INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University
Microfilms
International
300 N. Zeeb Road
Ann Arbor, MI 48106

Automatic control and data analysis of a multi-channel millimeter wave radiometer

Zielinskie, David Alphonse, M.S.

The University of Arizona, 1988

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

PLEASE NOTE:

In all cases this material has been filmed in the best possible way from the available copy. Problems encountered with this document have been identified here with a check mark _✓_.

1. Glossy photographs or pages _____
2. Colored illustrations, paper or print _____
3. Photographs with dark background _____
4. Illustrations are poor copy _____
5. Pages with black marks, not original copy _✓_
6. Print shows through as there is text on both sides of page _____
7. Indistinct, broken or small print on several pages _✓_
8. Print exceeds margin requirements _____
9. Tightly bound copy with print lost in spine _____
10. Computer printout pages with indistinct print _____
11. Page(s) _____ lacking when material received, and not available from school or author.
12. Page(s) _____ seem to be missing in numbering only as text follows.
13. Two pages numbered _____. Text follows.
14. Curling and wrinkled pages _____
15. Dissertation contains pages with print at a slant, filmed as received _____
16. Other _____

UMI

AUTOMATIC CONTROL AND DATA ANALYSIS
OF A
MULTI-CHANNEL MILLIMETER WAVE RADIOMETER

by

David Alphonse Zielinskie

A Thesis Submitted to the Faculty of the

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE
WITH A MAJOR IN ELECTRICAL ENGINEERING

In the Graduate College

THE UNIVERSITY OF ARIZONA

1988

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment or requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED:

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

John A. Reagan
J. A. REAGAN
Professor of Electrical and
Computer Engineering

July 26, 1988
Date

DEDICATION

To my parents,

Al and Rita Zielinskie

ACKNOWLEDGMENTS

The author wishes to thank Bruce A. Bachoffer for introducing to him the field of Electrical Engineering, and for the encouragement that he provided. Special thanks are especially due to Dr. John A. Reagan, he took a bewildered graduate student and provided him with the opportunity to take an interesting and challenging project from conception to realization. In addition, Dr. Reagan helped in ways too numerous to mention, I am indebted to him for all of his support and guidance. Thanks go to Barbara A. Boscoe for her help in taking hand sketched figures and transforming them into the superb illustrations appearing in this thesis. In addition, Barb provided a sympathetic ear when things went haywire. I wish to thank my management and friends at Hughes Aircraft Company for allowing me the opportunity to attend the University of Arizona. Thanks especially to Gary W. Anderson, Don J. Strittmatter, R. Ward Stevens, Jeff S. Supp and Mike A. Krause, who provided encouragement and also put up with my strange work hours.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS--Continued

ABSTRACT

This thesis describes a multi-channel millimeter wave radiometer control system that will be used for atmospheric water vapor and temperature profile retrievals. The system consists of four subsystems which provide a total of nine frequency channels, a steerable reflector to permit slant-path measurements at different zenith angles, and a host computer for analyzing the data. The nine channels span the water vapor and oxygen absorption lines in the 20 to 60 GHz range. A distributed processing architecture is implemented to control the system. Each of the subsystems employs a signal processor and a microcontroller, which are configurable from the host. The signal processor filters the receiver's output, while the microcontroller oversees the radiometer, accepts data from the signal processor and communicates with the host. The host executes a custom shell that allows it to concurrently accept data from the subsystems, position the reflector and execute user analysis programs.

CHAPTER 1

INTRODUCTION

The goal of remote sensing is to extract information about an entity without being in physical contact with it or without being in close proximity to it. The science or art of remote sensing deals with measuring and analyzing the interactions between an object and its surroundings. This information is gathered by studying how the object influences its environment. These influences may include measuring changes in electromagnetic, acoustical or gravitational fields, observing acoustic waves caused by the object's presence, or by receiving the electromagnetic energy emitted or scattered from the object (Elachi 1987).

Radiometry is one example of remote sensing measurements; it deals with the analysis of electromagnetic or acoustic radiation. When pertaining to meteorology, passive radiometry is the measurement of emitted radiation from the atmosphere. These emissions are caused by the interaction between solar radiation and the atmosphere. More precisely, atmospheric gases absorb radiation at specific frequencies and then, after undergoing energy state changes, re-radiate this excess energy as infrared and microwave radiation, which can be picked up by a receiver. These interactions occur at frequencies where the atmospheric gases are at resonance. Two primary gases involved with atmospheric absorption are oxygen and water vapor. Figure 1 shows the

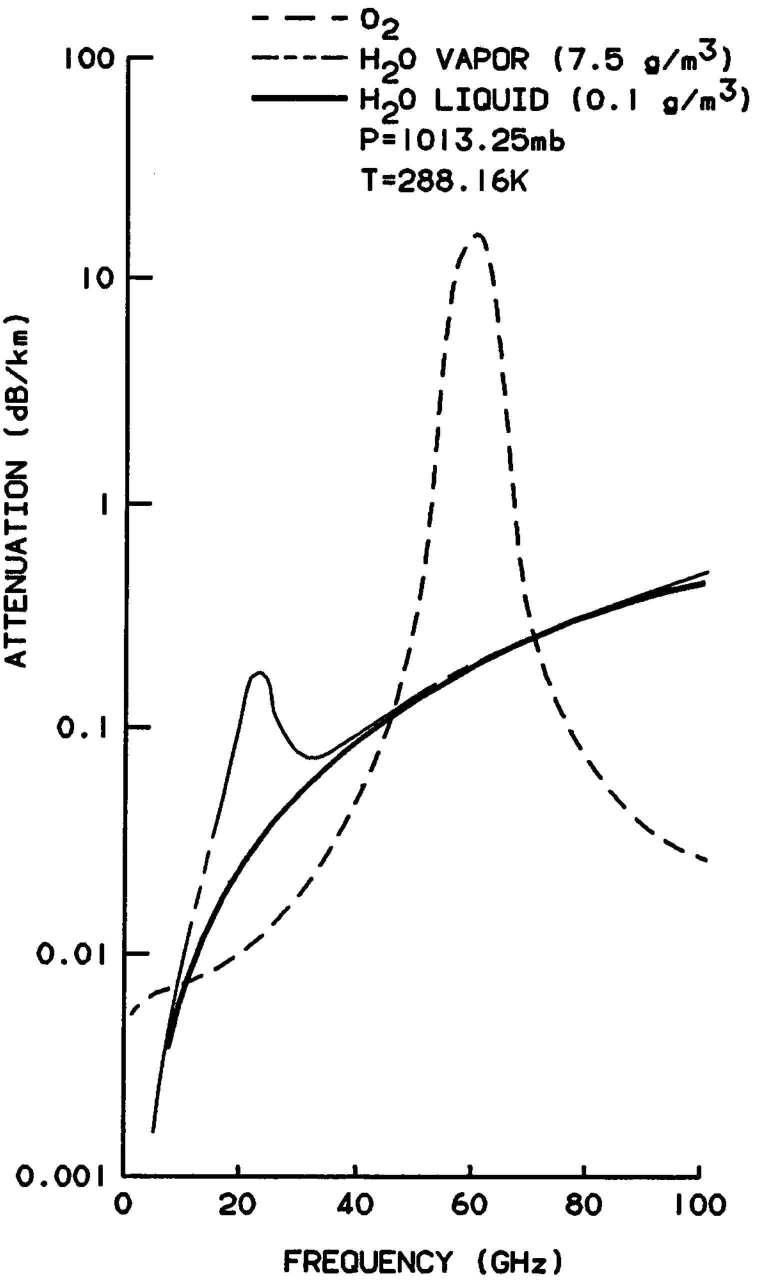


Fig. 1 Atmospheric Absorption of Oxygen and Water Vapor

atmospheric absorption curve for these gases in the microwave region (Blaskovic, Box and Rogers 1987). Note that oxygen absorption peaks at 60 GHz and that water vapor peaks at 22 GHz.

The objective of atmospheric microwave remote sensing is to determine the emitted radiance of the atmosphere, from which the atmospheric brightness temperature may be obtained. The radiation emitted by the atmosphere is described by Planck's law. This characterization may be approximated by the Rayleigh-Jeans law for frequencies below about 100 GHz, in which case the radiance is linearly related to radiative temperature or brightness temperature (Elachi 1987). This brightness temperature is used in inversion schemes which attempt to retrieve profiles of atmospheric temperature and water vapor concentration from brightness temperature measurements at various frequencies.

The use of radiometry to measure atmospheric parameters was first conceived by Dicke (Blaskovic and others 1987). He drew upon two diverse theories, that of thermal radiation and Johnson noise of resistors, to determine atmospheric brightness temperatures. Consider a system in thermal equilibrium comprised of an antenna connected to a resistor by a perfect conductor (Fig. 2). In this system the antenna collects radiation which in turn is absorbed by the resistor. Since the antenna-resistor system exists in thermal equilibrium, all of the heat generated by the resistor must be radiated by the antenna. Thus, the noise power gathered by the antenna must be equivalent to the noise power in the resistor.

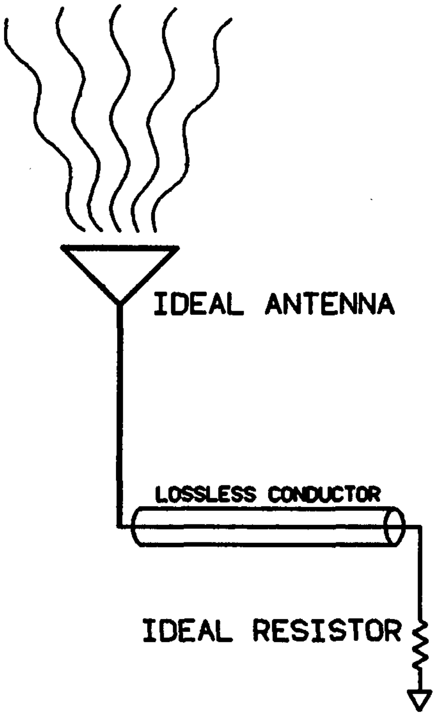


Fig. 2 System in Thermal Equilibrium

In general, the noise power per unit bandwidth, Pn, at the terminals of the resistor is described by Nyquist's theorem given by, $Pn = k\,T$ (watts per Hertz), where k is Boltzmann's constant and T is the absolute temperature of the resistor (Blaskovic and others 1987). We can define the radiation received by the antenna in terms of antenna temperature, Ta, which can be substituted for T in the above equation.

The antenna temperature contains the value of the brightness temperature. Unfortunately, the antenna temperature also contains other effects such as cosmic and background radiation, in addition to the self-generated system noise. These effects must be eliminated before an accurate determination of the brightness temperature can be made. Once the brightness temperature is known, further processing will be employed to extract atmospheric temperature and water vapor profiles.

Objectives

The goal of the project described in this thesis was to construct an adaptable controller for a radiometer system. The controller had to be able to manage the data collection task of the radiometers. The system also had to be easily adaptable as the data and knowledge bases matured. Special attention was given to ease-of-use and system flexibility, two typically mutually exclusive choices in system construction. The following guidelines were established in the design of the system:

a. symmetrical hardware and software design,
b. adaptable hardware and software,
c. user friendliness,
d. automatic setup and operation.

Strategy

The following chapters develop various aspects of the specific radiometer system that was proposed and constructed. The Perspective chapter describes the entire system starting with the incoming radiation and ends with storage of the brightness temperature data. Emphasis is placed on data processing and radiometer control. Topics such as microwave reception, system sensitivities, and system calibration procedures are covered in another thesis devoted to these types of problems.

The ensuing four chapters describe, in detail, hardware and software elements for the sub-systems of the radiometer processing and control system. Following this, a detailed explanation is given of how the three sub-systems interact to form the complete system, including the results of a modeling simulation of the system. The conclusion follows which summarizes the systems characteristics. The appendices contain a complete set of schematics; details on the custom hardware; JAIL, the radiometer interface language; and the results of a hardware/software simulation of the system.

CHAPTER 2

PERSPECTIVE

This thesis describes a multi-channel millimeter wave radiometer system that has been developed at the University of Arizona in conjunction with a joint research program with Pennsylvania State University (Thomson 1988). The system consists of four microwave radiometer subsystems which provide a total of nine frequency channels, a steerable reflector to permit slant-path measurements at different zenith angles, and a host computer for recording and evaluating the data. The radiome-ters employ three-way Dicke switching including hot and warm reference loads for calibration and gain control. The nine channels span the water vapor and oxygen absorption lines in the 20 to 60 GHz range with the lowest frequencies (20.6, 22.235 and 31.65 GHz) to be used for water vapor and liquid water retrievals and the six highest frequencies (50.3, 52.85, 53.85, 55.45, 58.8 and 61.15 GHz) to be used for temperature profile retrievals. The zenith angle scanning capability is for the purpose of both calibration, tipping curve procedure, and the study of temporal/spatial variations in atmospheric water vapor and temperature.

The conceived system is composed of a central computer that manages the four radiometers and controls a steerable reflector as shown in Figure 3. The steerable reflector is used to select the

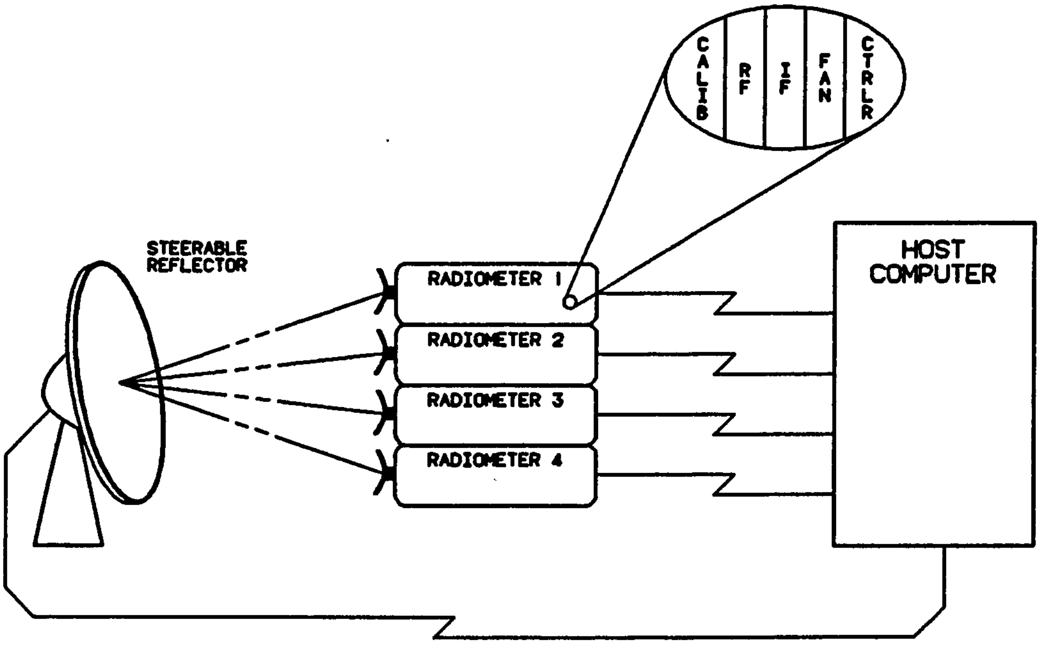


Fig. 3 System Block Diagram

region of the atmosphere that the radiometer receives signals from. Each of the radiometers is comprised of the following equipment:

1. receiving antenna,
2. calibration assembly,
3. RF/Downconverter section,
4. IF section,
5. imbedded micro-controller.

Items one to four were purchased from MILLITECH, INC of Merrifield, MA. Item five is made up of both custom circuit cards and cards purchased from THE MICROMINT, INC of Vernon, CT.

The choice of a receiving antenna is a critical part of the radiometer design. It must be chosen to provide high gain, narrow beam width, and low side-lobe levels. This is necessary to ensure that only energy in the volume of interest is received and converted, thereby insuring measurement validity.

The antenna chosen for the system is a corrugated conical waveguide horn. This type of antenna is designed to produce high gains and low side-lobe levels. To further improve directivity and gain, and decrease the side lobes, a dielectric Gaussian, lens matched to the feed horn, is also included in the antenna assembly.

The calibration assembly employs the Dicke switching technique to calibrate and condition the incoming signal. This method relies on the linear relationship between noise power and temperature. In Dicke switching, the incoming radiation is compared to signals from the two reference loads. The reference sources are absorptive loads maintained at constant, elevated temperatures.

Each of the radiometers has three controllable circulators for the purpose of selecting which of three possible sources enter the RF section. The three sources are the sky, hot load, and warm load. By alternating among these sources at a specific rate, the Dicke switching technique can be employed to deduce the sky brightness temperature.

The RF section employs a superheterodyning technique to down-convert the microwave signal to an easier handled IF signal. A double balanced/low conversion loss mixer performs the frequency conversion. Multiple local oscillator sources can be switched in and out to convert signals at different frequencies using only one receiver. GUNN oscillators are used as the local oscillator sources because of their extremely stable operation.

The IF section performs three important functions. It amplifies the signal from the RF section, performs filtering of the resultant signal, and then converts the signal to a DC level. The filtering element is a bandpass filter with cut-off frequencies of 50 and 200 MHz. A square law detector is used to sense the IF signal and convert it to a DC voltage that is proportional to the power received by the detector.

An error introduced by the square law detector is commonly called one-over-f-noise, written as 1/f noise (Manassewitsch 1980). This noise looks like a slowly varying sinusoidal wave which is added to the converted DC signal. For measurements to be accurate over long times, this 1/f noise must be eliminated before the brightness temperature can be extracted.

An imbedded micro-controller, the Intermediate Data Collector-IDC, interfaces each radiometer to the host computer (Fig. 4). It is made up of a general processor, an Intel 8052, and a digital signal processor, a Texas Instruments TMS 32010. The 8052 handles communications with the host, measures internal temperatures, and collects brightness temperatures from the DSP. The DSP interfaces to the detector and performs the signal conditioning.

