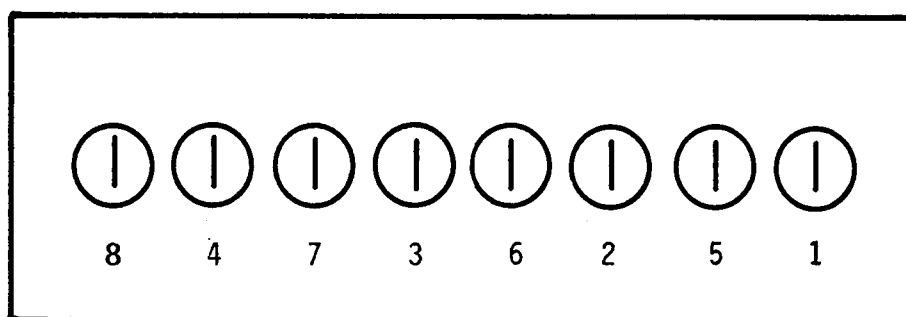
Figure 2.3. Spectroradiometer Detector Assembly.

1/2-inch intervals. There are four detectors, spaced at 1-inch intervals, that are filtered to operate in the first order of the dispersed spectrum. Four additional detectors are filtered to operate in the second order and are physically interspersed between the first order detectors. The assembly is mounted in an evacuated housing on a liquid nitrogen (LN_2) cooled baseplate. The vacuum interface window (not shown in Figure 2.3) is made of sapphire.

The peak-to-peak excursions of the scan mirror are increased to the point where the concatenated ranges of the eight detector channels provide effectively continuous coverage over a broad spectral range. The effect is that eight different spectroradiometers operate simultaneously, with each covering a slightly different spectral range. Figure 2.4 shows the relative positions of the eight channels as seen from the output image mirror. The nominal spectral range of each channel is also shown.

The grating has 76 lines/mm and a 3-micron blaze wavelength. It provides an inverse linear dispersion of 0.0256 micron/mm in the first order and 0.0128 micron/mm in the second order. The detector slits, shown in Figure 2.3, were adjusted to the width of 0.254 mm to produce a calculated FWHM passband of 0.0065 micron in the first order and 0.00325 micron in the second order. The slit widths can be less if higher resolution is required. Each first-order detector covers a 0.65-micron span and each second-order detector covers half as much. The total span is 4 microns, nominally, starting at about 1.1 micron.

Detector Mounting Arrangement



Channel No.	Nominal Wavelength Ranges
1	1.11 - 1.43
2	1.44 - 1.77
3	1.79 - 2.09
4	2.11 - 2.41
5	2.56 - 3.21
6	3.22 - 3.88
7	3.92 - 4.52
8	4.59 - 5.18

Figure 2.4. Detector Locations and Nominal Spectral Range.

Each InSb detector is equipped with an autoranging amplifier/demodulator which synchronously rectifies the chopped detector signal and generates an analog detector signal for input to the data acquisition system. Also, the instantaneous amplifier gain is coded and passed to the data acquisition system.

A review of the amplifier/demodulator circuit design, Figure 2.5, will clarify the detector signal format. The original rectangular-wave, detector signal is generated by chopping the input light with a circular chopper wheel (see Figure 2.2). The wheel simultaneously chops the light beam in an optical switch that outputs a reference signal as indicated in Figure 2.5. The reference signal is of the same frequency and phase as the detector signal. The current-to-voltage transformation at the detector pre-amplifier (IC-1 in the figure) is controlled at either 1 or 100 by transistor-transistor-logic (TTL) control signal E. Control signals A through D allow only one of the amplifiers IC-3 through IC-6 to be active at a time. The control signals are generated in the 16HD8 programmable array logic (PAL) device. Depending on the state of the control signals A through E, the amplification of the detector signal is either 1, 3, 10, 30, 100, 300, 1000, or 3000. The amplified signal is synchronously rectified in the demodulator, filtered at IC-8, DC-amplified at IC-9 and fed out as the analog detector signal. The signal voltage is proportional to one-half of the peak-to-peak detector current.

The circuitry consisting of IC-10 through IC-14 senses the over-ranged ($> 9V$) or under-ranged ($< 1.5V$) conditions of the analog

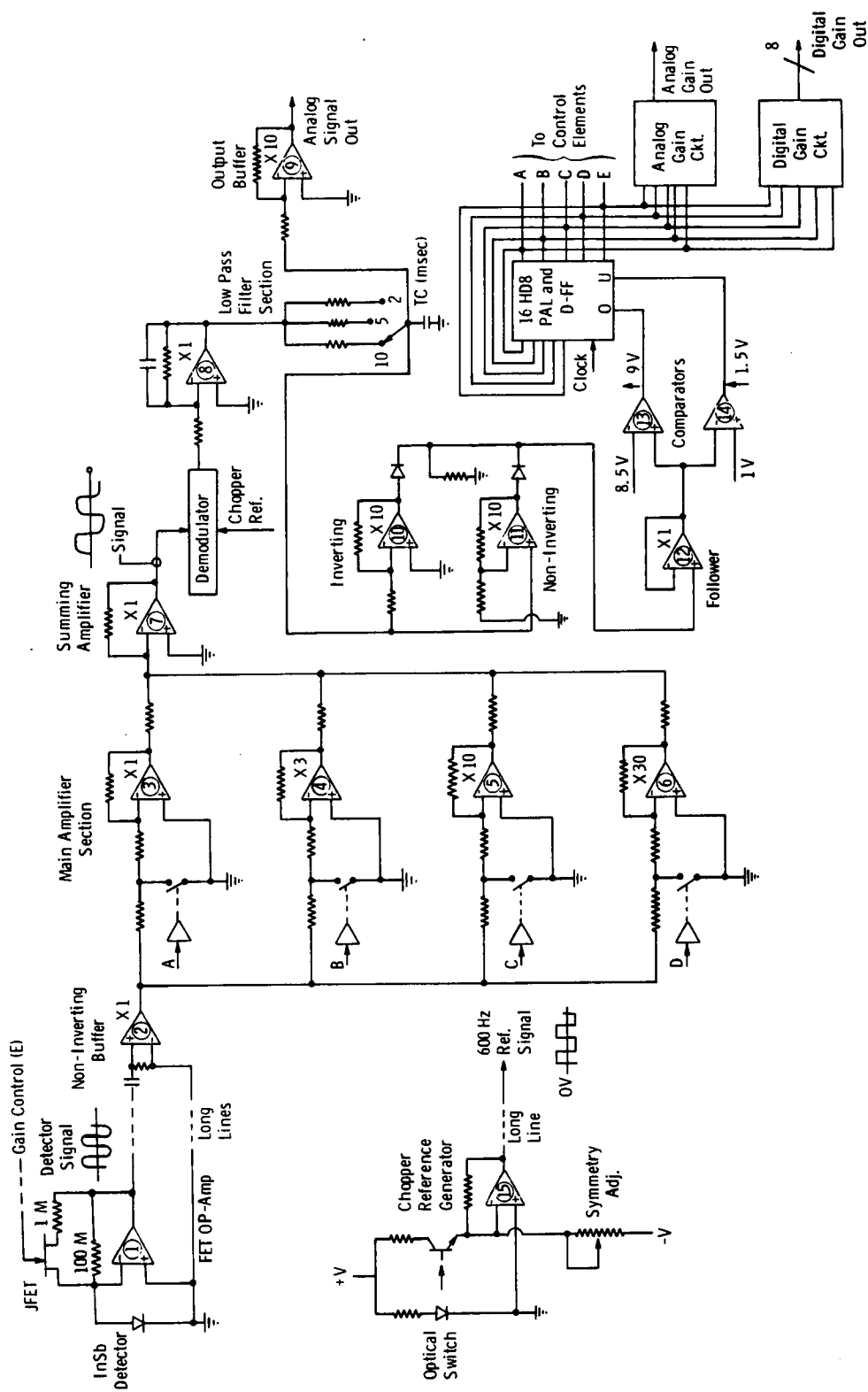


Figure 2.5. Autoranging Amplifier/Demodulator Circuit Design.

signal and generates TTL signals to the PAL at the O or U inputs. The PAL generates a new control state to bring the signal back on scale [10]. Simultaneously, the new gain state is output on the eight lines from the digital gain circuit as shown. These eight TTL signals are subsequently encoded to three TTL bits to indicate one of the eight possible analog gain states (see Chapter III).

In conclusion, the data acquisition system must simultaneously acquire the analog detector signal, the instantaneous 3-bit gain indication, and the mirror position voltage. To acquire a single analog data sample means to sample, hold, digitize, and latch each voltage for reading. After the data sample is read, it must be stored on disk. Subsequently, in processing the stored digital data, programs must decode the 3-bit gain data and factor the indicated gain into the detector data at each indicated wavelength. This process must be repeated for each of the eight detector channels. The data reduction process will be discussed in Chapter IV.

CHAPTER III

DIGITAL DATA ACQUISITION SYSTEM DESIGN

1. DATA SYSTEM DESIGN CONCEPT

The essential design criteria for the data acquisition system originate in the operational requirements of the spectroradiometric system. Specifically, the hazardous environments of rocket test facilities usually preclude operator presence during the data acquisition processes. Therefore, the first design requirement is that the data acquisition process must be completely self-controlled, starting with a simple remotely activated signal.

Secondly, the data system must be capable of storing the spectral data on a removable storage medium for the purpose of permanent data retention. Also, the raw data storage capacity shall allow data run periods of 10 minutes or longer.

Thirdly, it is required that spectral scan times be on the order of 1 second. This translates into data rates up to 36 Kbytes per second for the system outlined in the two previous chapters.

Finally, it is required that the turn-around time for producing reduced radiometric data from raw data be minimized. To help realize this general requirement, certain hardware features were designed into the data acquisition system. The features include the digitization of data at constant, uniform wavelength intervals. This feature simplifies and shortens the data reduction process, because calibration data and raw radiometric data are referenced to a standard set of wavelengths. Another feature is that the data acquisition process

is rigorously controlled by the combination of hardware and a software routine to record and partition data on a scan-by-scan basis. This feature provides for selecting and reducing specific spectral scans without the necessity of reducing all of the data in a data run. The advantages and implementations of these features will become more obvious as the system is described in the following sections.

The simplified block diagram of the data acquisition system is shown in Figure 3.1. The original raw data consist of the eight detector signals and the mirror position signal. The data are digitized by nine analog-to-digital converters (ADCs) operating in parallel. A single convert command activates all ADCs. The digital data pass through dual-ported, first-in-first-out (FIFO) buffers to digital input ports on the microcomputer. During the data acquisition process, the microcomputer executes a program called RECORDAT.C (see Chapter IV) to acquire the data. The program is loaded and started prior to the actual data run, and is remotely activated to complete the data run. The program is designed to first accumulate the data from one spectral scan in memory and then write the complete scan to the disk. A scan partition longword hexadecimal (hex) number FF0000FF is attached to the end of each individual scan record. In this design, there is no limit to the number of scans that can be recorded, except for the capacity of the storage disk.

The time at the start of each successive spectral scan is acquired from the IRIG time code generator and attached to a header on each scan record.

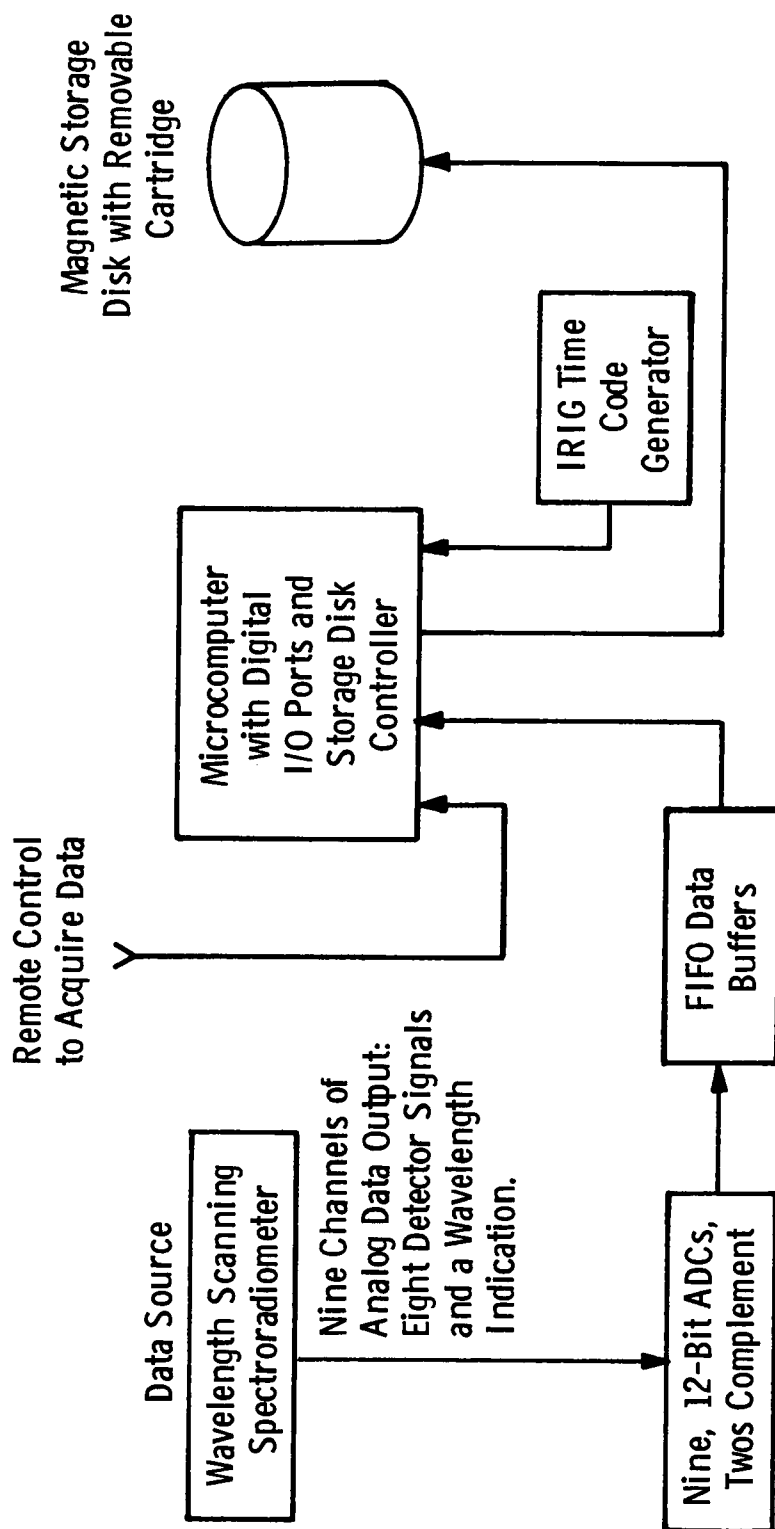


Figure 3.1. Major Components of the Data Acquisition System.

The function of the FIFOs is to provide 200 to 300 milliseconds of time buffering for the disk write operations without losing spectroradiometer data. The spectroradiometer generates data continuously except during the scan-mirror flyback and restart time. This interval is about 50 milliseconds and is not long enough for a disk write of one scan record. Data are written into the FIFOs in synchronism with the spectroradiometer scans and independently of the microcomputer operation. The RECORDAT program reads the data out of the FIFOs until they are empty. Therefore, the FIFO read and write operations are asynchronous and the overall spectral scan rate is determined by the speed of the microcomputer and its ability to store the previous record and start reading the current record before the FIFOs are filled. Scan times on the order of 1 second are typical.

The block diagram of Figure 3.2 shows the flow path typical of each of the eight detector signals. The analog signal goes to the ADC block which actually includes a sample and hold (S/H) amplifier, a 12-bit ADC, and digital data latches. The digitization is initiated by the "Clock," which simultaneously latches the 4-bit analog gain data. (Bit 4 of the gain data was not mentioned previously; it is a cold field-of-view bit which indicates a negative detector signal.) After the digitization is complete, the ADC block outputs a "Data Ready" pulse. The new digital data word is then written into the two, 1024-word deep FIFOs by the low-true "Write" pulse. Data are read out one word at a time by the low-true "Read" pulse; and the FIFO output data are latched for microcomputer reading via two, 8-bit digital data input ports. As will be discussed shortly, the

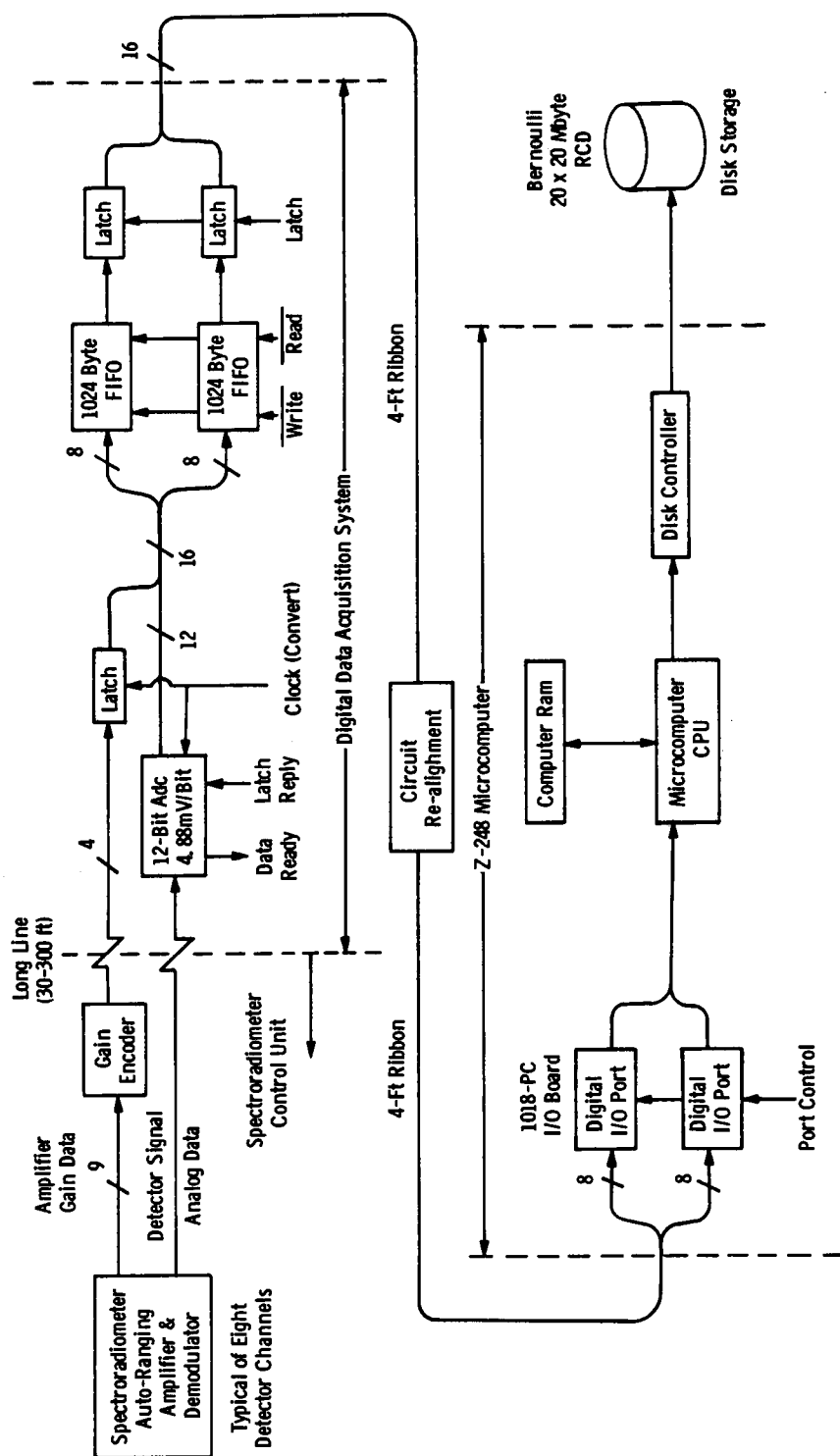


Figure 3.2. Basic Concept Design of One Detector Data Acquisition Channel.

control pulses required for this implementation are generated in data flow control circuitry which communicates with all major components in the system.

The circuit realignment indicated in Figure 3.2 is required to match the input format of the Model 1018-PC I/O board, which has one ground wire for each data bit.

The bit pattern at the microcomputer input ports is shown in Figure 3.3. Each new word consists of a least significant byte (LSB) and a most significant byte (MSB), and carries the information shown. The data are read and stored sequentially under software control. The number of Clock pulses generated per spectral scan depends on the data interrupt wavelength interval setting (see the following section) and is selectable at 2000, 1000, 500, or 250 pulses per scan (nominal values). Therefore, the amount of data generated per scan is 36 Kbytes, 18 Kbytes, 9 Kbytes, or 4.5 Kbytes, respectively. These numbers follow from the fact that a single Clock pulse gives rise to 16 bytes of detector data and 2 bytes of mirror position data.

The recovery of the information in the detector data word, Figure 3.3, will be discussed in Chapter IV, Section 3.

To complete the review of the data acquisition concept in Figure 3.2, the data are read and stored in random access memory (RAM) under software control. By the end of each scan, the FIFOs have been emptied and all data have been transferred to the RAM. After the FIFOs become empty, the accumulated data are written to the disk. As discussed earlier, the next spectral scan starts before the disk write is finished. New data accumulate in the FIFOs until the microcomputer resumes read operations.

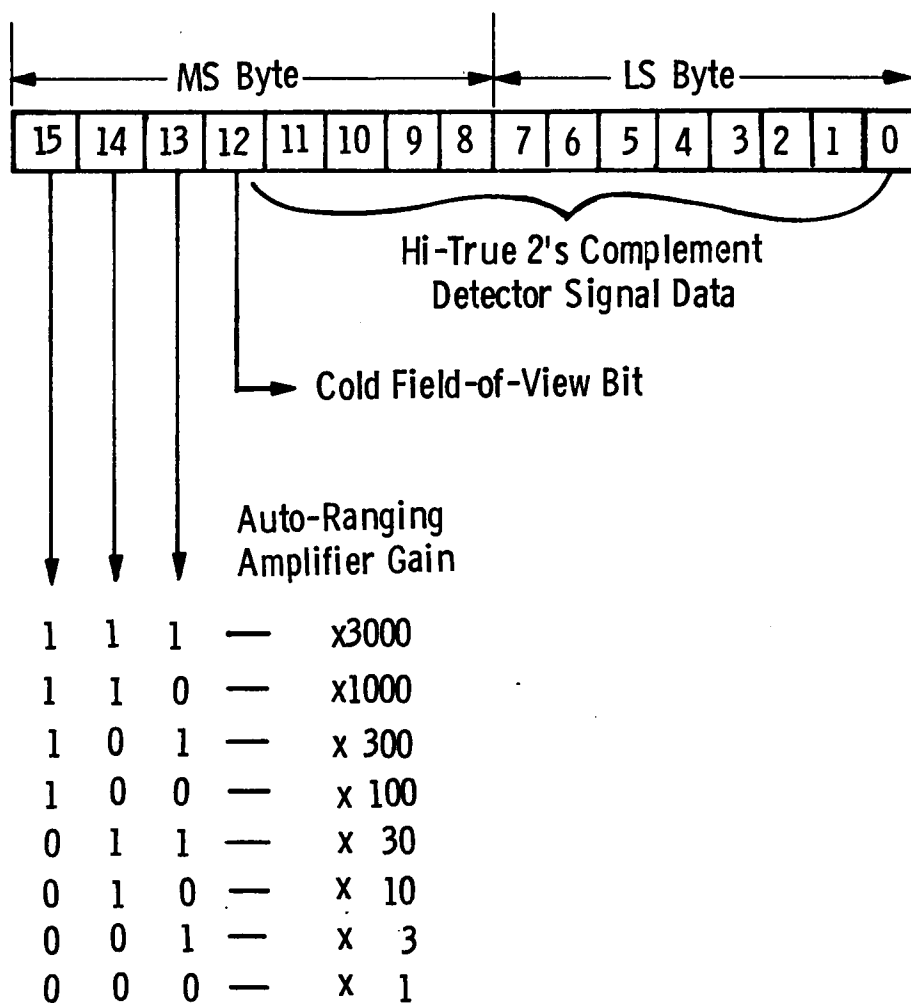


Figure 3.3. Bit Pattern Typical of All Eight Detector Channels.

This data acquisition system was implemented with the Zenith Z-248 microcomputer and the Iomega Bernoulli Removable Cartridge Disk (RCD) Recorder. The applications programs (see Chapter IV) were written in C programming language, using the compiler C86 from Computer Innovations, Inc. The operating system is Microsoft, MS-DOS, Version 3.

The data acquisition system concept design is now presented in its fully developed form in Figure 3.4. This figure shows a simplified block diagram of all circuitry up to the microcomputer I/O ports. The circuitry across the top of the figure was discussed above and is the same circuitry shown across the top of Figure 3.2. This circuitry is repeated for each of the eight detector channels. The function of Figure 3.4 is to introduce two major components in the data acquisition system design. The first component is the circuit board containing the data interrupt generator and the mirror position ADC. This circuitry monitors the scan mirror sync pulse and the mirror position signal, and it generates the system sync pulses and data interrupts which control all data acquisition functions. A simplified circuit diagram of this board is shown in Figure 3.5, and the functions of the circuit are described in the next section.

The second major component introduced in Figure 3.4 is the data control board. This board controls the data flow in the system. As indicated in the figure, it communicates with the microcomputer via the control status line and the control commands line. It processes the interrupts from the data interrupt generator and issues the Data Clock pulses to the ADCs. After all nine ADCs respond with

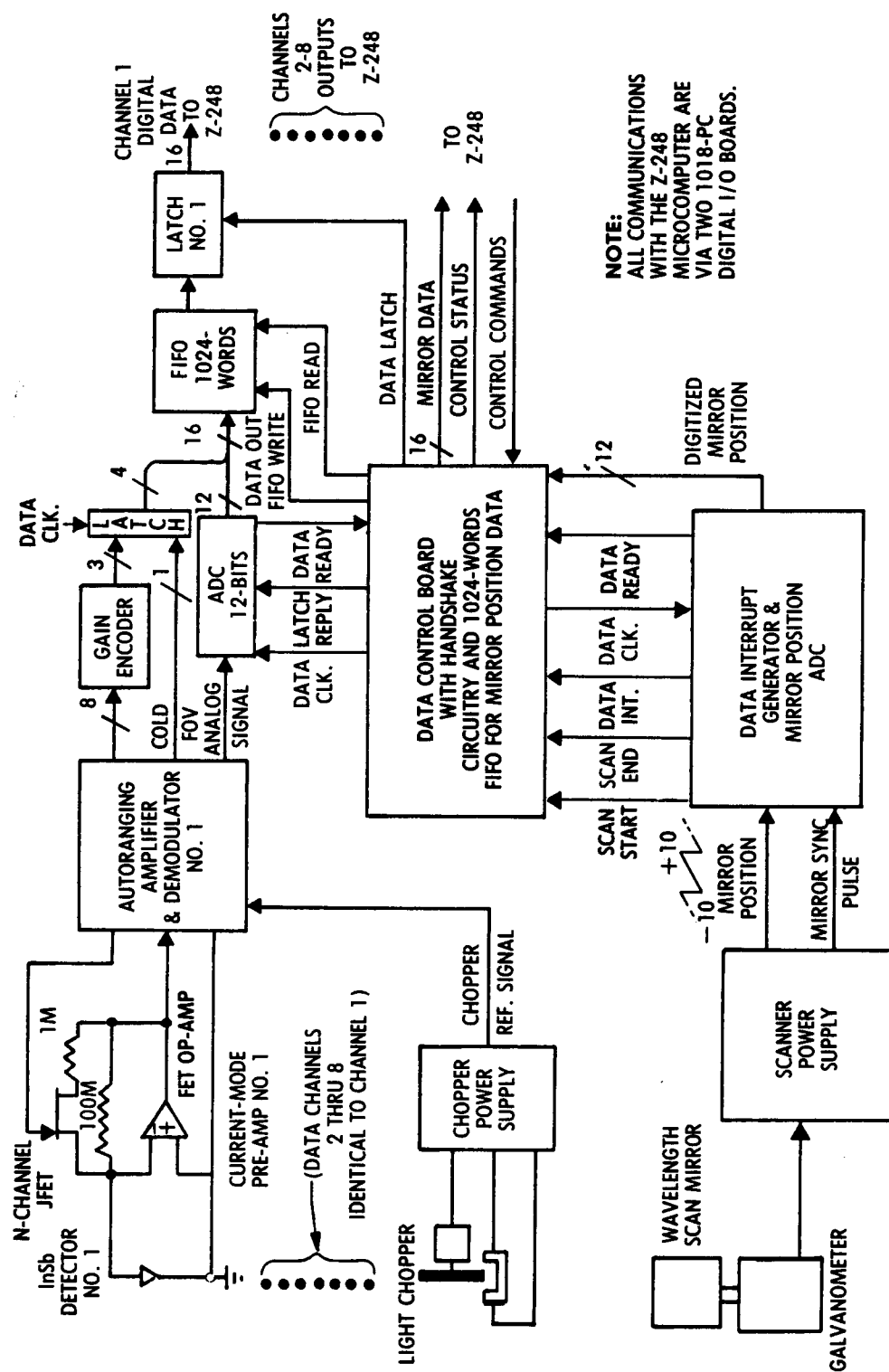


Figure 3.4. NIR/IR Spectroradiometer Data System Concept Design.

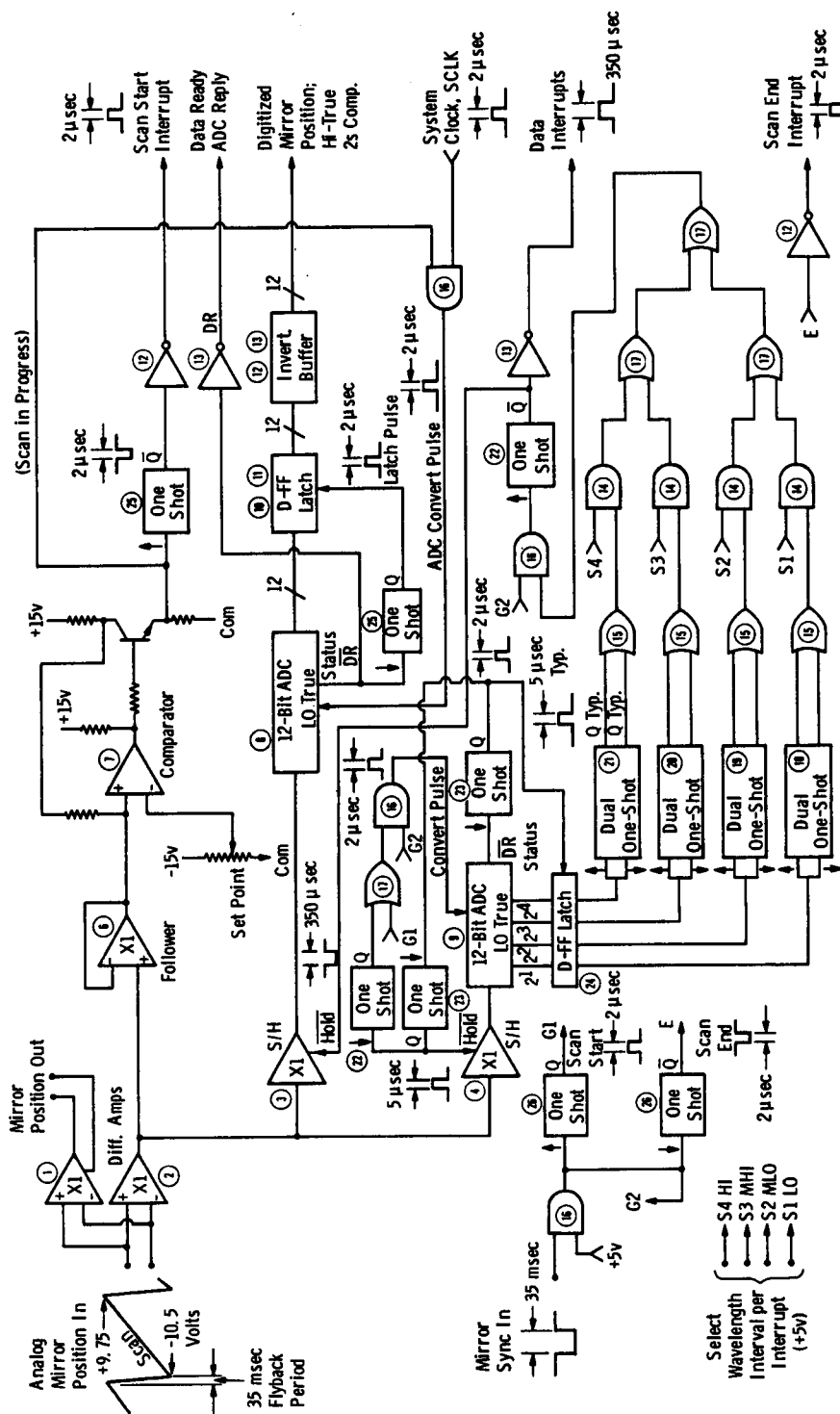


Figure 3.5. Data Interrupt Generator Concept Design.

Data Ready pulses, the data control board issues FIFO write pulses to all 18 FIFOs. It also processes data read commands from the microcomputer and issues FIFO read and latch pulses to output 18 new bytes of data to the microcomputer input ports. The functions of the data control board are described more fully in Section 3 of this chapter.

2. DATA INTERRUPT GENERATION

The data acquisition process is initiated by TTL pulses (data interrupts) generated by the circuitry shown in Figure 3.5. The circuit has three primary inputs: (1) the mirror position ramp voltage; (2) the mirror sync (TTL) signal, both shown on the left of the figure, and (3) the system clock (SCLK), shown on the right side of the figure.

To inject a few words about nomenclature, all integrated circuits (IC) will be referred to by number, which is the circled number in or near the component in the figure. This is also the IC number on the circuit board. Most TTL integrated circuits have several sections (e.g., IC-15 has four sections). This is the reason some TTL gates have the same IC number in the figure.

Much of the data flow control in this system is accomplished using the edge-triggered, one-shot multivibrators. An example is the SN74LS221, which is a TTL part with two sections on one IC. In the block diagrams, such a one-shot is always shown with an up-arrow or a down-arrow at its input to indicate that it is wired for rising or falling edge triggering, respectively. A Q or $\bar{Q}$ is shown at its output to indicate the use of either the positive or negative output

pulse, respectively. Also, the duration of the Q or $\bar{Q}$ is indicated on the output line.

The interrupt generator circuit is designed to generate data interrupts at constant, uniform intervals of wavelength. This is realized by generating interrupts at constant, uniform intervals of the mirror position voltage. As shown in Figure 3.5, the voltage varies linearly from -10.5 volts to +9.75 volts. The sweep period is adjustable from 0.5 seconds to about 10 seconds. The sweep is followed by the mirror flyback, which takes about 35 milliseconds. The sweep signal passes through buffer amplifiers, IC-2 and IC-6, to comparator IC-7. The function of the comparator set point is to provide a constant voltage (wavelength) reference at which data acquisition for each new scan is to commence. The comparator trigger point is set near -9.5 volts in order to "skip over" some ramp nonlinearities in the -10.5 to -9.5 range. When the comparator triggers, it causes the one-shot IC-25 to trigger and generate the Scan Start Interrupt at the output. This signals the data control board that a new scan has started.

The Scan End Interrupt is generated by IC-26 on the falling edge of the mirror sync as shown at the lower left of Figure 3.5. It is output by IC-12 at the lower right. A scan start pulse (not a data interrupt) is generated in the other section of IC-26, on the rising edge of the mirror sync signal. This pulse is the first convert command going to the ADC (IC-9) at the start of each scan.