Hardware Components

The system is comprised of a host computer which manages four radiometer units and controls a pointable reflector; the host also stores and analyzes the captured data. Each of the units is a stand alone radiometer with a digital signal processor, DSP, and a single board computer, SBC. Together the DSP and SBC form the intermediate data collector, IDC. The radiometers have identical hardware and software with the difference between units being the frequency range that each radiometer operates over and the number of specific frequencies, one to three, that each receiver is designed to handle.

Host Computer

The host is able to accept and store data from the four radiometers, control the reflector, and analyze the accumulated data. The host is an IBM PC/AT equivalent computer with four serial ports, motor controller, 640k of RAM memory and a 30 Mb hard disk (Fig. 5). Each of the radiometers is attached to a separate serial port on the

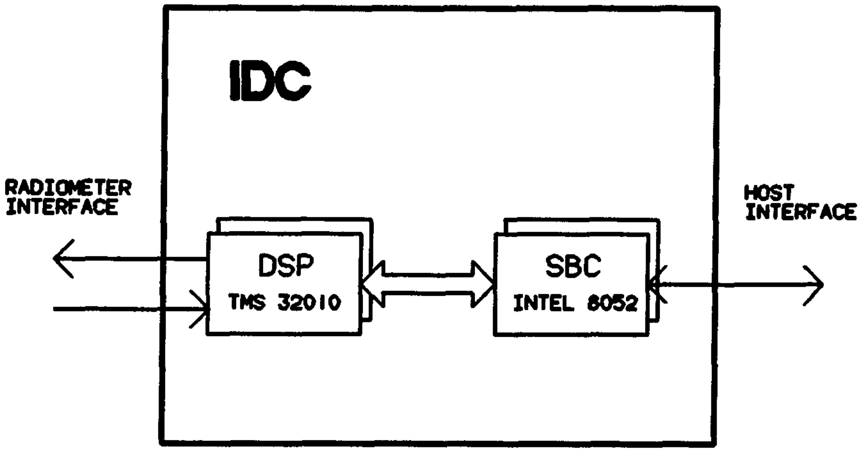


Fig. 4 Intermediate Data Collector

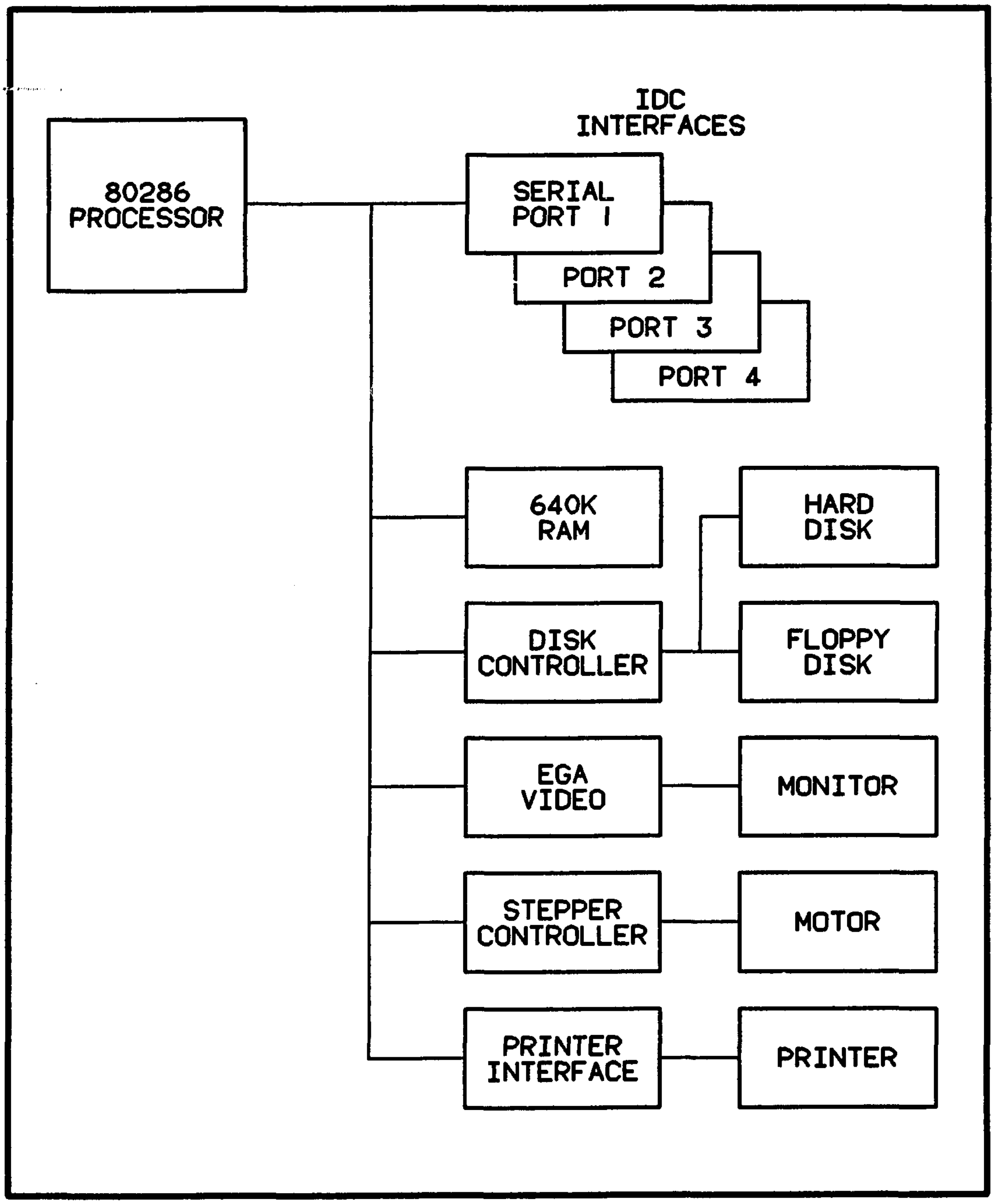


Fig. 5 Host Computer

host. All data from the radiometers is transmitted to the host over a 9600 baud serial link. The reflector is governed by a smart stepper motor controller that accepts position commands from the host and then performs the desired action.

Signal Processing

The digital signal processor, DSP, is used to remove the 1/f noise characteristic of the receiver and average the signals used to calculate the brightness temperature. The heart of the DSP is the TMS32010 signal processor from Texas Instruments. The TMS32010 is able to perform the data reduction from the receiver in real time. The DSP also manipulates the two calibration loads that are used in the Dicke switching. In addition, on radiometers with multiple receiver channels, the DSP selects the frequency of the receiver (Fig. 6).

Communication with the SBC, by the DSP, is accomplished through the IDC's bus using a large scale integrated circuit, LSI, register/buffer. Data on the DSP side of the buffer is as a sixteen bit word, while the SBC side is organized as eight bit words.

The memory in the DSP is all RAM, which means that the host must load a control program prior to operation. Hardware has been added to ensure that the DSP comes up in a reset state when power is first applied, thus preventing erratic operation. This scheme was chosen because it allows modification of the program without opening up the unit and replacing the PROMs. This enables the DSP's role in the system to be entirely redefined, even when operated in a remote location.

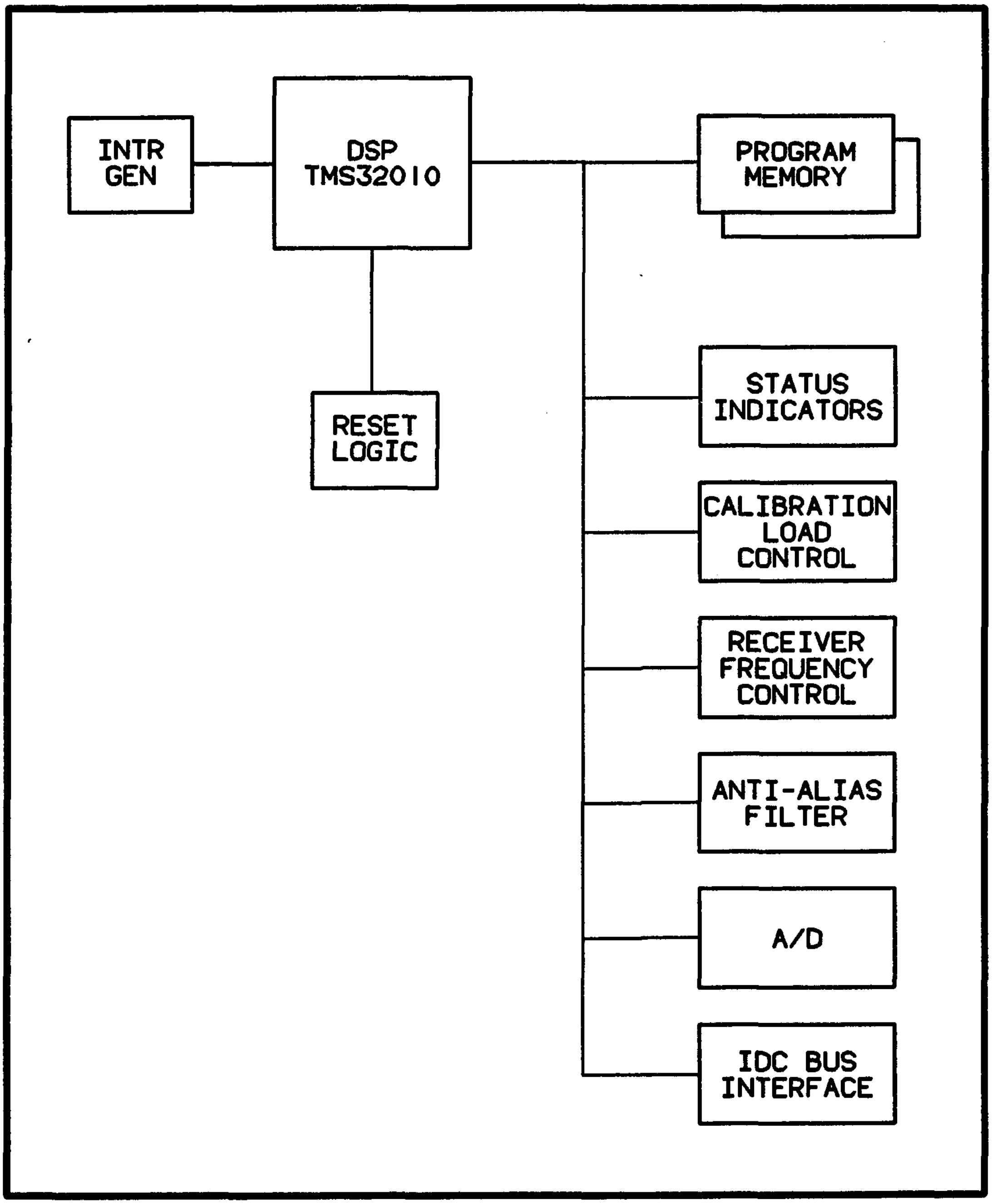


Fig. 6 Signal Processing

Interface with the radiometer is made using an analog to digital, A/D, converter. An analog anti-alias filter and sample and hold amplifier make up the pre-filtering circuit of the raw radiometer video data. The A/D has twelve bits of resolution plus a sign bit. The inputs are able to convert a plus five volt to minus five volt signal, with a full twelve bits of resolution; more precisely, the A/D converter has a resolution of 1.2207 millivolts per bit.

Four software addressable latched status LEDs provide indication of program status, or during built-in-test, act as a diagnostic aid. Four other latched bits are available for additional process control or status monitoring.

General Processing

The single board computer, SBC, is used to monitor and control various radiometer parameters, download the DSP control program, accept data from the DSP, further process the DSP data, communicate with the host and control a heat exchanger. The SBC is comprised of a general purpose computer and a sixteen channel A/D converter (Fig. 7).

The SBC is a purchased board with an Intel 8052-BASIC microcontroller. The board has a twenty four bit peripheral interface register, 32 Kb of RAM, 16 Kb of EPROM and two serial ports. The board executes a subset of the BASIC language which has been tailored to real-time process control.

The A/D board is used to monitor temperatures and voltages in the radiometer. Several thermocouples are placed inside the radiometer for use in controlling the heat exchanger. Any voltage or temperature

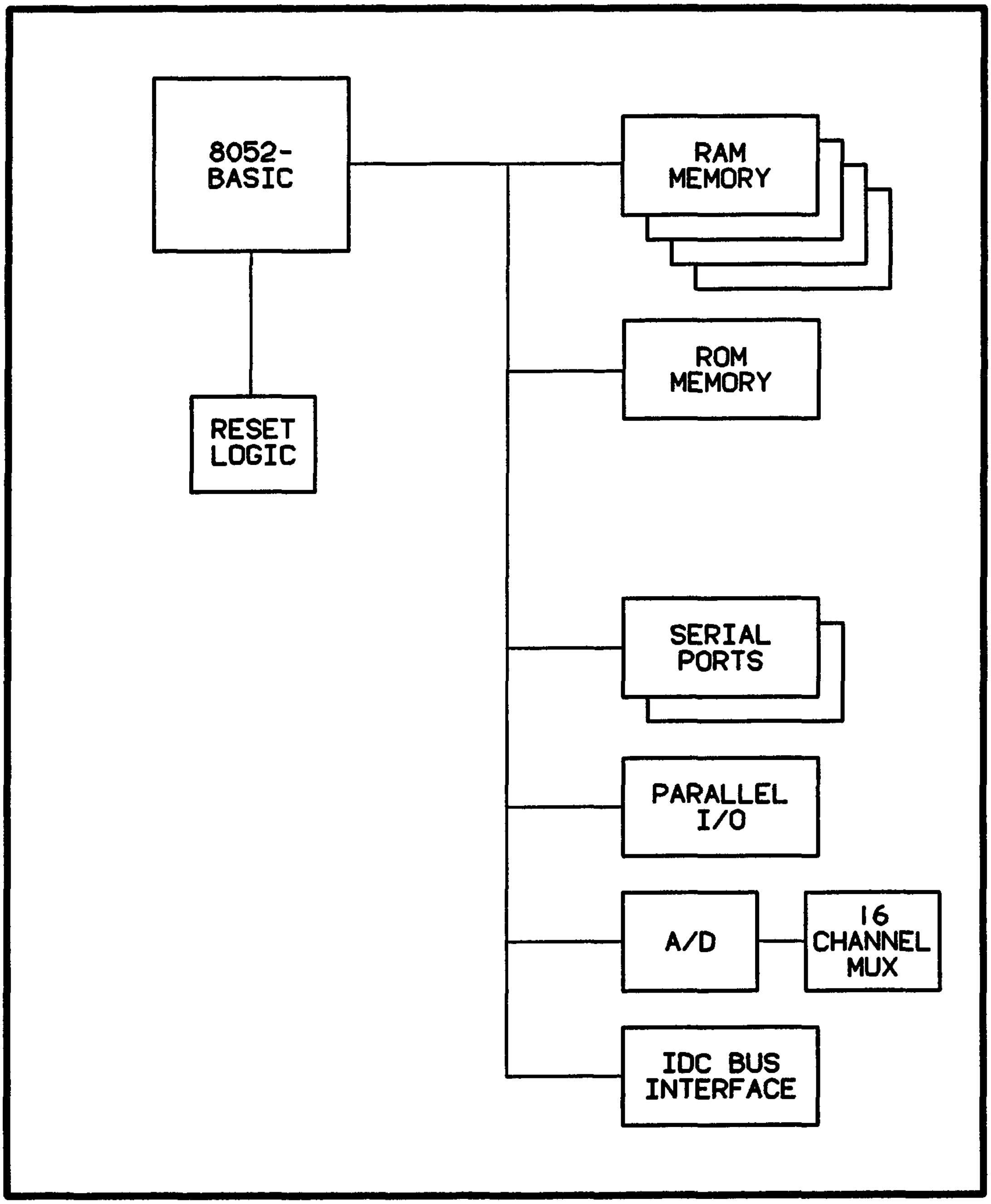


Fig. 7 General Processing

which is out of tolerance will cause the host to be alerted and to take appropriate action.

Communications with the DSP is accomplished using eight predefined addresses. Manipulating these various addresses controls the different states of the DSP, such as reading data or downloading a new program. Communications with the host are made using one of the serial ports. Messages contain various news items ranging from system health to processed radiometer data.

Software Components

The software components are the means through which the hardware is controlled. The software for the host, SBC and DSP were written to utilize the full potential of the hardware. This was accomplished by allowing customization of the hardware via the software. All aspects of the system can be re-programmed under host control.