The function of IC-9 and the other circuitry on its output is to generate data interrupts at constant, uniform intervals of the

mirror position voltage. As shown in Figure 3.5, the voltage is fed to IC-9 through the S/H amplifier IC-4. The IC-9 is a 12-bit ADC whose least significant bit (LSB), 2^0 , represents 4.88 millivolts of analog voltage. The ADC repeatedly digitizes the input voltage as fast as it can operate. The first convert command of each scan is the scan start pulse fed in through connection G1 on IC-17. Successive convert commands are generated by the ADC's own data ready ($\overline{DR}$) output and IC-23. As shown, the output of IC-23 triggers the data latch at IC-24 and triggers the other half of IC-23 to generate the next 5-microsecond sample period for the S/H amplifier. At the end of the sample period, IC-22 is triggered to generate the next convert pulse. The ADC conversion time is about 25 microseconds.

At the end of the scan, the mirror sync goes low, G2 on the output of IC-16 goes low, and the stream of convert pulses is disrupted. Consequently, IC-9 remains dormant until the next scan start pulse occurs.

During the scan, the dual one-shots, IC-18 through IC-21, continually monitor the latched ADC bits, 2^1 , 2^2 , 2^3 , and 2^4 . A bit toggle on any line generates a 5-microsecond pulse which is fed through the OR gates in IC-15 and to AND gates in IC-14. Switch contacts S1 through S4 determine the data interrupt wavelength interval by putting a HI (+5 volts) on one, and only one, of the AND gates. The data interrupt wavelength intervals are designated as L0, ML0, MHI, and HI, corresponding to voltage intervals of 9.76, 19.5, 39.1, and 78.1 millivolts, respectively.

The selected pulse train is fed to a one-shot, IC-22, which outputs 350 microsecond data interrupts. These pulses are fed to the data control board, which triggers on the rising edges to generate the system clock (SCLK). The period of SCLK is 1 to 4 milliseconds and is dependent on both the mirror scan time and the data interrupt wavelength interval setting.

The SCLK pulses are used as the ADC convert commands throughout the system. They are fed back to the Data Interrupt Generator to trigger the mirror position ADC, IC-8, as shown in Figure 3.5. The digitized mirror position data are then fed back to the data control board (see next section) where they are read into a FIFO.

The experimental results show that the mirror position voltage of the first data point in a scan depends on the data interrupt wavelength interval. For the four options, LO, MLO, MHI, and HI, the voltages at the first data point are -9.4629, -9.4629, -9.4238, and -9.3847. To simplify the data reduction software, some leading data points are discarded in the first three options, and all data reduction starts at the mirror position of -9.3847 volts. Table 3.1 shows the mirror position voltages observed for the first few data points in each of the four data interval options. As expected, the intervals are 9.8, 19.5, 39.1, and 78.1 millivolts, respectively. Occasionally a data point will be 4.8 millivolts off the pattern. An example is the first point in the MHI column. This error amounts to one least significant ADC bit and is corrected by the software. The number of data points generated with each of the four options is 1984, 992, 496, and 248, respectively. Therefore, it is possible to

Table 3.1. Mirror Position Voltages for Each Data Interrupt Option.

Data Interrupt Wavelength Interval (mv)			
LO	MLO	MHI	HI
-9384.7	-9384.7	(-9379.9)	-9384.7
-9375.0			
-9365.2	-9365.2		
-9355.4			
-9345.7	-9345.7	-9345.7	
-9335.9			
-9326.1	-9326.1		
-9316.4			
-9306.6	-9306.6	-9306.6	-9306.6

trade spectral resolution for faster scan rates and extended run times, as discussed in the introduction.

To emphasize the important points of this section, data interrupts are generated at the same set of mirror positions during successive mirror scans, and the data interrupt intervals are stepwise variable by the factor of two.

3. DATA FLOW CONTROL

The data control board shown in the system block diagram of Figure 3.4 (page 32) controls all digital data generation and data flow up to the I/O ports of the microcomputer. The board interacts with the computer when the data acquisition program RECORDAT is executing. Figure 3.6 (sheets 1, 2, and 3) presents a simplified schematic of the circuitry on the board. A review of the board functions is a necessary prerequisite for understanding the data acquisition software described in the next chapter.

The data control board does a considerable amount of "hand-shaking" with its various communicants by monitoring the state (i.e., HI or LO) of input communication lines and by issuing appropriate control levels on output lines. All circuitry is TTL and all hand-shaking functions are implemented with either edge-triggered, one-shot multivibrators or set-reset (S-R) flip-flops. The functions of the circuitry in Figure 3.6 will be reviewed with the aid of the diagram in Figure 3.7 (sheets 1 through 4). The diagrams define and assign names to the various control signals. The vectors indicate source edges that trigger the generation of control pulses or level

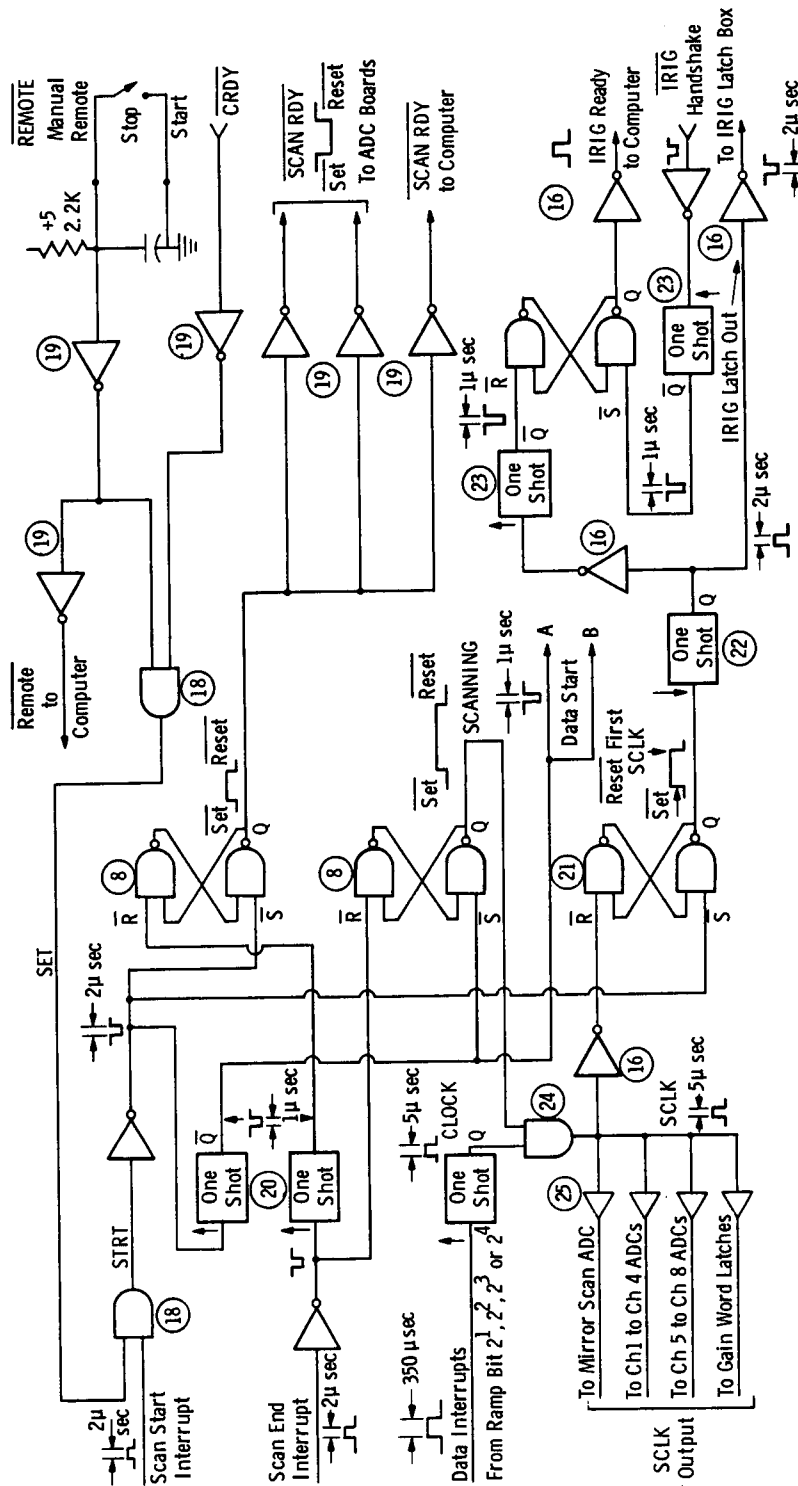


Figure 3.6. Data Control Board Design (Sheet 1 of 3).

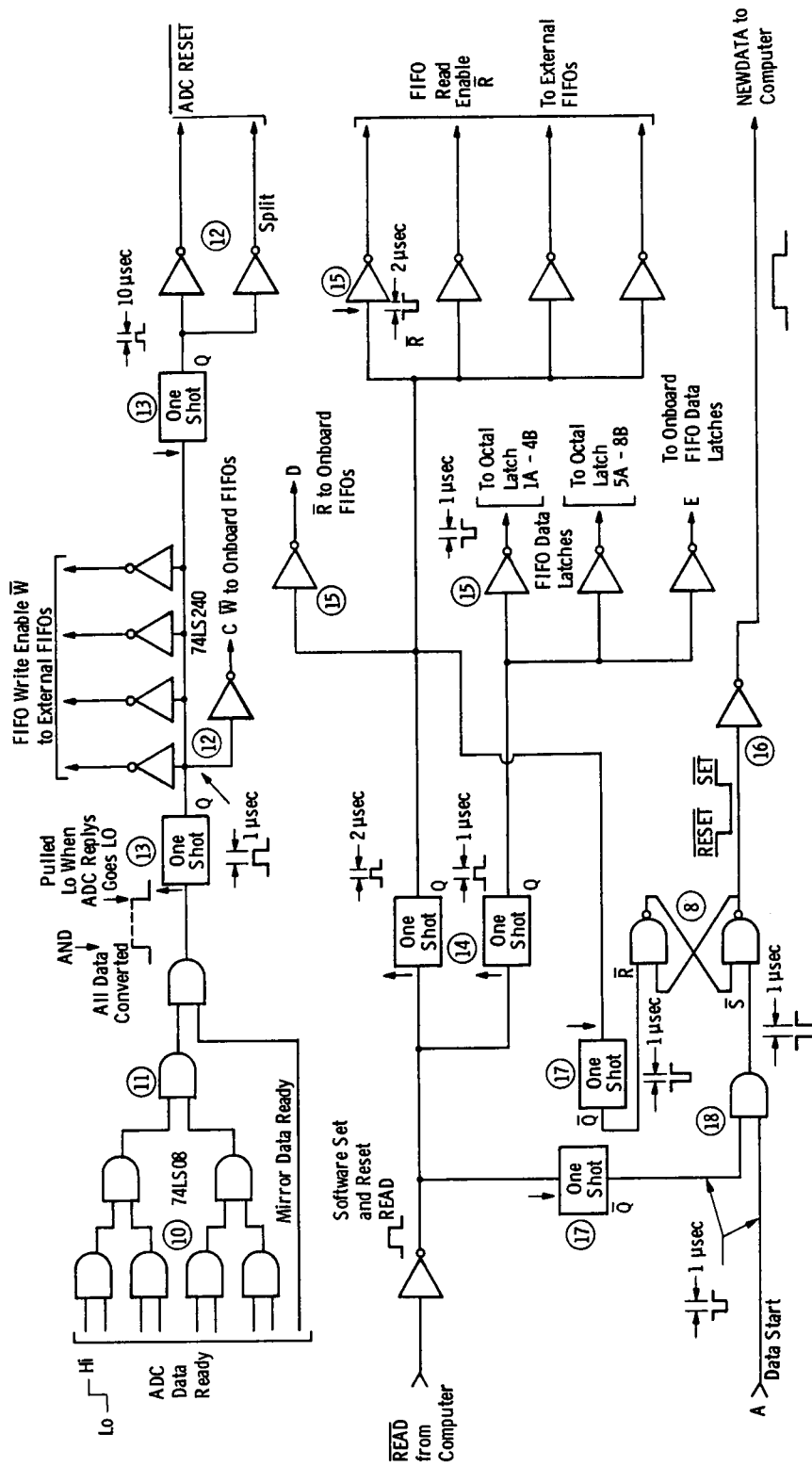


Figure 3.6 (Continued; Sheet 2 of 3)

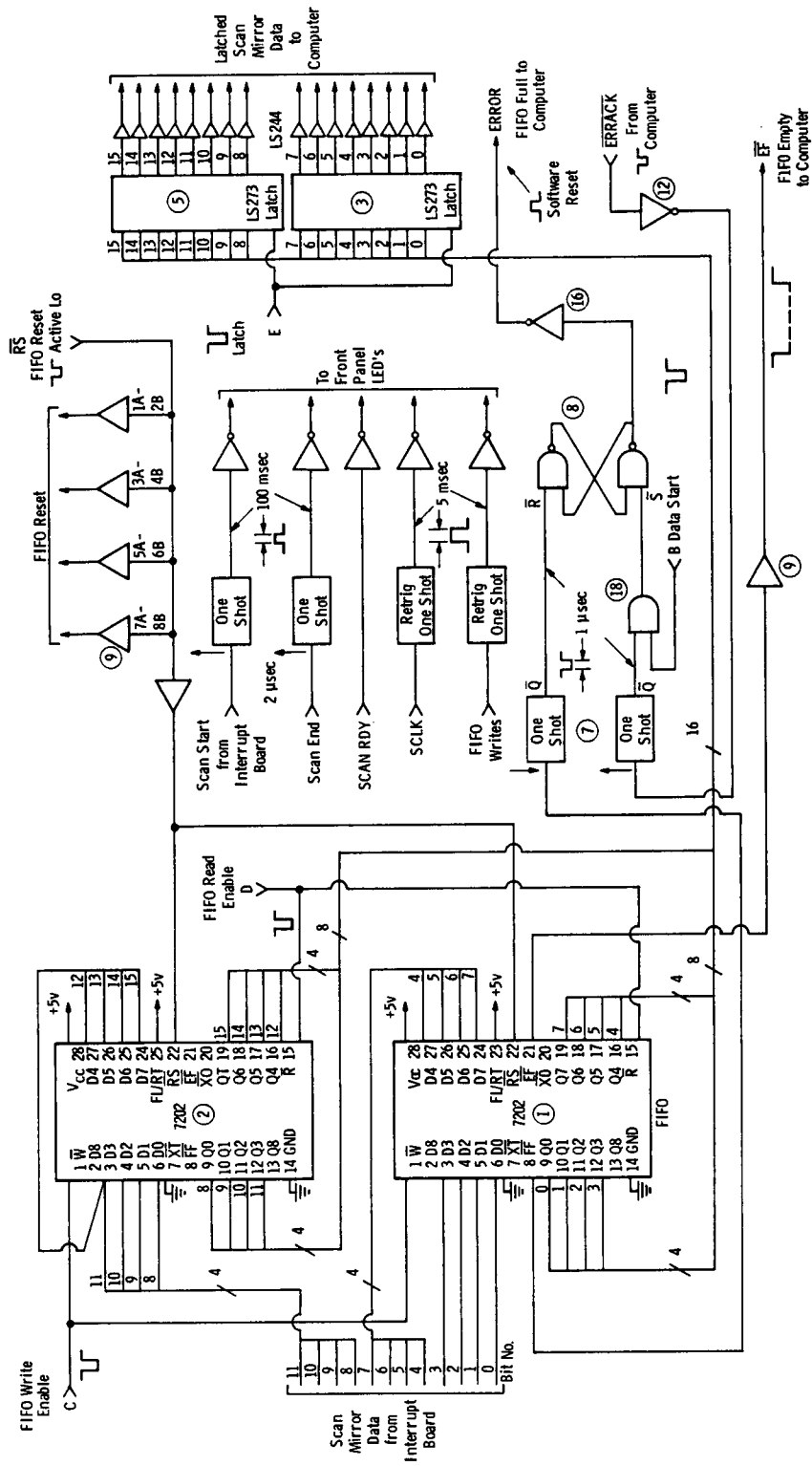


Figure 3.6 (Continued; Sheet 3 of 3)

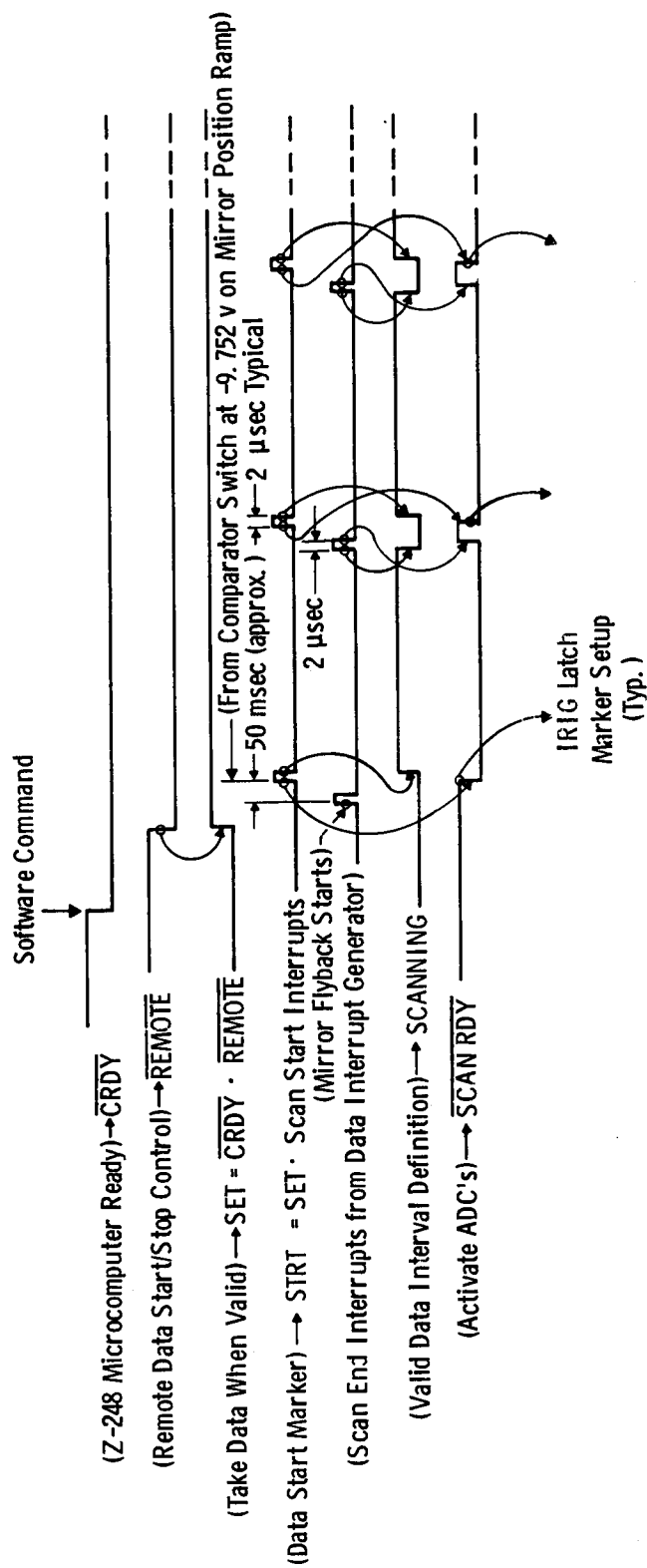


Figure 3.7. Data Flow Control and Handshake Functions (Sheet 1 of 4).

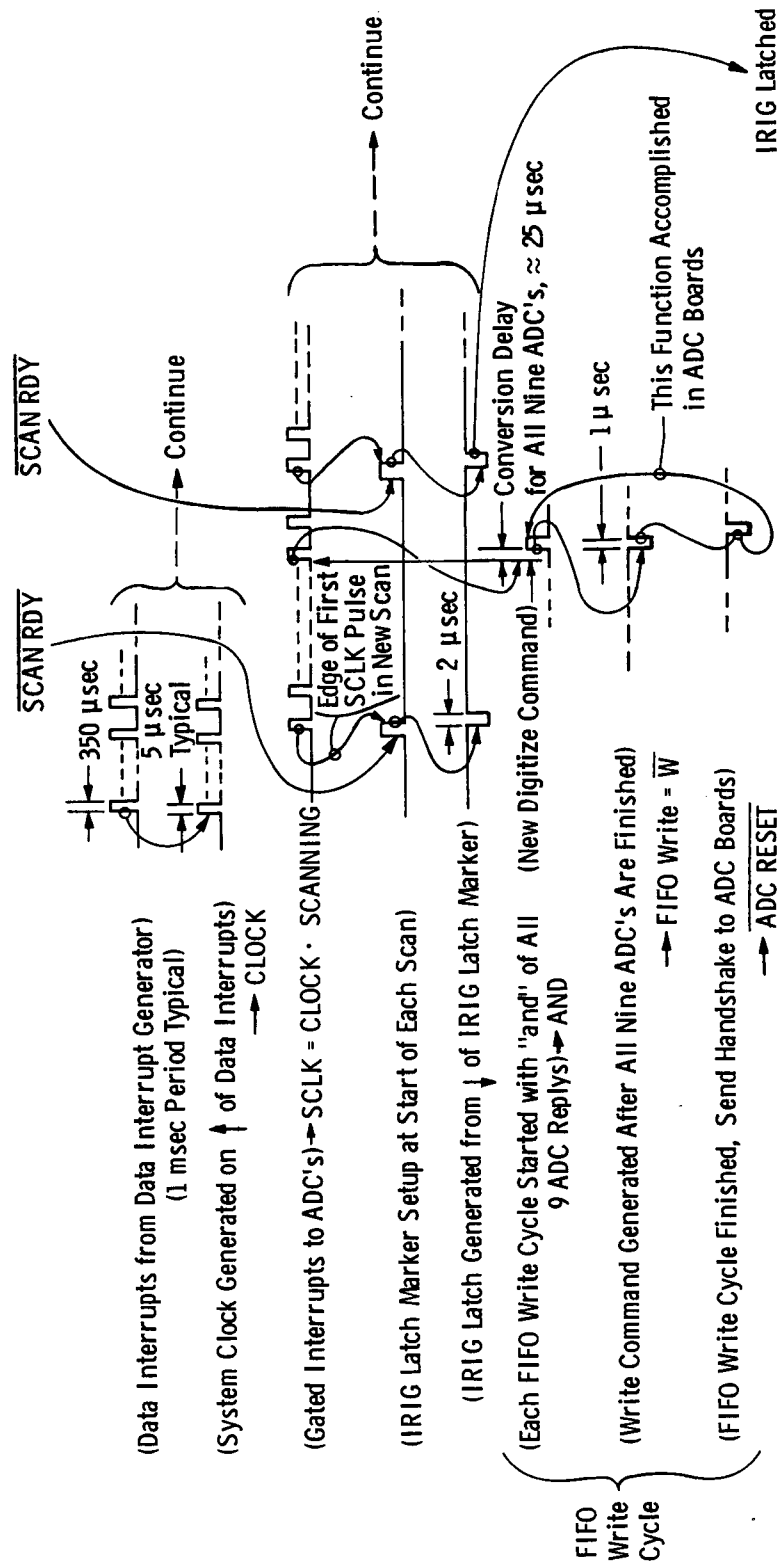


Figure 3.7 (Continued; Sheet 2 of 4)

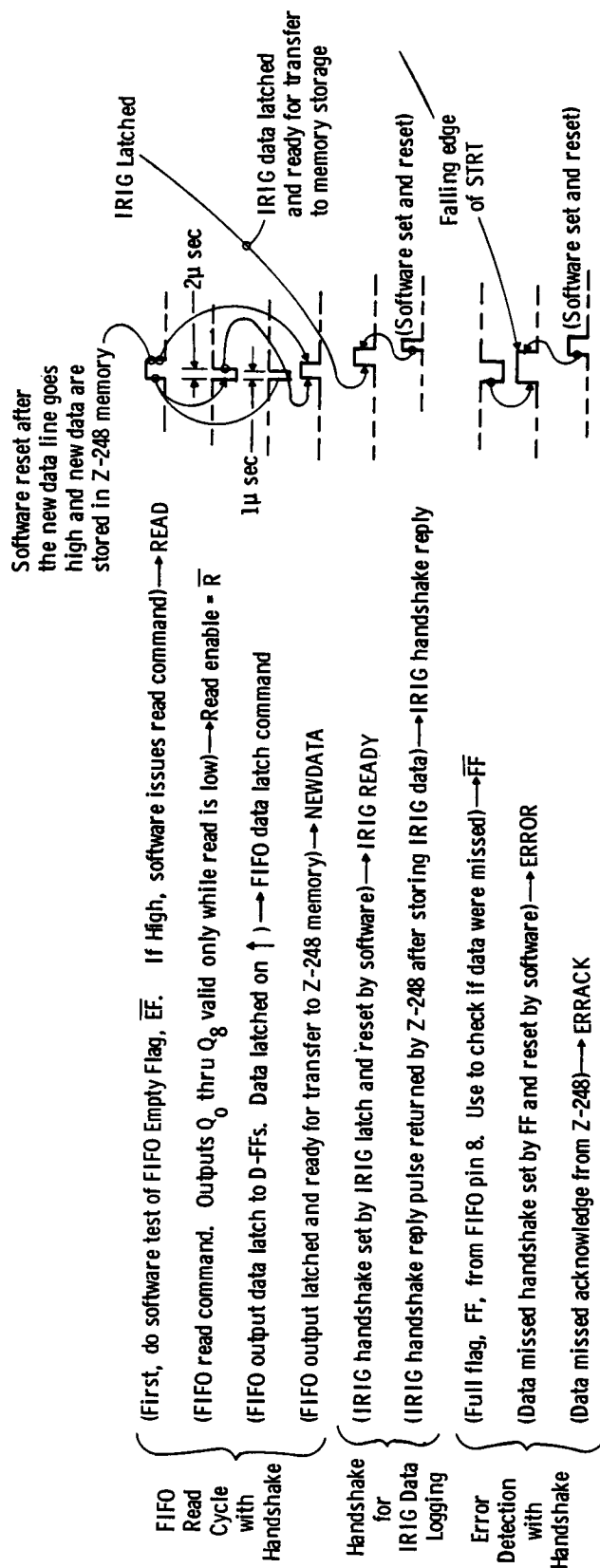
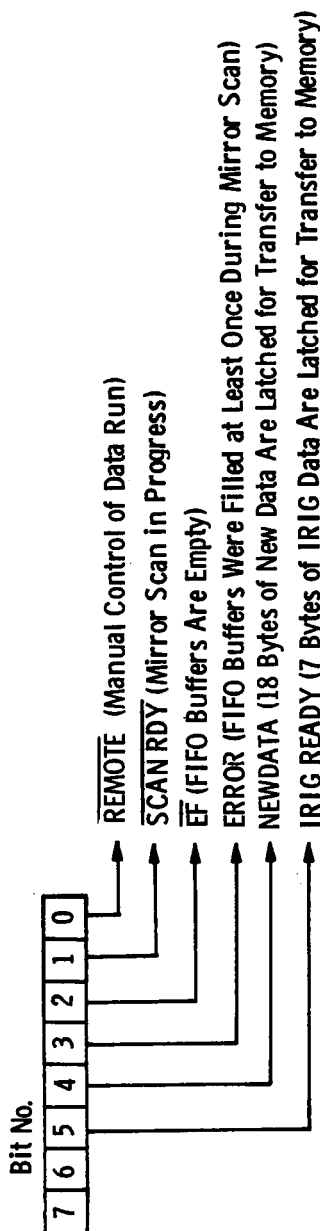
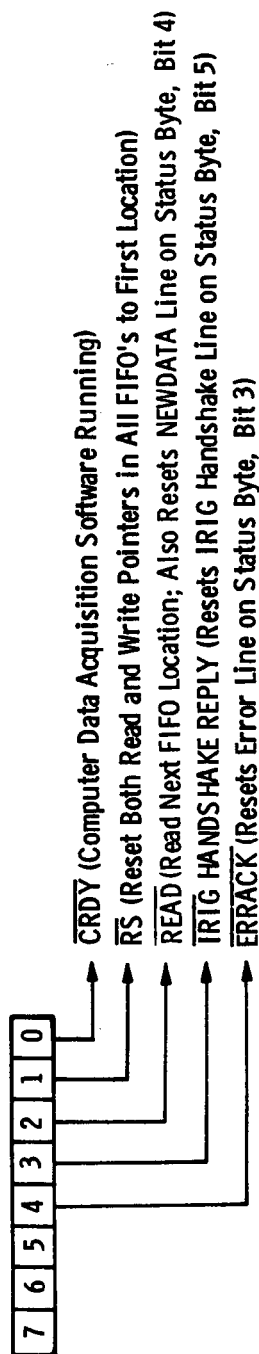


Figure 3.7 (Continued; Sheet 3 of 4)

Status Byte Specification; Data Transmitted to Microcomputer:



Control Byte Specifications; Data Returned by Microcomputer:



Notes: (1) $\overline{\text{READ}}$, $\overline{\text{IRIG HANDSHAKE}}$ and $\overline{\text{ERRACK}}$ Are All Inverted at the Input Before Application As Shown in Timing Diagrams.

Figure 3.7 (Continued; Sheet 4 of 4)

transitions. In general, there is a one-to-one correspondence between vectors and sub-circuits on the control board.

The data control board communicates with the microcomputer by transmitting a status byte and receiving a control byte. Figure 3.7 (sheet 4) assigns names and bit numbers to the various status and control signals. The bar over a variable name indicates that the specified condition is true for a low (0 volts) signal level.

The four generalized control board functions are:

1. it initiates the data acquisition process after sensing $\overline{\text{CRDY}}$ and $\overline{\text{REMOTE}}$;
2. it issues the system clock (SCLK) that causes all nine system ADCs to digitize and latch new data (a total of 18 bytes);
3. it monitors the nine ADC replies; and when the new data are digitized, it generates and issues a FIFO write ($\overline{\text{W}}$) command to transfer the new data to the FIFOs;
4. it monitors the FIFO READ command from the computer; and when true, it issues FIFO read enable ($\overline{\text{R}}$) pulses and data latch pulses; and after the data are latched, it issues a NEWDATA status to the microcomputer.

A data run starts by executing program RECORDAT which pulls $\overline{\text{CRDY}}$ low as shown in Figure 3.7 (sheet 1). After the operator manually activates $\overline{\text{REMOTE}}$, the SET level goes high and the scan start markers (STRT) are gated on. These functions are evident in the circuitry across the top of Figure 3.6 (sheet 1). The rising edges of the STRT pulses (1 per scan) set IC-8 (top section) and generate

the $\overline{\text{SCAN RDY}}$ outputs to the ADCs and the microcomputer. The scan end interrupt causes the reset of IC-8.

The falling edge of each STRT pulse generates a trigger that sets IC-8 (bottom section) and generates the SCANNING gate. This gate turns on IC-24 to pass the CLOCK pulses as shown at the lower left of Figure 3.6 (sheet 1). The SCANNING gate is reset by the scan end interrupt as shown in the figure. These functions lead to the output of the system clock, SCLK, as illustrated in the top three lines of Figure 3.7 (sheet 2).

The next two lines on sheet 2 of Figure 3.7 illustrate the generation of the IRIG data latch. An IRIG latch marker is set at the same time as $\overline{\text{SCAN RDY}}$ and reset on occurrence of the first SCLK pulse in each scan. This is the function of the S-R flip-flop, IC-21, in Figure 3.6 (sheet 1). When reset occurs, an IRIG latch pulse is generated and fed out to latch the current IRIG time. Simultaneously, an IRIG Ready is issued to the microcomputer as shown at the lower right of Figure 3.6 (sheet 1). These functions are further illustrated in Figure 3.7 (sheet 3).

The FIFO write cycle is illustrated in Figure 3.7 (sheet 2). After each SCLK pulse, there is a wait period of about 25 microseconds for all ADCs to report that digitization and data latching are complete. When true, the AND goes high and triggers the generation of the write enable ($\overline{W}$) pulse. It is output to all 18 FIFOs in the system. The trailing edge of the $\overline{W}$ pulse generates the $\overline{\text{ADC RESET}}$ output which releases the system ADCs for the next SCLK convert

command. The circuitry for the FIFO write cycle is shown across the top of Figure 3.6 (sheet 2).

The FIFO read cycle is illustrated in Figure 3.7 (sheet 3) and implemented by the circuitry on the lower part of Figure 3.6 (sheet 2). When the RECORDAT program is ready for the next data point (18 bytes of data), it issues a $\overline{\text{READ}}$ command that triggers a FIFO read enable ($\overline{\text{R}}$) pulse and a data latch pulse. After latching, the NEWDATA line is set. After the microcomputer stores the new data, it resets the $\overline{\text{READ}}$ line which resets the NEWDATA output.

The only digital data passing through the data control board are the mirror position data as shown in Figure 3.6 (sheet 3). All of the detector data FIFOs are on separate system boards. Since all FIFOs are clocked by the same read, write, and reset pulses, error checking can be accomplished by monitoring only one of the FIFO's error flags. Therefore, the empty flag ($\overline{\text{EF}}$) and full flag ($\overline{\text{FF}}$) are monitored at IC-1 (Figure 3.6, sheet 3). The $\overline{\text{EF}}$ is output directly through a line buffer (IC-9) and the $\overline{\text{FF}}$ triggers IC-7 and IC-8 to generate an output ERROR status. The ERROR status is software reset by $\overline{\text{ERRACK}}$ on the control byte.

The final point about the control board circuitry is that the program RECORDAT initializes the FIFO pointers by issuing a reset ($\overline{\text{RS}}$) pulse at the start of execution. The reset pulse is fed to all 18 FIFOs as indicated in Figure 3.6 (sheet 3).

4. DATA INPUT/OUTPUT IMPLEMENTATION

The data acquisition circuitry and the microcomputer communicate through two multi-channel, digital input/output (I/O) boards, the Model 1018-PC manufactured by Industrial Computer Designs, Inc. Each board has 96 I/O lines (12 by 8 bit ports) in addition to 32 input-only lines (4 by 8 bit ports). Each 8-bit port is individually addressable and can be software configured for either input or output [11]. The C86 library function "outportb(portno,value)" is used to write a hexadecimal value to a specific output port. Similarly, the function "inportb(portno)" is used to read a hexadecimal value fed into a specified port [12].

Figures 3.8 and 3.9 show the operational configuration of I/O boards 1 and 2, respectively. The data of one detector channel requires two input ports as discussed earlier. The functions of the I/O ports are determined by writing specific bit patterns to the control ports [11]. The control port numbers are shown on the right sides of Figures 3.8 and 3.9. One control port configures three I/O ports. For example, writing 9B (hex) to port number 11 sets ports 8, 9, and 10 for input only. Writing 8B (hex) to port 23 configures port 20 for output and ports 21 and 22 for inputs, as shown in the middle of Figure 3.9. Ports 4, 5, 6, and 7 are hardware configured for input only.

The data system status byte and the data system control byte go through ports 22 and 20, respectively, as shown in Figure 3.9. The information in the bit patterns of the control and status bytes was defined earlier in Figure 3.7 (sheet 4).

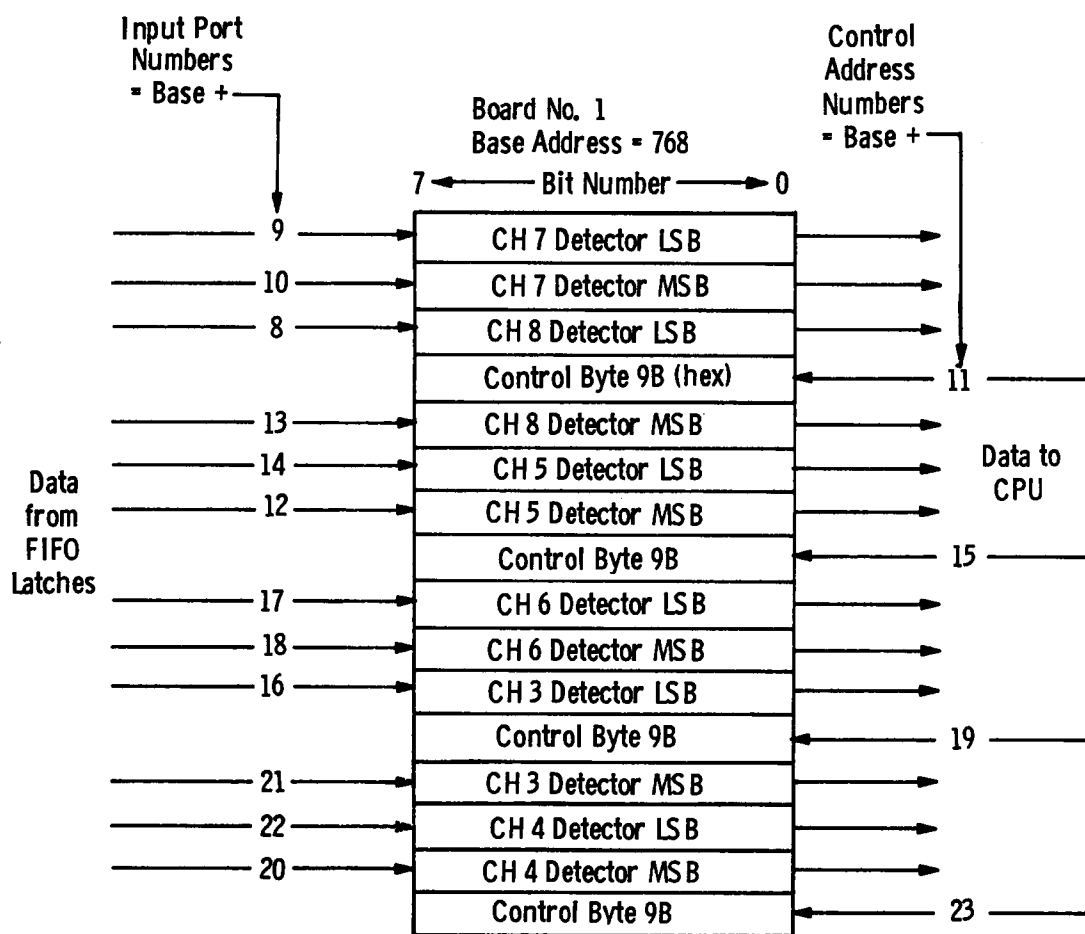


Figure 3.8. Operational Configuration of I/O Board No. 1.

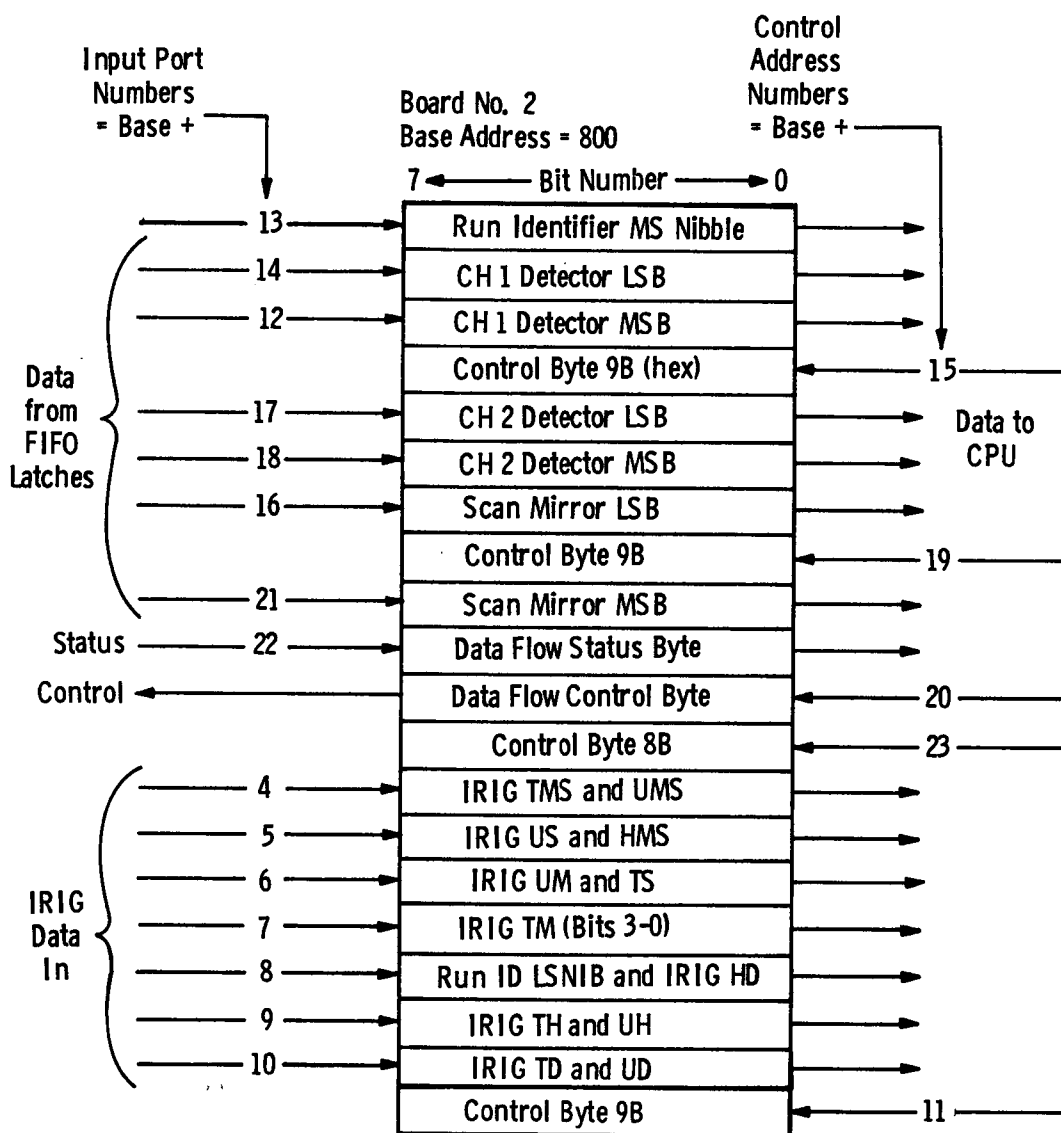


Figure 3.9. Operational Configuration of I/O Board No. 2.

Program RECORDAT configures the I/O ports at the start of its execution, and the configuration remains constant during the execution. The program reads the status byte port during execution and takes appropriate action. It also writes specific bit patterns to the control byte port. Table 3.2 defines the bit patterns by name and hexadecimal (OX) value. The first three patterns are used to initialize the data acquisition circuitry. CRDYLO starts the data run, READFIFO causes reads of 1 byte from all 18 FIFOs in the system, and IRIGRET is the IRIG handshake signal. A required bit pattern not shown in the table is EC (hex), which simultaneously acknowledges a FIFO full error ($\overline{\text{ERRACK}}$) and resets the FIFO pointers ($\overline{\text{RS}}$). The description of RECORDAT in the next chapter will show the utilization of the bits in the control and status bytes.

Table 3.2. Control Bit Patterns for Data Flow Control.

Bit Number and Control Function (X = Not Used)							Control Byte Name and Hex Value
7	6	5	4	3	2	1	
X	X	X	$\overline{\text{ERRACK}}$	$\overline{\text{IRIG}}$	$\overline{\text{READ}}$	$\overline{\text{RS}}$	0 $\overline{\text{CRDY}}$
1	1	1	1	1	1	1	INITHI = 0XFF
0	0	0	0	0	0	1	INITLO = 0X03
1	1	1	1	1	1	0	INITFIFO = 0XFD
1	1	1	1	1	1	1	CRDYLO = 0XFE
1	1	1	1	1	0	1	READFIFO = 0XFA
1	1	1	1	0	1	1	IRIGRET = 0XF6

CHAPTER IV

DATA ACQUISITION SOFTWARE DESCRIPTION

1. DESIGN CONCEPT OVERVIEW

A customized set of C-language programs was developed to acquire and reduce spectroradiometer data. The total software package contains approximately 2100 lines of code. The data acquisition programs will be described in this chapter, and the data reduction programs in Chapter V. Figure 4.1 shows the design concept for the data acquisition software. There are three main programs:

(1) RECORDAT.C; (2) RECOVDAT.C; and (3) COADD.C.

Program RECORDAT is the real-time data acquisition program. It reads the status byte, writes the control byte, and reads the spectroradiometer data at the input ports. It organizes the data and stores it in the file E:DATARUN.DAT. All spectral scans are stored sequentially with the hexadecimal longword FF0000FF used as the partition between scans. The program returns to operating system control after the data collection is finished. The number of scans to be read and stored in E:DATARUN.DAT must be specified by the operator when the program execution is started. When the program execution terminates, it creates and writes E:SCNRECRD.DAT, which is a binary record of the scan numbers and corresponding byte totals of all individual scans in E:DATARUN.DAT. The operator can run the program SHOWRECS to display this information.

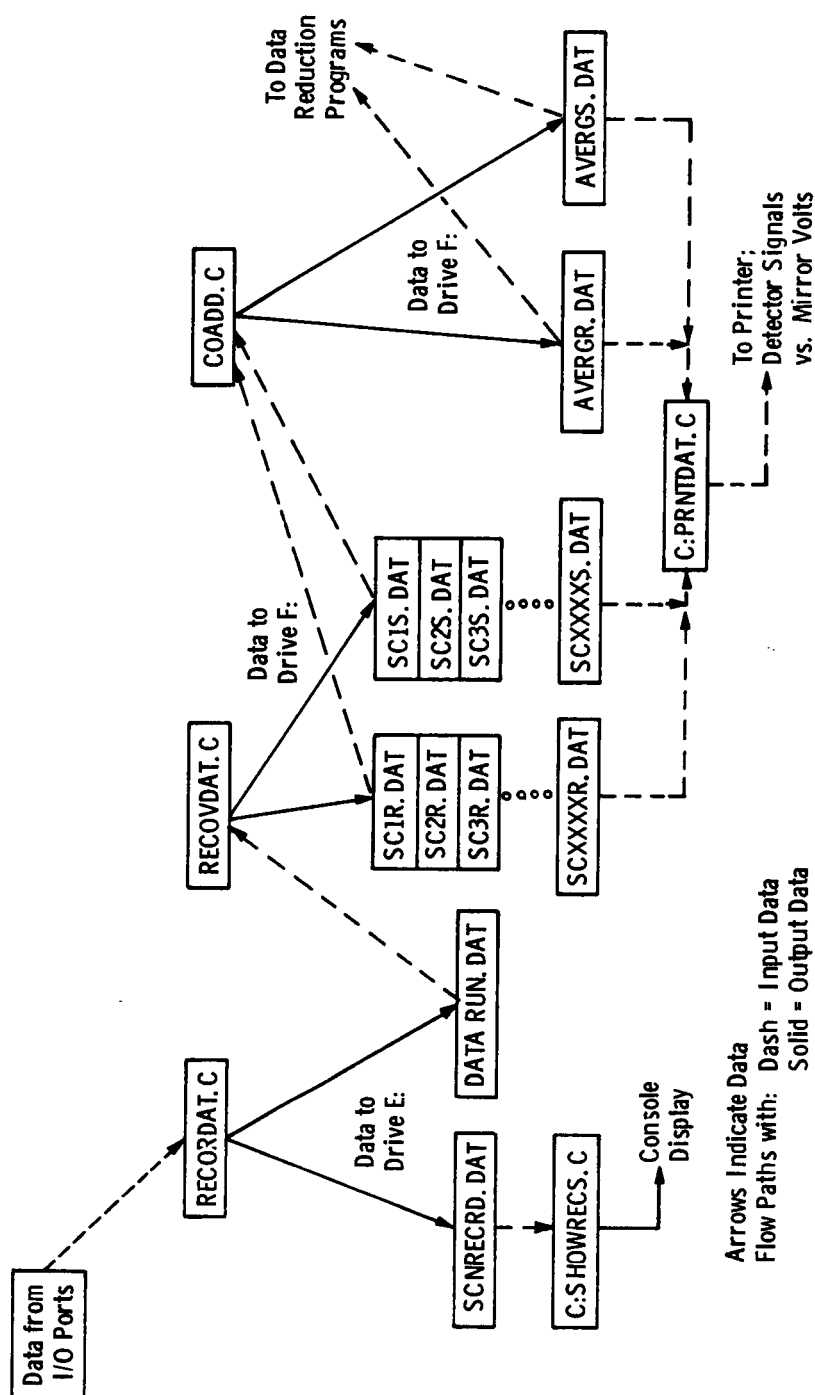


Figure 4.1. Data Acquisition Programs and Data Flow Paths.

Program RECOVDAT is run at a convenient time after the data run is finished. This program disassembles the DATARUN.DAT file, scan by scan, and writes the data in each scan to a separate DOS file as shown in Figure 4.1. For instance, if a data run has 500 spectral scans, then the value of XXXX in the figure will be 500. When the execution of RECOVDAT starts, the operator is requested to identify the data as sample (S) or reference (R) type, and the DOS filenames are formulated to include the S or R designation as shown in the figure. This nomenclature is used to distinguish sample and reference data required in the wavelength calibration phase, where the ratio of sample-to-reference data produces the spectral transmittance of a calibration sample (e.g., polystyrene). The sample data are taken with the sample located in the input radiation path. The reference data is simply a clear-beam scan. A data run is either sample or reference, and it cannot be both simultaneously. Therefore, in wavelength calibrations, the programs RECORDAT and RECOVDAT must be repeated to produce both type S and type R data scans. Spectroradiometer data acquisition requires running the programs only once and these data are designated either type R for radiometric calibration or type S for plume test data.

Program COADD.C calculates the ensemble average of a specified number of reference or sample scans and writes the result to AVERGR.DAT or AVERGS.DAT, as shown in Figure 4.1. The calculations consist of averaging the detector readings at corresponding data points (mirror positions) in a specified sequence of spectral scans.

The starting scan and the sequence length must be specified by the operator at the start of COADD execution.

Program COADD is used to improve the signal-to-noise ratio (i.e., data accuracy) of very noisy data. Fluctuations in rocket plume radiance, as well as the noise and vibrations encountered during the data run, can generate electrical noise on the detector signals and on the mirror position readings. The mirror may actually backtrack or jump ahead momentarily because of mechanical vibrations. Program COADD is designed to counter these effects. In addition to calculating detector signal averages, it checks the data quality by making certain that data points are in the proper sequence with respect to mirror position. If any data points are found to be out of order, the program "patches" the data. All patching is completed prior to calculating the average scan data. Program COADD is the most complicated program in this project.

Program PRNTDAT.C (Figure 4.1) is used to print the contents of a single data file specified by the operator. The data are printed on 14-inch printer paper. Programs PRNTDAT.C and SHOWRECS.C are relatively simple programs and will not be discussed in detail. Programs RECORDAT, RECOVDAT, and COADD will be described using high-level flow diagrams in the following three sections.

2. DESCRIPTION OF RECORDAT.C

The product of program RECORDAT is the raw data file DATARUN.DAT. The organization of this data file will be reviewed first, in order to help clarify the functions of RECORDAT. Figure 4.2

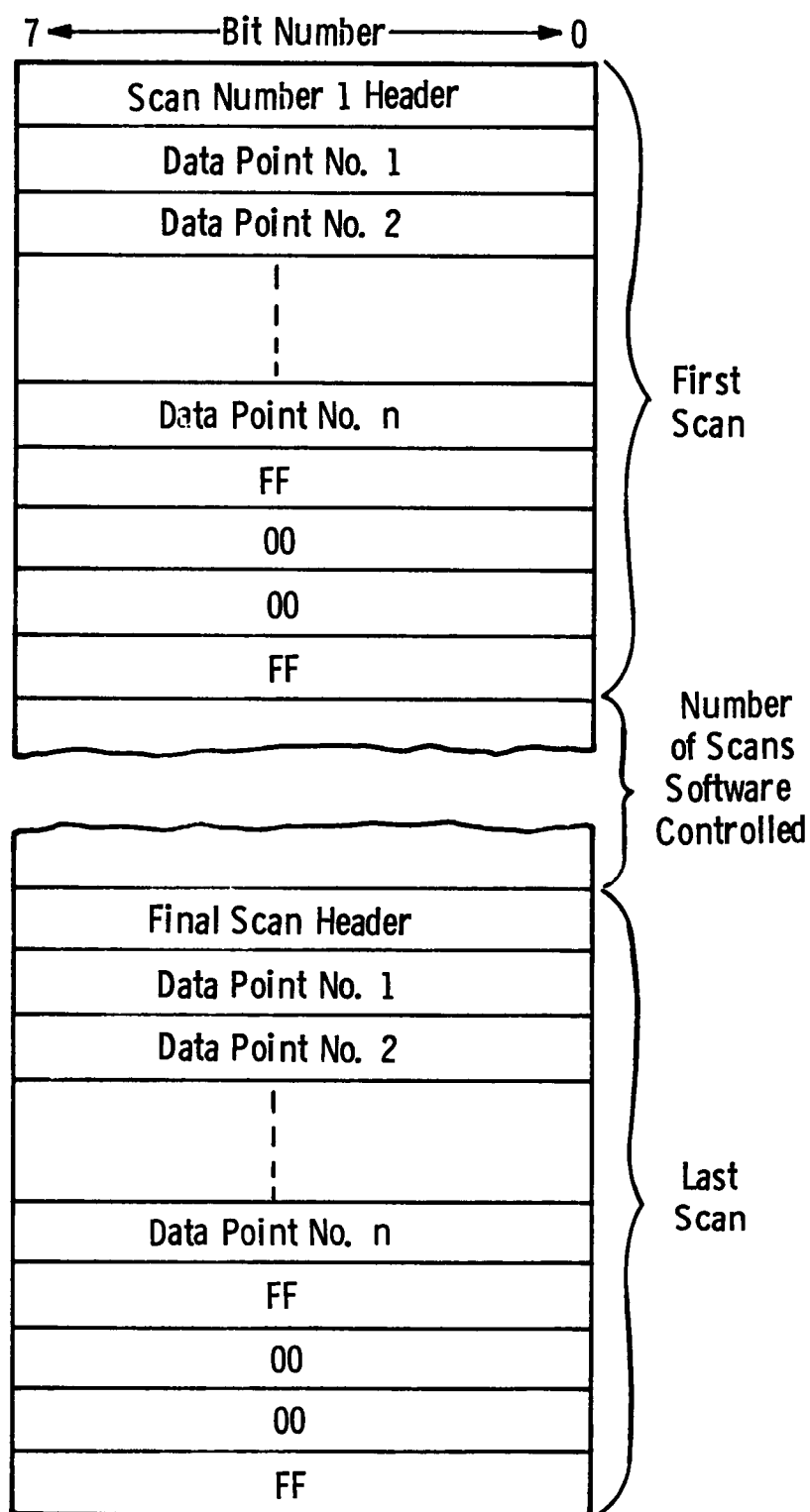


Figure 4.2. Data Structure of DATARUN.DAT.

shows the overall structure of the raw data file. The first entry in each scan is the header block which consists of the 10 bytes defined in Figure 4.3. The header contains the scan number in the first two bytes, the 8-bit run identifier in half of byte 3 and half of byte 4, and IRIG data from the low nibble of byte 4 through byte 10. The IRIG data are in BCD format, and one of the functions of RECOVDAT (see next section) is to convert it to character format by inserting the ASCII bits.

After the header block, there are n data points in a single spectral scan (Figure 4.2). Each individual scan is terminated by the hex longword FFO000FF. The number n is determined by the data interrupt wavelength interval setting as explained in Section 2 of Chapter III. The nominal values of n are 2000, 1000, 500, or 250 corresponding to the interval settings of LO, MLO, MHI, and HI, respectively. Each data point contains 18 bytes of data organized as shown in Figure 4.4. The mirror position word is first, followed by the eight detector signals. The information in each detector signal word was defined earlier in Figure 3.3.

Finally, the length of the raw data file (Figure 4.2) is determined by the required scan total as entered by the operator when RECORDAT is started. The program terminates data collection when the required number of scans has been written to DATARUN.DAT.

With all of the foregoing introductory material in mind, the program RECORDAT.C can now be discussed. Figure 4.5 presents a flow diagram of the entire program. While the figure is fully annotated and largely self-explanatory, a functional overview is useful to

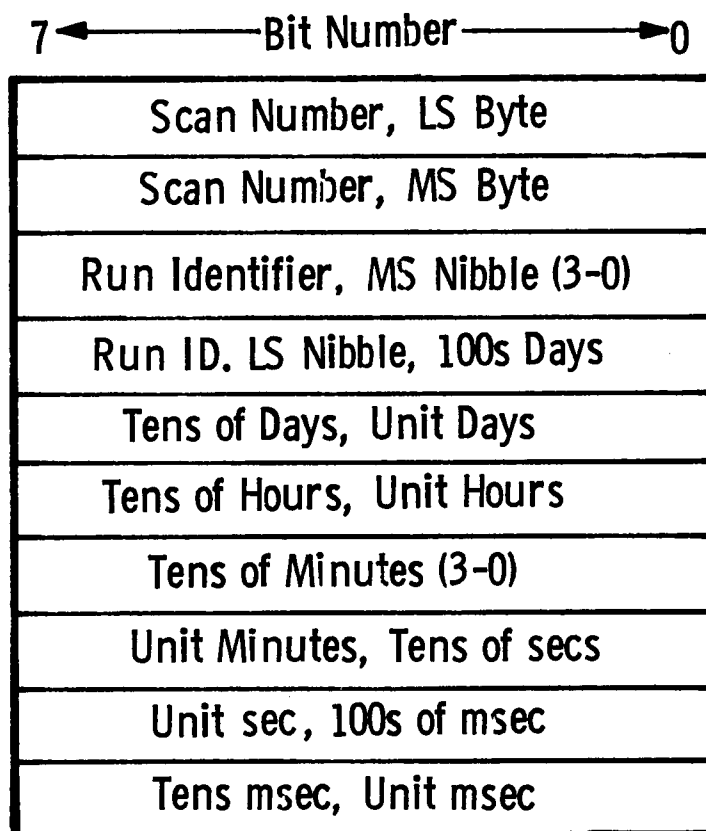


Figure 4.3. Header Data Format in DATARUN.DAT.

7 ← BIT NUMBER → 0

Scan Mirror Position, LSB		
Scan Mirror Position, MSB		
CH 1 Detector Signal, LSB		
CH 1		, MSB
CH 2		, LSB
CH 2		, MSB
CH 3		, LSB
CH 3		, MSB
CH 4		, LSB
CH 4		, MSB
CH 5		, LSB
CH 5		, MSB
CH 6		, LSB
CH 6		, MSB
CH 7		, LSB
CH 7		, MSB
CH 8		, LSB
CH 8	↓	↓, MSB

Figure 4.4. One Data Point in DATARUN.DAT.

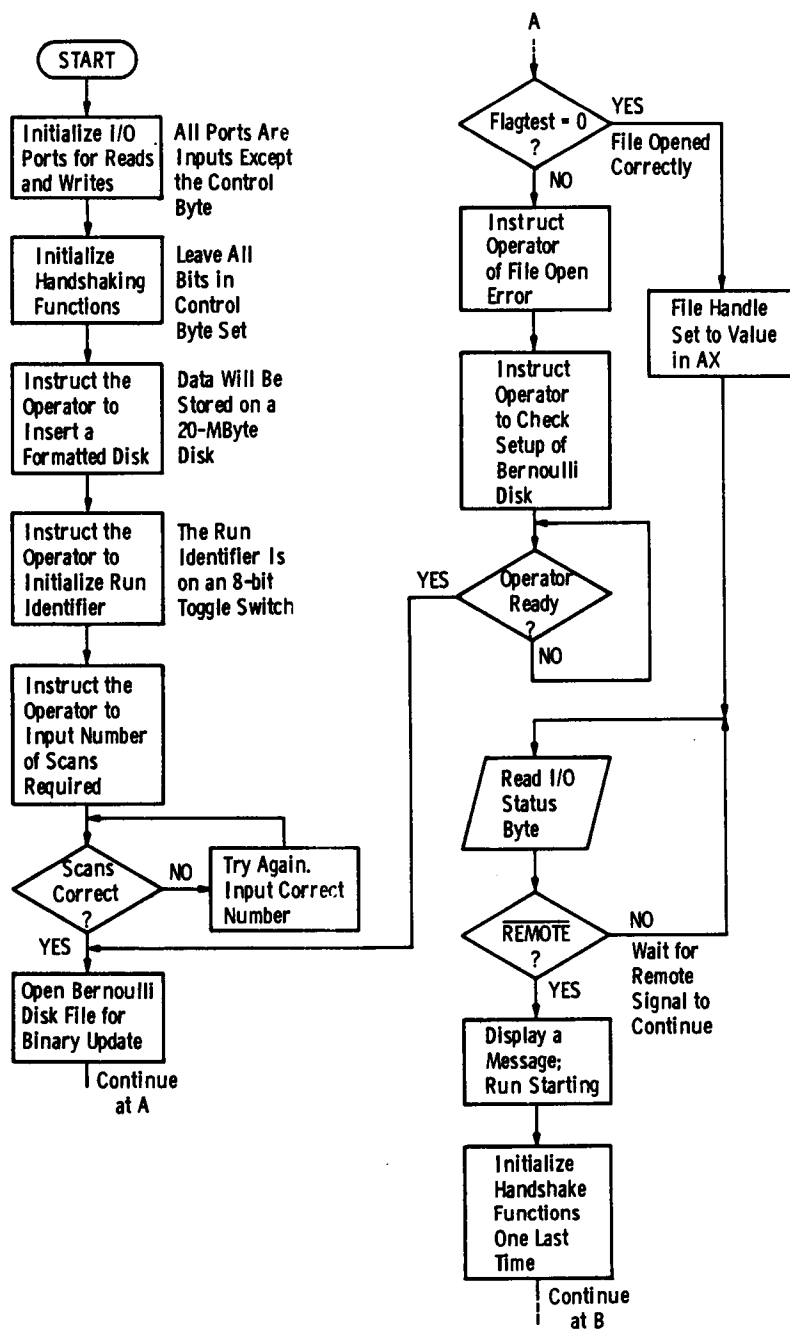


Figure 4.5. Flow Diagram of Program RECORDAT.C (Sheet 1 of 5).

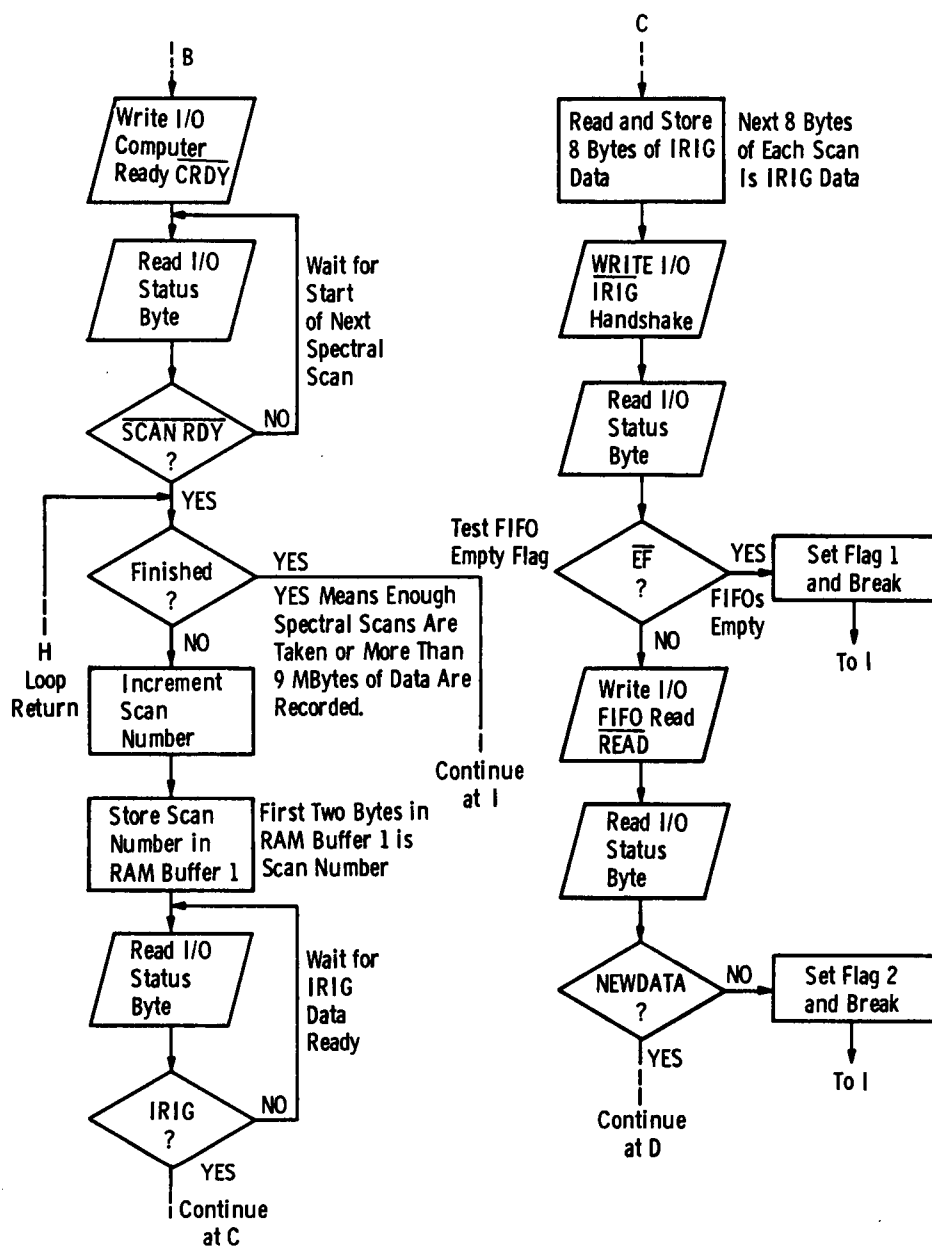


Figure 4.5 (Continued; Sheet 2 of 5)

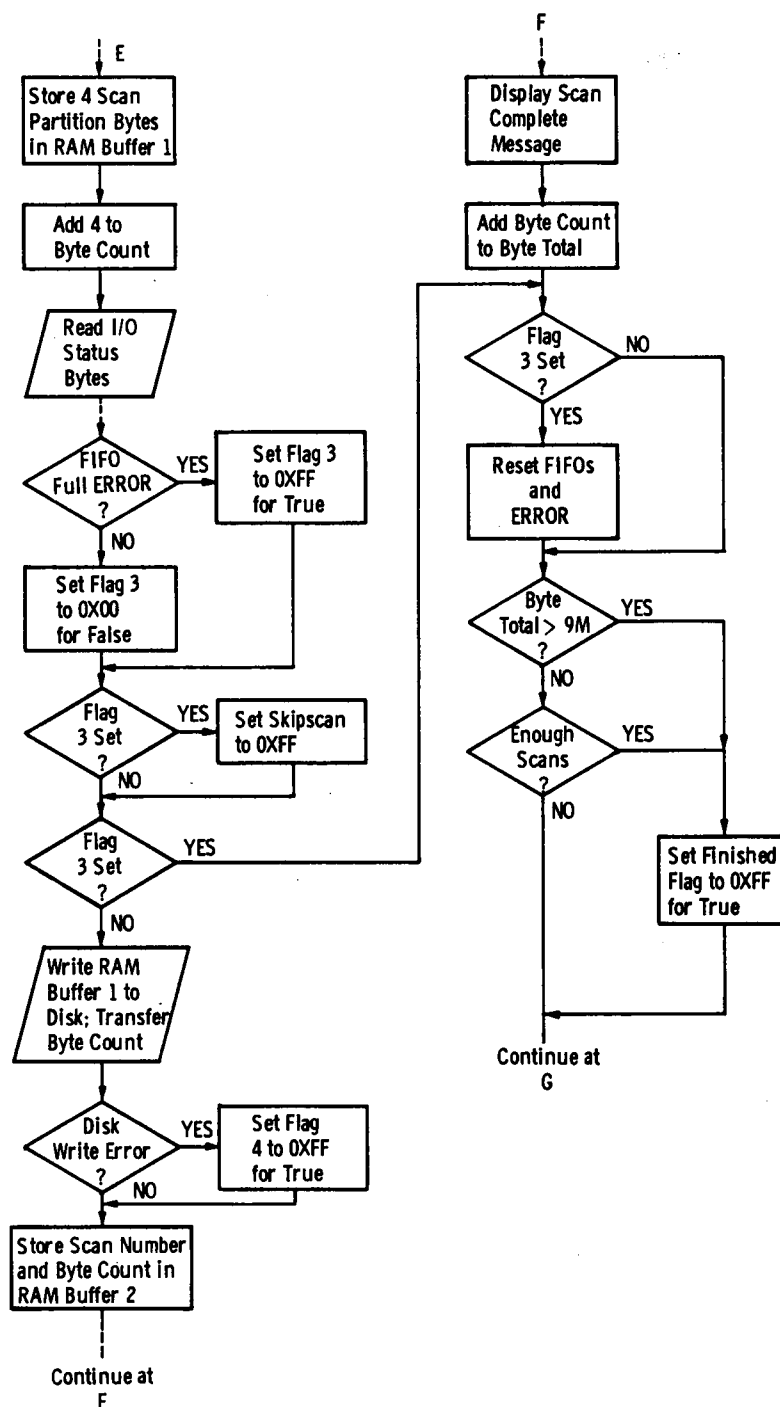


Figure 4.5 (Continued; Sheet 4 of 5)

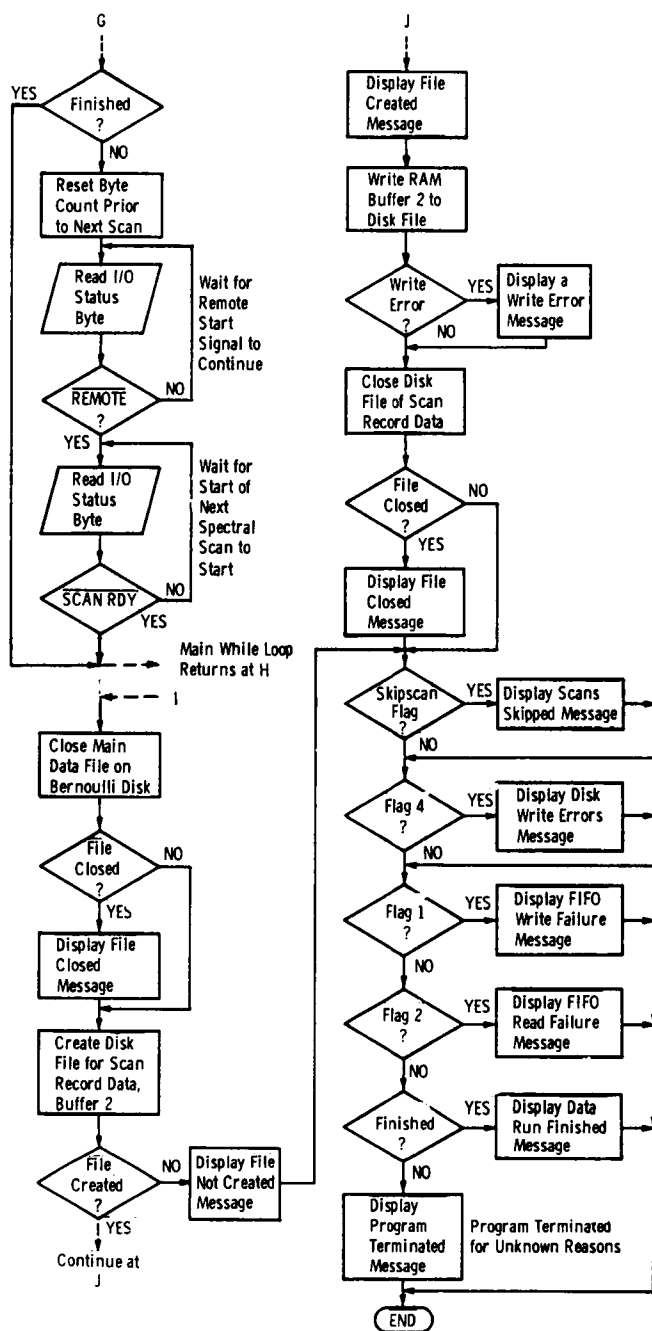


Figure 4.5 (Continued; Sheet 5 of 5)

understanding the program. On Figure 4.5 (sheet 1), the I/O ports are configured (as described in Chapter III), the required number of scans is stored, and the DATARUN.DAT file on drive E is opened. The program then waits for the remote control signal to continue. On Figure 4.5 (sheet 2), it issues the $\overline{\text{CRDY}}$ bit to activate the data acquisition circuitry and waits for the start of the next spectral scan. Data acquisition starts when the $\overline{\text{SCAN RDY}}$ status bit goes low. The process of acquiring a single spectral scan continues to the middle of the left column on sheet 5, where there is a loop return to point H on sheet 2. Scans are recorded until finished, after which there is a branch to point I, as shown on sheet 2. Point I is located on sheet 5.