Host Computer

PC based software has been developed to permit quasi real-time retrieval of atmospheric brightness temperature. These data may be used later to create a data/knowledge base that could be used to modify the system operation in order to optimize the retrieval of significant temporal/spatial features. There are two distinct separations in the host's software function, they are application programs and radiometer control (Fig 8). The radiometer control is split into two areas, communications processing and reflector positioning. Both of these routines are interrupt driven, meaning that they will remain idle until

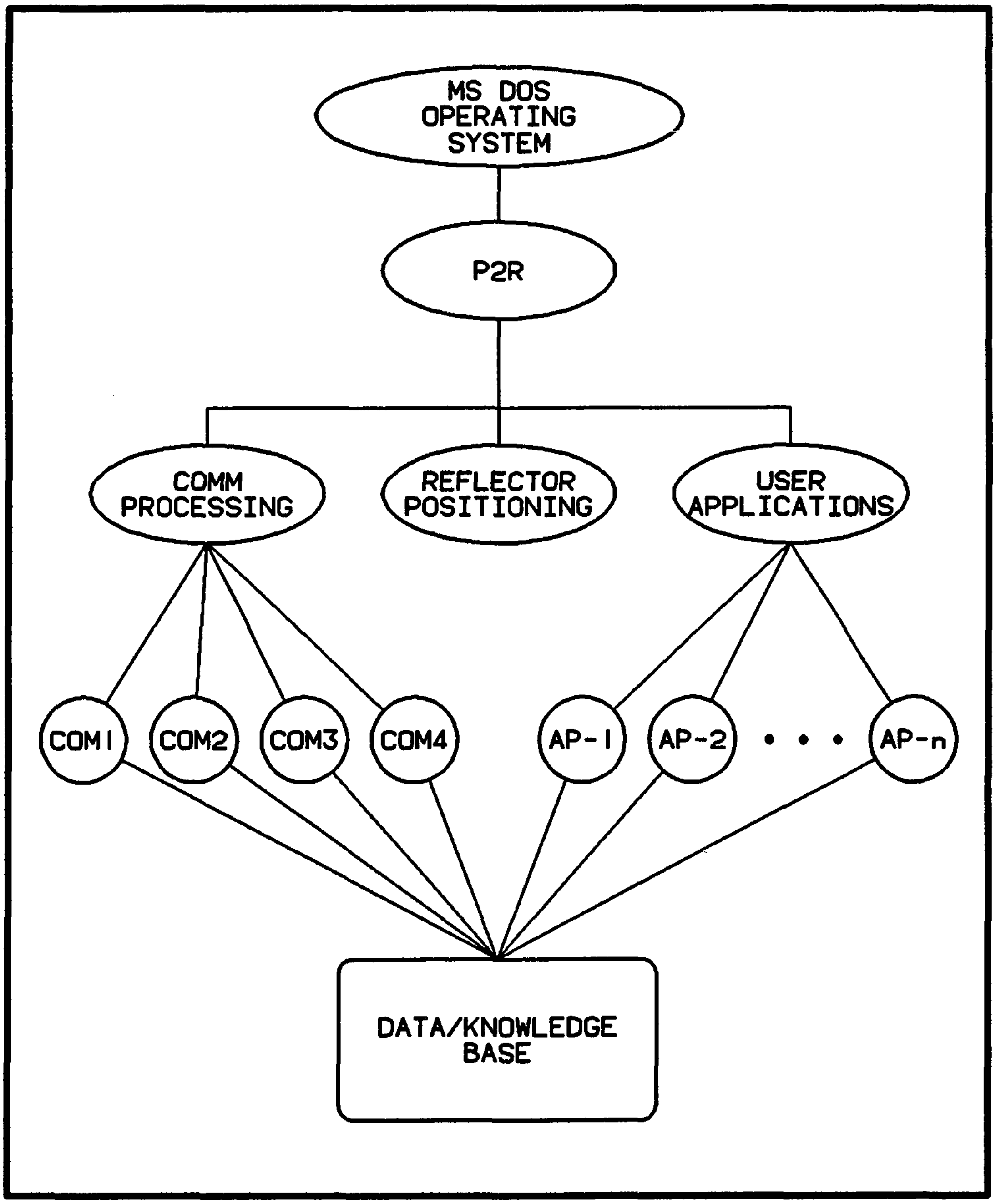


Fig. 8 Host Software Diagram

activated by an external event. Once activated, they interrupt the currently active task and preform a service, reading data from a radiometer or positioning the reflector, before returning control to the pre-empted task. The user application routines are stand alone programs that are executed under the P2R program and perform operations on the data base. The majority of the software for the program is written in Microsoft C version 5.1, with the remainder written in Microsoft Macro-assembler version 5.

The communications processor task is activated whenever a communications interrupt is generated, each time data is received from a radiometer. When operative, this task accepts the data, stores it in a buffer, then checks the buffer's length. If the buffer is eighty percent full, it is written to hard disk and then reset. This feature was not part of the operating system, so it had to be added, and, because this is a very time critical piece of software, assembly language was chosen.

Like the communications processing task, the reflector positioner task is interrupt driven and it is activated by the system clock interrupt. After a predefined time, the reflector is moved to a new position. The time interval and the movement increment are read from a configuration file during the initialization of P2R.

User application programs can be any MS-DOS compatible program. After installing the communications and reflector positioner interrupt service routines, P2R is ready to perform analysis operations on the data base. The analysis operations are accomplished with user supplied programs which are spawned or called by P2R, under user direction. The names of these programs are stored in the configuration file.

The host software was written to be extremely adaptable and easily modifiable. This system is a research tool, and, as such, not all of the situations that the system may find itself in have been defined, let alone solved. What has been done is to provide a modular, expandable framework that can be easily adapted to solve these situations as they arise.

SIGNAL PROCESSING

The DSP software must be able to perform the filtering, control the radiometric sources, select the desired receiver frequency, and communicate with the SBC. All software for the DSP is written in its native assembly language, which has been highly optimized for digital filtering applications, while still having the necessary instructions for the other tasks. The DSP software can be divided into two major parts; they are the event scheduler and the background task (Fig. 9).

The event scheduler executes in the foreground. The system time is read and compared to a master schedule and when a match is found, the appropriate subroutine is executed. The subroutines will handle the raw video filtering, detector source control, frequency selection, brightness temperature calculation, data reduction and communications. The interrupt task will read the A/D converter, when appropriate, and maintain the system clock.

The use of a scheduler was chosen to allow easy modification of the program. Each routine seems to execute until completion, even as the data from the A/D converter is read and processed. This approach

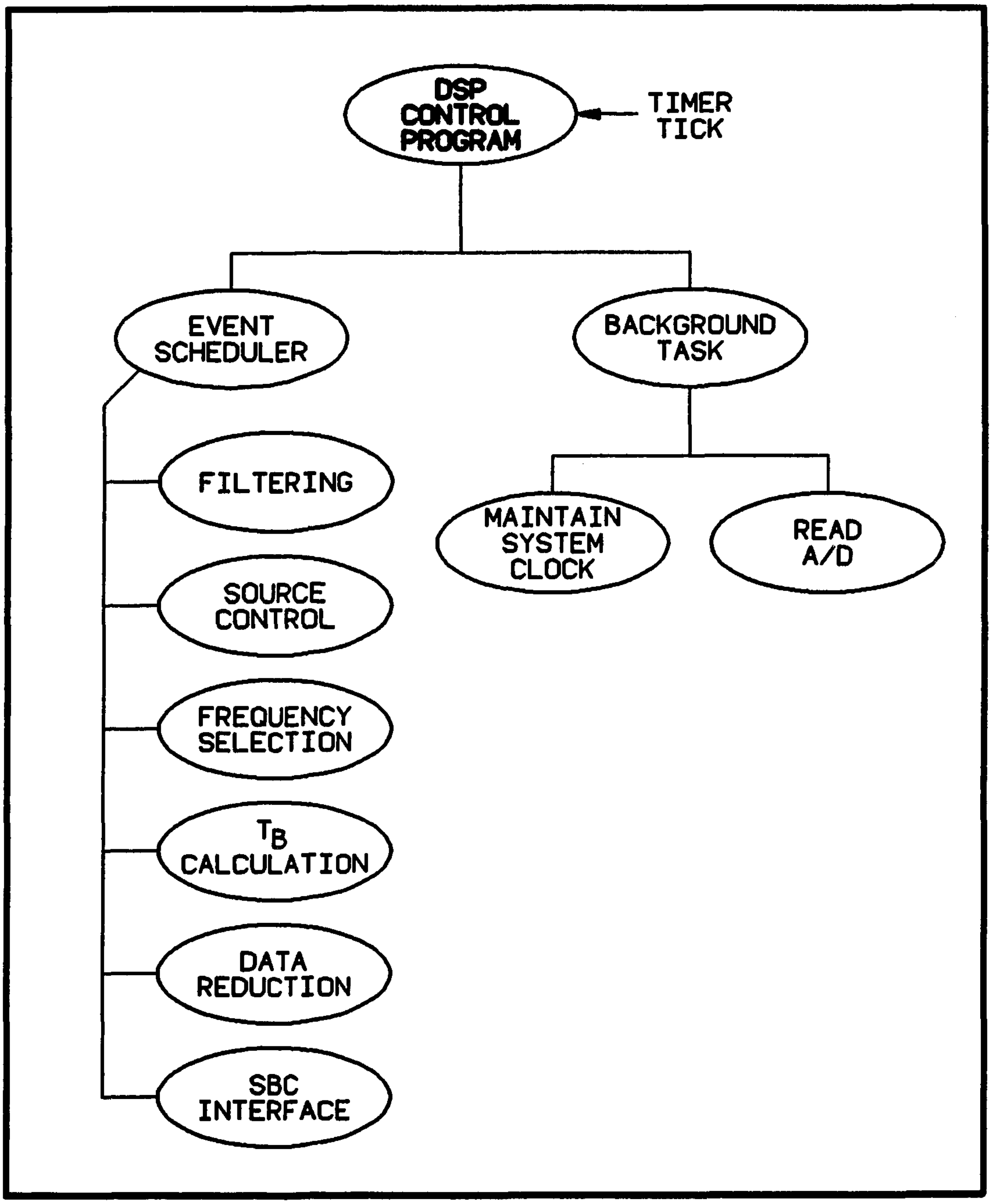


Fig. 9 DSP Software Diagram

allows persons inexperienced in real-time processing to modify the program and get valid results.

One design assumption made was that the program will not need to be modified often, if at all. The part that probably will change will be the filtering requirements. With this in mind, the filter coefficientsand sampling interval may be downloaded separately, without re-compiling the entire program. This modification is accomplished via the host computer and can be ordered remotely.

General Processing

The processor in the SBC, executes a subset of BASIC. The task of the SBC is highly specialized. In BASIC this detail is easily lost. Keeping with the outlined objectives, a specialized language was written. This language is called JAIL, for Just Another Interface Language. JAIL accepts English commands and translates them into the appropriate SBC actions, most of which are obscure references to a particular memory address. This natural language approach frees the user wishing to modify the SBC's operation from worrying about address references, permitting the user to concentrate on the system level application (Fig. 10).

The JAIL program executes in all of the SBCs. JAIL is divided into two sections, a command interpreter and a communications controller. The command interpreter reads commands from a personality module that was sent to the individual SBCs from the host. This module defines the specific requirements for the specific radiometers. The communications controller manages contact with the DSP and the host computers as well as maintains the system clock. Functions are split

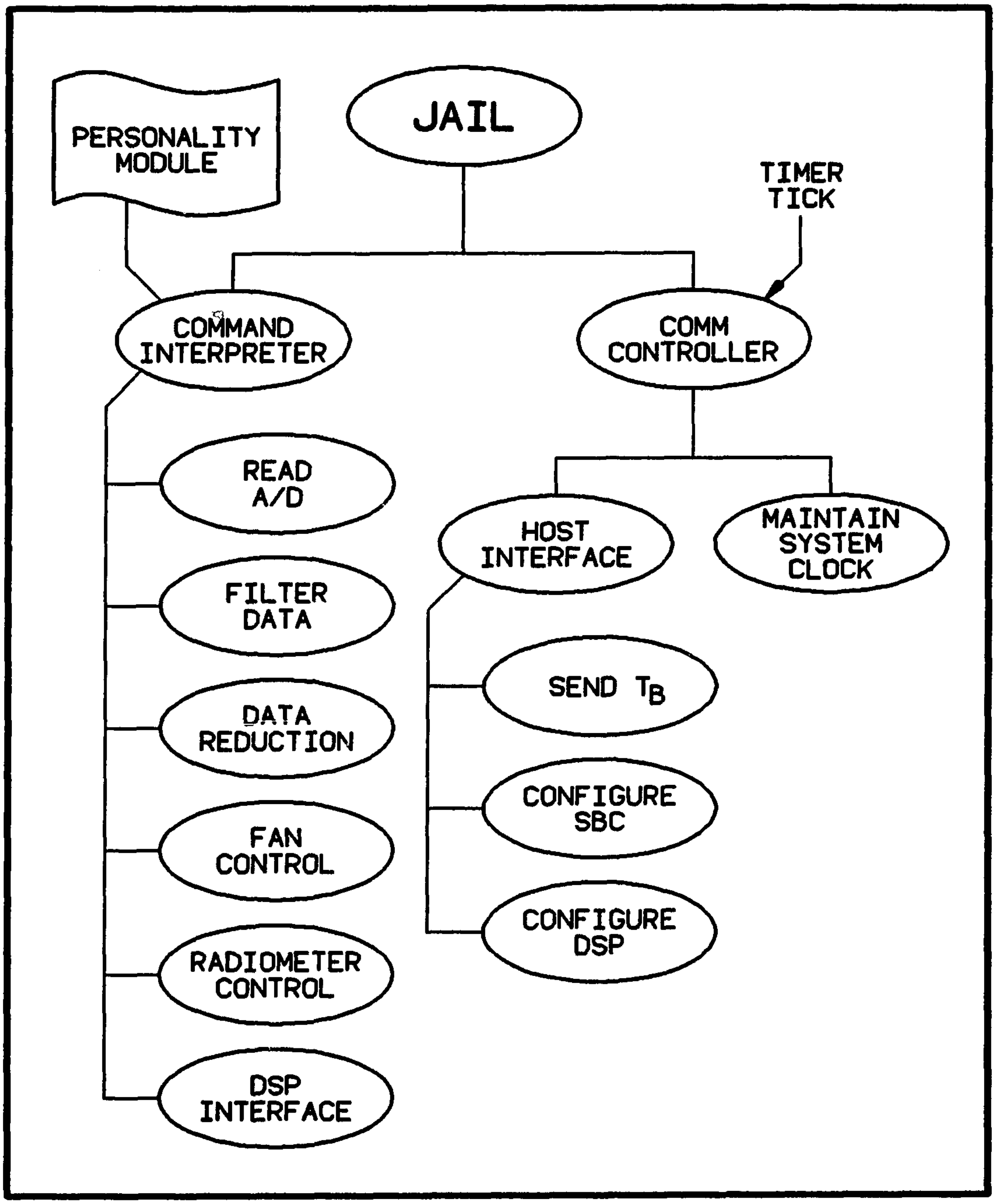


Fig. 10 SBC Operating Environment

in this manner to prevent a user personality module from locking out host communications. This is especially important when the unit is located in a remote location.

Design Philosophy

The goal of this project is to construct a flexible research tool for use in studying atmospheric phenomena. Ergonomics were given an extremely high priority because even if a tool is useful, if it is too difficult to use or too hard to change, it will soon become obsolete. With this in mind, several hardware/software design approaches were simulated prior to arriving at the present scheme.

The Network II simulation language was used to explore the various hardware/software scenarios without constructing hardware or software. Network II was able to focus on loading or utilization of the numerous hardware and software components that comprise the full system, prior to construction, to identify and correct any critical resource allocations. The results from these simulations were then used to resolve any conflicts that were uncovered. Also, the simulation documented the hardware, the software and their interactions.

CHAPTER 3

INTERMEDIATE DATA COLLECTION

The intermediate data collector, IDC, interfaces the host computer to the radiometer. The IDC is comprised of a single board computer, a sixteen channel analog interface board, the digital signal processor and a radiometer interface board. All of the boards reside on the IDC's backplane. Figure 11 describes the signals for the bus.

The IDC physically resides in the radiometer. The four boards are connected by means of a passive mother board and are housed in a card cage. There are four power forms available from the backplane which are +5V, -5V, +12V, and -12V. They are supplied by a switching power supply operating off of a standard 120 VAC power outlet.

The link with the host is by an RS-232C serial data port on the SBC. Interface with the radiometer is through an analog to digital converter on the radiometer interface board. The digital signal processor controls the receiver frequency of the radiometer and also selects the signal for downconversion by manipulating the circulators. Additional monitoring of the radiometer is performed by the single board computer in conjunction with the sixteen channel analog converter board. Together they are able to monitor the temperature inside the unit, using thermocouples, and to directly measure the various power forms located inside the unit. If the temperature in the unit is high the single board computer turns a fan on, and if the temperature remains high, the single board computer can shut down the radiometer. Likewise, if one or more of the power forms is out of tolerance, the radiometer can be shut down.

PIN	Description	PIN	Description
A	-12 Volts	1	+5 Volts
B	+12 Volts	2	Ground
C	Reset	3	No Connection
D	No Connection	4	Address/Data 3
E	Timer Zero Input	5	Address/Data 4
F	!RD	6	Address/Data 5
H	Timer One Input	7	Address/Data 6
J	!PSEN	8	Address/Data 7
K	Timer Two Input	9	Address/Data 0
L	Timer Two Trigger	10	Address/Data 1
M	TTL Serial Out	11	Address/Data 2
N	TTL Serial In	12	Pulse Width Modulation
P	!Int 0 - DMA Req	13	!Int 1
R	!DMA Acknowledge	14	!Program Enable
S	Address 15	15	!Program Pulse
T	Address 14	16	No Connection
U	Address 13	17	No Connection
V	Address 12	18	No Connection
W	Address 11	19	No Connection
X	Address 10	20	RD/!WR
Y	Address 9	21	!DS
Z	Address 8	22	!AS

Fig. 11 IDC Bus Diagram

CHAPTER 4

RADIOMETER INTERFACE

The digital processing hardware interfaces the radiometer to the data processing system. It performs the task of selecting the input source that will be downconverted, choosing the receiver frequency, sampling the raw analog data from the receiver and then calculating the brightness temperature. The digital processing hardware consists of two boards which reside in the IDC; these boards are the digital signal processor, DSP, and the radiometer interface board, RIB (Fig. 12). While these boards are housed in the IDC, they are interfaced using a ribbon cable. The DSP does use the IDC's bus for power and communications with the single board computer, while the radiometer interface board only uses the IDC's bus for power. The DSP controls the data acquisition hardware on the radiometer interface board. Once a value has been read, the input source is changed and read. This process repeats until the three input sources have been read, at which time a brightness temperature is calculated. After several brightness temperatures are calculated, they are averaged before passing them on to the SBC.

Before the computation of the brightness temperature can be accomplished, the raw data must first be conditioned due to noise added to the signal in the downconversion process. This noise is referred to

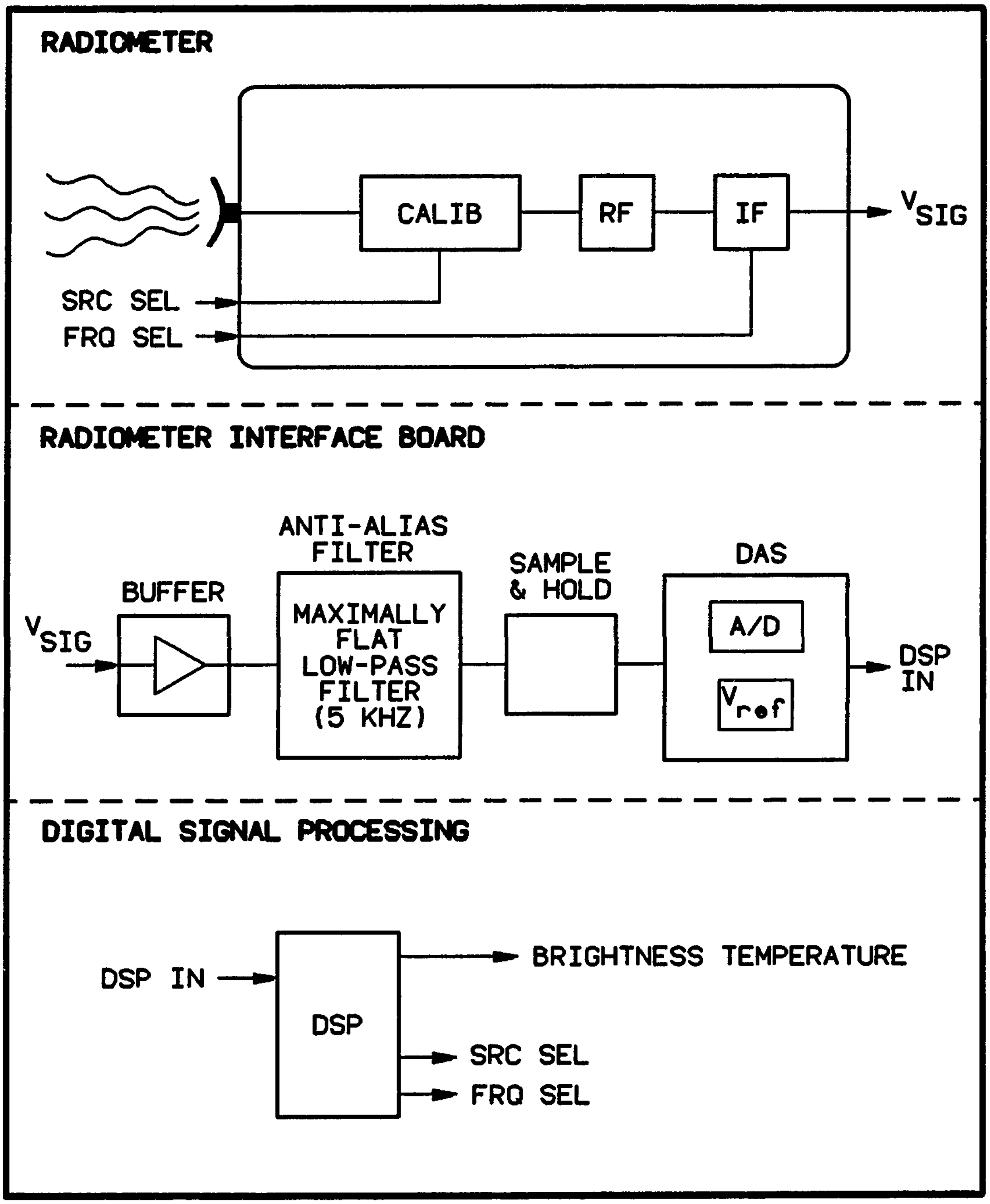


Fig. 12 Radiometer Signal Flow

as 1/f noise because of the characteristic fall off as frequency increases (Fig 13).

A digital filtering approach was chosen to condition the signal because of the ease at which the filter can be altered. The use of a digital filter requires that an analog filter be used to eliminate possible aliasing in the signal reconstruction process. The Nyquist theorem states, in order to recover a sinusoidal waveform, the sampling frequency must be twice as high as the highest frequency of interest (Oppenheim and Schafer 1975). The 1/f noise is not necessarily sinusoidal, so in order to recover the signal, the sampling frequency must be increased to account for the higher frequency components present in the waveform.

The signal processor reads the radiometer's output via the analog to digital converter and filters the signal. This is done for the hot load, the warm load and the sky. When all three sources are read and filtered, the brightness temperature can be calculated. The extraction of brightness temperature proceeds as outlined below.

The received signal out of the radiometer (Fig. 12) is of the form,

$$Vsig = K (Tsig + Tsys). \tag{4.1}$$

where Vsig is the received signal,
K is the system gain constant,
Tsig is the signal temperature,
Tsys is the basic system temperature.

Substituting the sky, warm load, and hot load for the signal yields,

$$Vsky = K (Tsky + Tsys), \tag{4.2}$$
$$Vwrm = K (Twrm + Tsys), \tag{4.3}$$
$$Vhot = K (Thot + Tsys). \tag{4.4}$$

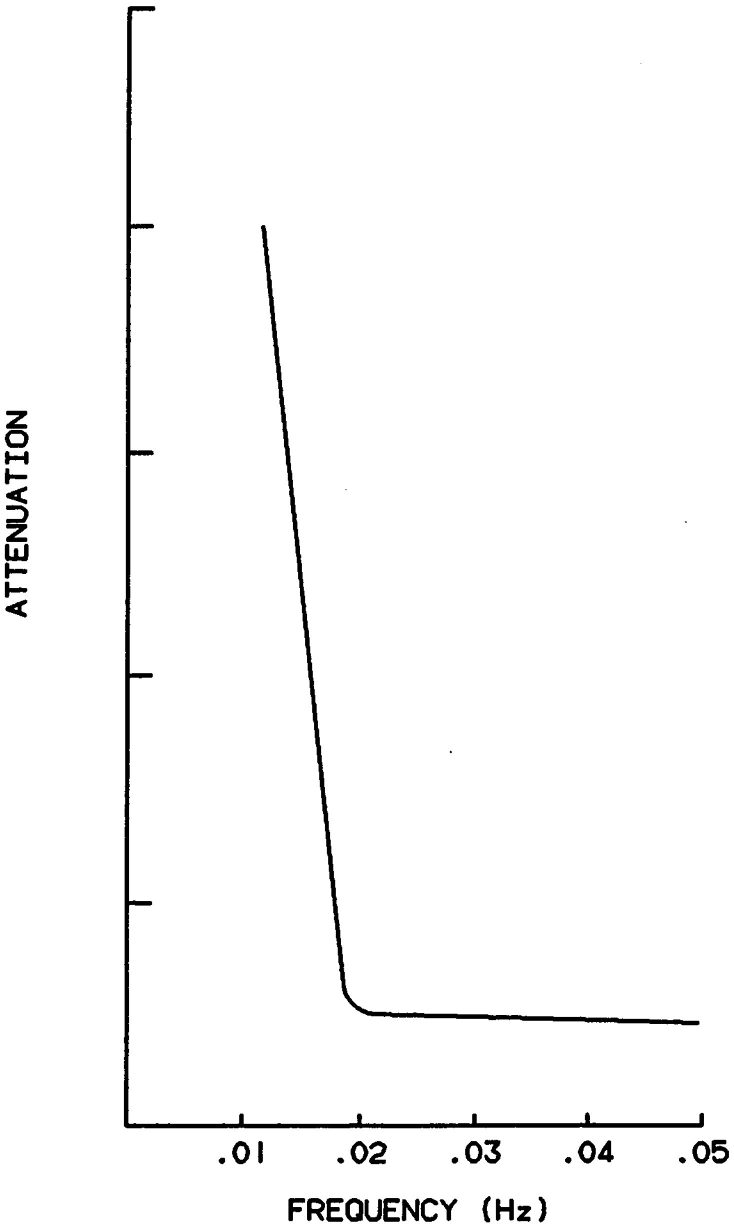


Fig. 13 1/f Noise Plot

Using equations 4.2, 4.3 and 4.4 to solve for Tsky results in:

$$\mathrm{Tsky} = \frac{(\mathrm{Vsky} - \mathrm{Vwrm})}{(\mathrm{Vhot} - \mathrm{Vwrm})}\,\mathrm{Thot} + \frac{(\mathrm{Vhot} - \mathrm{Vsky})}{(\mathrm{Vhot} - \mathrm{Vwrm})}\,\mathrm{Twrm}. \qquad (4.5)$$

Tsky is the desired brightness temperature of the sky and is used to determine atmospheric temperature profiles.

This calculation is then repeated ten times, an algebraic average is computed, and the data is then sent to the SBC. On radiometers with multiple frequencies, the frequency is changed, and this process is repeated for each of the receiver frequencies.

Analog Signal Conditioning

The analog signal conditioning circuitry on the radiometer interface board consists of a buffer amplifier, an anti-alias filter, a sample and hold buffer and an analog to digital converter with a precision voltage reference (Fig. 14). The output from the radiometer's receiver is treated as a differential signal to improve noise immunity. Because of this, the amplifiers used in the buffer and the filter are referenced to the radiometers receiver's signal return. In addition to this, two sample and hold circuits are used to further improve the conversion accuracy, one for the actual receiver output and the other for the signal return.

Buffer Amplifier

The buffer amplifier presents a one hundred kilohm load to the radiometer detector while providing a gain of five. This is done because the detector signal ranges between zero and one volt, while the analog to digital, A/D, converter can accept signals to five volts. An

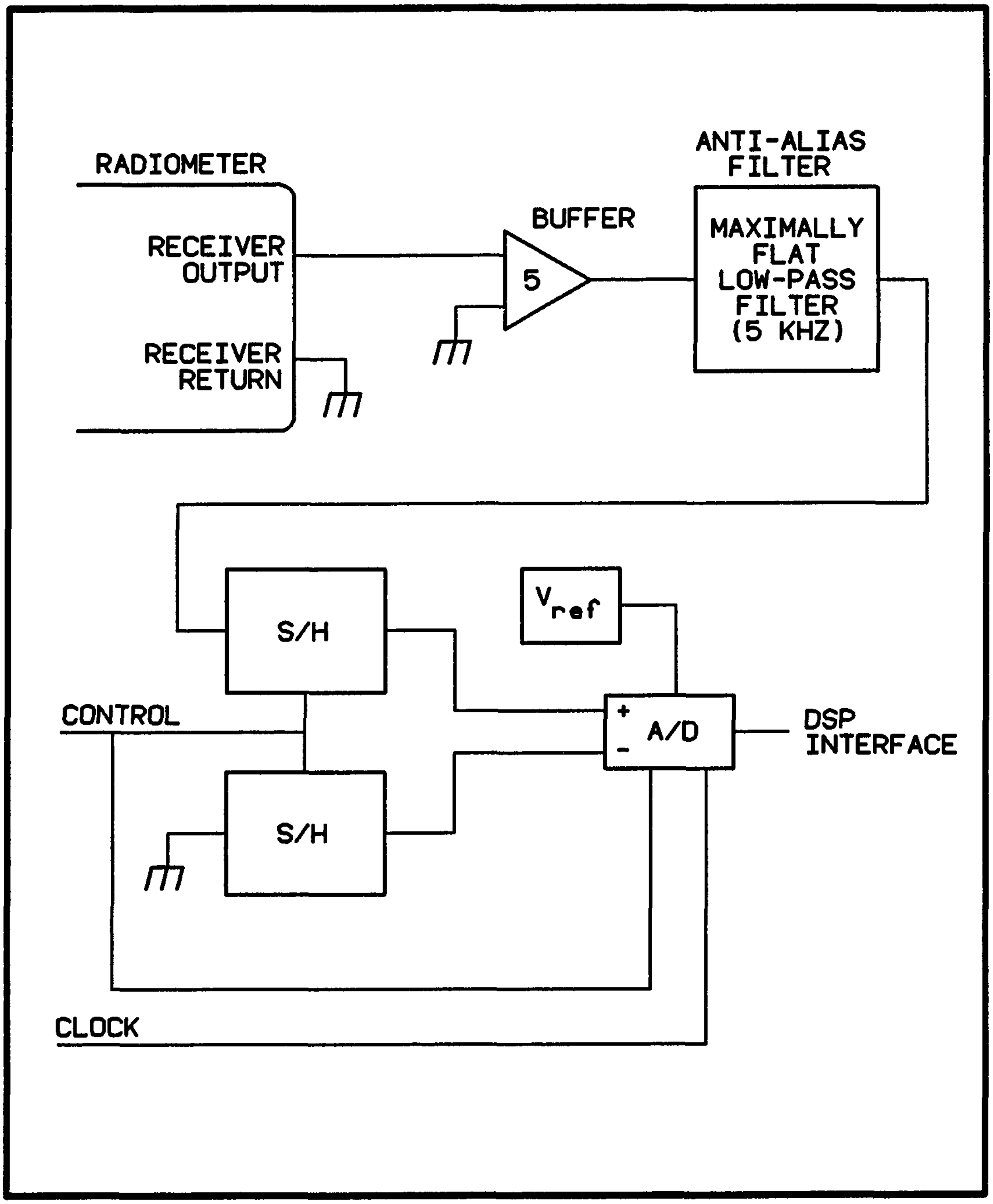


Fig. 14 Analog Signal Conditioning

LM108A operational amplifier was chosen as the buffer because of its extremely low voltage and current offsets and drift over temperature as well as its very high common mode and supply voltage rejection ratios.

Anti-Alias Filter

The 1/f noise present in the radiometers output has been determined to be most severe at frequencies of less than one hertz. The anti-alias filter's cutoff was chosen to be five kilohertz. The selection of this cutoff frequency allows the filter ample time to settle during switching between the three sources and more than satisfies the Nyquist requirement for the waveform reconstruction.

The anti-alias filter was chosen to be an eighth order maximally flat low-pass filter because of the sharp roll-off of this type of filter. The amplifiers used in the filter are LM108A's because of their high input impedance and low offset currents. All resistors are one percent, metal film of the RLR05 series, and the capacitors are ceramic, one percent of the CKR22 series. Figure 15 shows a gain plot for the filter.

Sample and Hold

A sample and hold circuit was employed to reduce skew effects of the A/D converter while it is performing the conversion. Both the signal and it's return are switched in order to accomplish this. The sample and hold, S/H, selected is an LH0023. The LH0023 was chosen because of its offset adjustment and low drift rate.

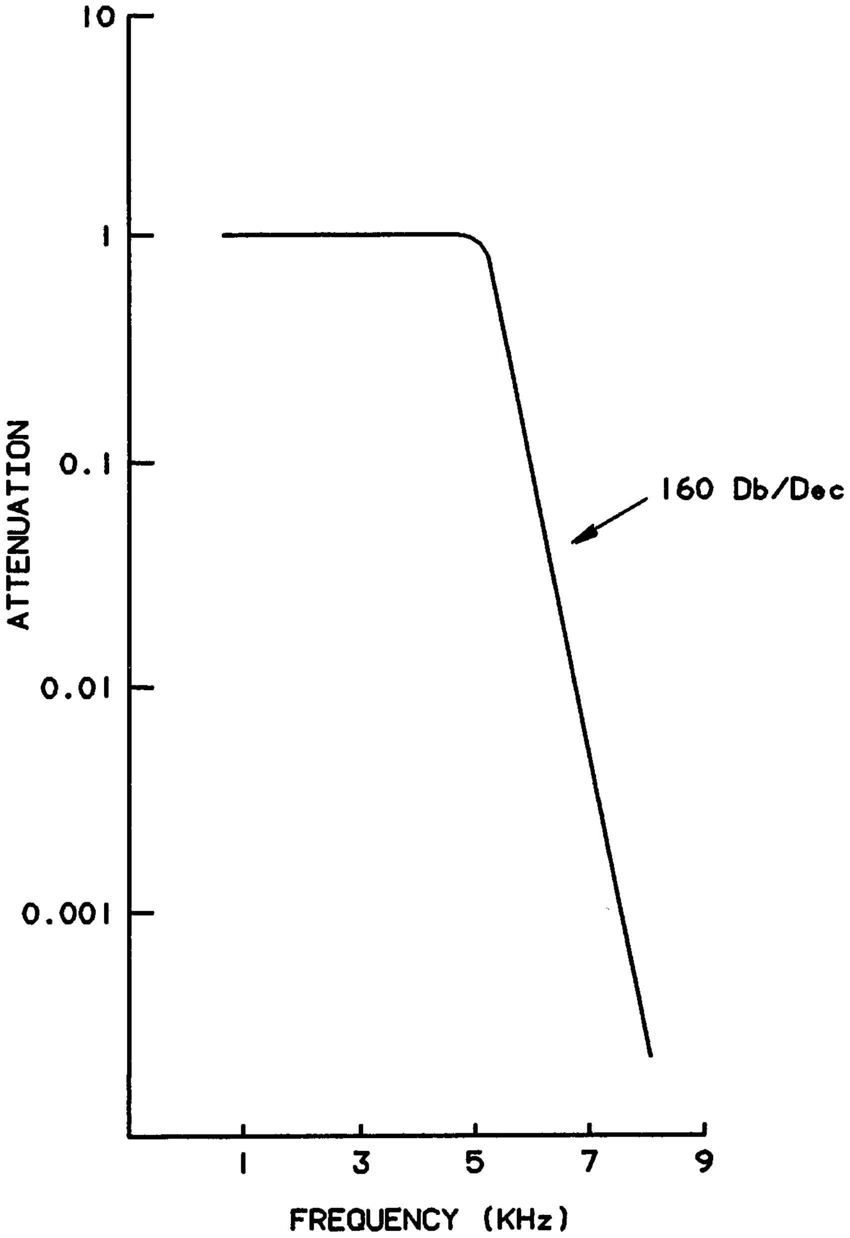


Fig. 15 Anti-Alias Filter Gain Plot

Analog to Digital Converter

The analog to digital converter is an ADC1225, which uses successive approximation to perform the conversion. It has twelve bits of resolution plus a sign bit, a linearity error of +/- 0.5 LSB and a conversion time of 100 microseconds. Although the ADC1225 is microprocessor bus compatible, the access time is not fast enough for the DSP. Because of this, the A/D's outputs are left active, and a high speed tri-stateable buffer is used for the DSP interface. A new signal conversion is started when the DSP reads the buffer. The old conversion data is stable on the converter's outputs until the new conversion is complete, which is ample time for the DSP to read the data. A status flag output also goes to the DSP which indicates that the conversion is complete.

To provide a stable reference throughout the conversion, an external voltage reference, an LM368-5, was utilized. The LM368 provides +5.000 volts +/- 0.02 percent. The reference has a temperature coefficient of +/- 11 parts per million per degree centigrade and excellent line and load regulation.