The process of acquiring a single spectral scan starts (sheet 2) with incrementing the scan number and recording the IRIG data. Next, some error testing is done to make sure the FIFOs are writing and reading data properly. If not, the appropriate error flag is set and the run is terminated. If the data acquisition circuitry is working properly, the first data point (18 bytes, as shown in Figure 4.4) is recorded by the function "Readdata()" at the top left of sheet 3. At this point, no further error testing is done during the scan, and the program goes into the main data acquisition loop shown on sheet 3. It stays in this loop until the $\overline{\text{SCAN RDY}}$ bit goes high and the FIFOs are empty (i.e., the $\overline{\text{EF}}$ bit goes low).

When the loop terminates, the program continues at point E (sheet 4) where it writes the scan partition longword (FF0000FF) and tests the ERROR bit from the FIFO full error. If there was no error

in the scan (the usual condition), the data from the scan are written to the DATARUN.DAT file. If an error did occur, the data are not recorded, the Skipskan flag is set for an output cue, and the FIFO pointers are reset. The program then tests for run finished; and if not, it tests the REMOTE and SCAN RDY status bits to branch back to point H for the next scan. Since the data write time is much longer than the mirror flyback, SCAN RDY will be true as soon as it is tested. However, REMOTE will not be true if it was manually deactivated during the scan. Therefore, the data run can be temporarily discontinued by the manual remote control.

The program termination starts at point I on Figure 4.5, sheet 5, where DATARUN.DAT is closed, and SCNRECRD.DAT is created, written, and closed. Finally, appropriate diagnostic messages are displayed depending on how the program is terminated.

As mentioned earlier, most of the data in a single scan are acquired in the main loop shown on sheet 3 of Figure 4.5. The characters "empty" and "scanend" are 1-byte flags which are used to control the branching. They are set to hex 00 for false or hex FF for true. This procedure is used because the C language does not allow Boolean variable declarations. Figure 4.6 presents the code segment that implements the main data acquisition loop starting at the FIFOs Empty Test and ending with the branch to point E. The code segment shows how control loops are implemented in C. The nested while-not-empty loop in the code segment is illustrated by the right column of the flow diagram, Figure 4.5 (sheet 3).

```

while ((!empty) || (!scanend)) { /*read and store data */
    portno = base + 22;          /*status byte port number */
    byte_in = inportb(portno); /*read status */
    tst_byte = 0x04; /*bit pattern 00000100 for
                        FIFO empty test */
    if (!(tst_byte & byte_in)) { /*expression is true for
                                empty FIFOs */
        tst_byte = 0x02; /*bit pattern for SCAN RDY test*/
        if (tst_byte & byte_in) /*true for scan NOT ready */
            scanend = 0xFF; } /*set scanend for true */
    else { /* FIFOs are not empty; read and store data */
        empty = 0x00; /*reset empty to 0x00 for false */
        while (!empty) { /*take data until FIFOs are empty */
            portno = base + 20; /*control byte port number */
            outportb(portno, READFIFO); /*initiate FIFO read */
            outportb(portno, CRDYLO); /*set read line hi again */
            i = readdata(byte_buf, i); /*acquire 18 bytes */
            portno = base + 22; /*status byte port number */
            byte_in = inportb(portno); /*read status */
            tst_byte = 0x04; /*bit pattern 00000100 for
                                FIFO empty test */
            if (!(tst_byte & byte_in)) /*expression true for
                                    empty FIFOs */
                empty = 0xFF; /*set empty to 0xFF for true */
        } /*end of while NOT empty loop */
    } /*end of else statements */
} /*end of while NOT empty OR NOT scanend loop */
empty = 0x00; /*reset for next scan */
scanend = 0x00; /*ditto */

```

Figure 4.6. Code Segment for the Primary Data Acquisition Function.

The code segment also shows how RECORDAT uses the `inportb( )` and `outportb( )` functions to communicate with the data acquisition system via the status and control byte ports. Also, on each pass through the nested while-not-empty loop, there is a call of the `Readdata( )` function, consisting of multiple (18) calls of `inportb( )` to read the 18 bytes of new data. The variables passed to function `Readdata( )` include the address of the data buffer (`byte_buf`) and the number (`i`) of bytes that have been written in the data buffer. `Readdata( )` returns an updated value of `i`, which is 18 bytes more than the input value.

As mentioned earlier, the product of RECORDAT is the raw data file DATARUN.DAT. It can be several megabytes long, depending on the number of spectral scans. The next section shows how the raw data are recovered and organized into manageable data files.

3. DESCRIPTION OF RECOVDAT.C

The product of program RECOVDAT is a series of individual binary data files with one MS-DOS file per spectral scan, as shown in Figure 4.1 (page 56). The C86 function `fopen( )` is called to open each file. The contents of each file are organized as shown in Figure 4.7. Since the spectral data are converted to longwords, the amount of data per scan will be twice the size of the raw data record. The nominal amount of data per file will be 72 Kbytes, 36 Kbytes, 18 Kbytes, or 9 Kbytes depending on the data interrupt wavelength interval. The end of file marker is provided by the operating system when

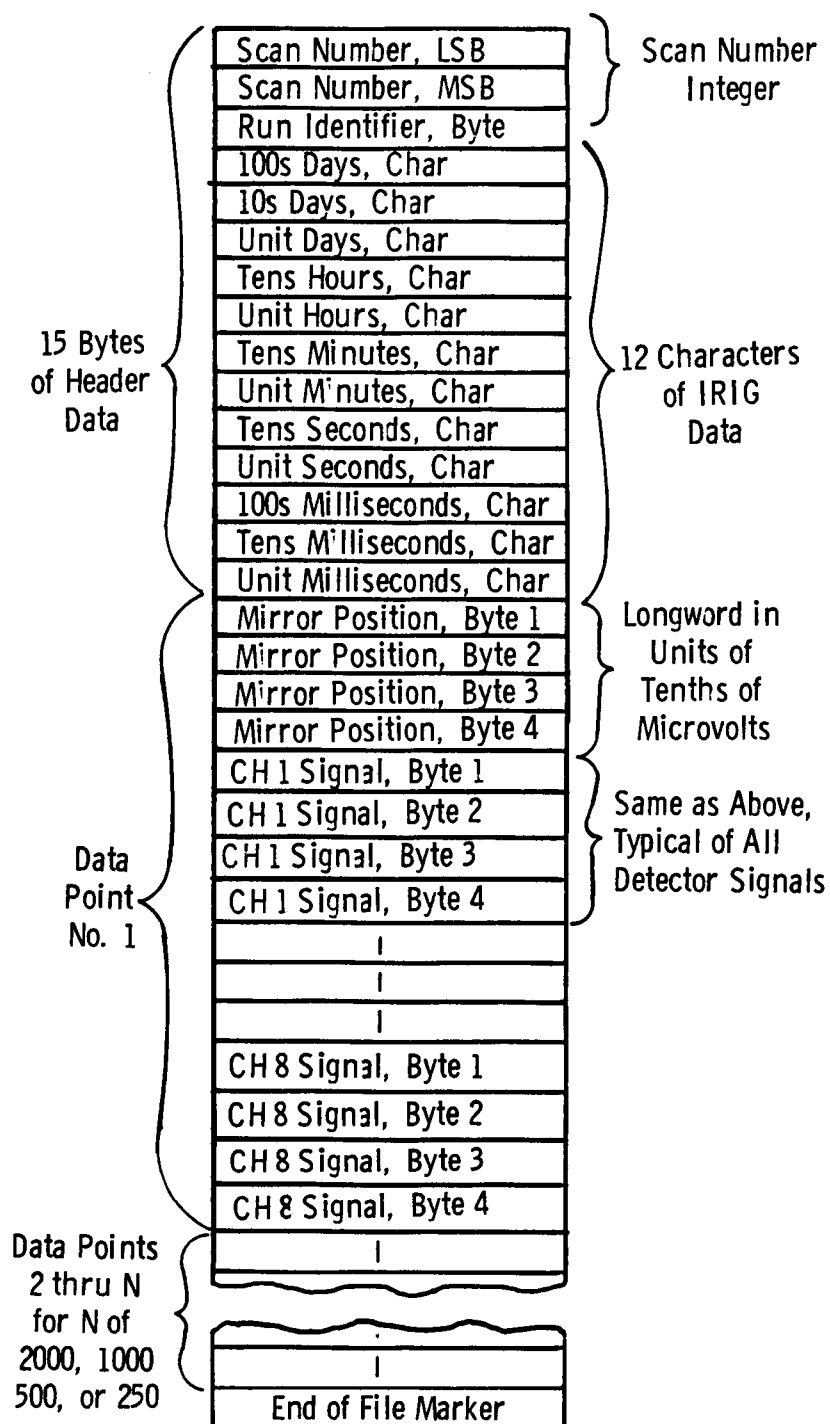


Figure 4.7. Data Content of All Files Produced by RECOVDAT.C.

the file is closed. The C86 function `fclose( )` is called to close the file.

The functions of RECOVDAT include converting the IRIG header data from BCD to character format and converting the mirror position and detector data to longword signed integers. The conversion of the detector data consists of multiplying each 12-bit signal (Figure 3.3 (page 30)) by 48828 (the value of 1 ADC bit in units of tenths of a microvolt), and dividing the product by the amplifier gain. Converting the mirror position and detector data to longword integers allows integer arithmetic to be used in subsequent data processing routines. Consequently, roundoff errors encountered in floating arithmetic are avoided. The range of signed longword integers ($+2.10^9$ to -2.10^9) is adequate for the spectroradiometer data.

The flow diagram of RECOVDAT is shown in Figure 4.8. The diagram is reasonably self-explanatory. At the start of the program, the operator is requested to enter the number of scans to be recovered from DATARUN.DAT and to indicate if the data type is sample (S) or reference (R). The file names are formulated to show the data type.

After RECOVDAT runs, it returns control to the microcomputer operating system. The operator then proceeds to run COADD.C.

4. DESCRIPTION OF COADD.C

Program COADD.C calculates the ensemble average of a specified sequence of the raw data scans, as indicated in Figure 4.1 (page 56). The general procedure consists of averaging the dependent variable data at corresponding values of the independent variable, which is

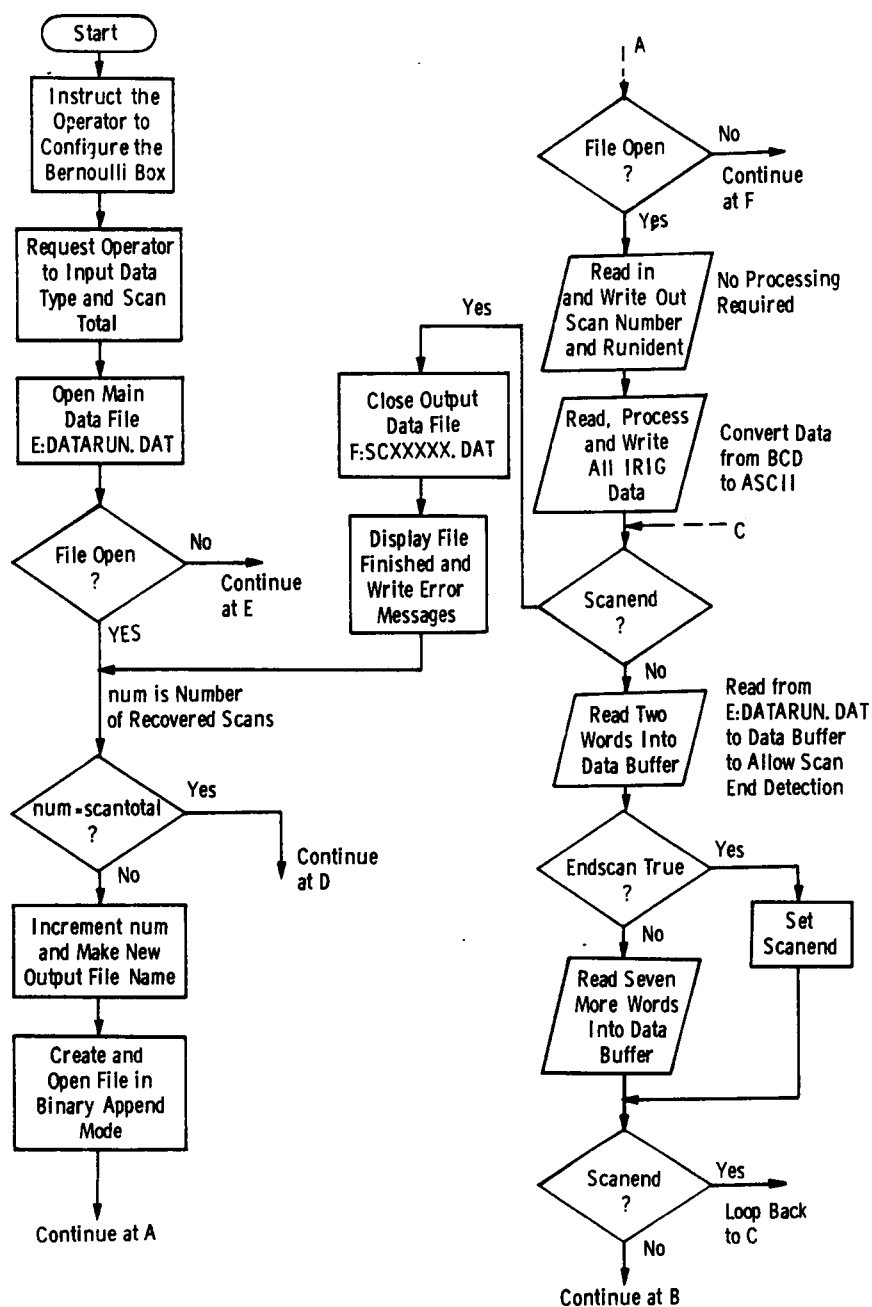


Figure 4.8. Flow Diagram of Program RECOVDAT.C (Sheet 1 of 2).

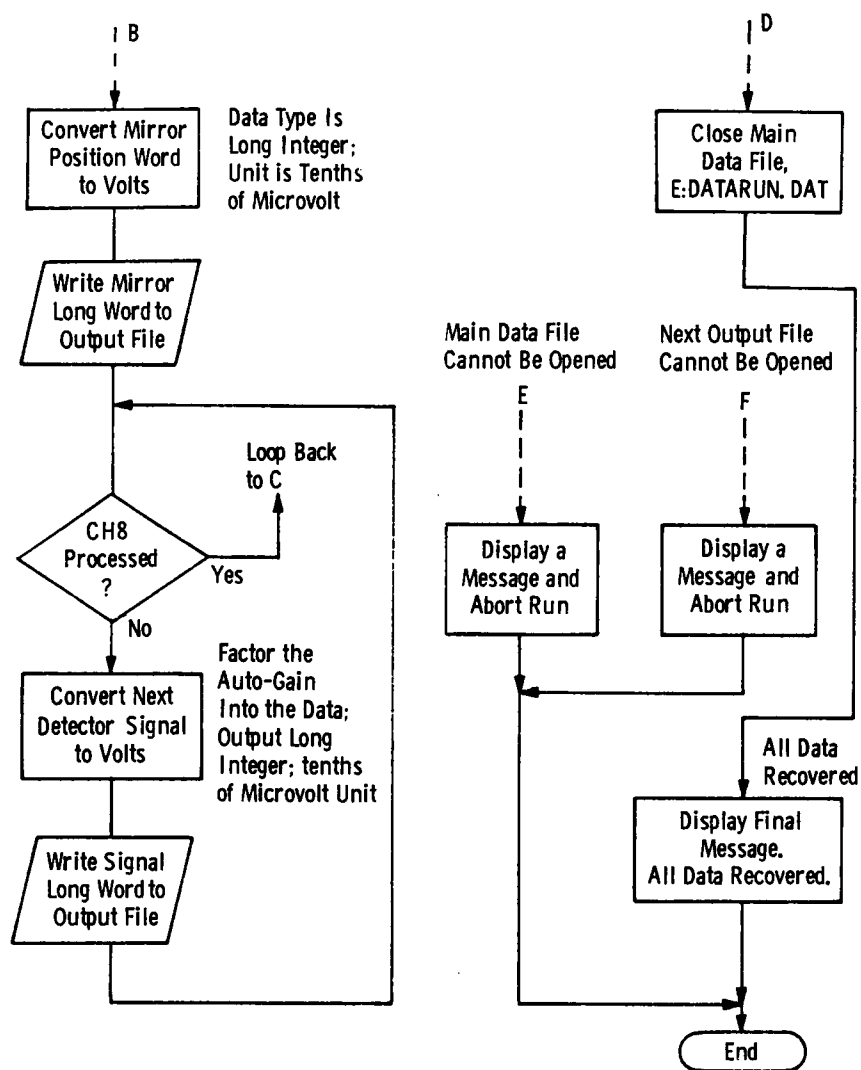


Figure 4.8 (Continued; Sheet 2 of 2)

usually time [13]. In this work, the independent variable is the mirror position (i.e., wavelength) and the detector signals are averaged at corresponding data points in a sequence of scans. This procedure is the main function of program COADD.C. The program reads a specified number of one or more raw data scans having the data structure shown in Figure 4.7. It then calculates the averages of the detector data and outputs an average scan as shown in Figure 4.1 (page 56). The structure of AVERGR.DAT or AVERGS.DAT is the same as the input data (Figure 4.7), with the header data being that of the first input scan. The averages are calculated at the mirror positions indicated earlier in Table 3.1 (page 38).

The program checks the data quality of each scan; and, if necessary, it does some "data patching" prior to calculating the ensemble average. Specifically, the program tests for the usual sequence of data points as defined by the mirror position values in Table 3.1. If any data points are repeated, they are discarded; and if any data points are missing, they are patched with extrapolated data. After the patching of each set is finished, the set is then added to the ensemble sum. This addition will be referred to as co-adding in the following discussions. The final operation is to divide the ensemble of summed data points by the number of input scans to acquire the averaged data. These various processes are accomplished by four passes through each scan's data set. The pseudo-code of Figure 4.9 outlines program COADD. Pass number five is the final process, in which the ensemble average values are calculated.

```

main()
{
    while all scans are not co-added {
        /*Process the next scan */
        store first scan header only;
        while not the end of current scan record {
            /*Do Pass No. 1 */
            read nine longwords into temporary buffer;
            if mirror voltage is on track or too high
                store mirror data in buffer 1 and
                store detector data in buffer 2;
        } /*End of Pass No. 1 */
        while not the end of buffer 1 {
            /*Do Pass No. 2 */
            transfer buffer 1 data to buffer 3 and
            simultaneously insert zeros into missing data;
            transfer buffer 2 data to buffer 4 and
            simultaneously insert zeros into missing data;
        } /*End of Pass No. 2 */
        while not the end of buffer 3 {
            /*Do Pass No. 3 */
            locate zeros in buffer 3 and patch mirror data;
            locate zeros in buffer 4 and patch detector data;
        } /*End of Pass No. 3 */
        while not the end of buffer 4 {
            /*Do Pass No. 4 */
            co-add buffer 4 data points to buffer 5 data points;
        } /*End of Pass No. 4 */
    } /*End of Loop; Return to process next scan record. */
    while not the end of buffer 5 {
        /*Do Pass No. 5 */
        divide each data point by the number of co-added scans;
    } /*End of Pass No. 5 */
    open output file;
    write buffer 3 and buffer 5 data;
    close output file;
} /*End of Main */

```

Figure 4.9. Pseudo Code for Program COADD.C.

The processes of pass number one are illustrated in Figure 4.10. Incoming data points are read into TEMPBUF. Each new data point consists of nine longwords with the structure of data point number one in Figure 4.7 (page 72). The mirror data are transferred to BUF1 and the detector data to BUF2. The mirror data are tested for repeated mirror position values; and if any such data points are found, they are discarded. At the end of pass one, the data in BUF1 meet the condition:

$$\text{BUF1}[i + 1] > \text{BUF1}[i], \quad 0 \leq i \leq m - 1. \quad (5.1)$$

When successive data points are in the proper sequence, the difference ($\text{BUF1}[i + 1] - \text{BUF1}[i]$) will be DELTAV, whose value depends on the data interrupt wavelength interval. In units of tenths of a microvolt, DELTAV will be 97,656 for LO; 195,312 for MLO; 390,624 for MHI; and 781,248 for HI. The mirror position of the first data point is a constant VREF = -93,847,416 in all cases.

The processes of pass two are illustrated by Figure 4.11. The program searches BUF1 for data points that are more than DELTAV apart, and any found are patched with zeros. An example is point ($i + 1$) in Figure 4.11. The patched BUF1 data are written into BUF3 as shown. The corresponding detector data in BUF2 are patched with zeros and written to BUF4 as indicated.

Pass three searches BUF3 for zeros and patches them with data. For instance, data point ($i + 1$) in Figure 4.11 is patched by:

$$\text{BUF3}[i + 1] = \text{BUF3}[i] + \text{DELTAV}. \quad (5.2)$$

The corresponding detector data in BUF4 are patched by extrapolations of the two previous data points in BUF4. Referring to buffer

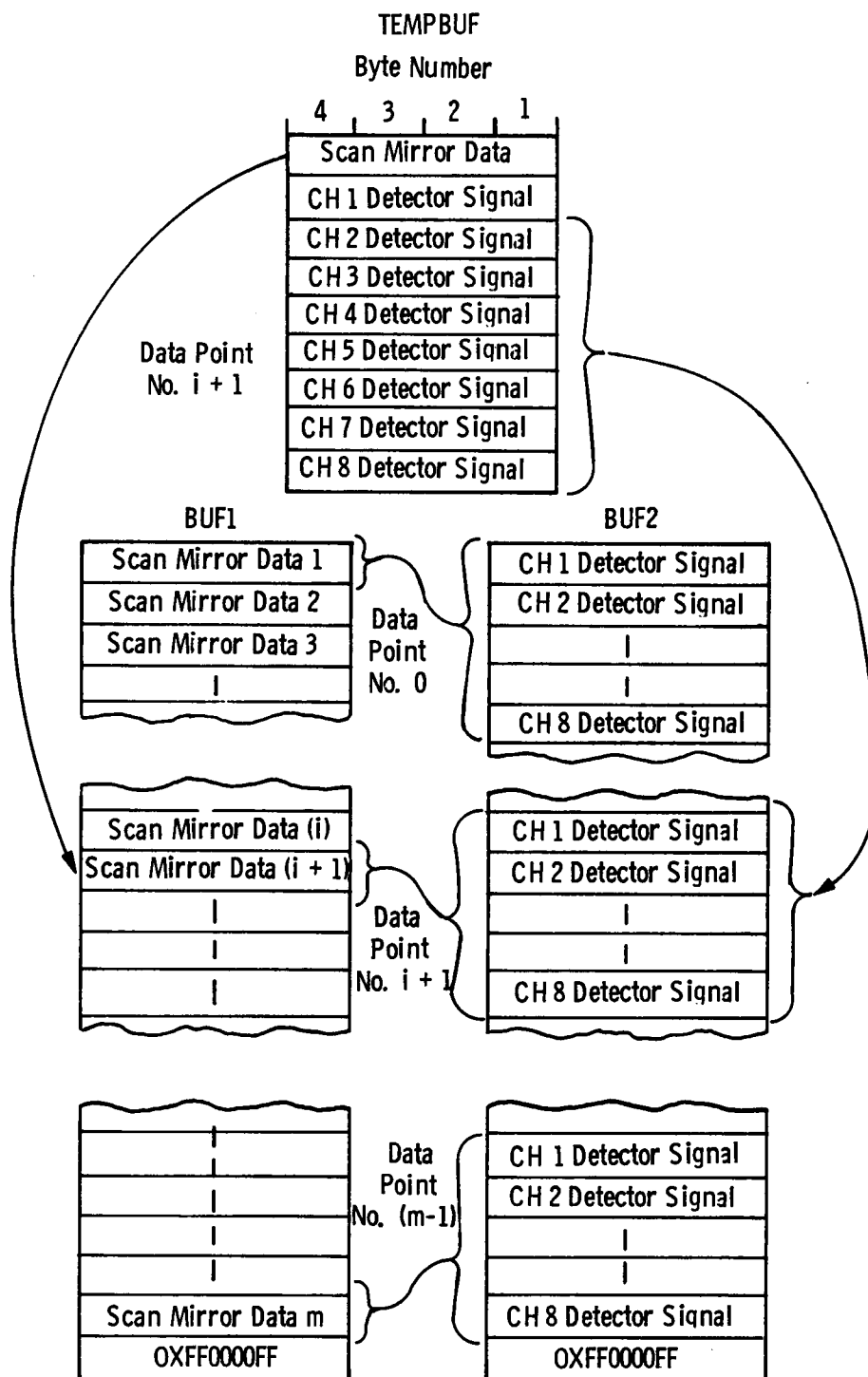


Figure 4.10. Processes of Pass No. 1.

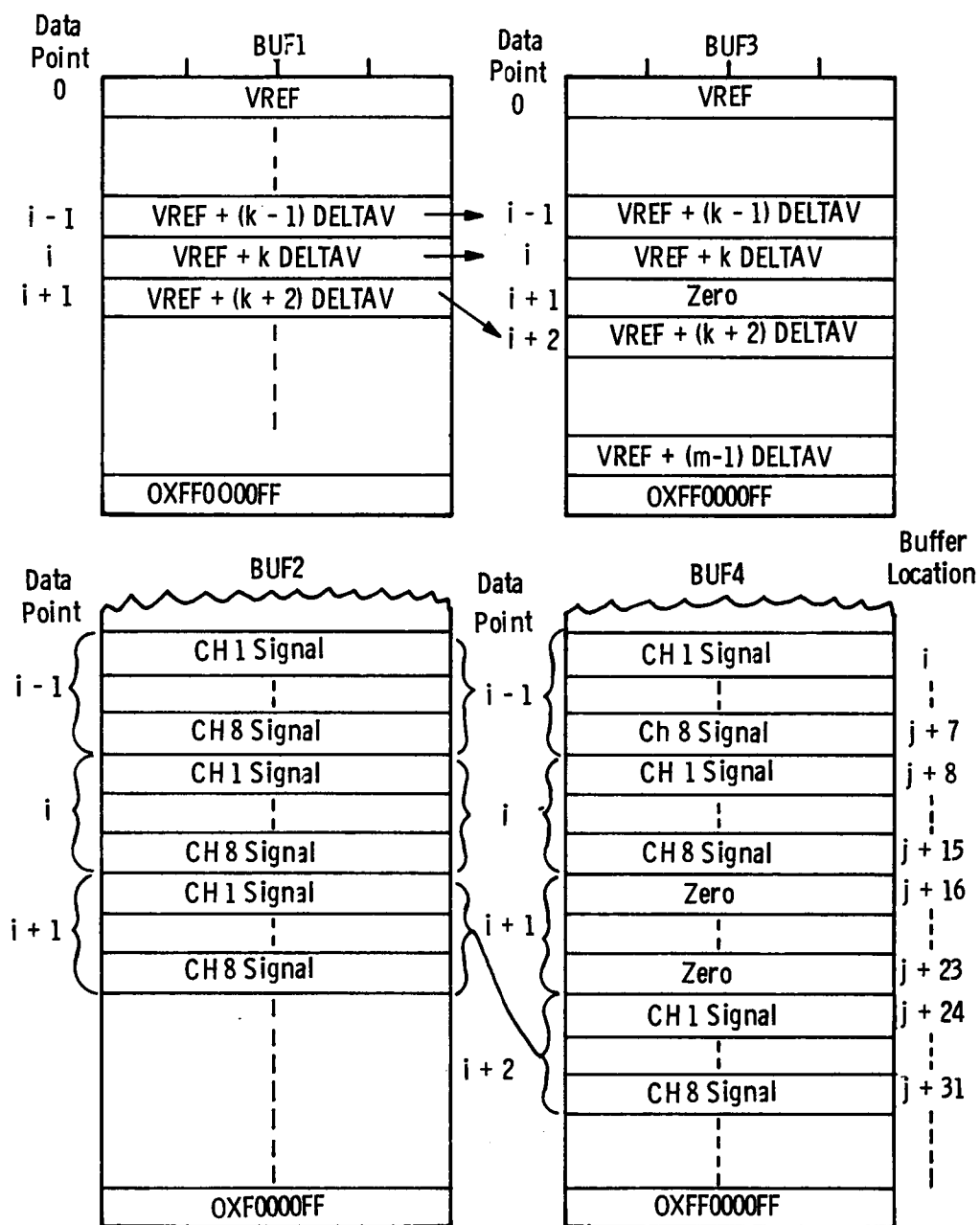


Figure 4.11. Processes of Pass No. 2.

location $j + 16$ through $j + 23$, the value of the data in location k in this range is calculated from

$$\text{BUF4}[k] = (\text{BUF4}[k - 8] - \text{BUF4}[k - 16]) + \text{BUF4}[k - 8]. \quad (5.3)$$

The code uses pointers to locate the referenced objects.

Pass four is the actual co-addition process in which the contents of BUF4 are added to BUF5, point by point:

$$\text{BUF5}[k] = \text{BUF5}[k] + \text{BUF4}[k], \quad 0 \leq k \leq (m - 1). \quad (5.4)$$

Pass five is also very simple, since it only divides each point in BUF5 by the number of co-added scans:

$$\text{BUF5}[k] = (\text{BUF5}[k]/N), \quad 0 \leq k \leq (m - 1). \quad (5.5)$$

In writing the output file AVERGS.DAT or AVERGR.DAT, the mirror data comes from BUF3 and the detector data from BUF5.

With this introduction to COADD, the program flow diagram can now be discussed. Figure 4.12 (sheets 1 through 5) shows the complete flow diagram. The program starts by querying the operator with regard to the number of scans to co-add, the starting scan number, the data interrupt interval, and the data type. The process of co-adding one scan starts at point I on sheet 1 of Figure 4.12 and continues to the end of sheet 4. Within this main loop, pass one is the procedure on sheet 2; pass two is completed by the operations in the left column of sheet 3; pass three starts in the right column of sheet 3 and ends in the right column of sheet 4; and finally, pass four is the operation at the lower right of sheet 4. After pass four is finished, the program loops back to point I to process the next scan.

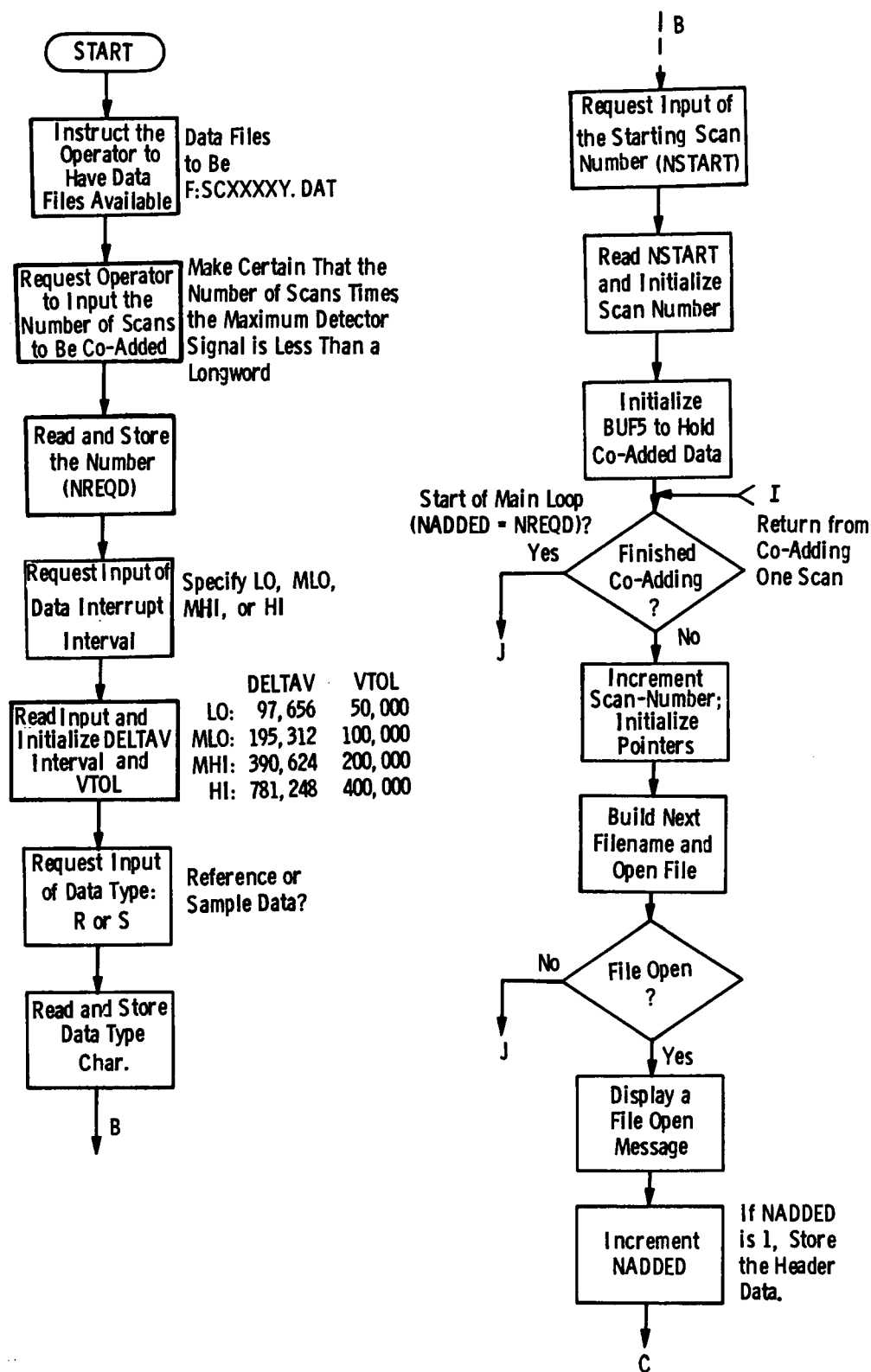


Figure 4.12. Flow Diagram for Program COADD.C (Sheet 1 of 5).

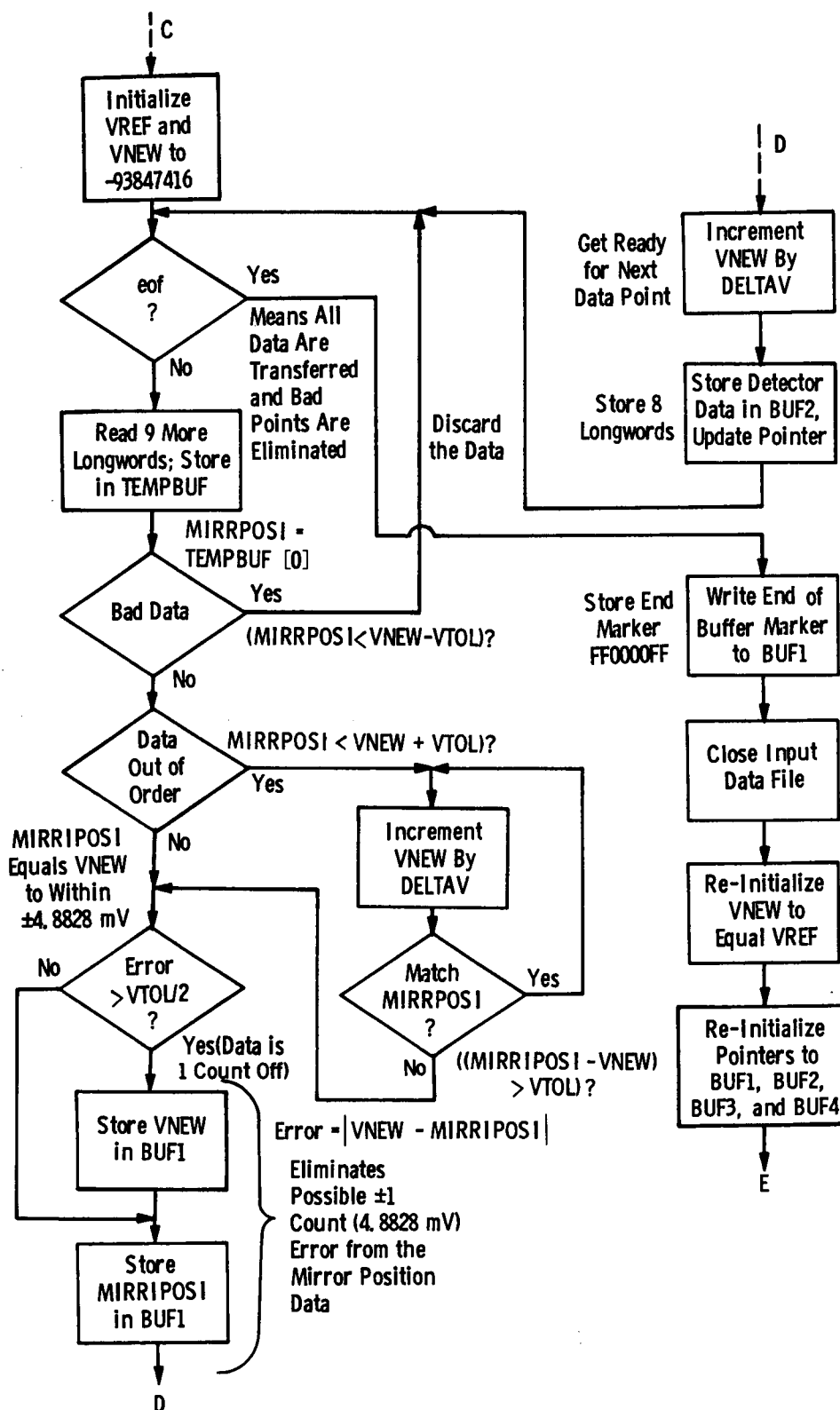


Figure 4.12 (Continued; Sheet 2 of 5)

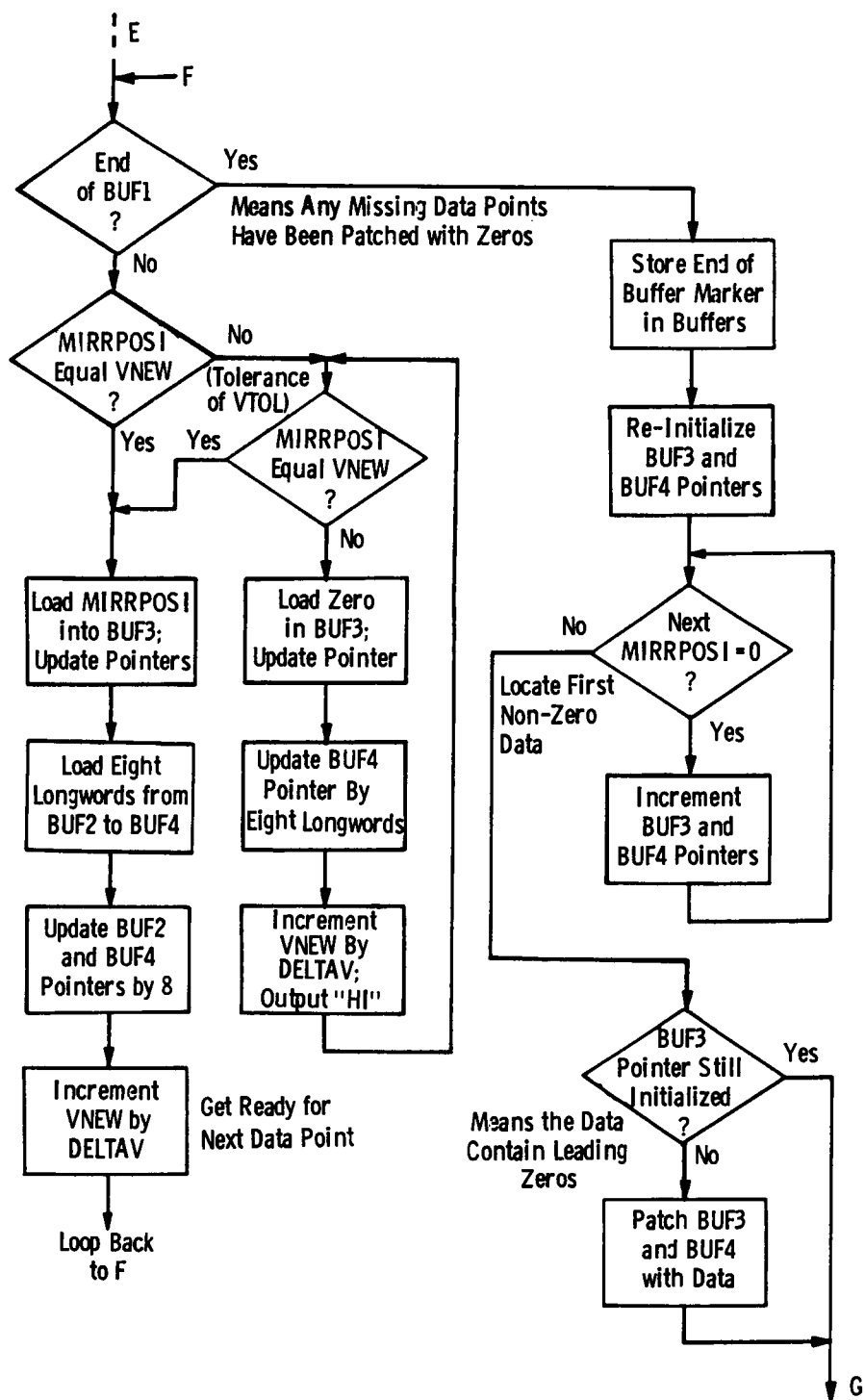


Figure 4.12 (Continued; Sheet 3 of 5)

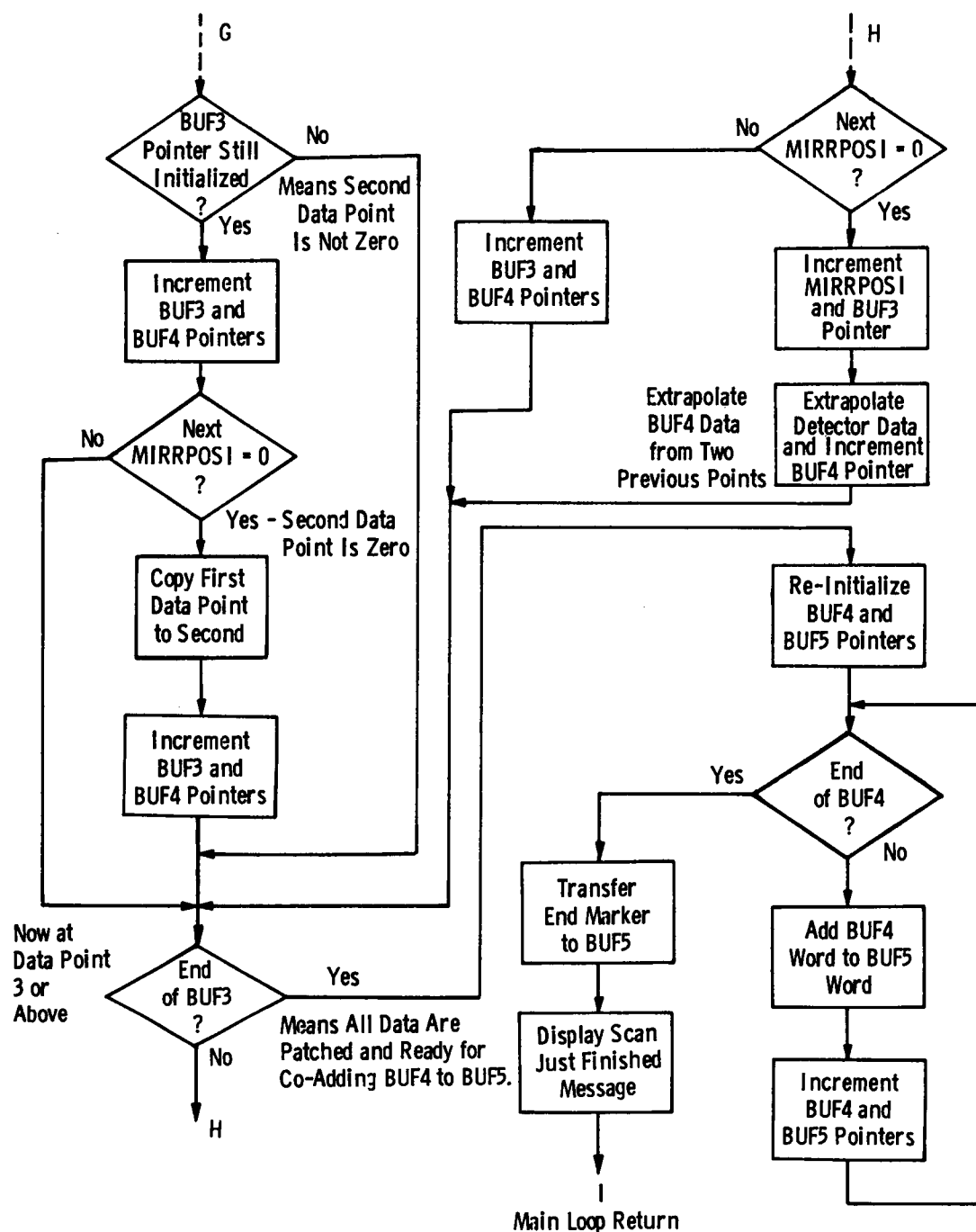


Figure 4.12 (Continued; Sheet 4 of 5)

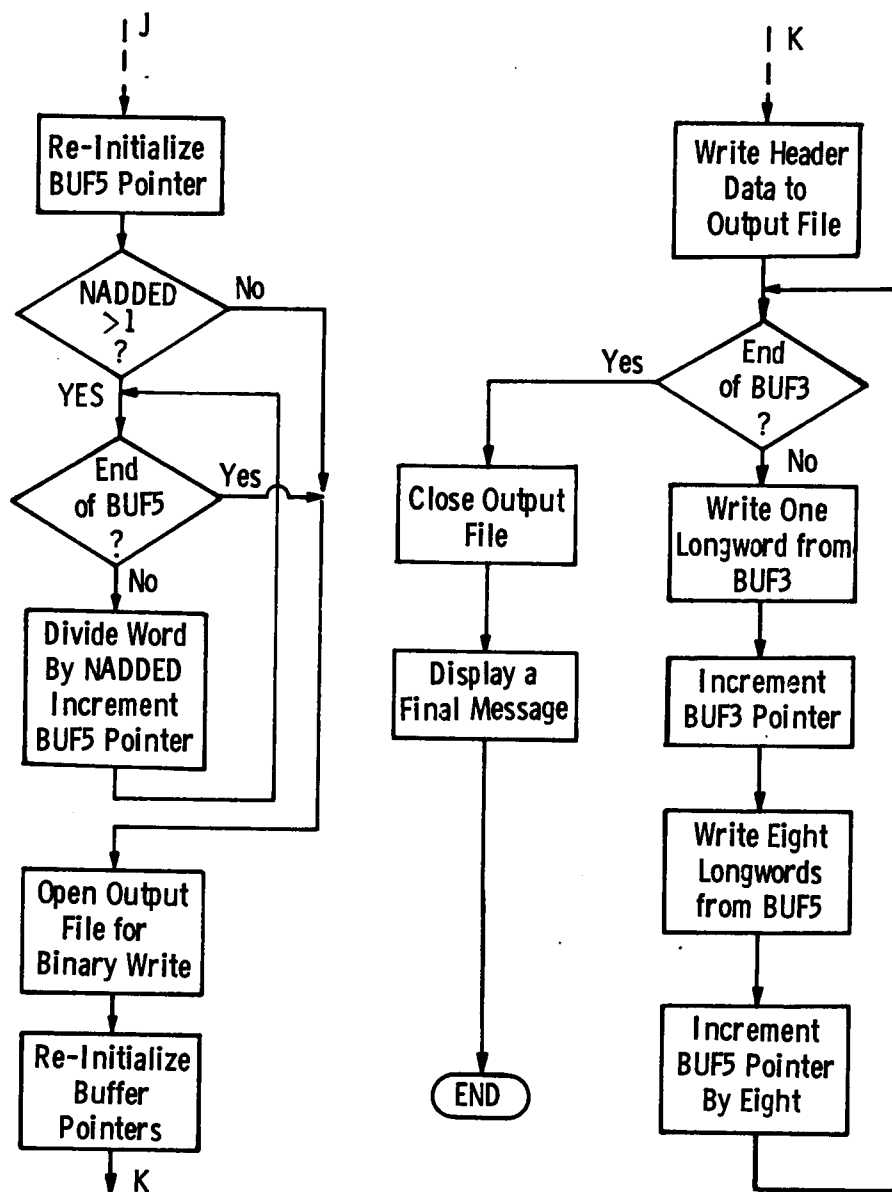


Figure 4.12 (Continued; Sheet 5 of 5)

Within pass three, most of the code is dedicated to testing for and patching leading zeros in the first two data points. This is necessary because the extrapolation algorithm (Equation (5.3)) cannot be used on the first two data points. The algorithm requires two preceding nonzero data points. Any leading zeros in BUF4 are patched with the first nonzero data in BUF4.

After all co-addition is finished, the program continues at point J on sheet 5 of Figure 4.12. Pass five is completed in the left column of sheet 5, and the program finishes by writing BUF3 and BUF5 to the output file. As stated earlier, the output file is AVERGS.DAT or AVERGR.DAT, depending on whether the data is type S or type R. This completes the data acquisition processes indicated in Figure 4.1 (page 56).

CHAPTER V

DATA REDUCTION SOFTWARE DESCRIPTION

The system for data reduction is illustrated in Figure 5.1. There are three programs: (1) XMISS.C; (2) RESPCALC.C; and (3) REDUCDAT.C. These programs are simpler than the data acquisition programs, and they will be described very generally using only code fragments.

Program XMISS is used to calculate the spectral transmission of transparent samples such as plexiglass® or polystyrene®. As will be explained shortly, transmittance data from several such samples are used to formulate the wavelength calibration of the spectroradiometer. Program XMISS reads data files F:AVERGR.DAT and F:AVERGS.DAT for reference and sample, respectively (see Figure 5.1). The program separates and stores the mirror data from both files in separate buffers. Similarly, it separates and stores all detector data from both files in separate buffers. The header data is discarded. The program must be compiled with the big memory switch, because it must allocate 26 buffers, each of which is 8 Kbytes long. The C-function malloc() is used to allocate memory space.

The program then proceeds to check the mirror data quality by making certain that the numbers in both sets of mirror data increase monotonically by 97656 (tenths of a microvolt) per data point from the reference value of -93847416 to the last data point. Program XMISS only works on data taken with the L0 data interrupt wavelength interval (i.e., maximum spectral resolution). If the mirror data

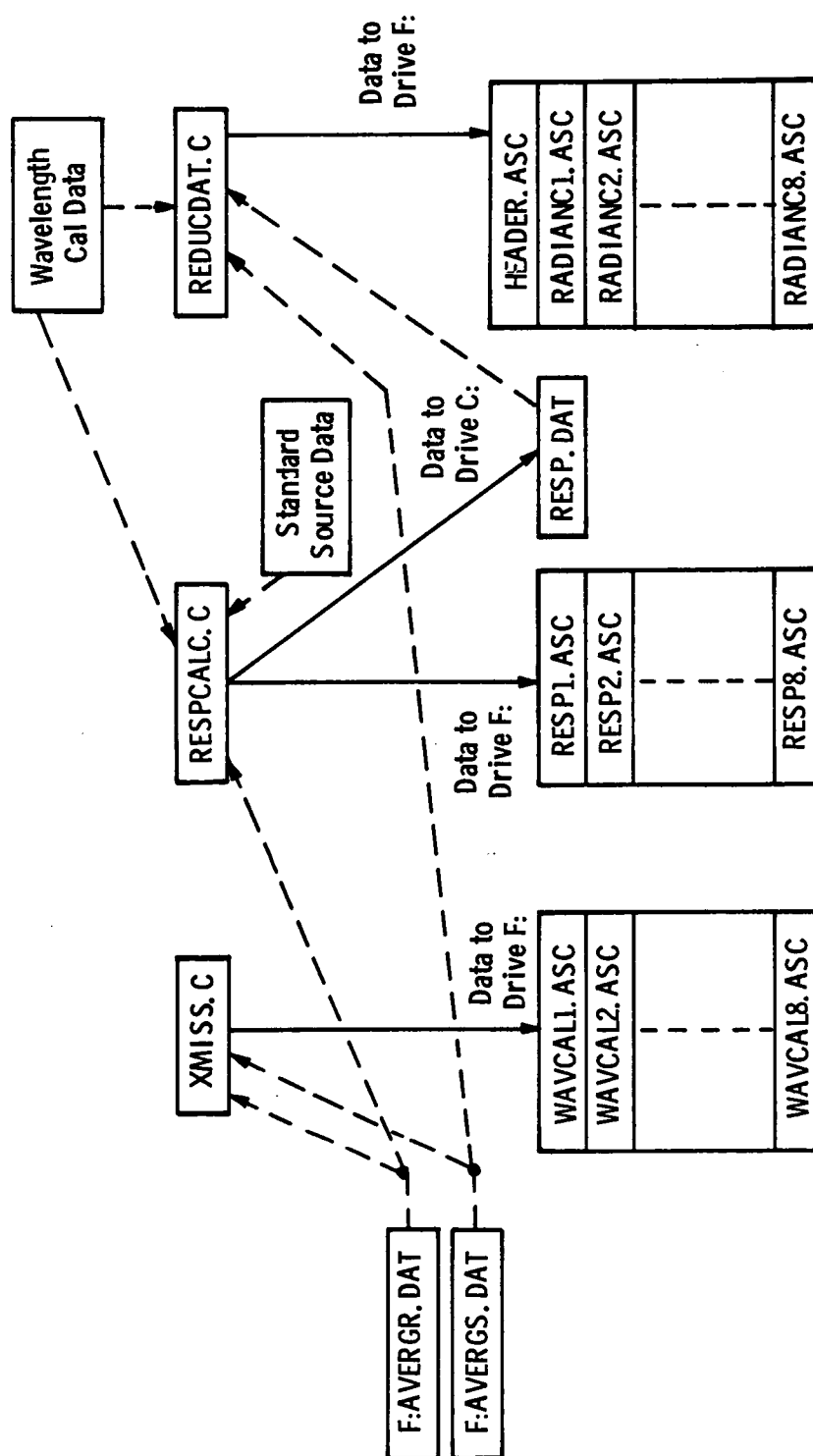


Figure 5.1. Data Reduction Programs and Data Flow Paths.

quality test fails, the program is aborted. If the test passes, the program neglects all further operations with the mirror data and works only with the data point number. The first data point (corresponding to Data Point No. 1 in Figure 4.7 (page 72)) is called number zero, and experimental results show the last data point to be number 1983. The wavelength calibration of the spectroradiometer is then formulated as wavelength versus data point number. This procedure is functionally equivalent to a "look-up table" for calibration data. Programs RESPCALC and REDUCDAT utilize the same data point reference when utilizing the wavelength calibration data.

Program XMISS then proceeds to calculate the percent of transmission versus data point number for each of the eight detector channels. The calculations consist of taking the ratio of corresponding detector data in the sample and reference data sets. The eight resulting data files are converted to ASCII format and stored in eight separate DOS files as indicated in Figure 5.1. The files are called WAVCALx.ASC, where x is the detector channel number. Figure 5.2a illustrates the code fragment used to implement the calculations for channel one, and Figure 5.2b shows the organization of each of the eight ASCII files.

The code fragment illustrates the utility of using auto-incremented array pointers in the C-language [14, 15]. Parameters `sptr1` and `rptr1` are pointers to the channel one sample and reference data, respectively. The data are of type signed long integer (32 bits) with units of tenths of a microvolt. The parameter "factor" is equal to 10,000,000.0 and the division operations convert the data to

```

while ((*sptr1 != 0xFF0000FF) && (*rptr1 != 0xFF0000FF)){
    ssig = ((*sptr1++)/factor)); /*convert sample data
                                to volts */
    rsig = ((*rptr1++)/factor)); /*convert reference data
                                to volts */
    *tptr1++ = 100.0*(ssig/rsig); /*ratio for CH 1 */
}/* end of while loop */

```

(a) Use of Auto-Incremented Pointers for Transmission Calculations.

```

0, %T ratio
1, %T ratio
  |      |
  |      |
1983, %T ratio

```

(b) Structure of all WAVCALx.ASC Files (x = 1 to 8).

Figure 5.2. Primary Functions of Program XMISS.C.

floating point voltages, with units of volts. The last line calculates the transmissions in percentage units (0 - 100%) and stores the result at the location of pointer `tptrl`. The pointers are post-incremented on each pass through the loop.

A similar code fragment is used for each of the other seven detector channels.

The resulting data are written one data point per line in the ASCII files, as shown in Figure 5.2b. A data point consists of the data point number and the corresponding percent transmission. The files may be printed or plotted like any ASCII file. Some plotted results will be presented in Chapter VI.

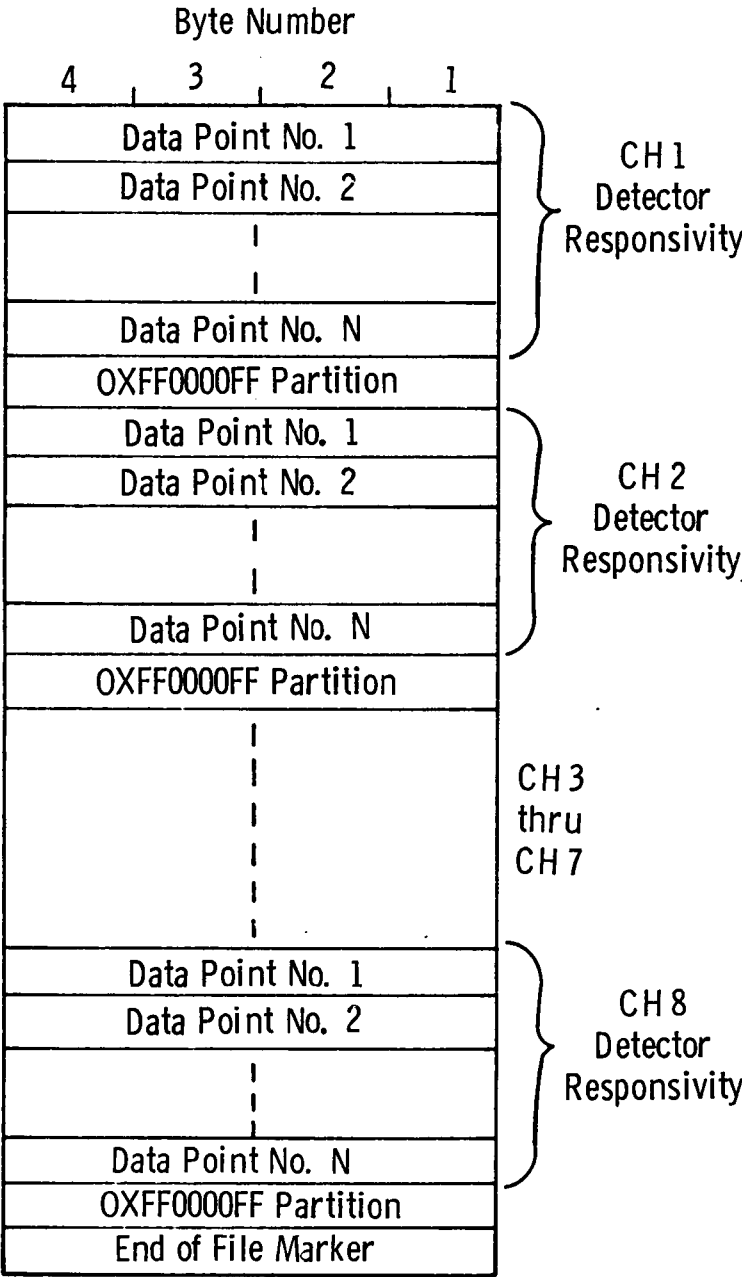
The wavelength calibration consists of measuring the spectral transmission of several samples which have absorption bands (i.e., transmission minima) at known wavelengths. From the accumulation of such data, a wavelength calibration equation of the form

$$\lambda(n) = \lambda_0 + n\Delta\lambda, \quad 0 \leq n \leq (N - 1) \quad (5.1)$$

is formulated for each detector channel. The parameter n is the data point number, and λ_0 is the wavelength at data point zero. The parameter $\Delta\lambda$ is the wavelength interval per data point, and N is the total number of data points in a LO interval scan. This procedure follows the concept presented in the introduction (see Equations (1.4) and (1.8)). Eight equations in the form of Equation (5.1) are formulated manually and are introduced into programs RESPCALC and REDUCDAT by editing initialized parameters. Each equation has unique values of λ_0 and $\Delta\lambda$, as will be shown in the Chapter VI.

Program RESPCALC calculates the "sensitivity profile" of the spectroradiometer. This is the radiometric calibration of the system. The actual parameter calculated is the inverse spectral responsivity with units of $\{(\text{watt/cm}^2 \cdot \text{micron} \cdot \text{Sr})/\text{tenth of a microvolt}\}$. This is the function $R(\lambda_i)$ discussed in the introduction (see Equation (1.5)). The function is calculated for each of the eight detector channels and the results are written to DOS ASCII files as indicated in Figure 5.1. Each ASCII file may be printed or plotted independently. The results are also stored in floating point format in a binary data file called RESP.DAT, which is organized as shown in Figure 5.3. This data file is subsequently used by REDUCDAT.C when reducing radiometric data to engineering units. The experimental results show that the number of data points per channel is $N = 1984$. Program RESPCALC works only for the LO data interrupt wavelength interval.

To experimentally run the radiometric calibration, the instrument is set up to "look" at a standard blackbody radiation source. The typical source temperature is 1000°C . Programs RECORDAT, RECOVDAT, and COADD are run to establish the calibration data file F: AVERGR.DAT as explained in Chapter IV. Program RESPCALC reads this calibration data and stores the mirror data and detector data in separate buffers. This requires nine buffers of 8 Kbytes each. The program then allocates eight buffers of 8 Kbytes each to hold the inverse responsivity data; additionally, eight buffers of 8 Kbytes each are allocated to hold the calculated wavelength values.



Note: Responsivity Data Are Type Float.

Figure 5.3. Structure of the Calibration Data in RESP.DAT.

The program then proceeds to check the mirror data quality by making certain that the mirror data increase monotonically by 97656 (tenths of a microvolt) per data point. If not, the program aborts; otherwise, the program proceeds to calculate wavelength based on a data point number ranging from 0 to 1983, as required by Equation (5.1). Figure 5.4a indicates how both the wavelength and inverse responsivity values are calculated and stored for the first detector channel. The data point number, j , is incremented on each pass through the loop and is used to calculate the wavelength $LAM1$. The parameters $LAM01$ and $DELAM1$ come from the wavelength calibration and correspond to λ_0 and $\Delta\lambda$ in Equation (5.1). The parameter $wavptr1$ is a pointer to the wavelength data buffer; parameter $rptr1$ is a pointer to the input calibration data for channel one. The function $calc()$ is called to perform the actual inverse responsivity calculation by evaluating Planck's function (Equation (1.1)) at wavelength $LAM1$ and temperature $TEMPK$, and dividing the result by the calibration data value $rcal$.

After all inverse responsivity calculations are complete, the wavelength data buffers and corresponding inverse responsivity data buffers are written to ASCII files with the structure shown in Figure 5.4b. Some plotted results will be presented in Chapter VI.

Program REDUCDAT.C is the final program required for operation of the spectroradiometer, as indicated in Figure 5.1. It reads the calibration data from RESP.DAT and the experimental test data from F:AVERGS.DAT and produces header and radiance data files in ASCII form. The radiance data are generated by taking the product of

```

j=0; /* data point count */
while ((*rp1++ != 0xFF0000FF) /*test for end of mirror
                                data */
    LAM1 = LAM01 + j * DELAM1; /*calculate next wavelength
                                value */
    *wavp1++ = LAM1; /* store the value */
    rcal = *rp1++; /* read next cal data value */
    *resp1++ = calc(LAM1, TEMPK, rcal); /*call function and
                                          store return */
    j += 1; /* increment data point count */
} /*end of while loop */

```

- (a) Calculations of Wavelength and Corresponding Inverse Responsivity for CH 1.

$$\begin{array}{cc}
 \lambda(0), R(0) \\
 \lambda(1), R(1) \\
 \vdots \quad \quad \vdots \\
 \lambda(n), R(n)
 \end{array}$$

- (b) Structure of all RESPx.ASC Files (x = 1 to 8).

Figure 5.4. Primary Functions of Program RESPCALC.C.

inverse responsivity data and the detector signal data at corresponding data points as discussed in introduction (see Equation (1.5)).

The program starts by reading F:AVERGS.DAT and storing the mirror data and detector data in separate data buffers. It also counts the total number of data points, where one data point consists of 36 bytes, as shown in Figure 4.7. The number of data points indicates the data interrupt wavelength interval of the experimental data. A parameter k is set equal to 1, 2, 4, or 8 depending on whether the interval was LO, MLO, MHI, or HI, respectively.

The program then proceeds to read and store the complete contents of RESP.DAT. It stores the calibration data for each detector channel in a separate data buffer. The program then calculates and writes wavelength and radiance values. The data are written out immediately with no intermediate buffering. Figure 5.5a shows the code fragment used to calculate the spectral radiance output for channel one. The figure also shows the structures of the data written in the header file and reduced radiance data files.

The parameter `sptr0` is the pointer to the mirror data buffer; `sptr1` is the pointer to the experimental sample data for detector channel one; `resp1` is a pointer to the inverse responsivity calibration data for channel one; and `optr` is a file pointer to the output ASCII file for channel one. In calculating the wavelength LAM , the data point number is evaluated first by taking the difference between the mirror position and $VREF$ (-93847416) and then dividing by $DELTA V$ (97656). This number will go as 0, 1, 2, ---, or 0, 2, 4, ---, or 0, 4, 8, ---, or 0, 8, 16, ---, depending on the data interrupt

```

while (*sptr0 != 0xFF0000FF) { /*test for end of mirror
                                data */
    LAM = LAM01 + (((*sptr0++) - VREF)/DELTAV) * DELAM1);
    RAD = (*sptr1++) * (*resp1); /*post-increment detector
                                data pointer */
    fprintf(optr, %6.4f, %11.3e\n", LAM,RAD/T1); /*write
                                                results */
    resp1 += k; /*index to next cal data point */
} /*end of while loop */

```

(a) Calculation of Spectral Radiance for CH 1.

Run Identification Byte: bbbbbbbb; Scan Number: ddd;
 IRIG Time at Data Point No. 1: ddD:ddh:ddm:dds:ddmS;

(b) Contents of the HEADER.ASC File.

$$\begin{array}{cc}
 \lambda(0), L(0) \\
 \lambda(1), L(1) \\
 \vdots \quad \quad \vdots \\
 \lambda(n), L(n)
 \end{array}$$

(c) Structure of all RADIANCx.ASC File (x = 1 to 8).

Figure 5.5. Primary Functions of REDUCDAT.C.

wavelength interval. After the data point number is evaluated, LAM is calculated using LAM01 and DELAM1 from the wavelength calibration, as explained earlier. At the end of each pass through the loop, the calibration data pointer, resp1, is incremented by the number k. Since sptr1 was post-incremented previously, the proper data value will be used when RAD is evaluated during the next pass through the loop. Again, this code fragment illustrates the convenience of using pointer arithmetic to access data structures.

The radiance value is corrected for a possible window loss by dividing the window transmittance T1 in the fprintf() statement. Window losses occur when measurements are taken through a test cell window. Since the transmittance is expressed as a decimal fraction, the division amounts to multiplying by a number greater than unity. The window transmittance is measured independently and values for T1 through T8 are input as initialized parameters, for channels 1 through 8. The values are taken as the window transmittance at the middle of each spectral span.

CHAPTER VI

EXPERIMENTAL RESULTS

The eight basic programs described in the two previous chapters were tested by going through the complete process of calibrating and applying the spectroradiometer. In this process, many software and hardware problems were discovered and eliminated. The essential experimental results will be reviewed in this chapter.

First, with regard to spectral scan rates, the fastest possible scan rate at the data interrupt wavelength interval of L0 was 2 seconds/scan. At faster scan rates, the disk write operation proved too slow to keep the FIFO buffers underfilled as required by the software design concept. At interrupt intervals of MLO, MHI, and HI, faster scan rates are possible. For this checkout, a procedure was adopted whereby all calibration data were run at the rate of 2 seconds/scan and spectroradiometer data were run at 1.5 seconds/scan.

Calibration data must be run at the L0 interval, but spectroradiometer data can be run at any of the four options. The latter feature is implemented in the design of REDUCDAT (see Figure 5.5a). The MLO interval was used in the radiance measurement demonstration of this checkout phase. In routine spectroradiometer applications, a scan rate of 1 second/scan and an interrupt interval of MLO may prove to be the optimum operation setup. These settings allow continuous spectroradiometer operation up to the limit of the storage disk capacity.

The software is designed to support the three distinct processes required to use the spectroradiometer: (1) wavelength calibrations; (2) inverse responsivity calibrations; and (3) spectroradiometric measurements of unknown sources. The programs, procedures, and experimental results of each of these three processes will be reviewed individually in the following discussion.

The programs that support wavelength calibrations are RECORDAT, RECOVDAT, COADD and XMISS. The first three must be executed two times for each calibration run. During the first execution, a wavelength calibration standard such as polystyrene is placed in the input light beam. The programs are run to produce the file F:AVERGS.DAT. A typical number of ten scans were co-added to produce the file. During the second execution, the calibration standard is removed and clear-beam spectral scans are taken. The same number of scans are co-added to produce F:AVERGR.DAT. Program XMISS is then executed. As described in Chapter V, transmission data are generated in ASCII format (see Figure 5.1) and can be printed or plotted. Figure 6.1 shows a plot of a typical result for channel 2, when using a plexiglass calibration sample. The transmission minimum at data point number 551 is known to be at 1.678 microns from an independent calibration of the plexiglass. Since the plotting routine can accommodate only 1000 data points, Figure 6.1 represents only approximately half of the data generated for channel 2. These are the data for data points 0 through 999.