Digital Filtering

The main function that the processor performs is implementing the digital filter. A finite impulse response, FIR, filter was chosen because of the ease with which this type filter can be implemented on the TMS 32010. In addition to performing the filtering, the processor must be able to manipulate the circulators in the radiometer to select between the sky, reference load and calibration load sources, to change receiver frequencies and to communicate with the SBC. The processor has the address map as shown in Figure 16. The digital signal processing circuitry is contained on one circuit card. The major functions of the card are divided up logically into processing, bus interfacing and interrupt generation (Fig 17).

Processing

The processor is a TMS 32010 digital signal processor running at 20 megahertz. The processor utilizes a modified Harvard architecture. In Harvard architectures the program memory and data memory are in two separate data areas allowing a full overlap of instruction execution and prefetch. This translates to a processor throughput of five million instructions per second. The chip contains 144 words of sixteen bit data memory. Program memory is external and is comprised of 2048 words of sixteen bit RAM.

At power up the DSP is placed into a reset state waiting for contact by the SBC before beginning processing. This is necessary because the DSP's program is stored in RAM, and at power up, the RAM's contents are in an undefined state. While in reset, the data bus is tri-stated, but not the address or control lines. Since the address and control lines are active, they must be disabled using a tri-state device, such as the 74LS244, to prevent an access conflict. After the SBC has finished modifying the RAM, the reset is canceled and the processing resumes.

ADDRESS	OUTPUT OPERATION
0	Status LEDs are updated using four LSBs of the bus
1	Data is latched into communications buffer
2	The SKY is selected as source for conversion
3	The REFERENCE LOAD is selected for conversion
4	The CALIBRATION LOAD is selected for conversion
5	Receiver frequency one is set
6	Receiver frequency two is set
7	Receiver frequency three is set

ADDRESS	INPUT OPERATION
0	The analog to digital converter is read
1	The data in the communications buffer is read

Fig 16 Address Map for the DSP

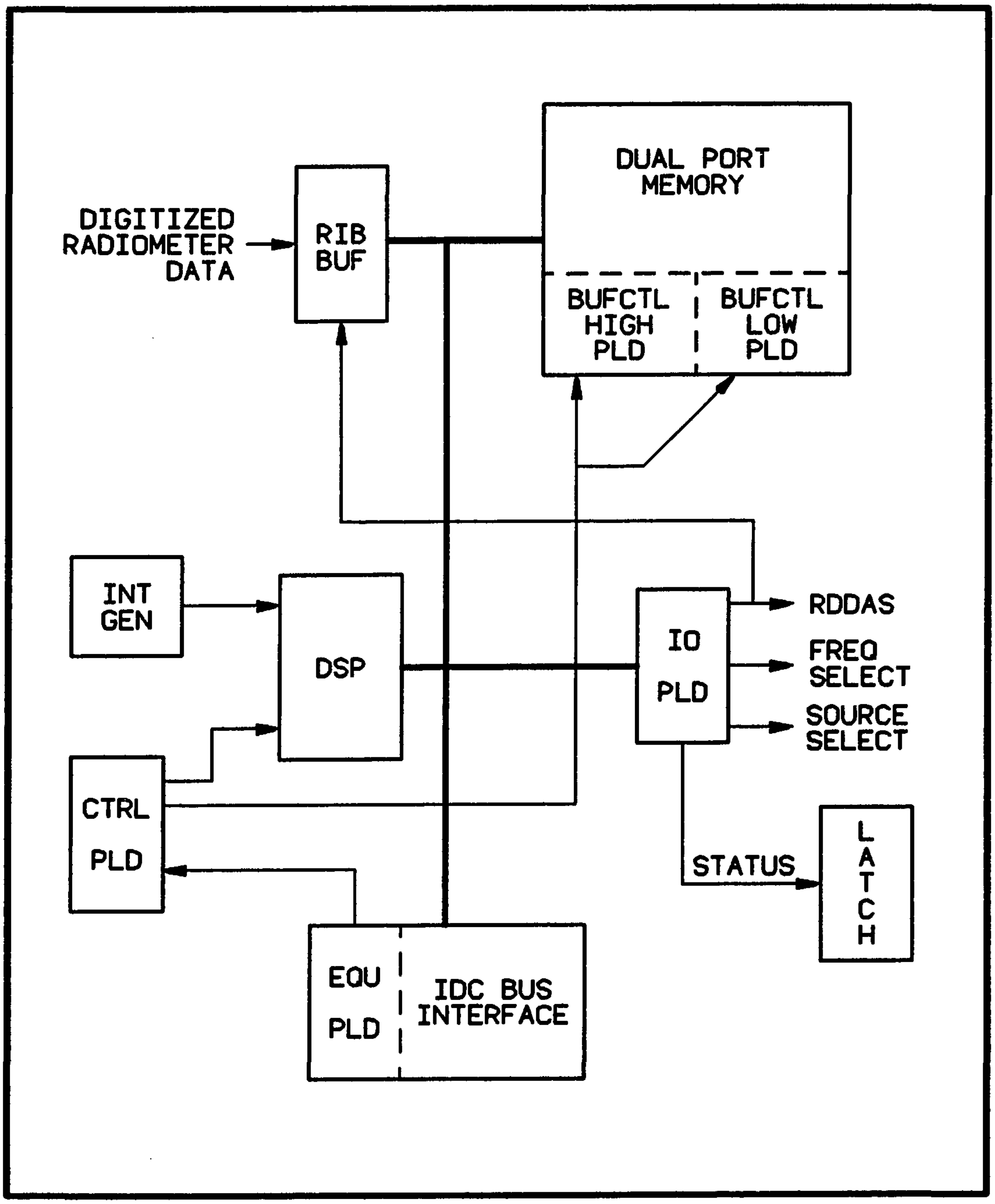


Fig. 17 DSP Block Diagram

The processor controls its local devices using PLD 16R6-IO, U19. This PLD contains six registered or latched outputs and two non-latched outputs. Requests to read the A/D or write to the LEDs use the nonregistered outputs, while requests to switch circulators or change receiver frequencies utilize the registered outputs. Even though the instructions for changing circulators or frequencies are output instructions, there is no actual information exchanged over the data bus. The instructions merely cause the registered outputs in the PLD to be updated appropriately. Interface to the microwave circulators require differential TTL drive; this is accomplished by using two 74LS04 inverters along with the original signal. Once changed, the circulators require one microsecond to settle. Selection of receiver frequencies is made by placing a low on the correct enable line of the receiver. After changing frequencies, the receivers require seven seconds to stabilize. Reading the A/D is accomplished by performing an IN zero instruction. Data from the A/D is accepted by the processor on the rising edge of RDDAS signal. Reading the A/D converter also causes a new conversion of the radiometer's output to take place. The four LEDs on the DSP circuit card, which are used to indicate the state of the processor, are lit by writing a zero in the lowest four bit positions and performing an OUT zero instruction.

Controlling the DSP, on a more global level, is accomplished by PLD 16L8-CTRL, U11. This PLD controls the reset state of the processor, upper/lower data word selection and direction of the SBC/DSP data buffer, U15. The DSP is placed into a reset state when power is first applied by C1 and R2. C1 and R2 provide a reset pulse of approximately one hundred milliseconds. Diode, D1, is also part of the reset circuit. It is used to quickly discharge C1 in the event of a transient power dropout on the five volt supply. In addition, the SBC may start or stop the DSP by writing to the correct address. Both of these requests are acted on by U11, and cause a pulse to be generated which sets the appropriate line on flip-flop U13B. Discussion of data word selection and buffer direction will be addressed in the Bus Interfacing section.

Bus Interfacing

Interface between the DSP and the SBC is physically accomplished through the IDC's backplane. The address space on the SBC is divided into two hundred fifty six pages of two-hundred-fifty-six bytes each. The DSP is mapped into page B8 of the SBC's memory space. PLD 16L8-EQU, U16, decodes the upper eight address lines into a page select, which is sent to PLD U11 as a valid address signal, VLDARD. Of the lower address byte, only the three least significant bits are needed and they are latched by U20. Latching of the address is required because the SBC has a multiplexed address/data bus. This allows the SBC to communicate with the DSP by manipulating locations B800 to B807, hex. U11 further decodes the address into high and low references on the DSP, remember that the SBC has an eight bit data path while the DSP has a sixteen bit data path. Odd address references cause operations to be performed on the upper byte of the DSP's data bus and even address references on the lower byte.

Four remaining signals used by the DSP for decoding, RR/W, RAS, DS and RD which are buffered by U6B. This is done to reduce loading of the

SBC's bus as well as to terminate the signal lines. All data to or from the SBC passes through U19, a 74LS245 bi-directional buffer, which is controlled by PLD U11.

Message passing or DSP RAM modification operations are accomplished using PLDs U7 and U8, 74LS652s U21 and U22, and 74LS374s U17 and U18. The 74LS652, a complex bi-directional buffer/register, accomplishes the task of allowing the SBC to read or write directly to the DSP's RAM or to pass messages between the two processors by latching eight bits of data internally. When altering the DSP's memory, the two 74LS374s, U17 and U18, are used to store the desired address to access in the RAM. The chore of decoding the control lines on the 74LS652s, 74LS374s and on the RAMs lies primarily with the two PLDs. Both PLDs, 16L8-BUFCTL, are programmed with the same algorithm. PLD, U7, is used to decode requests to the low byte and U8 the high byte, with the function of pin one being to select between high and low order byte operations.

Interrupt Generation

An interrupt generator is required by the DSP for accurate sampling, and signal reconstruction, of the radiometer's output. An eight bit counter, a flip-flop and an inverter are used to implement this function. The five megahertz clock, from the processor, is divided by two hundred fifty six to provide a 19,531.25 hertz waveform. The waveform is then shaped, using the flip-flop, to form an eight hundred nanosecond active low pulse. This pulse is then routed to the processors interrupt pin of the processor, where a low going pulse will cause the processor to service the interrupt.

Software Aspects

The hardware of the radiometer interface is very flexible. Almost all of the hardware is, in fact, controlled through software. This approach is in keeping with the design goals of this thesis. Because of this, there is no one, correct program to execute on the DSP. Instead, there exists a pool of routines that can be combined to form a program. This section will describe a framework for a typical program. Discussion of the program that was originally written to execute on the DSP will be deferred until chapter seven.

The software for the DSP is written in the processors native assembly language because a high level language was not available for this processor. Even if one were available, it still might be preferable to write in assemble language because of the ease at which FIR filters are implemented on the TMS 32010.

The software in the DSP is separated into a background task and a foreground task. The responsibility of the background task is to update the system clock and read the A/D converter. The foreground task, or event scheduler, processes the data and communicates with the SBC. Figure 18 shows a pseudo code representation of the DSP software.

The background task is executed each time an interrupt occurs, every 51.3 microseconds. The variable, TIMER, is incremented each time the background task is executed. TIMER serves as the system clock, which is used by the event scheduler. The A/D converter is also read during the interrupt service, but not each time the interrupt occurs.

```
        B       START
        B       INTERRUPT

* SYSTEM CONSTANTS AND FILTER COEFFICIENTS STORAGE
              .
              .
              .

START:  INITIALIZATION

LOOP:   EVENT SCHEDULER

                IF TIME = FILT_INTRVL
                        DO FILTERING

                IF TIME = SOURCE_INTRVL
                        DO SELECT_SOURCE

                IF TIME = FREQ_INTRVL
                        DO FREQ_SELECT

                    .
                    .
                    .

                IF TIME = COM_INTRVL
                        DO COMMUNICATIONS

        B       LOOP

INTERRUPT:

        UPDATE SYSTEM CLOCK
                TIME = TIME + 1
                IF TIME > MAX_TIME
                        TIME = 0

        READ A/D CONVERTER

        RETi
```

Fig. 18 DSP Pseudo Code

The A/D converter is read only after the number of interrupts stored in variable ADCNT has occurred. The data from the A/D is then stored in variable ADVAL.

The foreground task is setup as a scheduler because of the ease with which alterations can be made in program flow. There exists a program library that contains such necessary routines as circulator selection, receiver frequency setting, filtering, communications and others. From this library a program can be constructed. When a change to the program is desired the order in which the schedular executes the routines changes. There is no need to modify the library, although a new function could be added. This modular approach minimizes the programming effort and keeps the software design at a system building block level.

Video Processing Summary

As discussed in the preceding sections the hardware provides a very adaptable environment for the software to execute on. The restrictions that the hardware imposes are minimal and are as follows:

2048 word maximum program size,
144 word data memory,
51.2 microsecond minimum resolvable time interval,
12 bit A/D conversion accuracy,
1 millisecond circulator switching time,
7 second channel switching time.

CHAPTER 5

GENERAL PROCESSING

The single board computer performs the general processing tasks in the IDC. The basic function of the SBC is to monitor the radiometer, supervise the DSP and communicate with the host computer (Fig. 19). On the SBC, there are three eight bit Input/Output ports, two serial ports, thirty-two kilobytes of RAM, eight kilobytes of EPROM, the Intel 8052-BASIC processor and bus interface circuitry. The I/O ports purpose is to control the power forms in the radiometer and to control a cooling fan. A sixteen channel A/D board rounds out the hardware resources of the SBC (Fig. 7).

Radiometer Monitoring

Monitoring the radiometer is accomplished using the sixteen channel A/D board. The analog interface board is addressed at 47360 decimal or B900 hexadecimal in the SBCs data memory. The board can accept up to sixteen single ended signals between plus and minus five volts. Caution, there is no input voltage protection on the analog signal lines on the board, voltages over 5.3 volts or below -5.3 volts will damage the A/D converter.

The A/D converter is a twelve bit, plus sign bit, successive approximation type analog to digital converter. Accessing the full thirteen bits of data using the eight bit data bus of the 8052 is accomplished by performing two successive reads of the converter. The

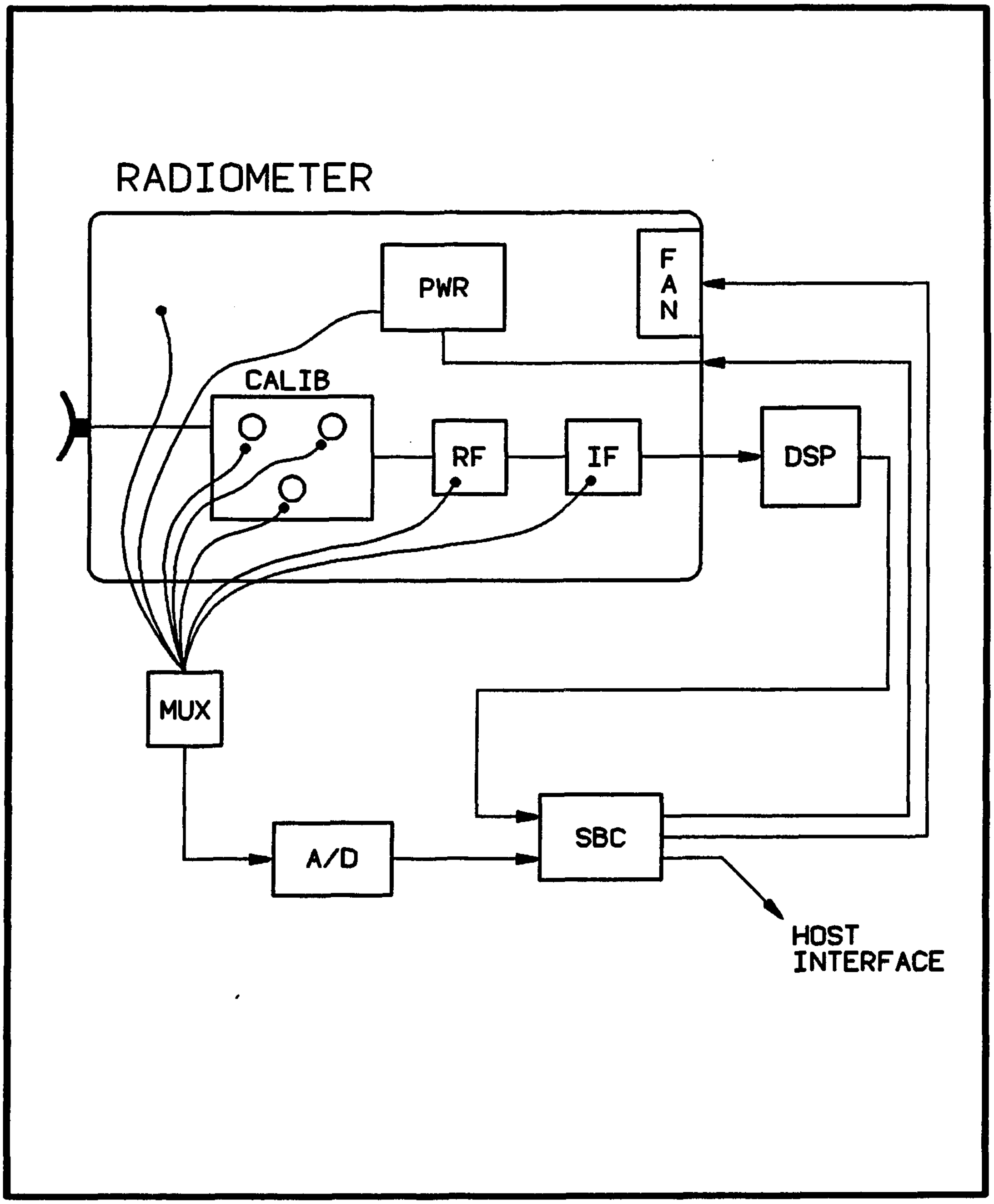


Fig. 19 SBC-Radiometer Interface

first time the converter is read, the high data byte is accessed and the second read causes the low byte to be transferred. A five volt, temperature compensated, voltage reference is provided on the A/D board to ensure a stable reference during the signal conversion. A one megahertz clock is also on the board which is required by the converter for process synchronization.

Starting a conversion and reading the data is accomplished by performing the following steps:

1. Select the desired input channel and set the status line high,
2. Bring status line low while retaining channel number,
3. Read most significant digit of data,
4. Read least significant digit of data.

The two digits can now be combined to form the correct result. There is no need to read the conversion status line of the A/D converter because BASIC is not able to access the converter quickly enough; the A/D converter is able to take 10,000 samples per second while BASIC executes approximately 10 statements per second. The following program fragment demonstrates channel selection and data reading.

```
XBY(47360) = 18 :REM Set mux to chan 2 & set status line high
XBY(47360) = 2  :REM Start the conversion
A1 = XBY(47360) :REM Read the MSB
A2 = XBY(47360) :REM Read the LSB
```

In addition to the A/D converter, the SBC has three eight bit ports. They are addressed at C800 through C803,for ports A, B, C and the control register. Port A has been defined to operate the various functions in the radiometer (Fig. 20). These functions include controlling a cooling fan and also controlling the power forms used by the radiometer, not by the IDC.

BIT	DESCRIPTION
0	Controls power to the radiometer
1	Controls the cooling fan

Fig. 20 Description of Port A on SBC

DSP Interfacing

Control over the DSP is accomplished by reading or writing to various addresses. The page address of the DSP control registers was selected to be 47104 decimal or page B8 hexadecimal. Figure 21 shows the offsets added to the page address to perform the various functions on the DSP.

During normal operation, data is passed between the SBC and DSP using a pair of LSI data registers on the DSP board. To the SBC they appear as two successive memory locations at the first two addresses on page B8. The SBC simply writes data to address B800, low byte of the DSP, and address B801, high byte of the DSP, and the data is latched in a message register used to communicate with the DSP.

The SBC can control the state of the DSP. To stop or reset the DSP, an input from B802 must be performed, and to start the DSP an input from B803 is required. The input instruction is decoded on the

ADDRESS	OUTPUT OPERATION
0	Store data in DSPs data latch, low byte
1	Store data in DSPs data latch, high byte
2	Latch low address byte in DSP (1)
3	Latch upper address in DSP (1)
4	Write data to DSP memory, low byte (2)
5	Write data to DSP memory, high byte (2)

ADDRESS	INPUT OPERATION
0	Read data from DSPs data latch, low byte
1	Read data from DSPs data latch, high byte
2	Halt the DSP
3	Start the DSP
4	Read data from DSP memory, low byte (2)
5	Read data from DSP memory, high byte (2)

Notes:

(1) DSP must be reset to have any effect, however data will be latched and will not change until written again.

(2) DSP must be reset to have any effect.

Fig. 21 DSP Interfacing Map

DSP board and controls a set/reset latch which is connected to the DSP reset line. In both cases, the data transferred during the read instruction may be discarded because no information is actually accessed. The only purpose of the input instruction is to generate an active low pulse which is used by the set/reset latch.

In order to use the remainder of the commands described in Figure 21, the DSP must be halted or be in a reset state. Once stopped, writing to addresses B802 and B803 sets the high and low addresses lines, respectively, of the RAM used as program memory on the DSP board. The data in the DSP's program memory can now be read or modified by accessing location B804 for the low data byte and B805 for the high byte. If the DSP is not stopped, data read will be meaningless and no alteration of the DSPs memory will occur. By accessing the DSP RAM, the SBC can download programs to the DSP, verify the DSP's program memory or download new filter coefficients.

Host Communication

Communication with the host is made using one of the two serial ports on the SBC. In order to communicate with the host, the port is configured for 9600 baud, eight data bits, one start bit, one stop bit and the parity bit disabled. The host is connected to the SBC just as an ordinary terminal would be. Using this connection scheme allows the use of the BASIC print and input statements by the SBC for exchanging data with the host. When the host has to contact the SBC, it sends a '!' character and waits for the SBC to respond with a '?'. Because the SBC has a one byte character buffer for incoming characters, there is no need for the host to repeatedly send the prompt; however, the response latency can be as much as two seconds. Once contact has been established, the communications between the host and the SBC proceed in real time. The SBC is not required to establish contact with the host before sending data. The host has a communications handler, which accepts data from the radiometers. The communications handler operates in the background, so the SBC is free to send data to the host at any time. On occasions when the host is not able to receive data, a control-S is sent to the SBC. This will suspend output from the SBC until a control-Q is received. In the event that the SBC does not respond, the host can send a control-C. When the SBC receives the control-C, it interrupts the running program and waits for instructions. At this time, the host can clear the SBC's program memory, download a new program or re-start the existing one.

SBC Software

The SBC executes a super-set of BASIC. This language is contained in the 8052's internal ROM memory. BASIC is a general purpose language and much detail can be lost when accessing memory addresses. Because of this, a language tailored to this specific interface task was developed to execute on the SBCs. The language is named JAIL and is written in native BASIC used by the 8052 processor. JAIL frees the user from remembering obscure memory locations and their functions allowing the user to concentrate on system level programming.

The JAIL interpreter is composed of two parts (Fig 10). The first part, the command interpreter, is fully accessible and customizable by using a personality module; the second part, the communications controller, is locked away. Denying direct access to the communications controller is done to prevent loss of contact with the host computer when using an untested user customization file. The communications controller task is responsible for maintaining the host interface and maintaining the system clock.

The user customization is accomplished through the use of a personality module which is downloaded via the host. The personality module is a text file that is processed with the JAIL compiler. The compiler takes the natural language source file and converts it into a pseudo code that is interpreted on the SBC. JAIL is a two pass compiler that reads all labeled references on the first pass and then generates complete p-code on the second pass. Figure 22 shows a diagram of the design and implementation phases for creating a JAIL file.

JAIL has approximately twenty commands which include instructions for contacting the host and DSP, reading the A/D board and conditional program branching. Labels and symbols can be up to ten characters long. Currently, there is space allocated for fifty labels and one hundred symbols; this may be expanded by altering the JAIL source code and re-compiling.

A complete description of the JAIL syntax is given in Appendix C.

The communications controller, which is not directly accessible by the user program, communicates with the host. An exception is generated every two seconds which interrupts the main routine and services the serial port. If data is not found in the serial port, the interrupt service is exited. However, if a character is found, direct

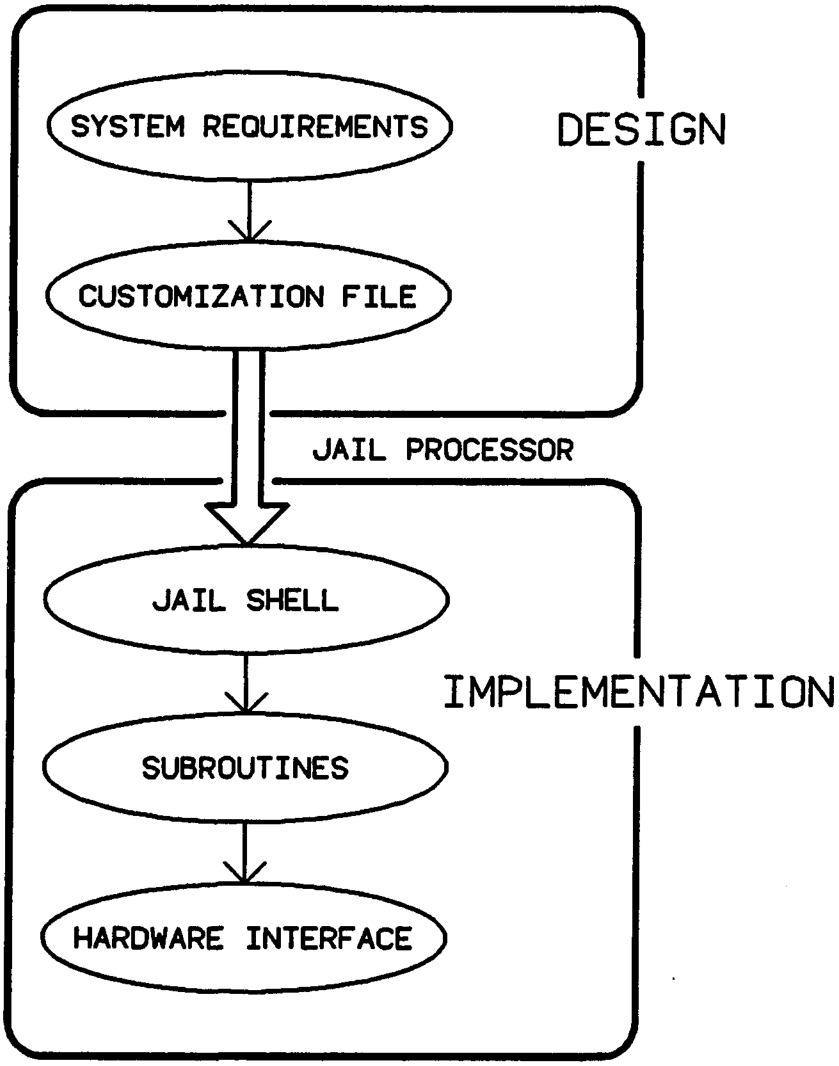


Fig. 22 JAIL Design Cycle

communications with the host is established. When the host is in contact with the service routine, there are six operations that the host may perform, they are:

1. Download another BASIC program,
2. Download a different JAIL file,
3. Change the communication interval,
4. Halt/start the SBC,
5. Halt/start the DSP,
6. Modify DSP program memory.

The communications interval time is verified to be between one second and ten minutes. This interval is the time that the SBC waits before sending brightness temperature data and radiometer health data, not the interrupt service time interval.

A short example program, written in JAIL, is presented below. It will demonstrate the basic command syntax and program flow.

```
;==============================================
;         Sample JAIL program 'TEST.RAD'       =
;==============================================
;

; DEFINE THE CONSTANTS

; define filter coefficients for the various channels

;   Channel 0 -- typical filter values for a thermocouple
#tc_1     0
#xb_1     0.1                ; initial voltage of integrator
#dxb_1    0                  ; initial dv/dt of integrator
#dt_1     0.1                ; sampling interval of detector
#a_1      1                  ; the filter parameters of
#b_1      1                  ;     a/(bs +c)
#c_1      1

;   Channel 1 -- thermocouple
#temp     1

; times 10, number of times to loop
#X_ten    10
```

```
;***************************************
;       start of the program          *
;***************************************

;   initialize parameters

        channel   tc_1
        ic        xb_1, dxb_1
        params    dt_1, a_1, b_1, c_1

:start  LOOP      X_ten
           channel tc_1
           filter             ; low pass filter the data
           wait                ; synchronize readings with the
                        ; interrupt
           ;
           ; check the temperature
           channel  temp
           read

           if > hi_temp
              fan on
           endif

           if < lo_temp
              fan off
           endif

        END LOOP

        host                ; contact the host
        goto      start     ; repeat over and over!

        Z                   ; fini
```

The first part of the example program defines various constants that will be used later on. A '#' precedes the constant's label, and label names are case insensitive. After the constants have been defined, the actual program begins. Channel zero, the thermocouple, is initialized and it will be filtered. A loop is defined to execute ten times. Statements inside the loop will read the thermocouple, filter the data and then wait for a system interrupt. After that, channel one, temp, is read. The value in temp is used to determine whether the fan will be turned on or off. After executing the loop ten times, the host is contacted. After the host returns control to the SBC, the program branches to 'start' where the process continues.

General Processing Summary

The SBC provides the link between the host and the radiometer. It gathers brightness temperature data from the DSP and reports on the state inside the radiometer, namely, temperatures and power-form voltages. The JAIL language provides a system level means for modifying the SBC's operation. JAIL, also, ensures that communications between the host and SBC will not be lost, which is crucial when operated remotely.

CHAPTER 6

HOST COMPUTER

The host computer has the responsibility of orchestrating all of the tasks in the system. This includes controlling the reflector, managing the individual radiometers and storing the data received from the radiometers. Before the data collection can begin, the host must initialize the radiometers since the SBC and DSP use RAM for program memory. By using RAM for program memory, the user can access and redefine any part of the radiometer system from the host. This capability enables the user to change the function of any or all parts of the radiometer without physically dismantling the unit. Because of the flexibility that the system possesses, special attention was given to ergonomics. The goal to the system design was to provide an extremely flexible hardware control structure and to provide software that was smart enough to be able to exploit all of the power that the hardware provides, without burdening the user with the details of that control.

Hardware

The host system is comprised of a NEC APC IV, which is a ten megahertz 80286 microprocessor running MS-DOS 3.2 as an operating system. The computer has 640 kilobytes of memory for running the operating system and user application programs, a thirty megabyte hard disk for program and data storage, a floppy disk drive, a real time clock/calender, four RS-232 serial ports, one Centronics parallel printer port, a stepper motor controller and a high resolution, EGA, type display with the appropriate color monitor. The computer hardware used in the host is available off-the-shelf.

Software

The program executing on the host was written in the 'C' programming language, with the exception of the communications handlers. The Microsoft C version 5.1 programming language was chosen as the compiler because of its execution speed, wide availability of utility libraries and popularity as a programming language. The ancillary communications functions were written in assembly language and compiled using the Microsoft Macro-assembler version 5.0. A library of windowing functions were used in construction of the program. The library is named, "The Window Boss", and it is from Star Guidance Consulting, Waterbury, Connecticut. Structured programming practices were employed in the creation of the program. The various parts of the program were coded as separate modules to make maintenance easier. A MAKE facility was utilized to ease in the creation of new versions of the program. The MAKE utility has the list of program modules that are required to re-construct the radiometer control program, and as any of the source or object files change, MAKE rebuilds the necessary files and then links all of the modules together which creates the program. The host control program is named P2R, for PC to Radiometer interface program.

P2R is an MS-DOS executable program (Fig. 8). P2R is divided into three main parts, which are communications processing, reflector positioning and user applications. Under the communications processor, there exist four serial port handlers. These handlers accept data from the SBC and store it into a data base. The data base is used by the user application programs to extract meteorological information items.

Serial Communications Handler

The serial communications handlers used in P2R replace the standard MS-DOS ones. There is one handler for each of the four serial ports. The new handlers read the serial port whenever a character is received and then store it in a buffer. When the buffer is about eighty percent full, it is appended to the data base.

The new communication handlers are substituted during the initialization process of the P2R program. When they are installed, user application programs cannot use the standard MS-DOS serial communication routines because they have been replaced. Attempting to use the MS-DOS serial port routines will cause erratic program operation to occur and may crash the program. Upon exiting from P2R, the standard MS-DOS communication port drivers are restored.

Timer/Reflector Positioner Handler

The reflector can be programmed to move at various time intervals and angular displacements. The MS-DOS time interrupt was employed to make this task transparent to user application programs. When this handler is installed, it accepts three pieces of information, the time interval between moves, the angular displacement, and the bi-directional movement flag. These values are obtained from the configuration file, which is accessed during the initialization phase of the P2R program. Once installed, the positioning of the reflector is performed automatically, until it is commanded to stop.

User Applications

User application programs perform data analysis operations on the data base. They can be any MS-DOS executable program. These programs may be run independently from P2R, however, data capture from the radiometers will not occur. When run from P2R, radiometer measurements are stored to the data base and the reflector is positioned, while an application program executes. The apparent multi-tasking from MS-DOS is accomplished by installing the data capture and reflector positioner tasks as interrupt handlers and then spawning a user application program as a sub-task under P2R. This is not true multitasking because when an application program is running, P2R is temporarily suspended from executing, and when the application program terminates, P2R resumes execution. If one application program desires to share data with another application program, it must write the information to an intermediate file because only one application program is able to execute at a time. When an application terminates, all data buffers and files are closed and this resource is returned to the system.

System Initialization

The system must be initialized before data capture can begin. The host must create buffers in memory for the communications handlers to use and open the data base file before installing the four communication modules. Once the communications drivers are operating, the IDC in all of the radiometers can be initialized.

Each of the four radiometer sub-systems has an IDC which has a general purpose processor, the SBC, and a digital signal processor, the DSP, which must be initialized. Because system flexibility is a major goal of the project, each of the sub-systems can execute different programs. To allow for this, a configuration file is read by the host for the purpose of initializing each of the processors independently and to allow for some configuration of the host as well. The configuration file is an ASCII file that can be modified or created using any word processor. It may also be updated from within the host control program when it is in the maintenance mode. The configuration file contains the names of the various programs used by the four radiometers and some host operating parameters. The configuration file has the structure which is shown in Figure 23.

The configuration file is organized into five sections. Four of the five parts contain specific file names to be downloaded to the radiometers. These file names are the BASIC program used by the SBC, the JAIL customization file, the TMS 32010 file for the DSP and the filter coefficient file. The fifth part contains the modifiable system operating parameters and the names of the user application programs. The hash sign, #, followed by the numerals zero to four is used to differentiate between the five parts being initialized. The profiles for the four radiometers are identified by the hash sign followed by the digits 1, 2, 3, 4, while system parameters use the digit 0. The digits 1, 2, 3 and 4 correspond to radiometers 1, 2, 3 and 4, respec

```
; Customization file for the host
;
; Setup the system profile
#0
20                              ;Start position in degrees
160                             ;Stop position in degrees
30                              ;Number of degrees per step
15                              ;Number of seconds between steps
1                               ;Allow bi-directional sweeping

; Define the user application programs, include drive and path
; specifier with the program name, if required.

c:\APPLIC\MEAN_DIS Perform a mean distribution on the data base
c:\APPLIC\STAT_AVG Perform a statistical average on the data base
c:\APPLIC\TREND Perform a trend study on the data base

; Radiometer 1 configuration
#1
C:\DATA_AREA\exec.bas           ;The SBC BASIC program
C:\DATA_AREA\rad_1.jl           ;The JAIL file
C:\DATA_AREA\dsp.lod            ;The TMS 32010 program
C:\DATA_AREA\filter_1.dsp       ;The filter coefficients

; Radiometer 2 configuration
#2
C:\DATA_AREA\exec.bas           ;The SBC BASIC program
C:\DATA_AREA\rad_2.jl           ;The JAIL file
C:\DATA_AREA\dsp.lod            ;The TMS 32010 program
C:\DATA_AREA\filter_2.dsp       ;The filter coefficients

; Radiometer 3 configuration
#3
C:\DATA_AREA\exec.bas           ;The SBC BASIC program
C:\DATA_AREA\rad_3.jl           ;The JAIL file
C:\DATA_AREA\dsp.lod            ;The TMS 32010 program
C:\DATA_AREA\filter_3.dsp       ;The filter coefficients

; Radiometer 4 configuration
#4
C:\DATA_AREA\exec.bas           ;The SBC BASIC program
C:\DATA_AREA\rad_4.jl           ;The JAIL file
C:\DATA_AREA\dsp.lod            ;The TMS 32010 program
C:\DATA_AREA\filter_4.dsp       ;The filter coefficients
```

Fig. 23 Host Configuration File Structure tively. Comments and blank lines can be freely used to improve readability of the configuration file. Comments are preceded by a semicolon; anything following the semicolon on that line will be ignored.