Another example, for data points 1000 through 1983, is shown in Figure 6.2. In this case, the sample is polystyrene and the data

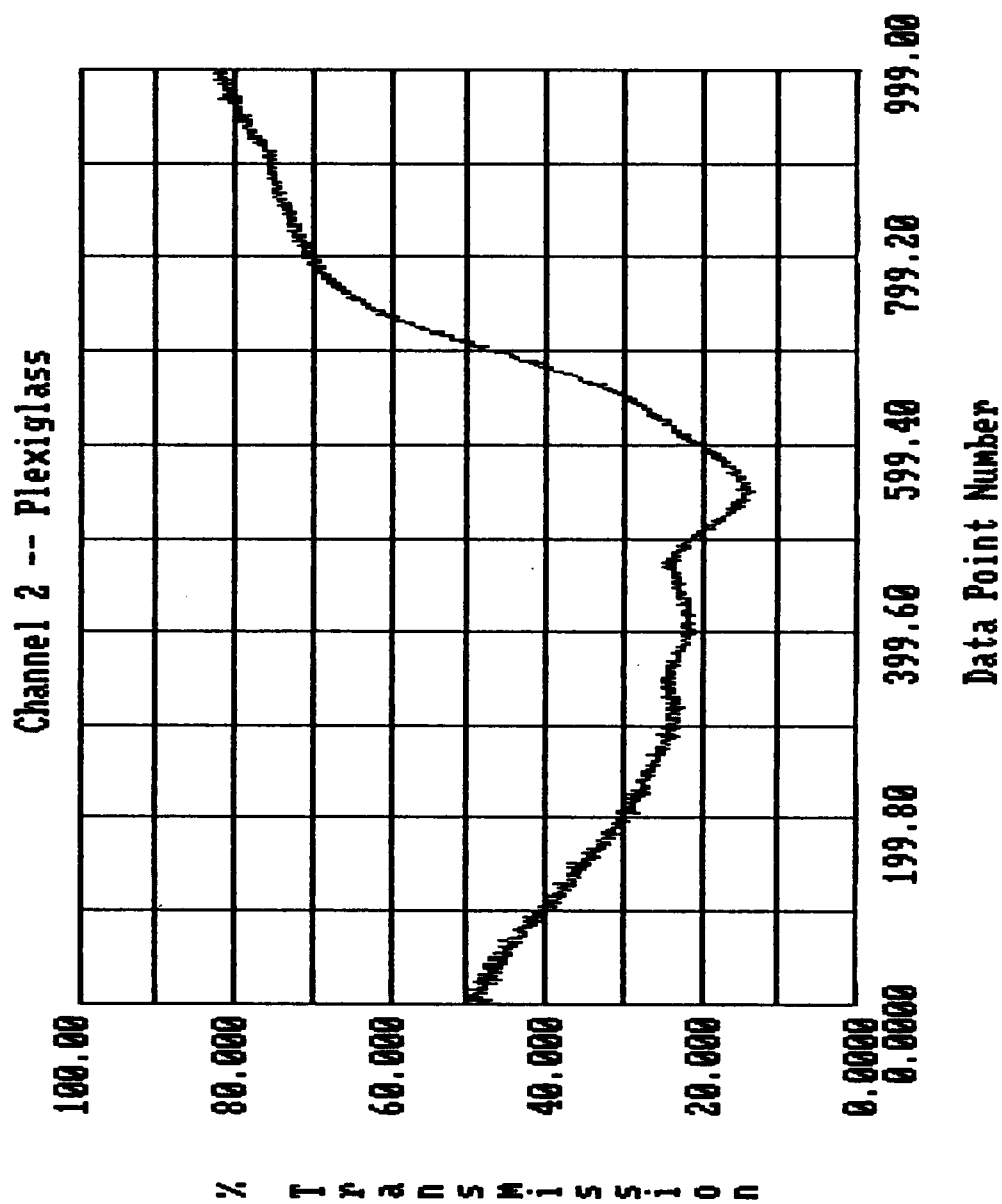


Figure 6.1. Transmission of Plexiglass in the Range of CH 2.

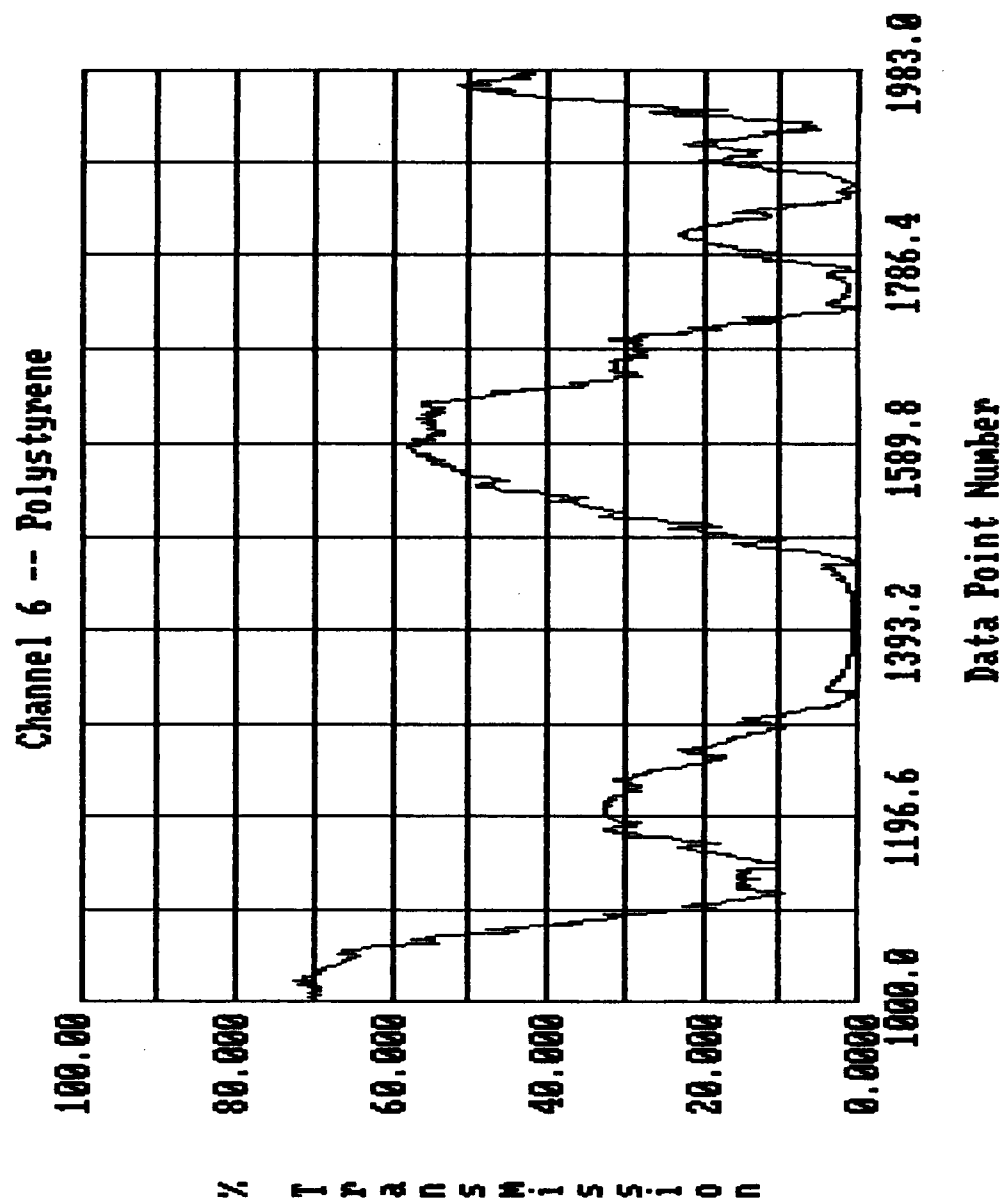


Figure 6.2. Transmission of Polystyrene in the Range of CH 6.

are for channel 6. The minima at data points 1128 and 1749 are known to be at 3.505 and 3.300 microns, respectively. Polystyrene is an industry-wide calibration standard. Samples are provided by Beckman Instruments, Inc.

The example in Figure 6.2 shows the presence of a troublesome, low-level "channel spectrum" superimposed on the data. The channel spectrum is the series of small signal peaks, which are almost uniformly spaced with respect to wavelength. This artifact is evident in much of the data and may be generated by inter-reflections in the sapphire vacuum-interface window on the detector attachment. It is more of a nuisance than a significant error source.

The wavelength calibration process consists of accumulating transmission data for plexiglass, polystyrene, Optosil®, and the atmosphere, all of which have known absorption lines. A system of eight linear calibration equations are then formulated manually. There must be at least one spectral line detected on each channel, and some channels must have more than one line in order to establish the slopes of the equations. The equations give transmitted wavelengths as functions of data point numbers for each of the eight channels, as follows:

$$\begin{aligned}
 \text{LAM1} &= 1.438 - 0.0001650n \\
 \text{LAM2} &= 1.769 - 0.0001650n \\
 \text{LAM3} &= 2.092 - 0.0001504n \\
 \text{LAM4} &= 2.407 - 0.0001504n \\
 \text{LAM5} &= 3.212 - 0.0003300n \\
 \text{LAM6} &= 3.877 - 0.0003300n \\
 \text{LAM7} &= 4.515 - 0.0003000n \\
 \text{LAM8} &= 5.184 - 0.0003000n
 \end{aligned}
 \quad 0 \leq n \leq 1983. \quad (6.1)$$

The equation slopes and intercepts are entered into programs RESPCALC and REDUCDAT, as indicated in Figure 5.1 (page 89). The data are entered manually by editing those programs.

The accuracies of the constants in Equation (6.1) have not been established, but they could undoubtedly be improved by using better standards, such as absorbing gases. Samples such as polystyrene are used because of the convenience.

The programs that support the instrument's spectral sensitivity (inverse responsivity) calibration process are RECORDAT, RECOVDAT, COADD and RESPCALC. The process starts by setting the instrument to "look" at a standard calibration source. The standard source used in this project was a 1000°C heated-cavity blackbody which generates the source curve shown in the introduction. The first three programs are run to produce data file F:AVERGR.DAT. Approximately 25 scans are taken (at L0 data interrupt interval) and co-added to produce the file. The maximum number of scans that can be co-added is limited by the upper range of a signed long integer (+2,147,483,659). The product of the number of co-added scans and the maximum detector signal (of any channel) must not exceed this number. The maximum possible detector signal is 10^8 in units of tenths of a microvolt. This corresponds to a detector signal of 10 volts with unity gain. If any of the detector signals go this high, only 21 scans can be co-added. In the calibration process, the maximum signal observed was approximately 2.5 volts at unity gain.

After acquiring F: AVERGR.DAT, program RESPCALC is executed to produce inverse responsivity in units of $((\text{watt}/\text{cm}^2) \cdot \text{micron} \cdot \text{sr})$ /tenth of a microvolt). The data are generated in both ASCII and binary files as discussed earlier. The ASCII files may be printed or plotted. The complete set of experimental inverse responsivity data is shown plotted in Appendix A. Since the plotting routine handles only 1000 data points, a total of 16 plots are required. The figures are presented with two plots for each of the eight channels. The data are plotted with auto-ranging log scaling on the ordinate.

A review of the data in Appendix A reveals a great deal of information about the instrument performance. Since the data are plotted as inverse responsivity, the instrument is less sensitive at the higher values. The data show that, in general, the instrument is less sensitive at the ends of spectral ranges of the various channels. This results from the fact that the grating (Figure 2.1 (page 12)) becomes underfilled at the ends of excursion of the scanning mirror. Other features evident in the plots are atmospheric absorption effects around 1.4, 2.7 and 4.3 microns. These effects are present in spite of efforts to purge the air out of the instrument during the calibration run. The data suggest that an evacuable instrument is very desirable. The data also reveal other instrument features beyond the scope of this discussion.

The final process, that of acquiring raw data and reducing it to engineering radiometric units, is supported by program RECORDAT, RECOVDAT, COADD, and REDUCDAT. The instrument is set to "look" at

an unknown source (e.g., a rocket plume) and RECORDAT is executed. When convenient, programs RECOVDAT and COADD are executed to produce data file F:AVERGS.DAT. Again, one or more scans can be co-added as discussed earlier. Finally, REDUCDAT is executed to generate reduced data, as indicated in Figure 5.1 (page 89). Since REDUCDAT reads the calibration data file RESP.DAT at the time of execution, the operator must be certain that RESP.DAT is present on drive C.

The reduced radiance data are in ASCII form and may be printed or plotted. A complete set of reduced data plots is presented in Appendix B. These data represent the measurement of the 1000°C black-body radiance taken at MLO interrupt interval. At this interval, which allows one plot per detector channel, there are 992 data points. The data were taken through an IRTRAN® 2 window, and corrections for window losses are included, as explained in Chapter V. The data were taken with a 10-foot air path. The eight plots taken as a unit should be identical to the 1000°C blackbody radiance plot shown in the introduction (see Figure 1.2 (page 5)). The agreement between the two sets of data is very good ($\pm 1\%$) except in a few trouble spots. With the inverse responsivity data being in error in certain spots, as described earlier, the radiance data is directly affected at the same spots. This is a reflection of the spectroradiometer's performance rather than a software deficiency.

The reduced data in Appendix B show that the data acquisition and data processing techniques produce high-quality data with errors on the order of $\pm 1\%$ or less.

In general, the data acquisition and data reduction processes work very well. An inverse responsivity calibration or a radiance measurement requires only a few minutes. Program RECOVDAT consumes most of the time, because it is required to isolate each spectral scan in a separate DOS file. However, this feature of RECOVDAT maximizes the accessibility of the data and makes the subsequent data processing fast and convenient.

CHAPTER VII

SUMMARY AND CONCLUSIONS

A complete system for acquiring and reducing spectral radiometric data was developed and demonstrated. The system consists of customized digital data acquisition circuitry that generates data automatically but outputs data as instructed by a microcomputer. The microcomputer reads, organizes, and stores the data for subsequent data processing. A customized software system was developed to acquire, store, and process the data.

The software system supports the operations of a grating spectroradiometer that scans the 1.1 to 5.1 micron range in one second. Very high data density was realized on the wavelength scale. The system provides the option of having up to 16,000 data points at the scan rate of 2 seconds per scan. At 1 second per scan, up to 8,000 data points can be acquired per scan. The software supports continuous data acquisition for periods of 10 minutes or longer, depending on the amount of data generated per scan.

The software system also supports the processes of wavelength calibration, instrument sensitivity (inverse responsivity) calibration, and radiance data reduction. Access to all processed data is maximized by writing the data of each process to an ASCII file in MS-DOS. Therefore, individual spectral scans may be selected and reduced to engineering units.

The individual programs interact with the operator and are highly communicative. This feature helps minimize errors in using

the system. In addition, as shown in the various program flow diagrams, a great deal of run-time condition testing is incorporated to eliminate execution errors.

The functionality of the software system was demonstrated by going through a complete calibration and subsequent test application of the spectroradiometer. The experimental results demonstrate excellent performance of the data acquisition and processing system. The reduced data from the radiance measurement demonstration show that the data acquisition and data processing techniques produce high-quality data with errors of $\pm 1\%$ or less.

The performance of the spectroradiometer was found to be good in most of its spectral range. However, the experimental results revealed several trouble spots where poor-quality data were generated. In general, these spots coincide with atmospheric absorption bands or the extremes of the scanning mirror excursions. The experimental results suggest the need for improvements in the grating spectrometer hardware. Specifically, the use of an evacuable spectrometer would eliminate calibration errors due to atmospheric absorption effects. Also, the use of a faster objective lens would improve the data accuracy at the extremes of the scan mirror excursions. In a qualitative sense, the performance of the data processing system exceeds the performance of the spectrometer; and, therefore, additional improvements in the spectrometer's performance are necessary to improve the total system performance.

LIST OF REFERENCES

LIST OF REFERENCES

1. Rochelle, W. C. "Review of Thermal Radiation From Liquid and Solid Propellant Rocket Exhaust." NASA TMX-53579, George C. Marshall Space Flight Center, February 1967.
2. Dereniak, E. L. and D. G. Crowe. Optical Radiation Detectors. New York: John Wiley and Sons, 1984.
3. Walker, R. A., et al. "Spectroradiometry: Its Benefits, Its Hardware, Its Uses," Optical Spectra, October 1981.
4. Sparrow, E. M. and R. D. Cess. Radiation Heat Transfer. Belmont, California: Brooks Cole Publishing Company, 1967.
5. James, J. F. and R. S. Stenberg. The Design of Optical Spectrometers. London, England: Chapman and Hall LTD., 1969.
6. Loewen, E. G., et al. "Diffraction Grating Handbook." Rochester, New York: Diffraction Grating Research Laboratory, May 1970.
7. Fastie, W. G. "A Small Plane Grating Monochromator," Journal of the Optical Society of America, Vol. 42(9), September 1952.
8. Fastie, W. G. "Image Forming Properties of the Ebert Monochromator," Journal of the Optical Society of America, Vol. 42(9), September 1952.
9. Fastie, W. G., et al. "Vacuum Ebert Grating Spectrometer," Journal of the Optical Society of America, Vol. 48(2), February 1958.
10. Sherrell, F. G. "Autoranging Radiometric System," Proceedings of the 33rd International Instrumentation Symposium, Instrument Society of America. Las Vegas, Nevada, May 1987.
11. "Model 1018-PC Owner's Manual, Version One." Industrial Computer Designs, Inc., Westlake Village, California.
12. "Optimizing C86 User's Manual, Version 2.3." Computer Innovations, Inc., Tinton Falls, New Jersey.
13. Newland, D. E. An Introduction to Random Vibrations and Spectral Analysis. London, England: Longman Group Limited, 1975.
14. Kernighan, B. W. and D. M. Ritchie. The C Programming Language. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1978.

15. Gehani, N. C: An Advanced Introduction. Rockville, Maryland:
Computer Science Press, 1985.

APPENDICES

APPENDIX A
INVERSE RESPONSIVITY DATA

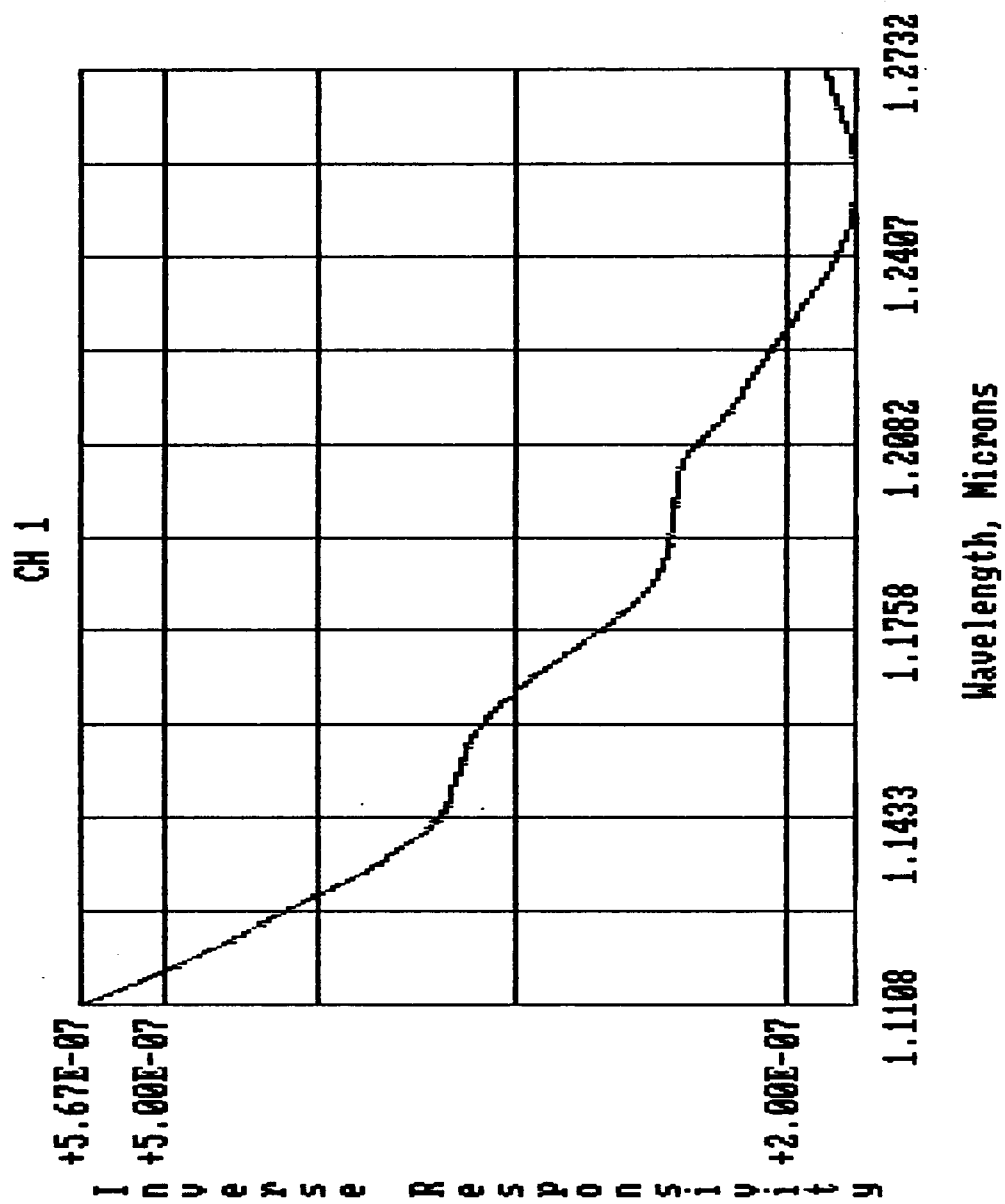


Figure A.1. Inverse Responsivity for Detector Channel 1.

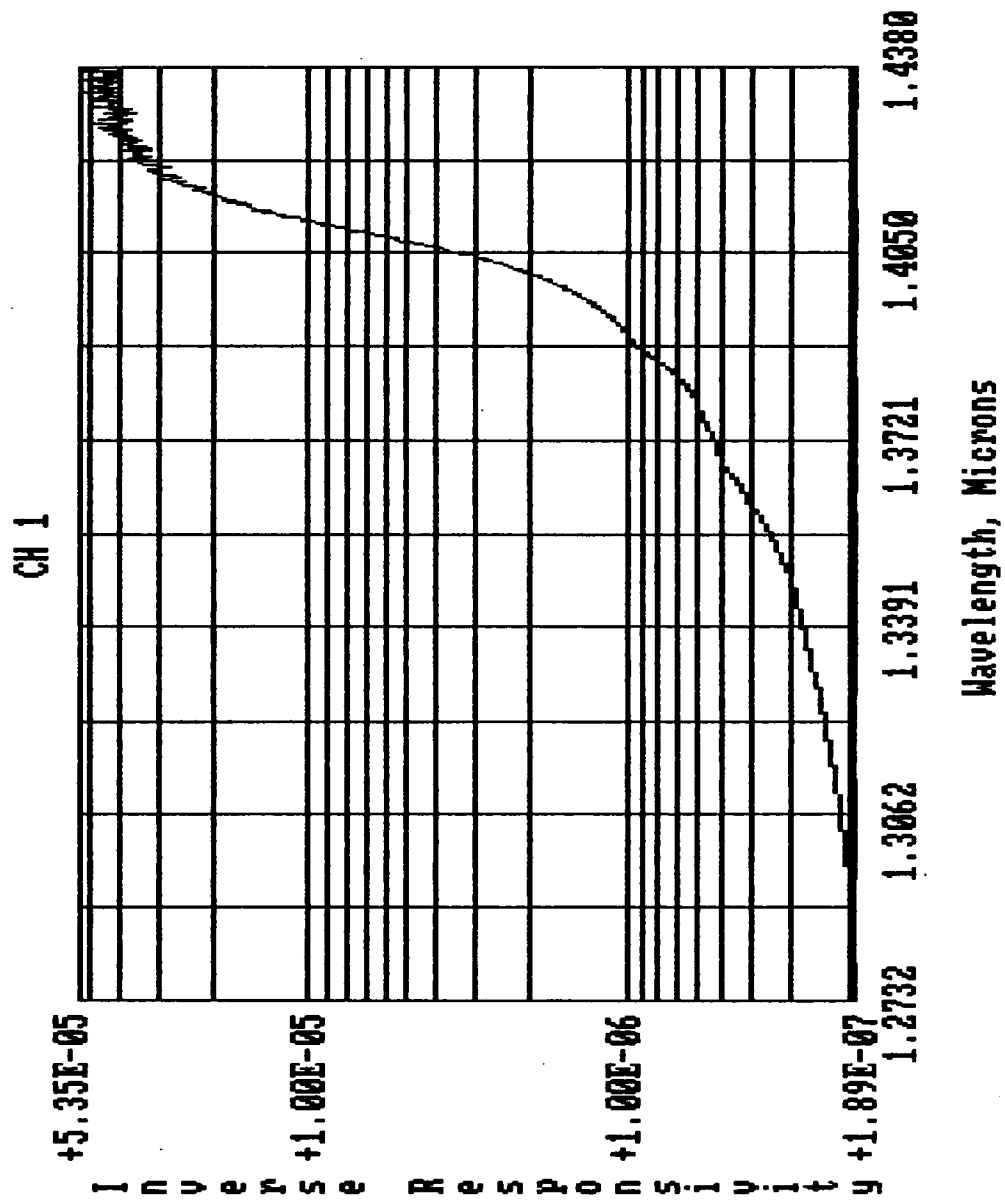


Figure A.1 (Continued)

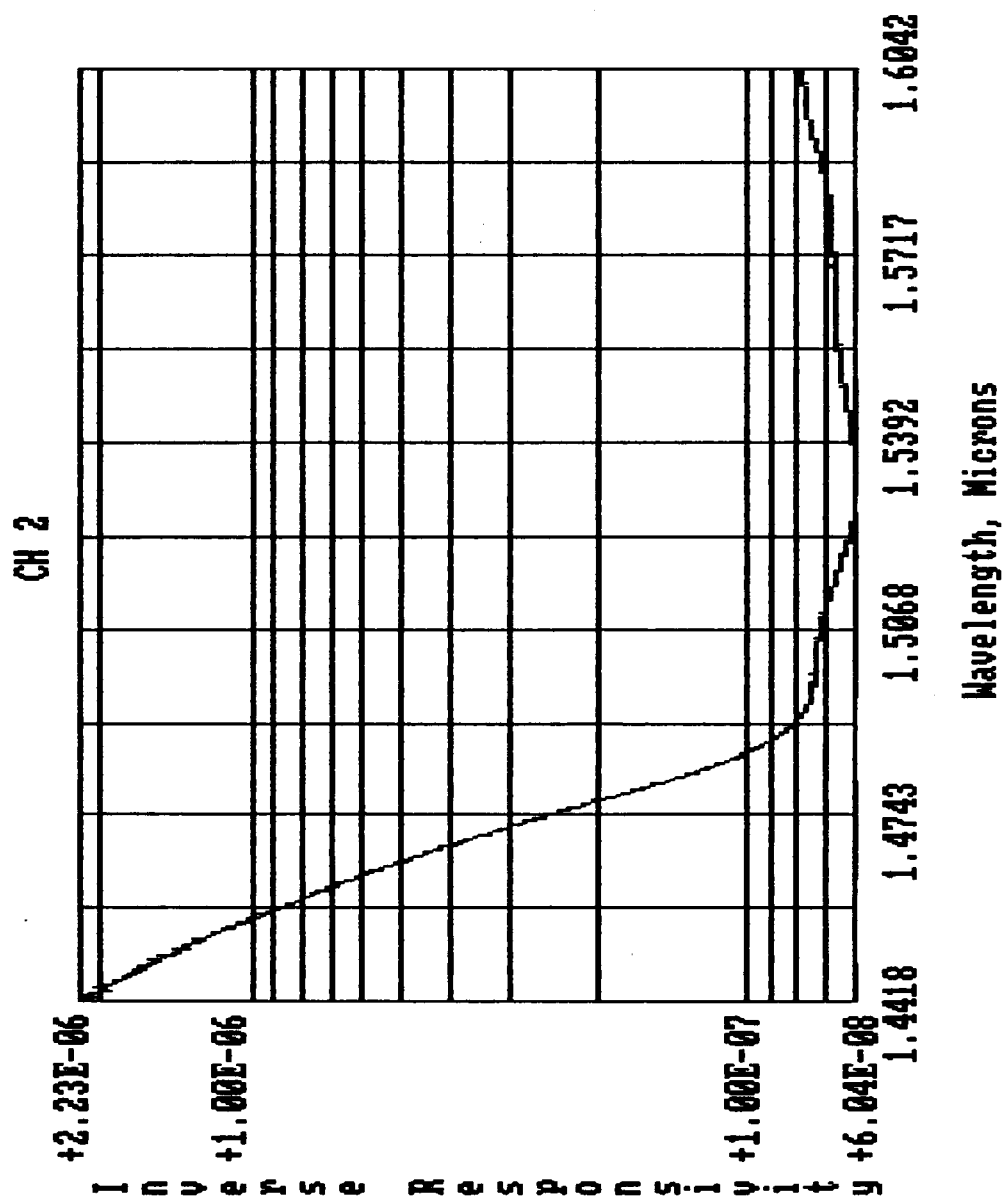


Figure A.2. Inverse Responsivity for Detector Channel 2.

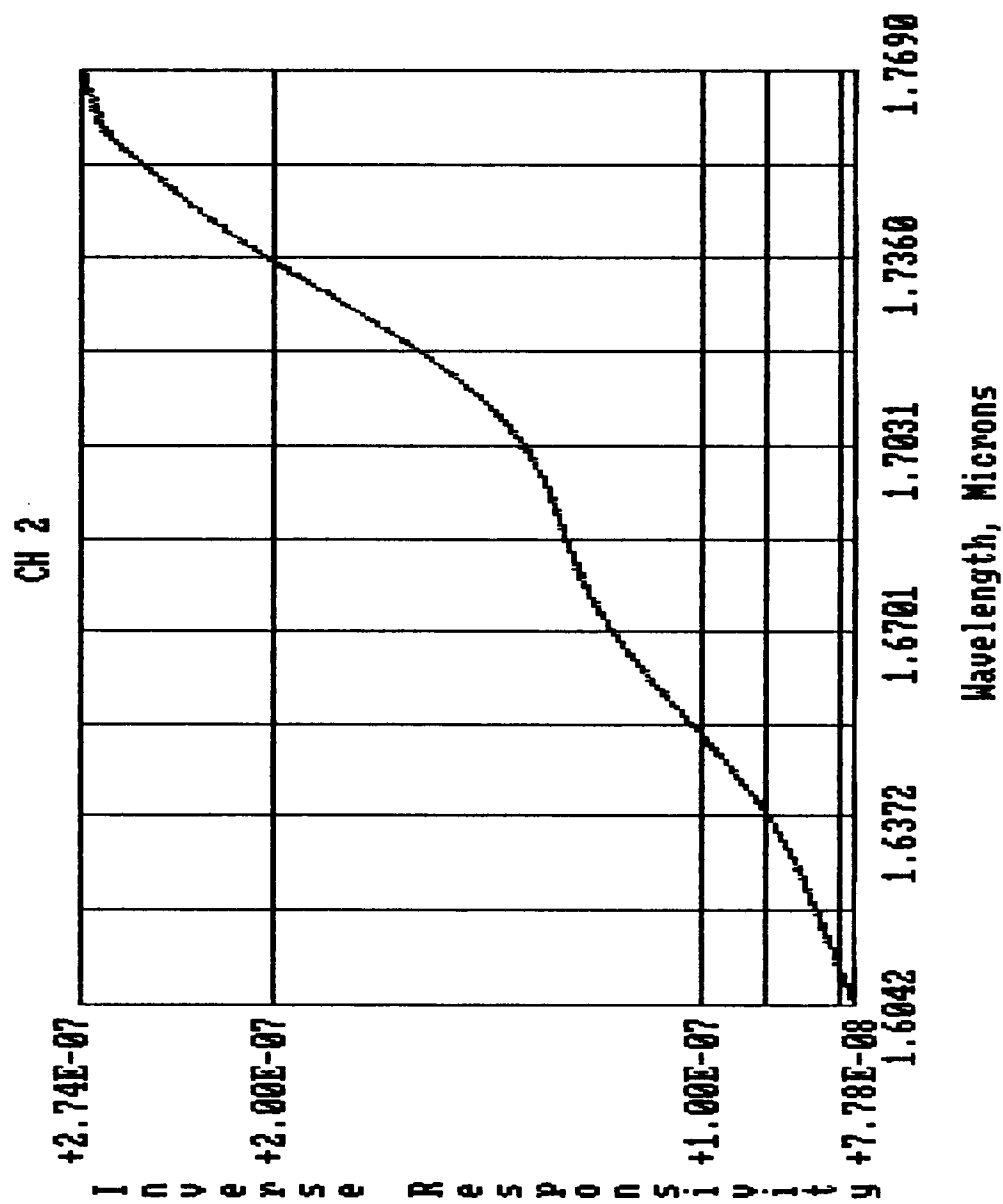


Figure A.2 (Continued)

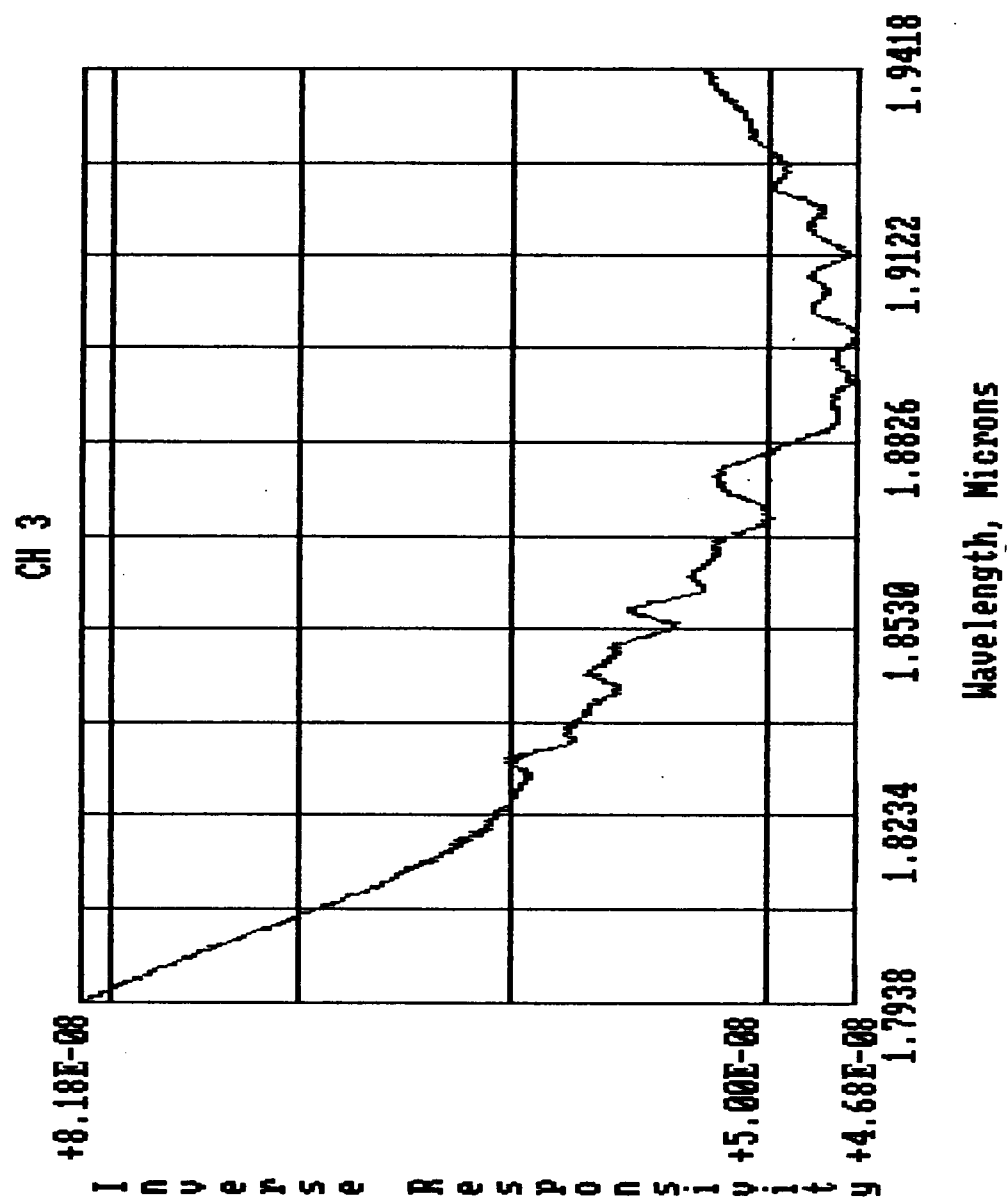


Figure A.3. Inverse Responsivity for Detector Channel 3.