On the lines following the definition of a part, specific parameters are defined. For the system they are the start position of the reflector, the stop position of the reflector, the number of degrees to move each time, the time between steps,the bi-directional movement flag and application program names. The units used for positioning are degrees, time between steps are specified in seconds, and the swapping interval is given in milliseconds. The bi-directional movement flag determines whether the reflector, after reaching the stop position, returns to the start position before continuing to take data or sweep back to the start position using the previously defined step interval in the negative direction. When the flag is a one, bi-directional movement is enabled, otherwise the reflector returns to the home position before data collection resumes. The swapping interval is the time the system allows data analysis tasks to run before being interrupted. In the event that one or more of the parameters is omitted, a warning message will be sent to the user and default values will be used. The default values are:

Start position	20 degrees
Stop position	160 degrees
Degrees per step	15 degrees
Time between steps	10 seconds
Bi-directional flag	Disabled

The customization file will then be updated to reflect these values. Following the reflector positioning definitions, the user application programs are identified. The complete drive and path name of the file should be included as part of the file name, the remainder of the line is used to hold a short description of what this application is used for.

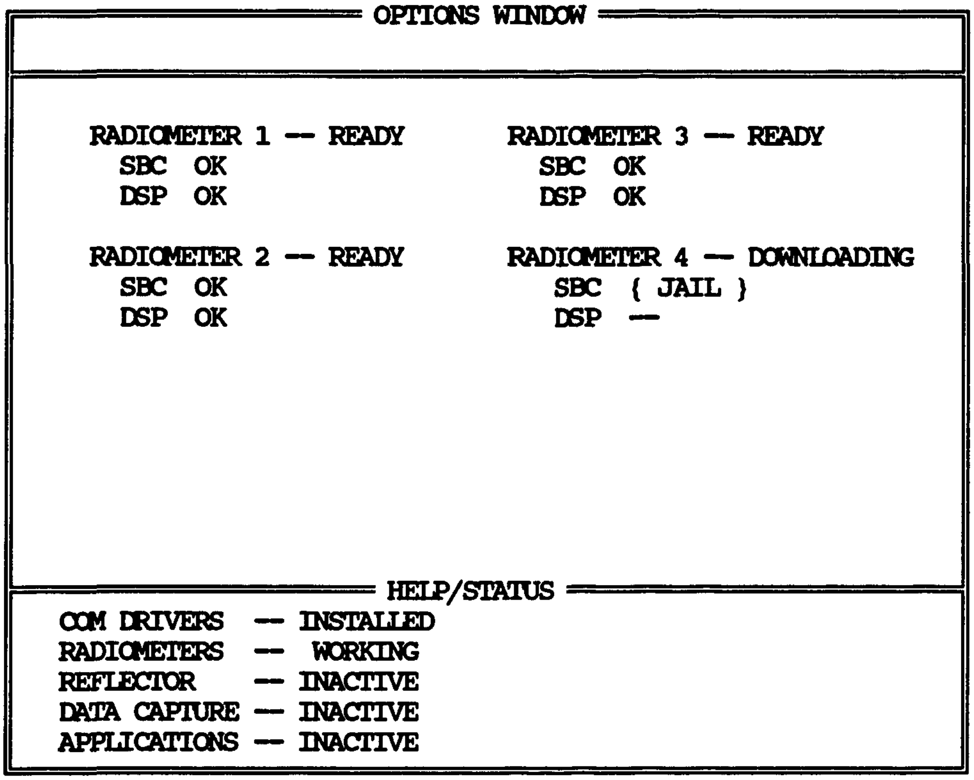
Parameters for radiometer definitions contain the names and paths of the files that will be downloaded to the SBC or the DSP. The ordering of the file names is: the name of the SBC executive, the name of the JAIL customization file, the name of the DSP executive, the name of the filter coefficients file. In the event that any file names are omitted, default names will be used; they are:

SBC executive	EXEC.BAS
JAIL file	DEFAULT.JL
DSP executive	DSP.LOD
Filter data	FILTER.DSP

The customization file will then be updated to reflect these name additions. If any of the files cannot be downloaded, the user will be informed, and the affected radiometer will be locked out until the user corrects the problem.

The customization file may be specified on the command line when the host's command program, P2R, is loaded. If it is not found, then the default file, DEFAULT.CFG, will be tried. Failing this attempt, the command program will create a copy of the default configuration file and inform the user.

During initialization, a window opens which will show the part of the system that is being affected. When system initialization has completed, the user will be informed. The program will continue to display the initialization information until a key is pressed. Any item that fails will be highlighted for easy identification (Fig. 24).



```
OPTIONS WINDOW

  RADIOMETER 1 -- READY        RADIOMETER 3 -- READY
    SBC  OK                      SBC  OK
    DSP  OK                      DSP  OK

  RADIOMETER 2 -- READY        RADIOMETER 4 -- DOWNLOADING
    SBC  OK                        SBC  { JAIL }
    DSP  OK                        DSP  --

HELP/STATUS
COM DRIVERS   -- INSTALLED
RADIOMETERS   --  WORKING
REFLECTOR     -- INACTIVE
DATA CAPTURE  -- INACTIVE
APPLICATIONS  -- INACTIVE
```

Fig. 24 Initialization Window

Automatic Data Capture

Upon completion of the initialization portion of the code, the command program enters the automatic data capture section of the code. The main body of this section deals with data storage and reflector positioning, which are hidden from the user. There is a basic framework to the program that allows for easy modification, yet protects the vital functions of data storage from inadvertent corruption.

The data storage portion of the code is completely hidden from the user. It has priority over the application program that may be running. An interrupt driven communications driver was written to replace the MS-DOS driver. The new driver creates a 1300 byte buffer for each of the four serial ports. The buffer is used for storing received data prior to storing them on the hard disk. The buffer is used to minimize the amount of disk access required to store data. When the buffer is about eighty percent full, the data in the buffer is transferred to the hard disk.

The reflector positioner task is likewise hidden from the user. After a pre-defined interval, the reflector is moved to another position. During the movement, no data will be taken by the SBCs or DSPs. Data in the host's buffer is sent to the data base along with any data still in the SBC. The system time and new reflector position are also stored and are used as a marker that documents the change of position.

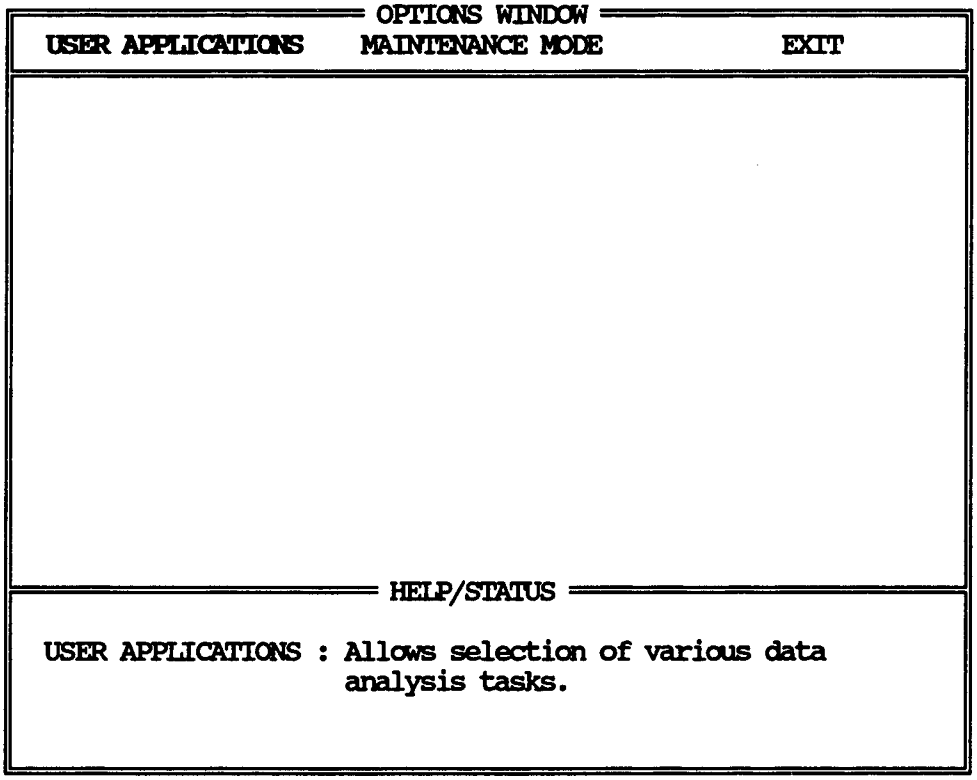
The data analysis task or tasks are routines written by the user. There is no need to link these programs with the command program, P2R. These tasks can access the data base to perform trend studies, plot brightness temperature profiles, and so on. The tasks are accessed by adding their names to the configuration file and selecting them from the P2R applications menu.

Ergonomics

The host is the interface to the radiometer system. Interface with the user is accomplished using windows. In this strategy, the screen is divided into three horizontal regions or windows. The first window is where the main operational modes are selected. It is located along the top three lines of the screen. This window is labeled, "Options Window". The options window is the root window and will always be displayed. The second window is a help window. It occupies the bottom five lines of the screen. The help window will display information about the various choices in the options window, and it will also be used to prompt the user for additional information, when required. The remaining screen area is where other windows are opened as they are needed. As options are selected, other windows will then be created, below and to the right of the preceding window. This is done to allow the user to know precisely where he is in the command hierarchy.

Maneuvering through the various windows is accomplished by pressing the appropriate arrow keys, up-arrow to go up one item, right-arrow to move right, and so on. Once a selection has been made, depicted by displaying the selected item in reverse video, the enter key is pressed. Depending on the item that was chosen, an additional window may open or the user may be prompted for more information. To close the current window and return to the previous window or to cancel a command, press the 'ESC' key. Pressing the 'ESC' key while in the root menu has no effect, and the program will not terminate.

The root menu displays the commands: User Applications, Maintenance Mode, and Exit. Figure 25 shows a typical start-up screen of the P2R program. In the user applications window, various user supplied typically access the data base and perform a statistical analysis on the data. The maintenance mode allows the P2R program to customize system operation. The final selection, EXIT, causes the system to shut-down; all buffers are flushed, the reflector is commanded to home position, and files are closed.

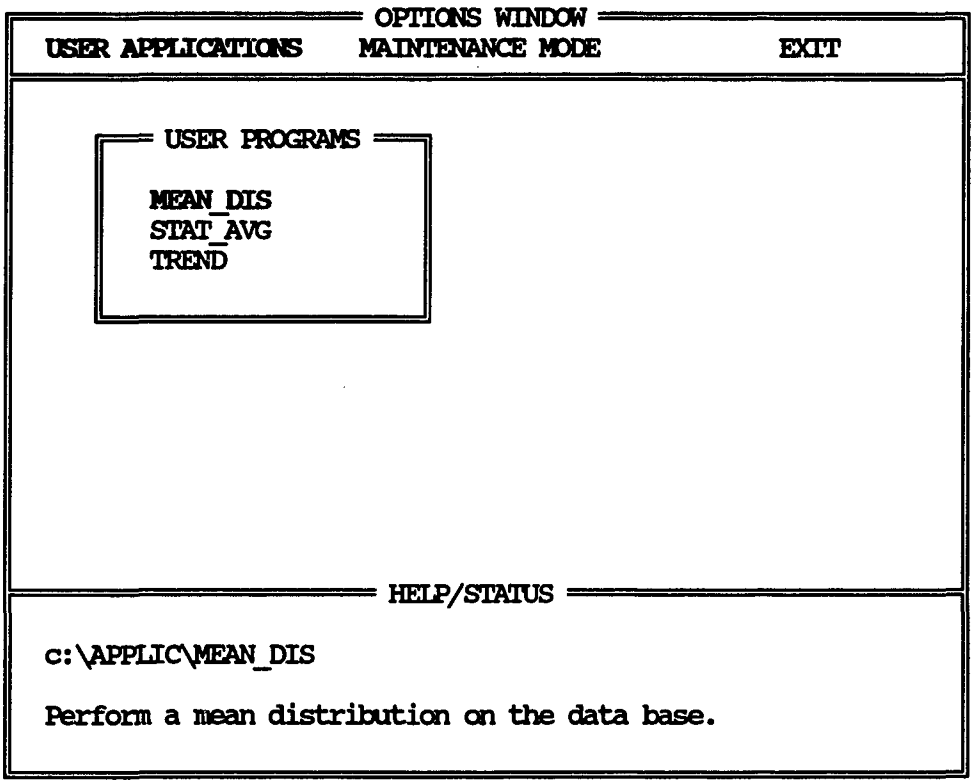


USER APPLICATIONS : Allows selection of various data analysis tasks.

Fig. 25 Start-up Screen

User Applications

Once the initialization of the system has been completed, the program starts capturing data from the radiometers. Several operations can be chosen from this window, Figure 26 displays a typical screen. From this menu, specific user programs can be executed. When the application program is running it appears to have total control of the system. The menus are removed, and the application program can make



c:\APPLIC\MEAN_DIS

Perform a mean distribution on the data base.

Fig. 26 Applications Screen

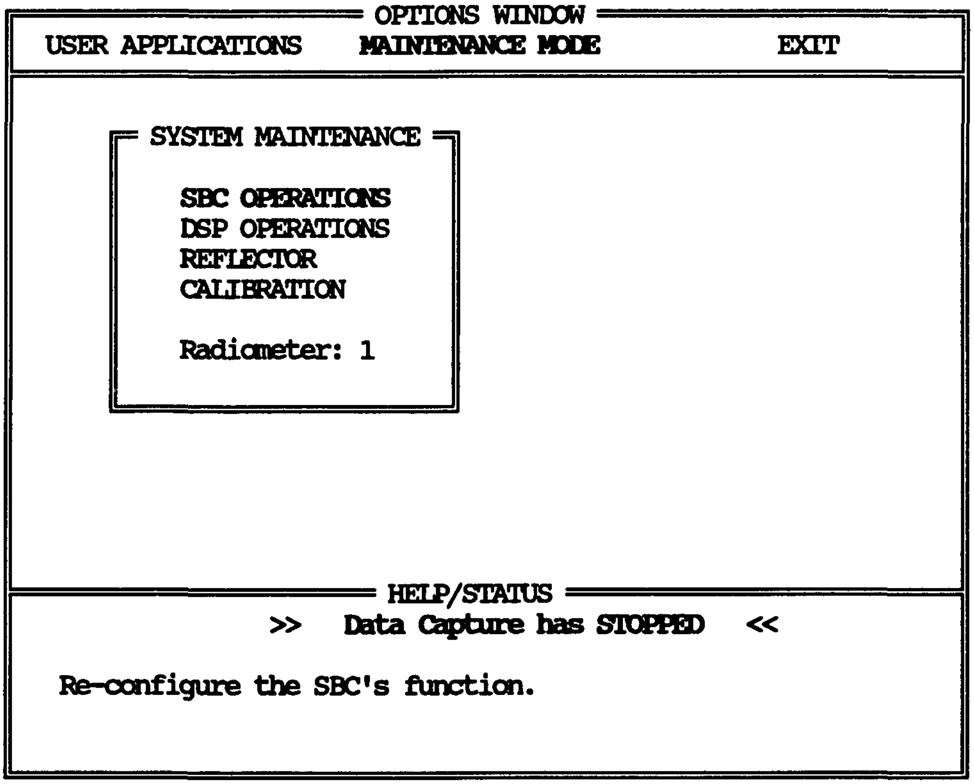
use of the screen in any manner, including using the high resolution graphics mode. When the application program terminates, P2R regains control of the system and the menu windows are restored.

When the user decides to exit from this window mode, he will be asked to verify his intention. This is due to the fact that the automatic data capture operates only when in this mode.

System Maintenance

After exiting the application window, the user is able to modify system parameters. With all of the customizability that is built into the system, there must be a simple way to manage the task of system maintenance, otherwise the flexibility of the system would be a liability and not an asset. There is a menu driven maintenance scheme employed to change the system configuration. It is called up from the root menu by selecting the Maintenance Mode option (Fig. 27).

System maintenance allows the user to alter the various parameters used by the initialization section of the code. It allows the user to manage the configuration file, download SBC or DSP programs



Radiometer: 1

>> Data Capture has STOPPED <<

Re-configure the SBC's function.

Fig. 27 Maintenance Screen

immediately, alter the reflector programming, or perform system calibration.

The items in this menu allow modification of SBC and DSP parameters for the four radiometers (Fig. 27). Radiometers are chosen by pressing the F1 through F4 keys. The selected radiometer is displayed at the top of the screen. Only one radiometer can be selected for alteration at a time.

The SBC alteration selection causes another window to be displayed with choices that allow for downloading a different BASIC command executive, implementing a new JAIL personality module, changing the communication interval and saving the changes. These operations will have an immediate effect on the remote SBC. The BASIC program will not be re-started until the applications window is entered. The final option presented will allow the user to save any of the modifications made. This will cause the program to use these new settings when the program is initialized.

Selection of DSP from the maintenance menu allows replacing the control program in the DSP, creating a new filter profile, downloading the profile and saving that profile to disk. Choosing a new filter program requires that it already exists in LOD file format, the default format of files linked using the TI cross-assembler. If the file specified does not exist, the user is informed and the file in the DSP remains intact.

The selection of a different filter does not mean that the DSP program must be modified and re-compiled, the new filter parameters are simply specified and then they are created from within the P2R program.