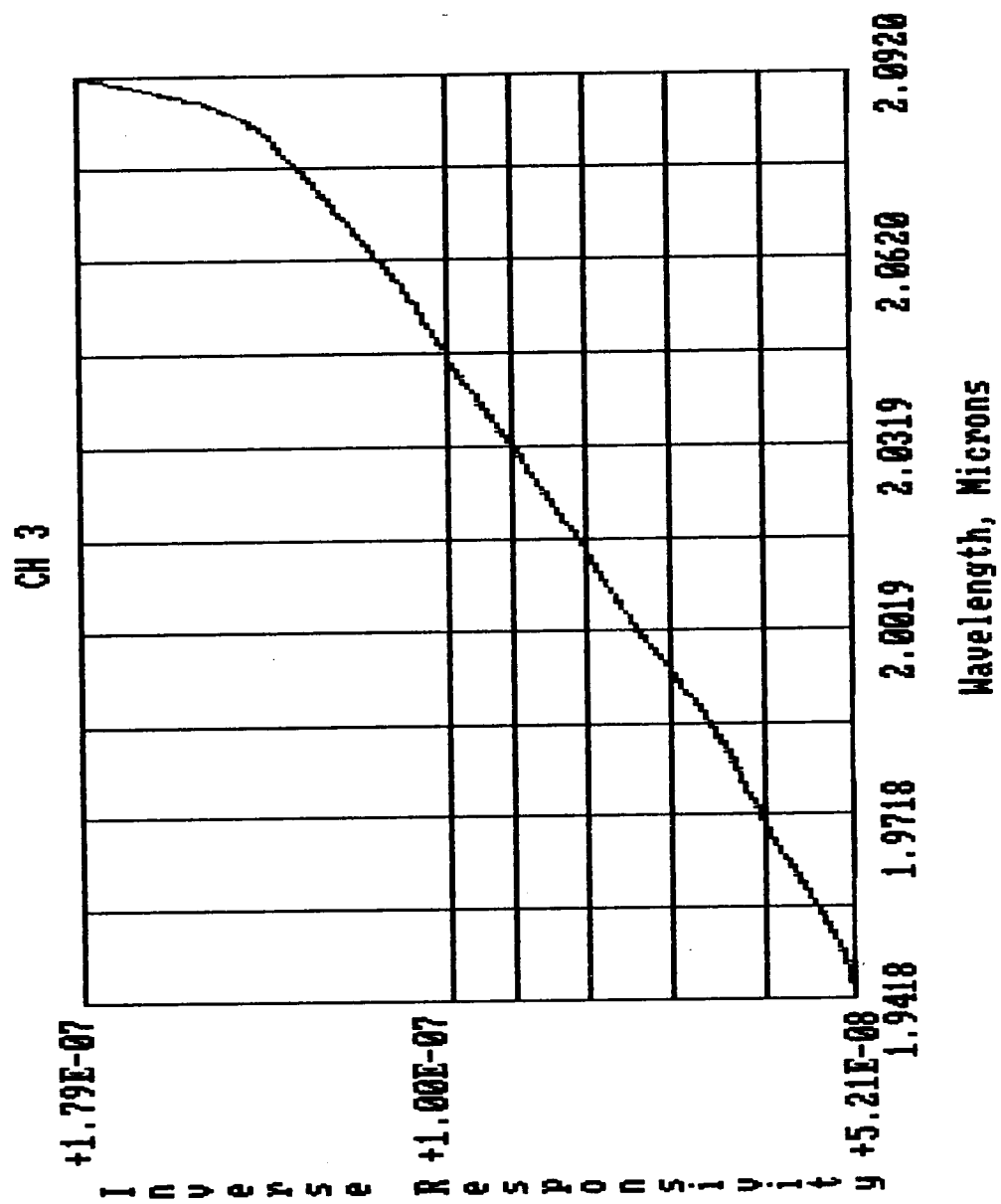


Figure A.3 (Continued)

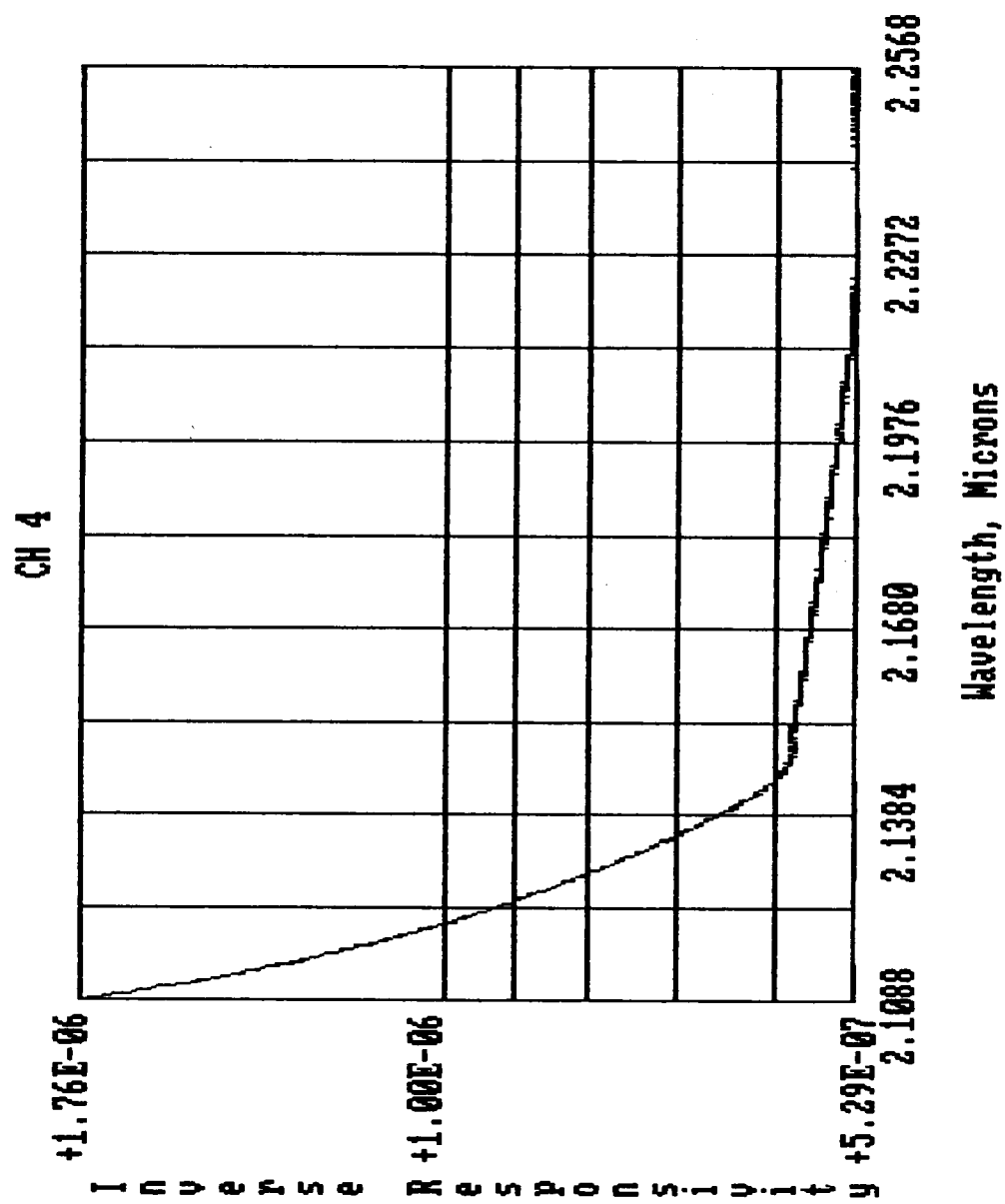


Figure A.4. Inverse Responsivity for Detector Channel 4.

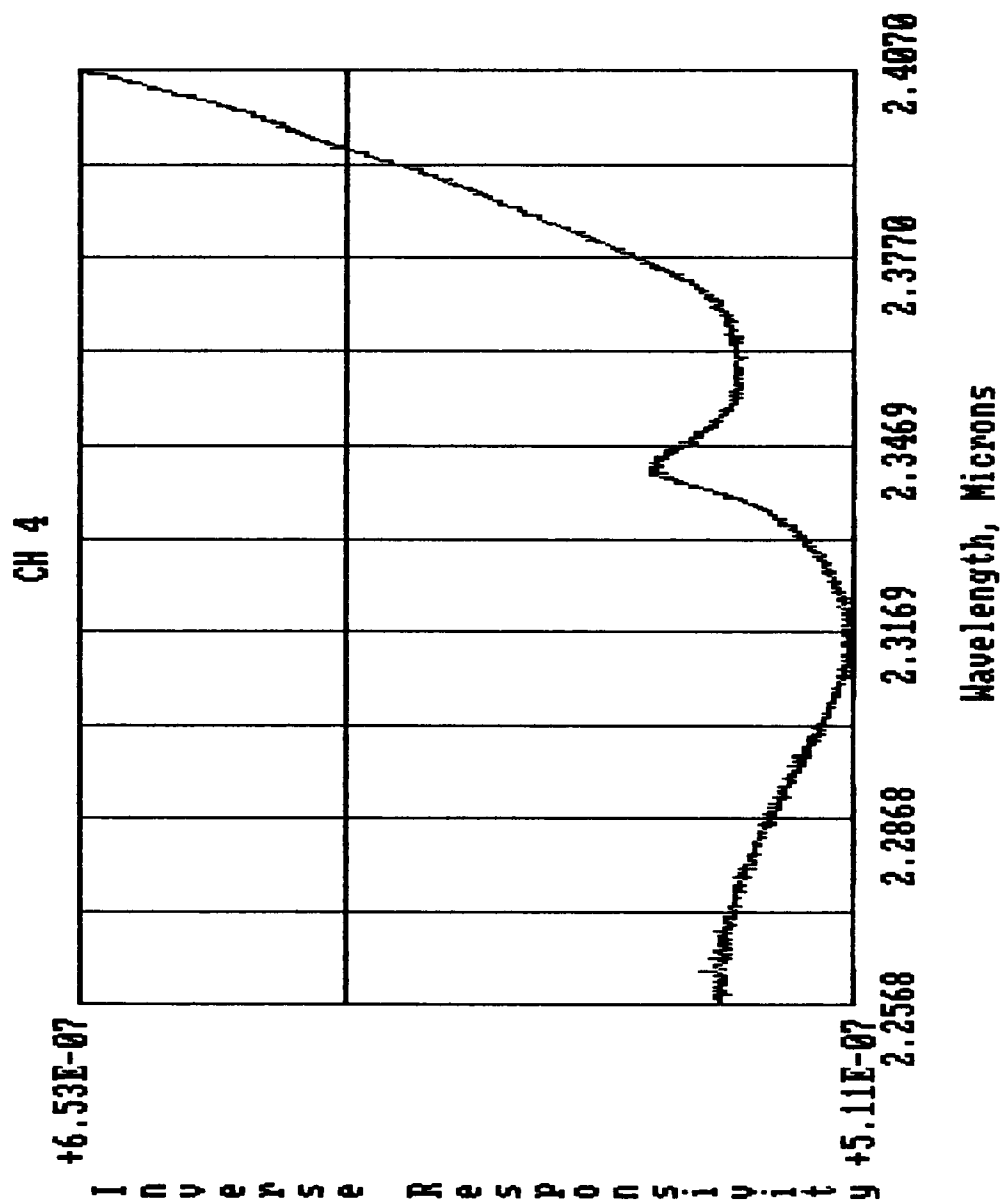


Figure A.4 (Continued)

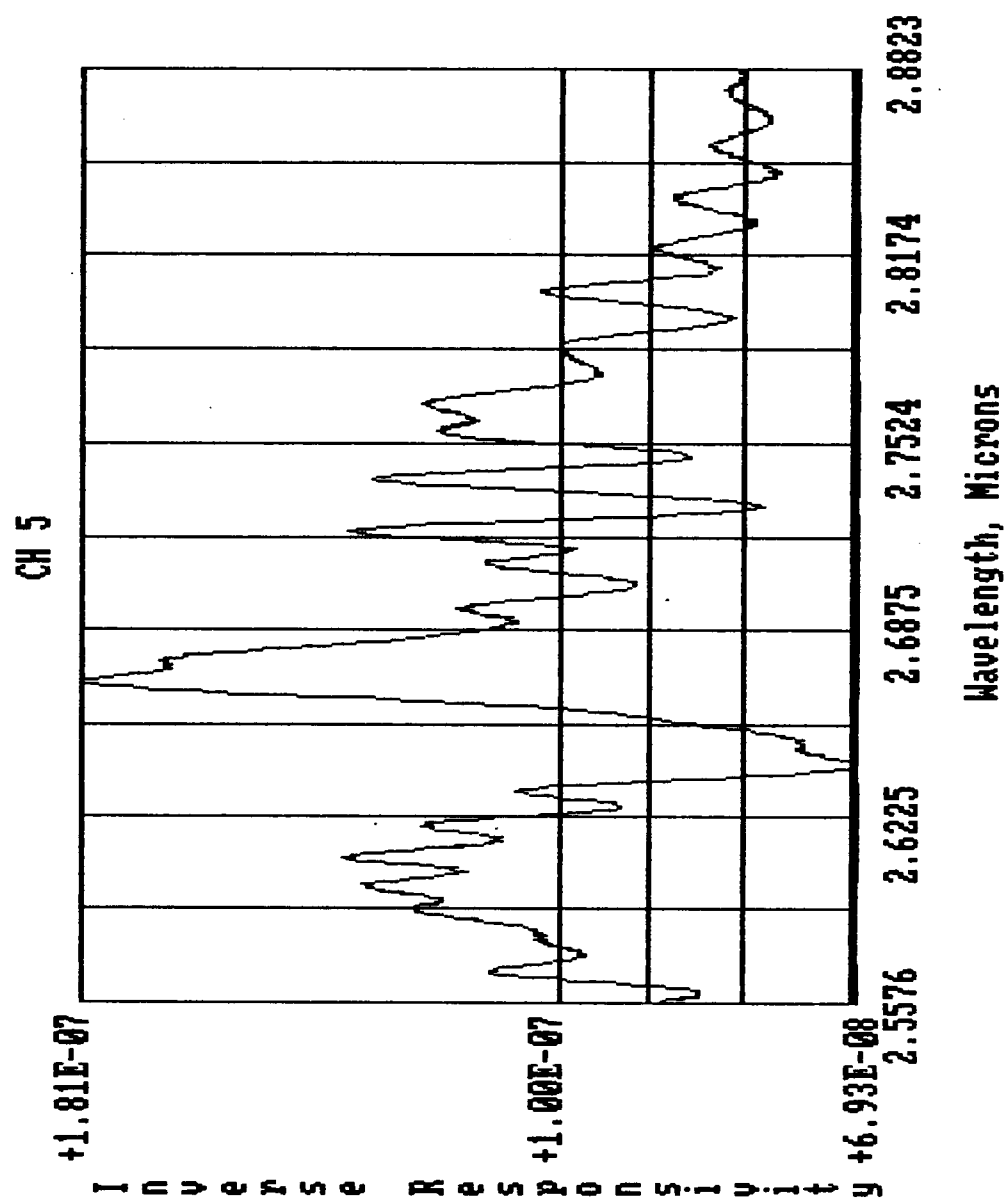


Figure A.5. Inverse Responsivity for Detector Channel 5.

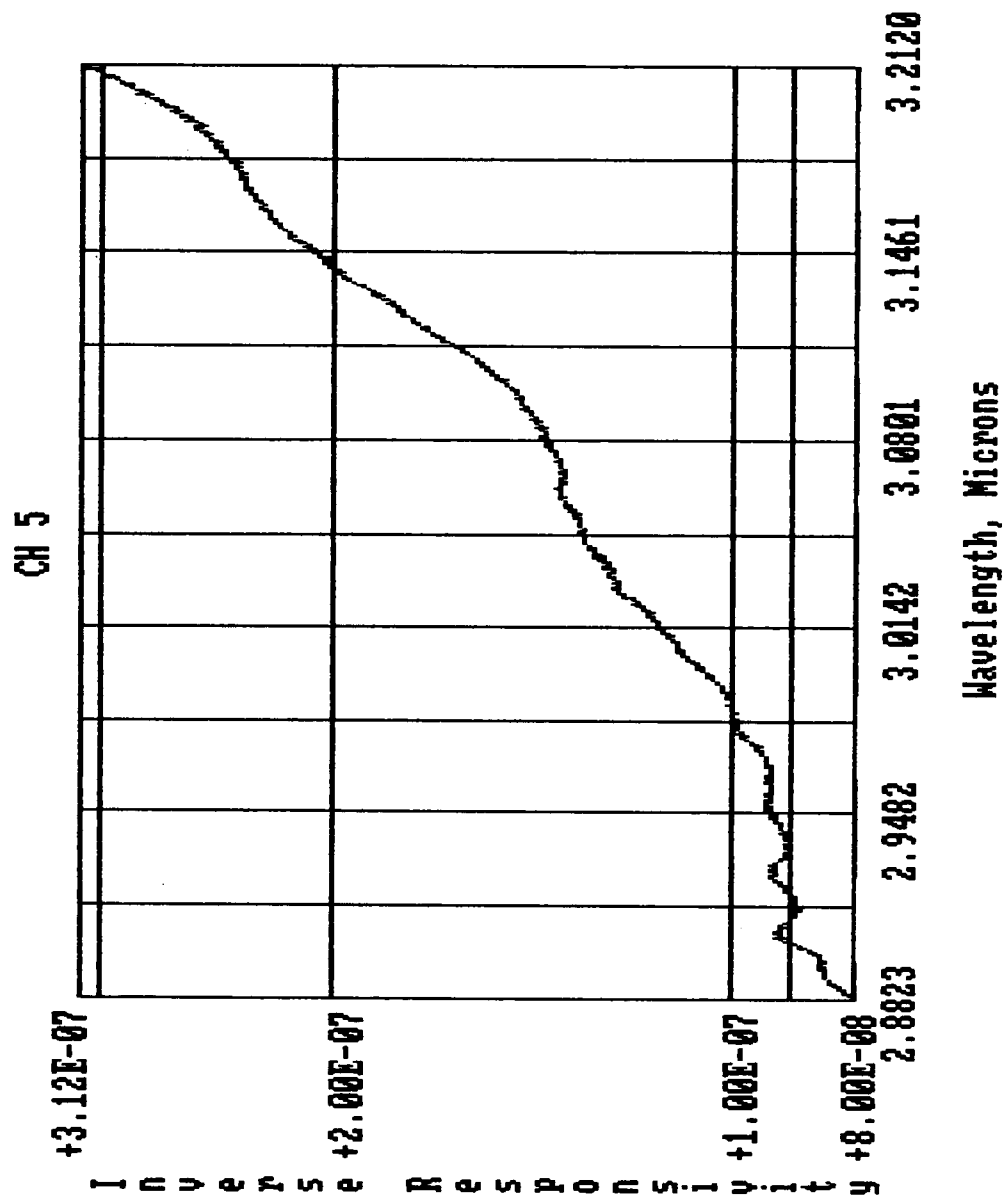


Figure A.5 (Continued)

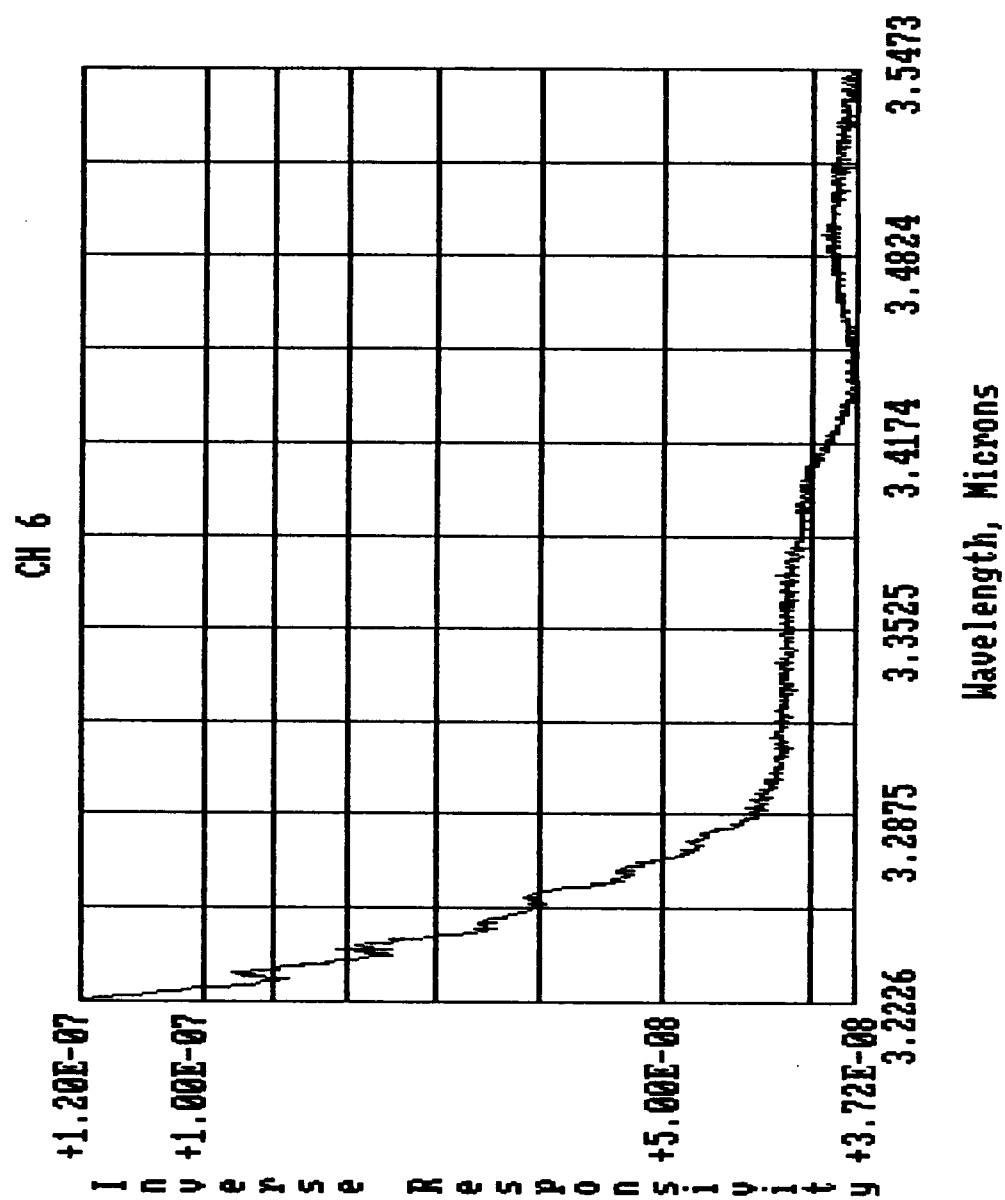


Figure A.6. Inverse Responsivity for Detector Channel 6.

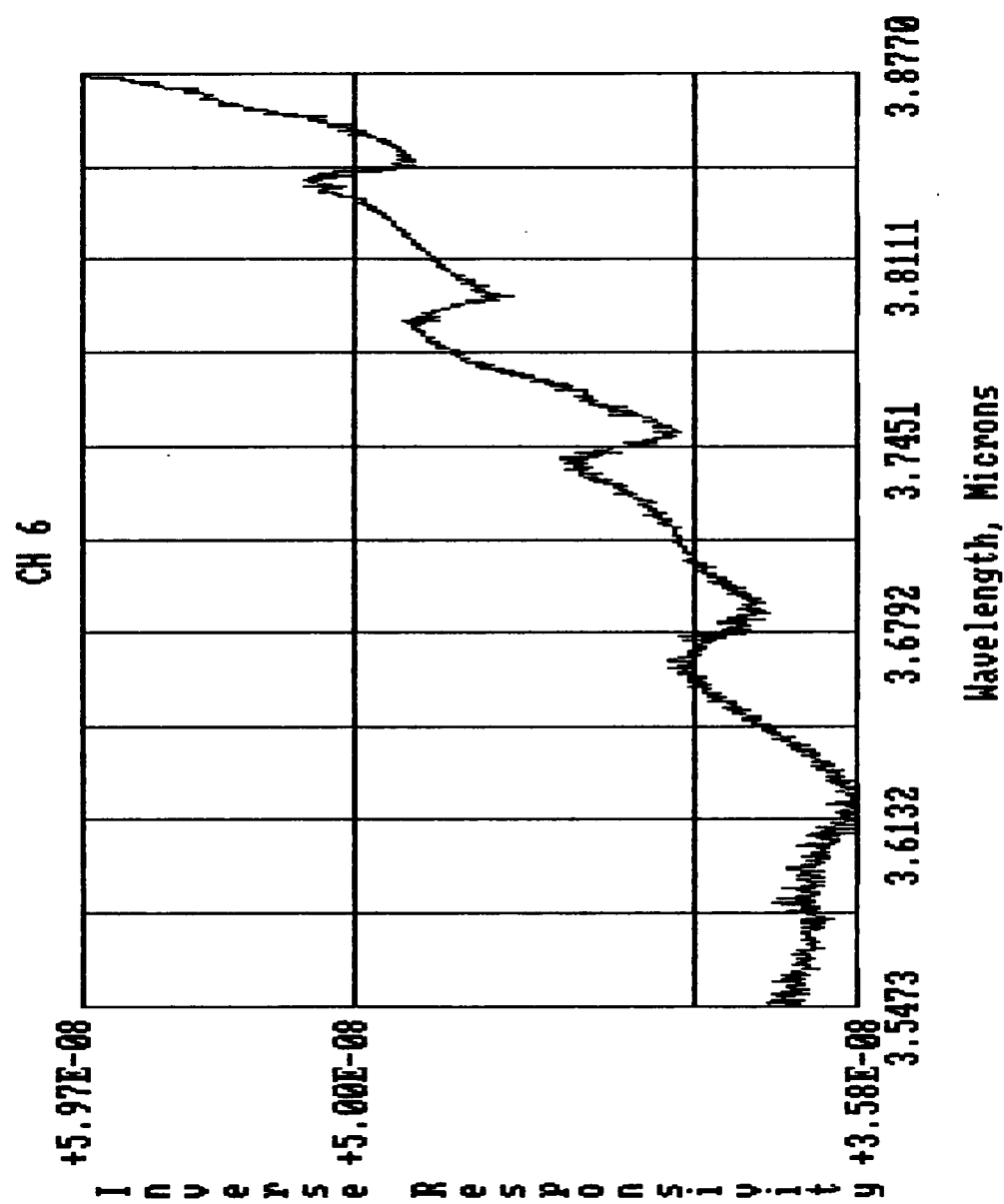


Figure A.6 (Continued)

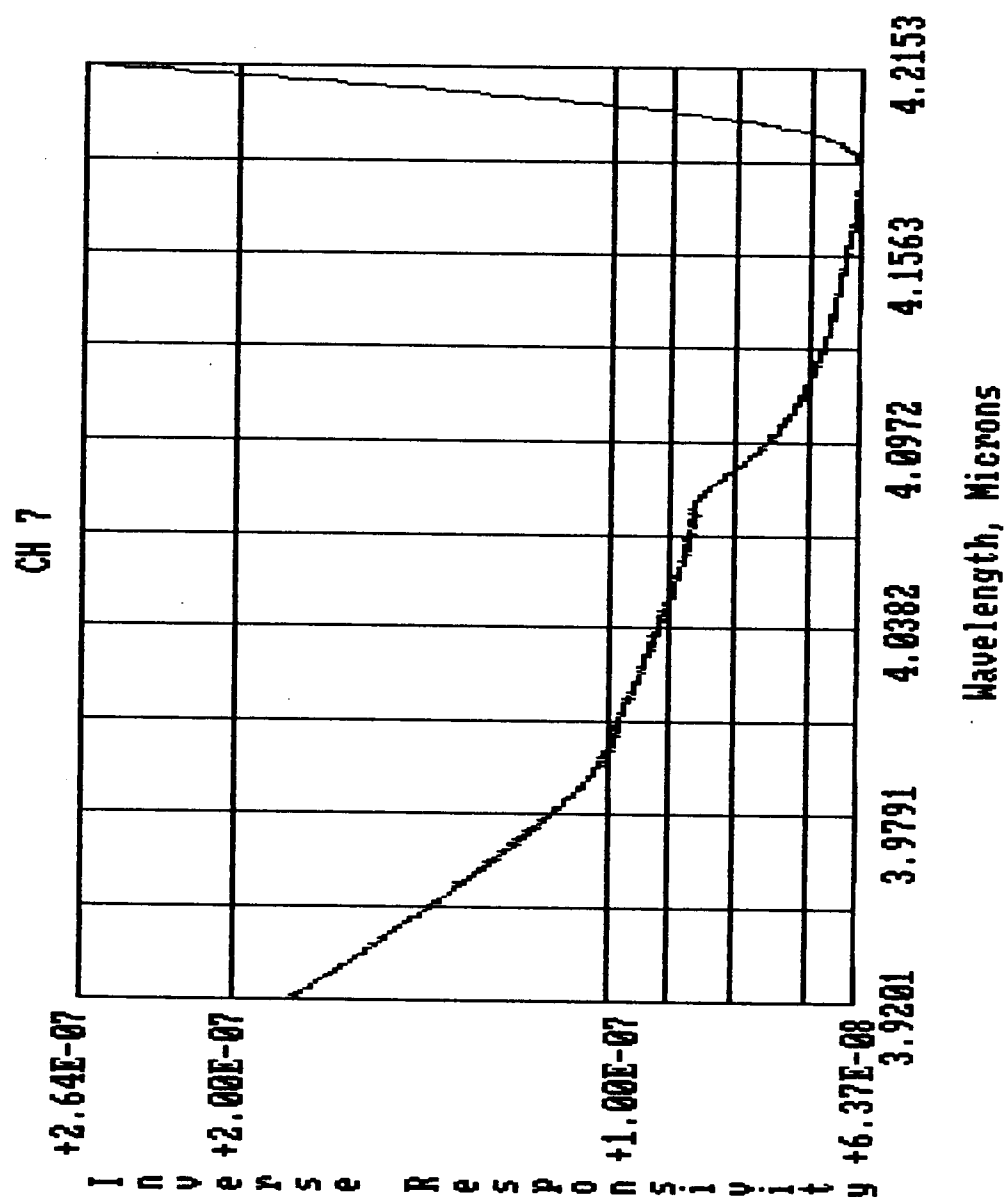


Figure A.7. Inverse Responsivity for Detector Channel 7.

CH 7

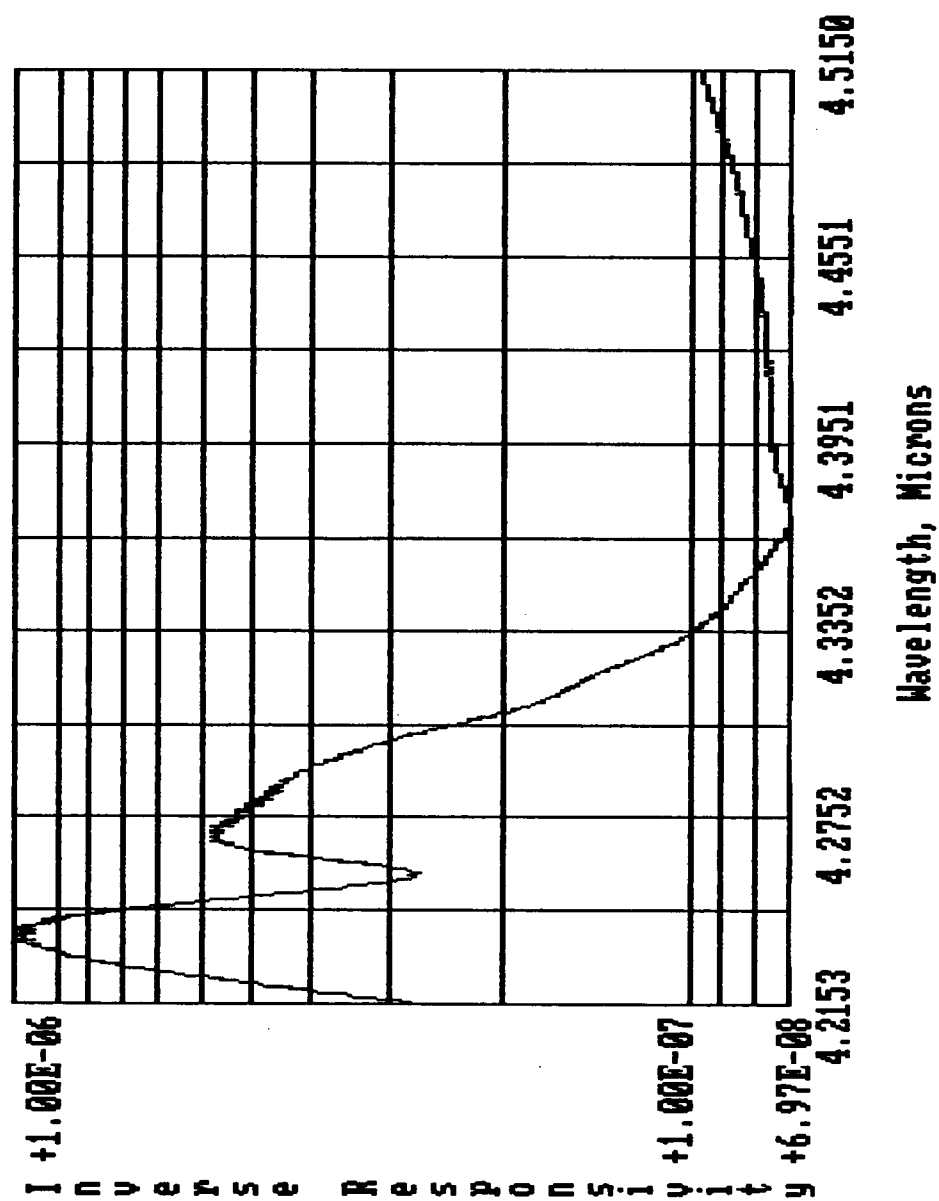


Figure A.7 (Continued)

CH 8

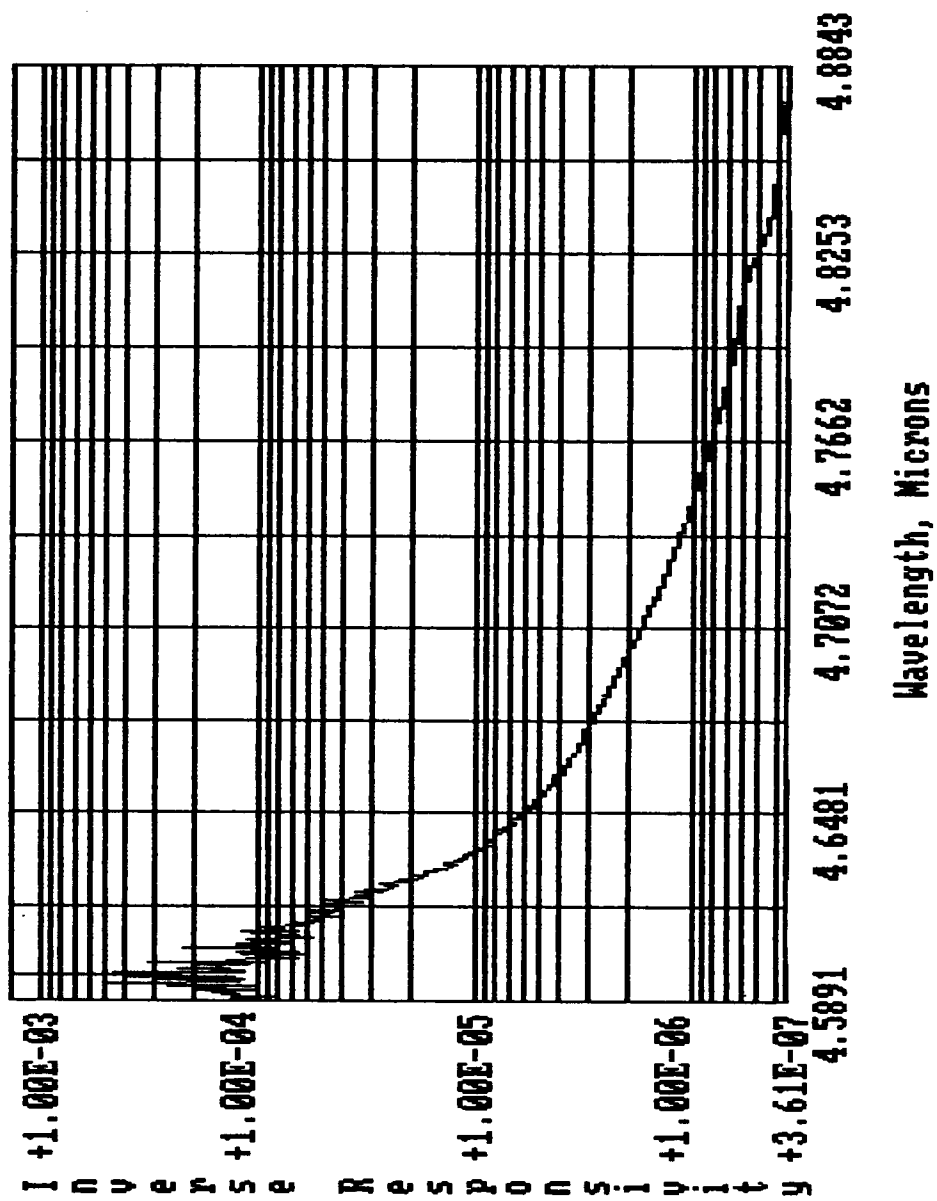


Figure A.8. Inverse Responsivity for Detector Channel 8.

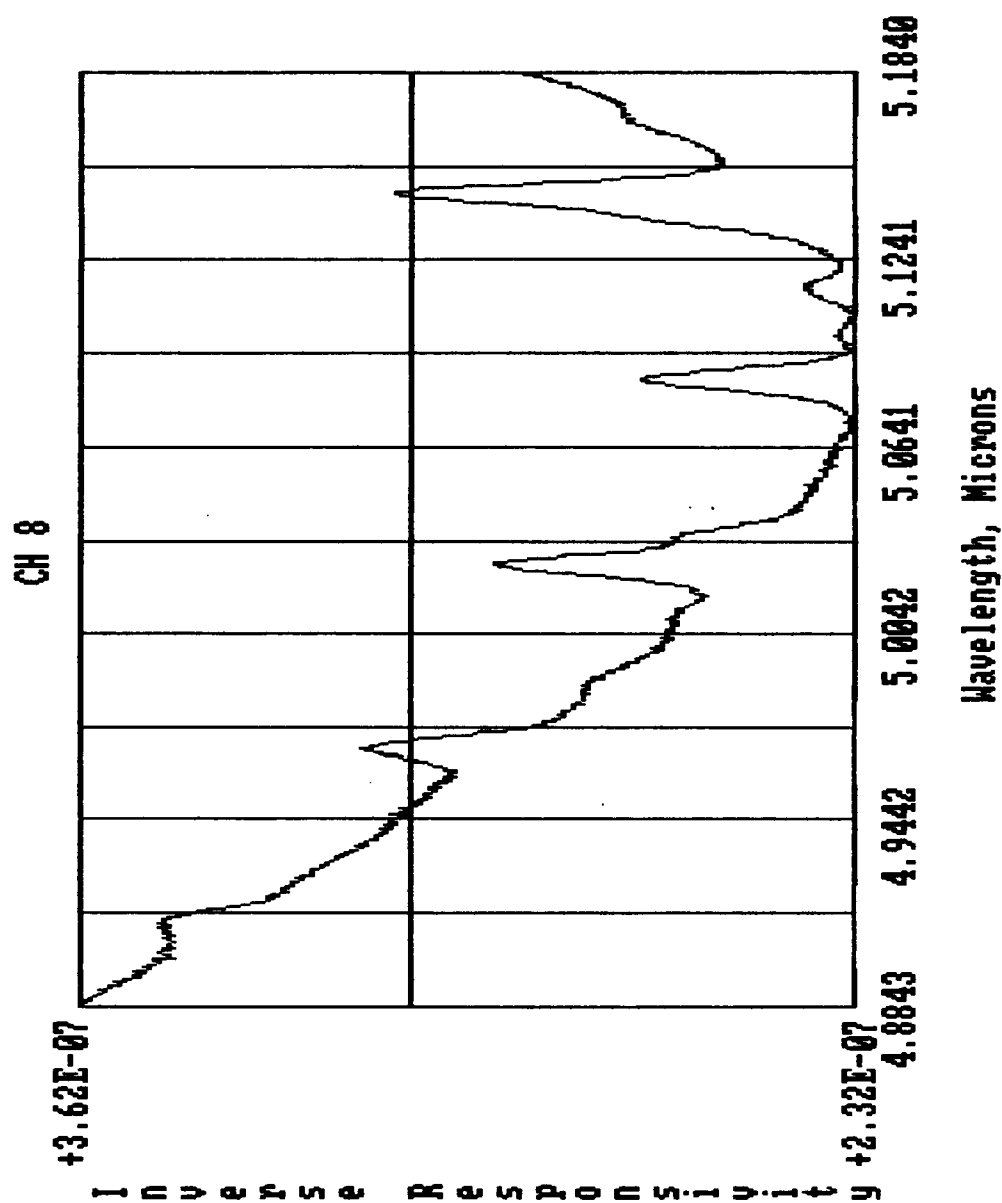


Figure A.8 (Continued)

APPENDIX B
RADIANCE MEASUREMENT DEMONSTRATION

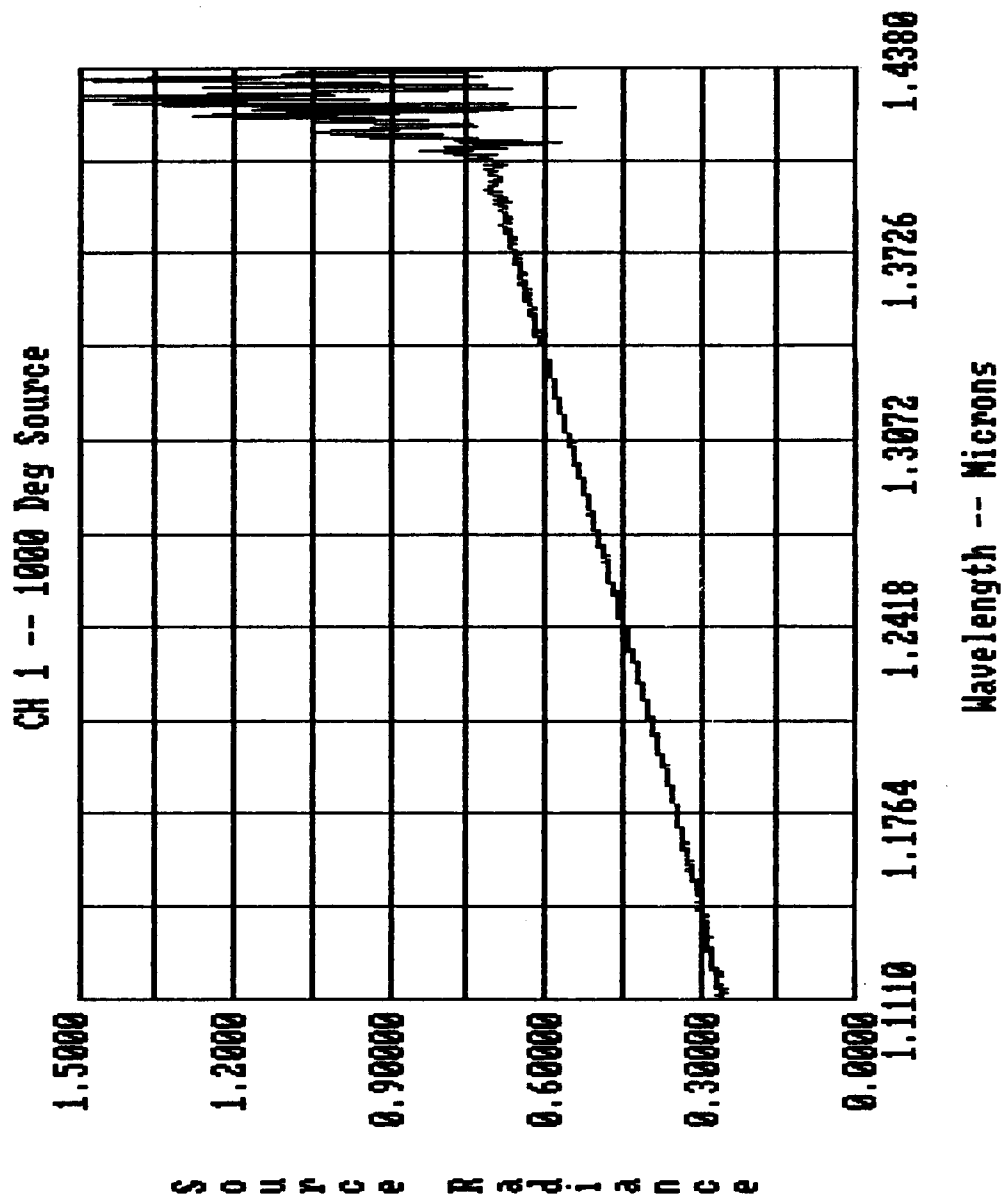


Figure B.1. Source Radiance for Channel 1.

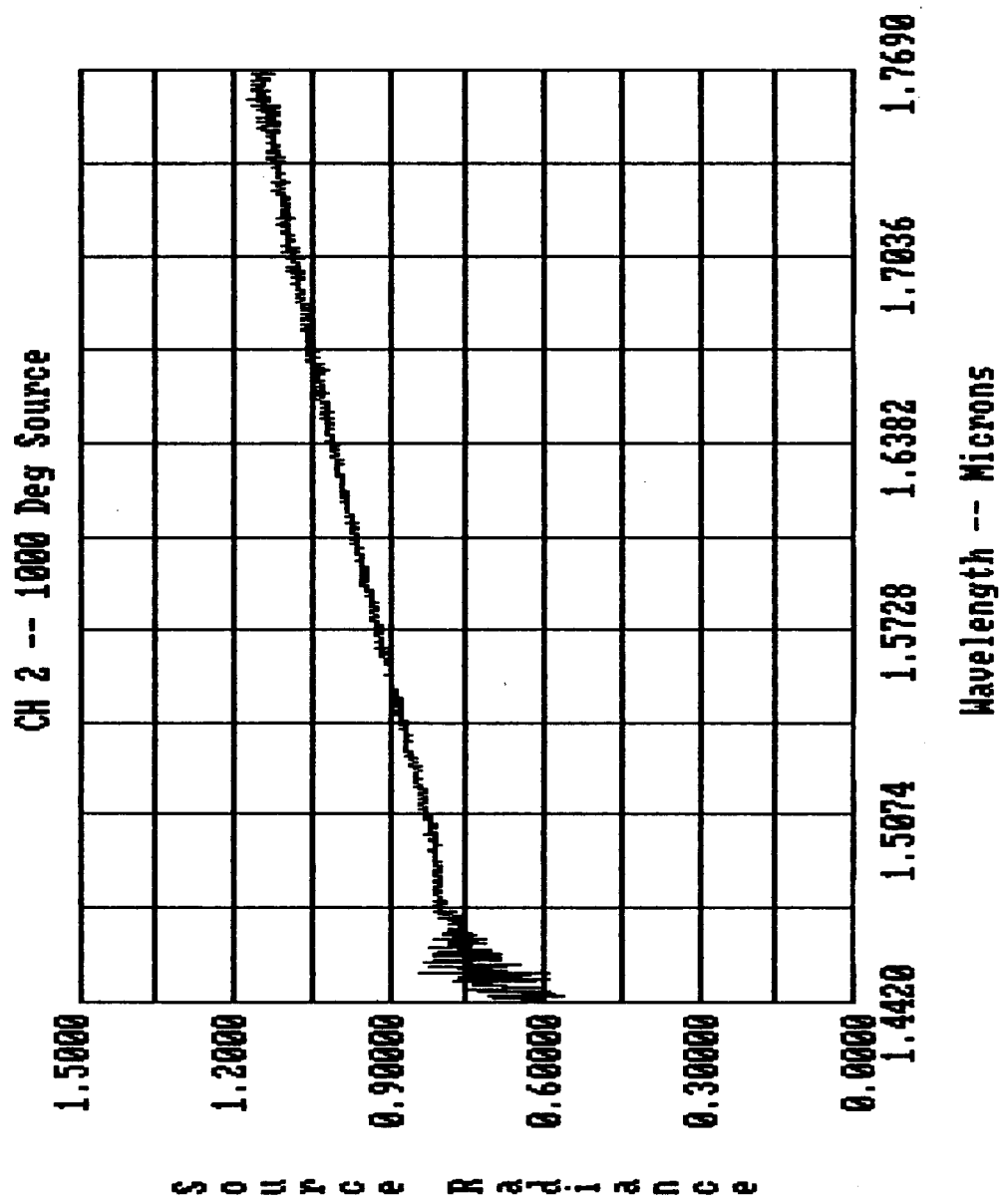


Figure B.2. Source Radiance for Channel 2.

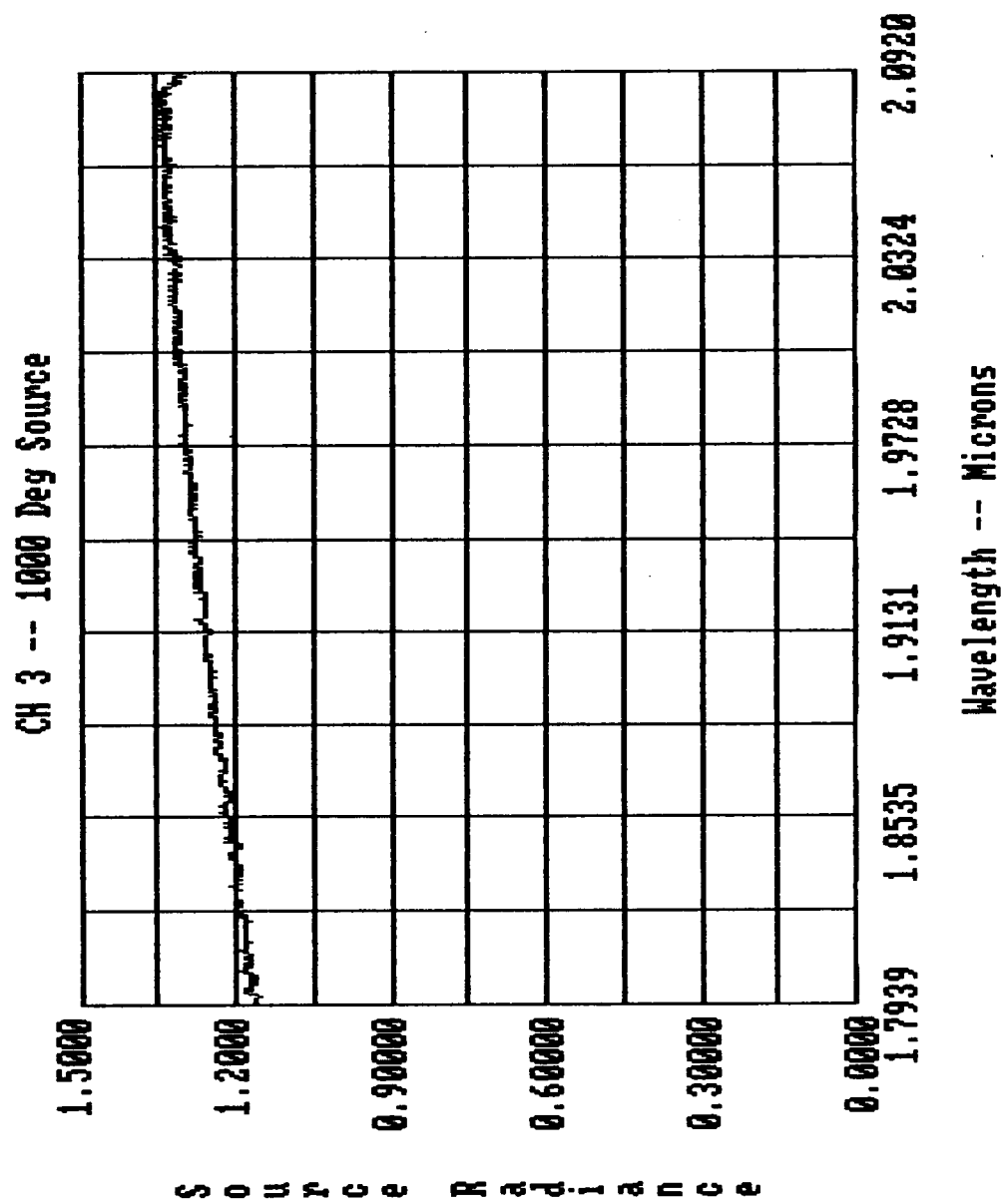


Figure B.3. Source Radiance for Channel 3.

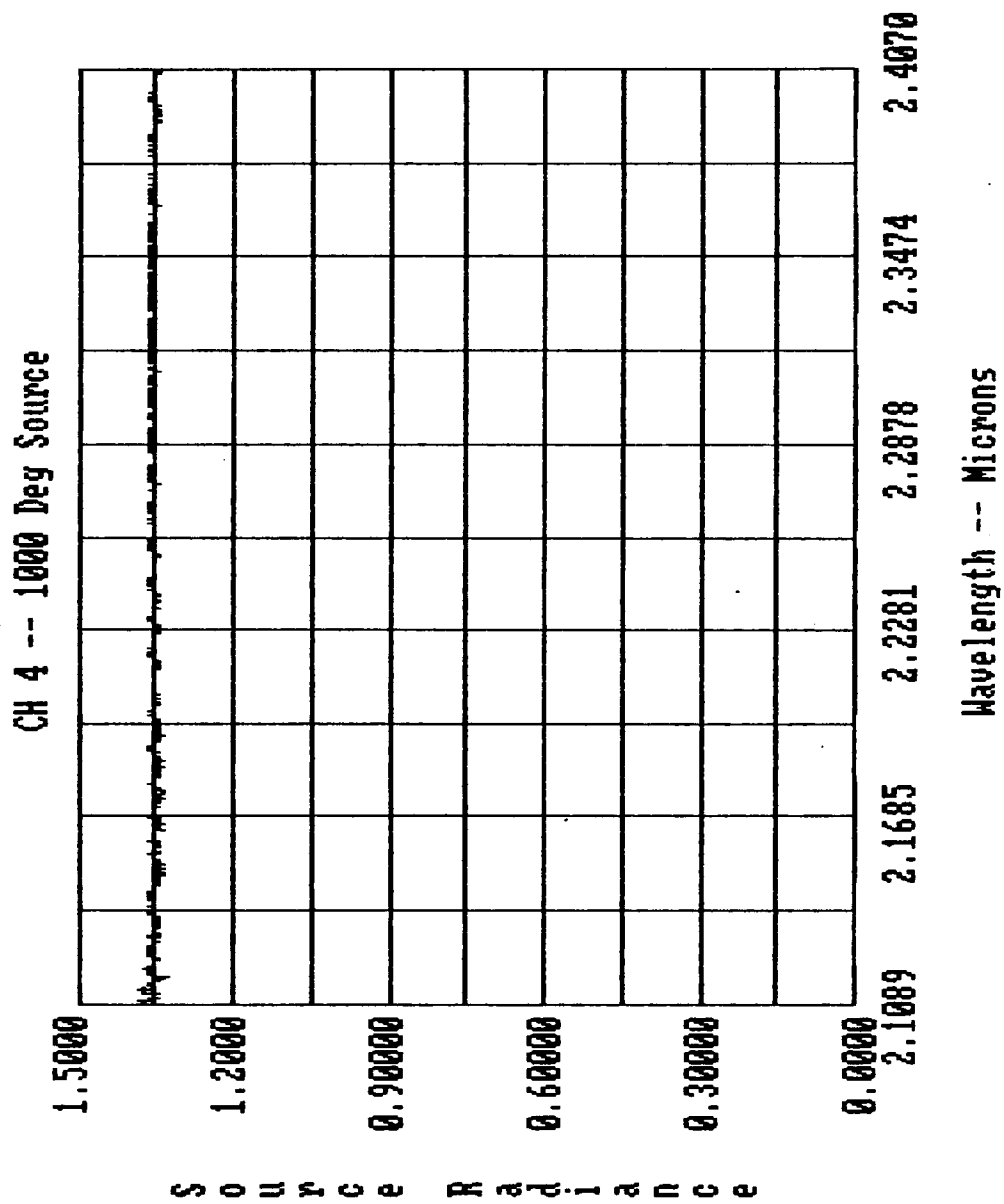


Figure B.4. Source Radiance for Channel 4.

CH 5 -- 1000 Deg Source

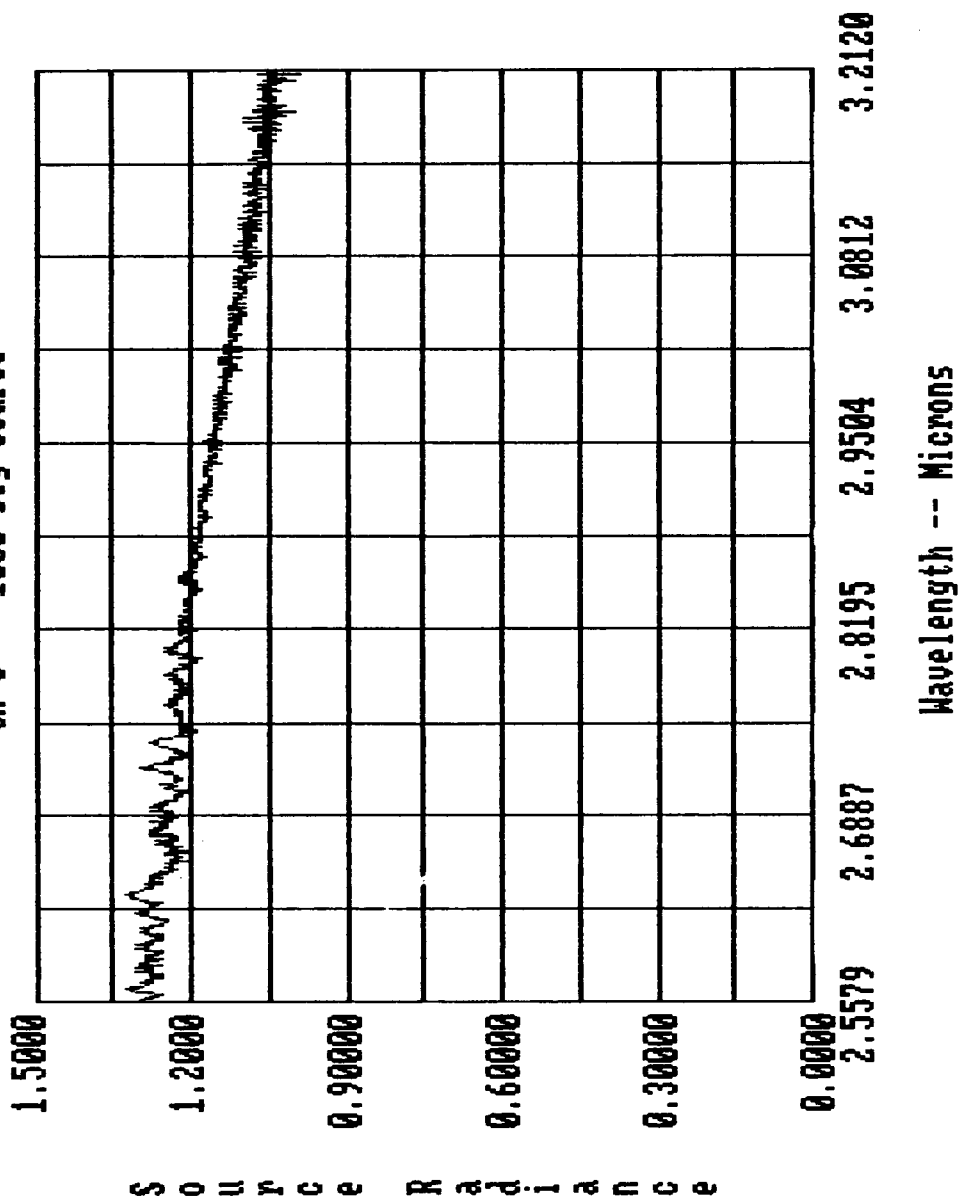


Figure B.5. Source Radiance for Channel 5.

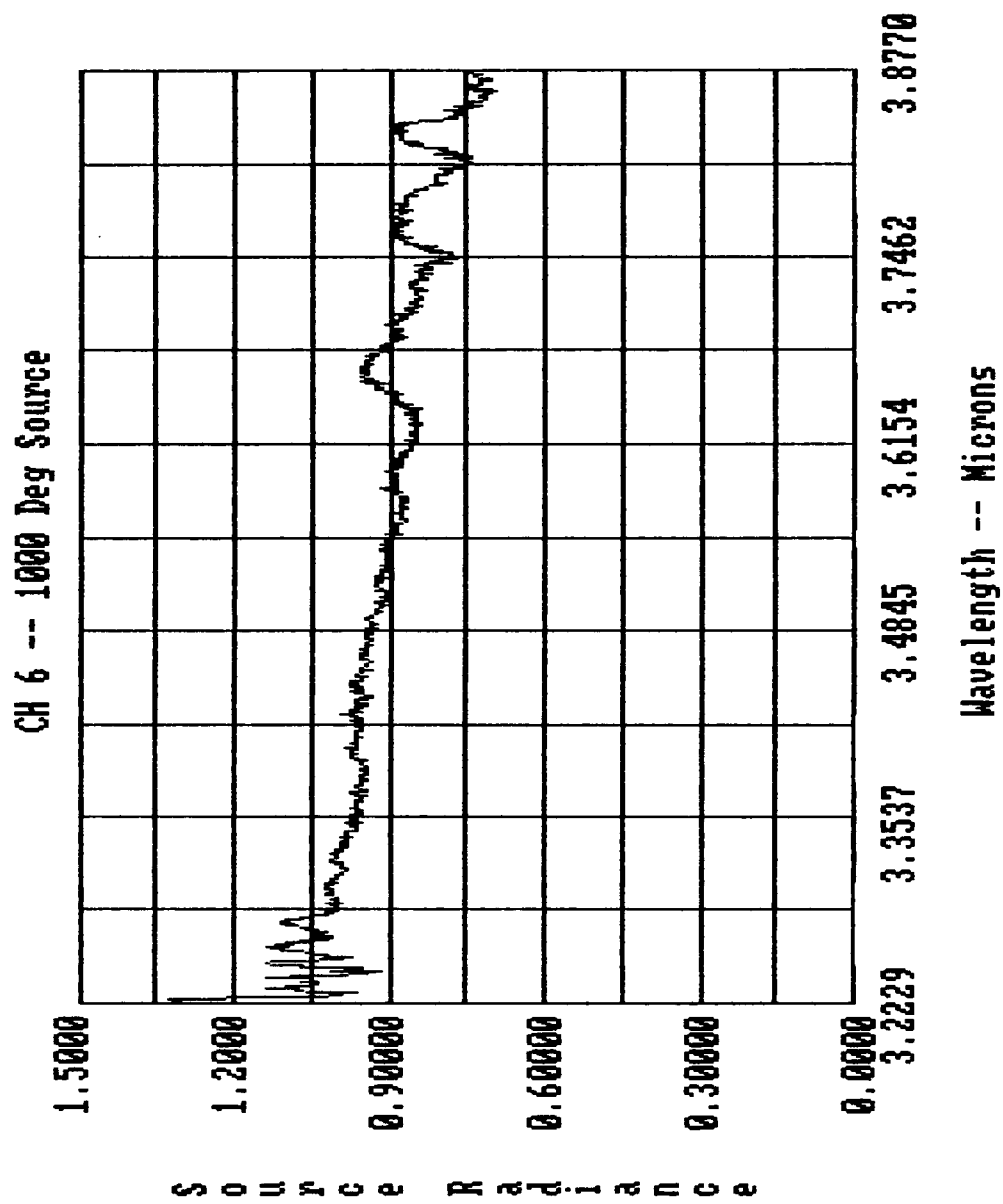


Figure B.6. Source Radiance for Channel 6.

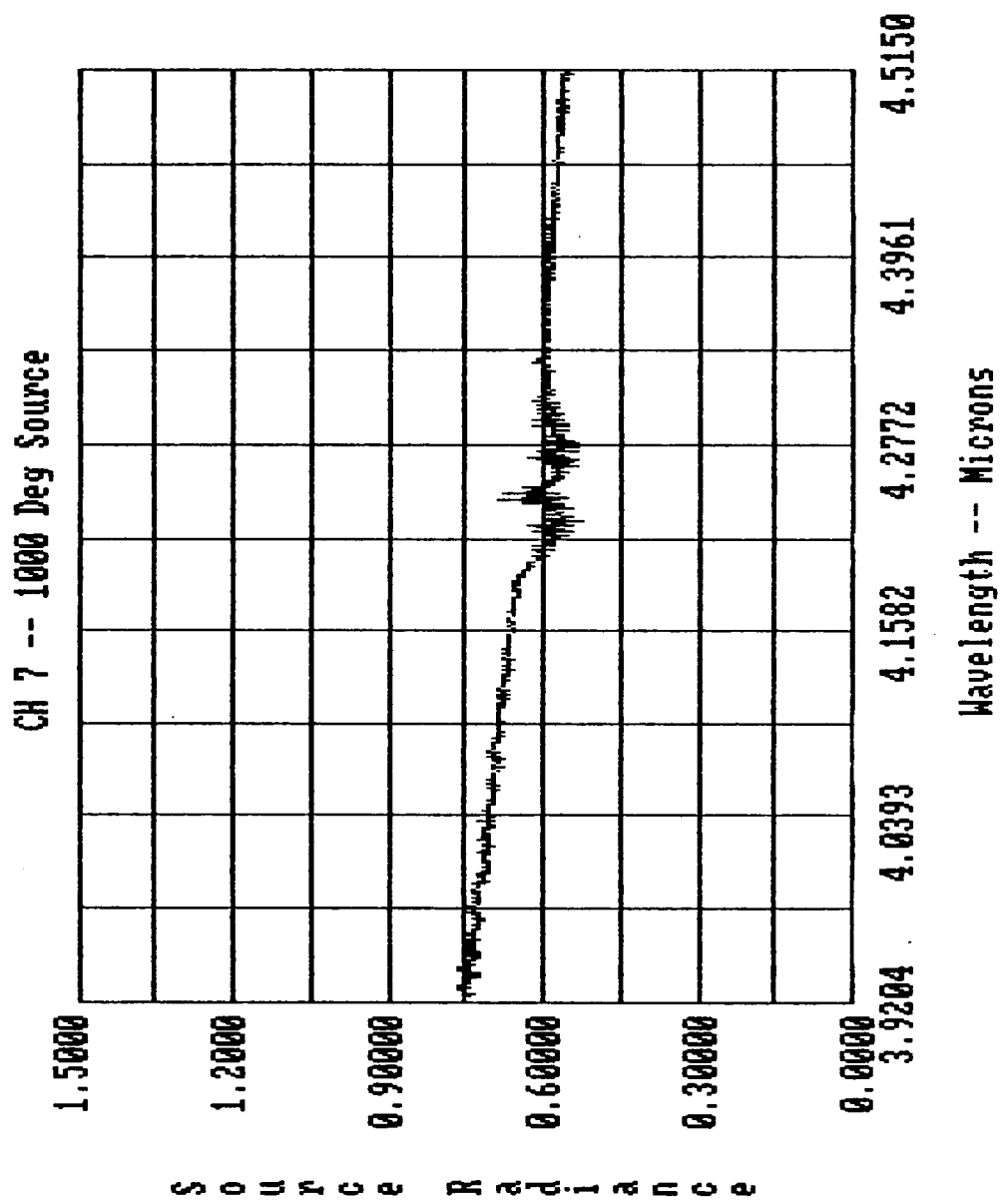


Figure B.7. Source Radiance for Channel 7.

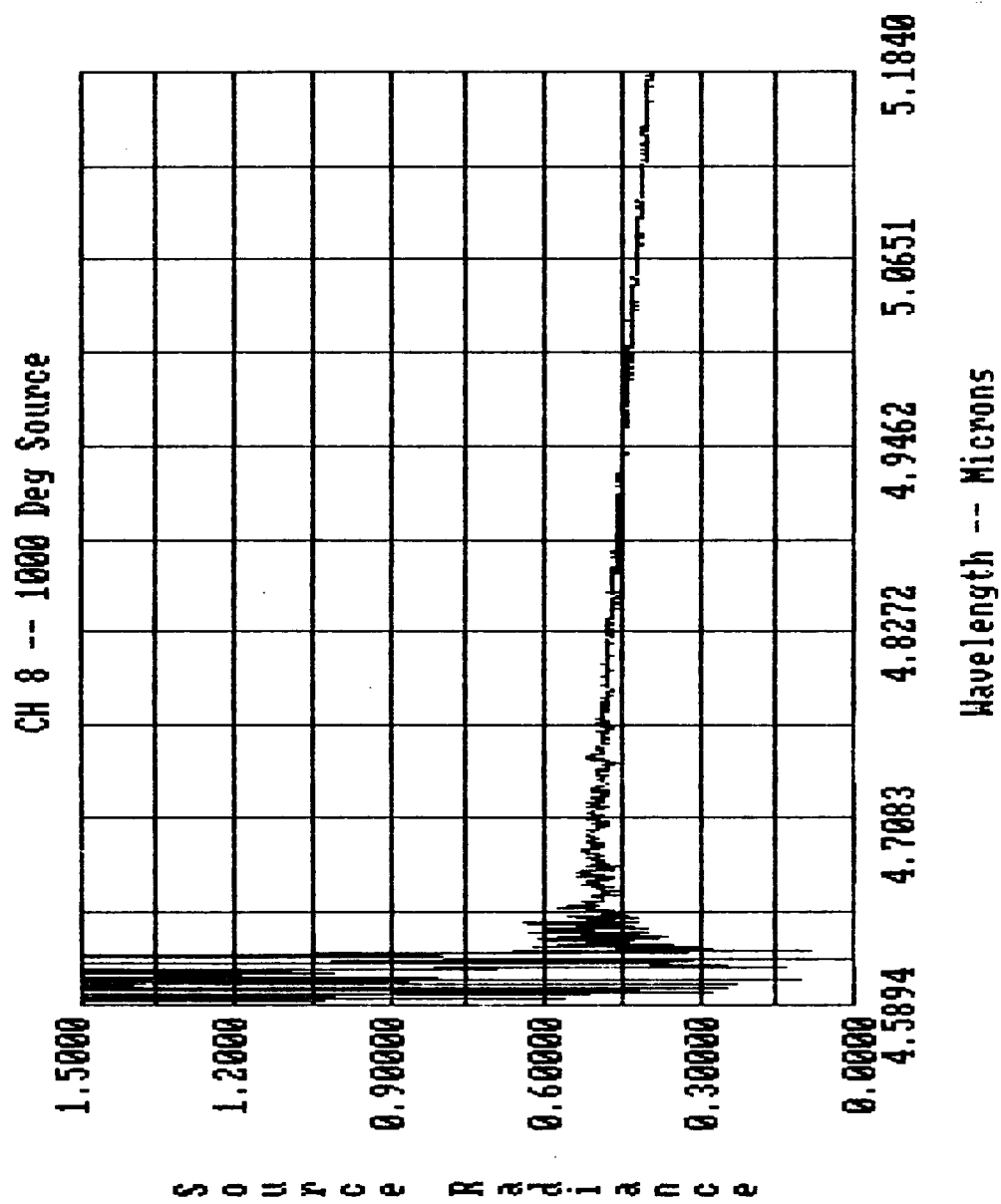


Figure B.8. Source Radiance for Channel 8.

VITA

Fred G. Sherrell, son of Mr. and Mrs. W. J. Sherrell, was born July 18, 1934, at Manchester, Tennessee. He attended Manchester City School and graduated from Coffee County Central High School in 1952. He joined the United States Air Force in 1952 and took training in electronic circuit theory and radar systems maintenance. He was assigned to the Strategic Air Command (SAC) during the period from 1953 to 1956. He worked at various SAC bases as an electronic service technician for B-47, bomb systems radar. After discharge from the Air Force, he attended The University of Tennessee in Knoxville. He graduated in June 1960 with a Bachelor of Science degree in Engineering Physics.

From June 1960 until the present, he has worked for Sverdrup Technology, Inc., at the Arnold Engineering Development Center, Arnold Air Force Base, Tennessee. He has worked at instrumentation engineering in several technical areas and is a registered electrical engineer in the state of Tennessee. In the period from 1963 to 1969, he attended The University of Tennessee Space Institute on a part-time basis, and graduated in 1969 with a Master of Science degree in Engineering Science. His area of interest at that time was radiation heat transfer.

In the period from 1981 to 1988 he again attended The University of Tennessee Space Institute part-time to earn a Master of Science degree in Computer Science.

He is married to the former Nina Louise Rogers of Manchester, Tennessee. They have three grown children, Roger, Janine, and Elise, and one granddaughter, Mallory Brink. The second grandchild is expected in the fall of 1988.