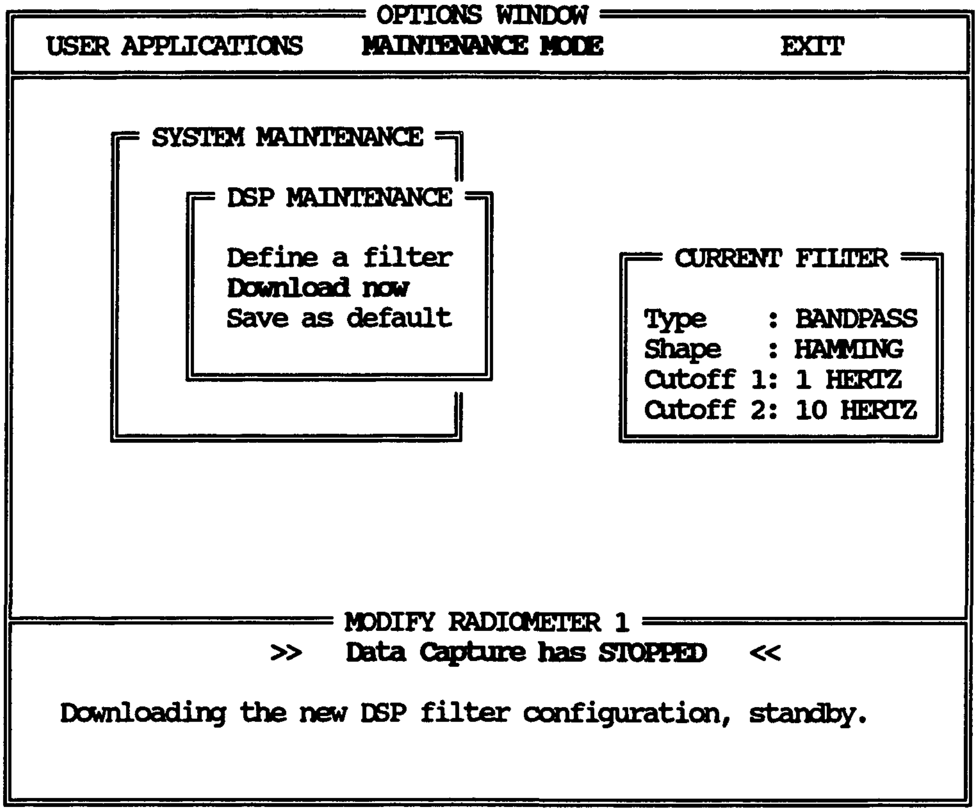
The creation of a different filter requires that the user choose the type of filter, bandpass, lowpass or notch; the shape of the filter, rectangular, Hamming or Keiser; and the cutoff frequency or frequencies. The filter coefficients will be formed, and the user will have a choice of saving them to a file or downloading them immediately. Figure 28 shows the window for the DSP alteration. In addition, a window is opened showing the values of the filter currently operating in the DSP.

Host Summary

Upon starting the command program, the host performs the following items:

1. System buffers and files are initialized,
2. Reflector is positioned,
3. The SBC in the radiometer is contacted,
4. The JAIL interpreter is downloaded and executed,
5. The DSP memory is tested,
6. The DSP control program is downloaded,
7. The JAIL customization program is downloaded,
8. The filter coefficients are downloaded,
9. Steps 3 - 8 are repeated for the remaining three radiometers,
10. The program enters the Applications window.

Once this has been accomplished, routine operation of the radiometer is started. The initialization is only performed when the radiometers are powered up. If, while initialization is being performed, a radiometer does not respond, the user is informed and initialization continues. The bad radiometer will be locked out of automatic data capture until the user services it. New application programs are added by inserting their names in the configuration file. This allows expansion of the radiometer interface program without re-compiling and linking.



Type : BANDPASS
Shape : HAMMING
Cutoff 1: 1 HERTZ
Cutoff 2: 10 HERTZ

>> Data Capture has STOPPED <<

Downloading the new DSP filter configuration, standby.

Fig. 28 DSP Maintenance

CHAPTER 7

CURRENT IMPLEMENTATION

The preceding chapters described the capabilities of the hardware and the control strategy of the software. The following discussion explains how the first version of the control software operates. Of course this version will certainly change as experience is gained with the system. The concept behind this thesis was to provide an adaptable framework under which the radiometer system operates.

The Host

The function of the host program, P2R, is to provide a means of collecting data from the radiometers, controlling the steerable reflector, and provide a means to analyze the stored data. P2R was designed to allow maximum flexibility with the application programs. It implements the data collection and reflector positioning tasks as interrupt service routines. This frees the application programmer from having to understand how to interface to P2R. Application programs are spawned from within P2R, and when they finish executing, control is returned to P2R. System Calibration

Before measurements are taken, the system must be calibrated to reduce the effects of system noise on the brightness temperature data. The host controls a steerable reflector. The reflector is used to make slant path measurements through the atmosphere and to calibrate the system. System calibration is performed by using the tipping curve method or by pointing the reflector toward a container of liquid nitrogen, measuring and recording the emissions from the container at the various wavelengths. This calibration data is store in the data base for reference purposes.

Host Simulation

During normal system operation data sent to the host, from the radiometers, will be stored on the hard disk for further processing by the application programs. The percentage of time that the system spends storing the radiometer data should, ideally, be small. The Network II simulator was used to determine the processor overhead.

The following describes the communication between the host and the four SBCs, as implemented in Network II. The host is able to accept radiometer data from the SBCs in the background, so the foreground task can analyze the data. The data capture routine, of the host, will be implemented as an interrupt service routine, preempting the data analysis task as required. In Network II, the interrupt service request will be modeled as a message from the SBC to the host, which has an input controller. The input controller will allow messages to be received by the host, even while it is busy with other jobs. Each SBC communicates with an individual task on the host with the data being stored in a buffer. When eighteen messages have been received, the data from the buffer will be transferred to hard disk.

The host will not generate a handshake to the SBCs when a message is received. This should not present a problem because of the benign conditions and the short interconnect distance that the system is located in. The simulation will show the loading on the host by the four SBCs. Figure 29 describes the physical makeup of the system. The host is responsible for archiving and analyzing data. The host is a ten megahertz 80286 based microcomputer. After studying the number of cycles per instruction and the instruction mix, it was determined that the basic cycle time, of the host, should be one microsecond. Figure 30 shows the software bubble chart and the Network II model fragment for the host processor.

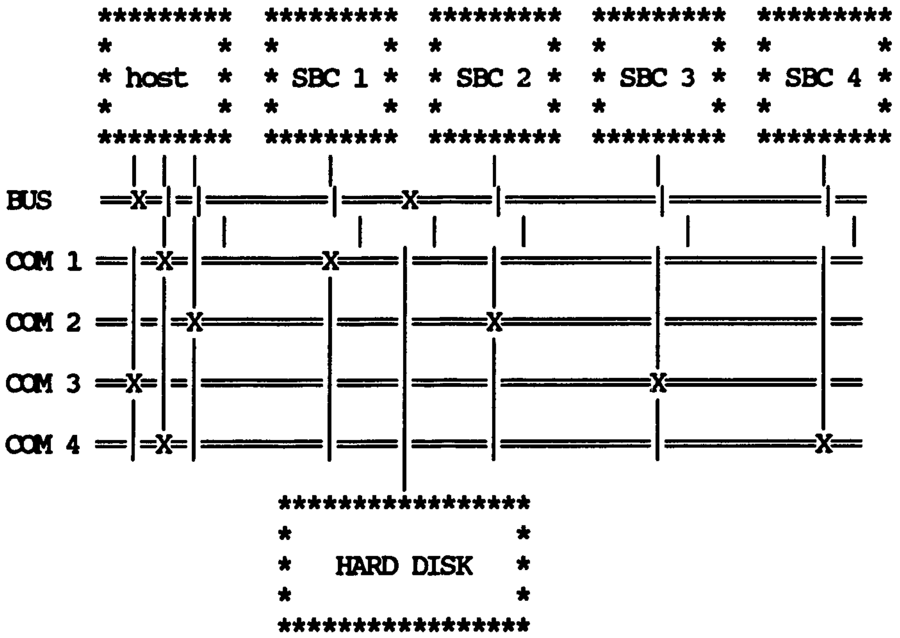


Fig. 29 Host-SBC Hardware Block Diagram

```
                      SBC x DATA
                      /
                     /
      ***************
     *               *  host
    *    ACCEPT x     *
    *                 * ACCEPT x
     *               *
      ***************
             |
             |CHAIN.COUNT :10
             |
             v
      ***************
     *               *
^   *   STORE DATA    *
|___*                 * STORE DATA
OP  *                 *
     *               *
      ***************

 HARDWARE TYPE = PROCESSING
   NAME = host
     BASIC CYCLE TIME =      1.000000 MICROSEC
     INSTRUCTION REPERTOIRE =
       INSTRUCTION TYPE = WRITE
         NAME ; STORE DATA
           STORAGE DEVICE TO ACCESS ; HARD DISK
           FILE ACCESSED ; RADIOMETER DATA
           NUMBER OF BITS TO TRANSMIT ; 6400
           ALLOWABLE BUSSES ;
             BUS
       INSTRUCTION TYPE = PROCESSING
         NAME ; ACCEPT 1
           TIME ; 100 CYCLES

 SOFTWARE TYPE = MODULE
   NAME = ACCEPT 1
     PRIORITY =     0
     CONCURRENT EXECUTION = NO
     ALLOWED PROCESSORS =
       host
     REQUIRED MESSAGES =
       SBC 1 DATA
     INSTRUCTION LIST =
       EXECUTE A TOTAL OF ; 1 ACCEPT 1
     ANDED SUCCESSORS =
       CHAIN TO ; STORE DATA
       WITH ITERATIONS THEN CHAIN COUNT OF ;     18
```

Fig. 30 Host Model Fragment

The SBC model must simulate sending typical messages to the host. Here, the four SBCs will generate fifty bytes of data, every ninety seconds. Figure 31 shows the software bubble chart and the Network II model fragment for SBC 1. This model is the same for each of the SBCs.

```
                 IP: +90 SECONDS
                    /
      ***************
     *               *  SBC 1
    *    PROCESS 1    *
    *                 * PROCESS 1
    *                 * SEND 1
    *                 *
     *               *
      ***************
                    \ SBC 1 DATA
```

```
  HARDWARE TYPE = PROCESSING
    NAME = SBC 1
      BASIC CYCLE TIME =   2000.000000 MICROSEC
      INPUT CONTROLLER = NO
      INSTRUCTION REPERTOIRE =
        INSTRUCTION TYPE = MESSAGE
          NAME ; SEND 1
            MESSAGE ; SBC 1 DATA
            LENGTH ; 500 BITS
            DESTINATION PROCESSOR ; host
            QUEUE FLAG ; NO
            ALLOWABLE BUSSES ;
              COM 1
        INSTRUCTION TYPE = PROCESSING
          NAME ; PROCESS 1
            TIME ; 600 CYCLES

  SOFTWARE TYPE = MODULE
     NAME = PROCESS 1
      PRIORITY =     0
      INTERRUPTABILITY FLAG = NO
      CONCURRENT EXECUTION = NO
      ITERATION PERIOD = +9.0000000E+07 MICROSEC
      ALLOWED PROCESSORS =
        SBC 1
      INSTRUCTION LIST =
        EXECUTE A TOTAL OF ; 18 PROCESS 1
        EXECUTE A TOTAL OF ; 1 SEND 1
```

Fig. 31 SBC Model Fragment

Running the simulation determined that the loading on the host by the four SBCs is three percent. This will leave ample processing time for the user application programs. Refer to Appendix D for a complete print out of the simulation results and the source deck.

The SBC

The SBC functions as an interface for the host to the radiometers. The SBC receives brightness temperature data from the DSP, monitors six thermocouples in the radiometer and monitors three power forms also in the radiometer. Currently, as brightness temperature data is received, the SBC sends it directly to the host, no conditioning or scaling is performed. After the system has been running for some time and analyses are performed on the data, some manipulation may be performed by the SBC on the data.

The thermocouples are used to calculate a temperature gradient, if one is present in the radiometer. Also under study is the effect of temperature on the various pieces of equipment located in the radiometer. This data will be sent to the host for storage every thirty minutes. Temperature data will have a 'S' and the channel number affixed preceding the actual data. The channels are setup to monitor the hardware as shown in Figure 32. There are seven spare analog channels on the A/D converter card.

A JAIL program has been written to implement the basic functions of transferring DSP brightness temperature data, controlling the fan and sending temperature and voltage profiles to the host. A

Channel	Description
0	Exit vent thermocouple
1	Circulator 1 thermocouple
2	Circulator 2 thermocouple
3	Circulator 3 thermocouple
4	RF module thermocouple
5	IF module thermocouple
6	+5 volt power form
7	+15 volt power form
8	-15 volt power form

Fig. 32 SBC Monitor Points

listing of the JAIL program can be found in the report entitled, "SBC PERSONALITY MODULE". The BASIC language JAIL interpreter is in the report entitled, "JAIL INTERPRETER". The JAIL compiler, used on the host, or any IBM PC compatible computer, is written in C. The source code is in the report entitled, "JAIL SOURCE CODE".

A Network II simulation was performed to document the interaction between the SBC and the DSP. The results of the simulation determined the amount of processing that the SBC was able to perform. Figure 33 shows the physical makeup of the DSP-SBC system under study.

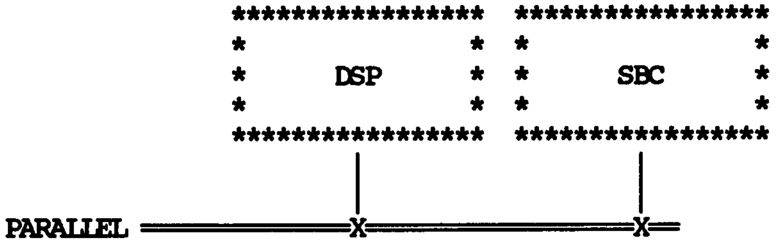


Fig. 33 DSP-SBC Hardware Block Diagram

Consider first the digital signal processor; its purpose is to perform bandpass filtering and scaling of the raw radiometer video. For this simulation, it is assumed that the video data from the DSP will be sent to the SBC once every 5 seconds as a 16 bit number. The DSP has a basic cycle time of 200 nsec. The filter program has been written and it takes 80 cycles or 16 microseconds to execute one iteration of the filter. The software model and the simulation fragment, for the DSP, is presented in Figure 34.

The SBC is responsible for communicating with the HOST, reading the video from the DSP and monitoring the radiometer status. In this section, the simulation will concentrait on everything except communication with the HOST. The SBC executes approximately 500 BASIC source lines per second. After monitoring the radiometer, the SBC will check a status port on the DSP board to see if data is available. Figure 35 describes the software and the model fragment for the SBC.

After running the simulation, it can be seen that the SBCs function can be expanded, if necessary. The processor loading of the SBC is twenty-three percent. Refer to Appendix E for a complete listing of the simulation and the source deck.

The DSP

The DSP is responsible for selecting the input source, selecting the receiver's frequency, filtering the data and computing the brightness temperature. Controlling the input source selection and selecting the receiver frequencies is done by performing an output

```
               IP: +5 SECONDS
                  /
                 /
  ***************
 *               *  DSP
*    DSP MODULE   *
*                 * FILTER
*                 * DATA READY
*                 *
 *               *
  ***************
                 \
                  \ DATA AVAILABLE
```

```
HARDWARE TYPE = PROCESSING
  NAME = DSP
    BASIC CYCLE TIME =      0.200000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; DATA READY
          MESSAGE ; DATA AVAILABLE
          LENGTH ; 16 BITS
          DESTINATION PROCESSOR ; SBC
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            PARALLEL
      INSTRUCTION TYPE = PROCESSING
        NAME ; FILTER
          TIME ; 80 CYCLES


SOFTWARE TYPE = MODULE
  NAME = DSP MODULE
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ITERATION PERIOD = +5.0000000E+06 MICROSEC
    ALLOWED PROCESSORS =
      DSP
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 FILTER
      EXECUTE A TOTAL OF ; 1 DATA READY
```

Fig. 34 DSP Model Fragment

```
                   DATA AVAILABLE
                  /
                 /
  ***************
 *               * SBC
*    SBC MODULE   *
*                 * READ SENSOR
*                 * STORE DATA
*                 * PROCESS SENSOR DATA
 *               * GET FILTERED VIDEO
  ***************   PROCESS VIDEO
```

```
HARDWARE TYPE = PROCESSING
  NAME = SBC
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = PROCESSING
        NAME ; READ SENSOR
          TIME ; 160 CYCLES
        NAME ; PROCESS SENSOR DATA
          TIME ; 200 CYCLES
        NAME ; STORE DATA
          TIME ; 20 CYCLES
        NAME ; GET FILTERED VIDEO
          TIME ; 5 CYCLES
        NAME ; PROCESS VIDEO
          TIME ; 200 CYCLES

SOFTWARE TYPE = MODULE
  NAME = SBC MODULE
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ALLOWED PROCESSORS =
      SBC
    REQUIRED MESSAGES =
      DATA AVAILABLE
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 READ SENSOR
      EXECUTE A TOTAL OF ; 1 STORE DATA
      EXECUTE A TOTAL OF ; 1 PROCESS SENSOR DATA
      EXECUTE A TOTAL OF ; 1 GET FILTERED VIDEO
      EXECUTE A TOTAL OF ; 1 PROCESS VIDEO
```

Fig. 35 SBC Model Fragment

instruction to the appropriate address, which are documented in Figure 16. The brightness temperature calculation is preformed as described in equations 4.1 through 4.5.

The filter algorithm implemented is a length-40 finite impulse response, FIR, filter. The type of filter is a rectangular, bandpass filter, with cutoff frequencies of one and ten hertz. Other types of filters with different cutoff frequencies can be implemented with this system. The largest cutoff frequency, that can be used, without altering the hardware, is two thousand five hundred hertz, and the fastest sampling rate of the analog to digital converter is one hundred microseconds.

A digital filter was used in this design mainly because of the frequency of the input signal; it is very close to zero hertz, DC. If a traditional filter were implemented, the passive components would be physically very large and would be apt to drift over temperature. Another reason for using a digital filter is the ease with which it can be changed. All that is necessary is to compute new filter coefficients and download them; no rework is required. The project, being a research tool, will probably have several different filter configurations implemented before the one that reduces the 1/f noise the most is found. The report entitled, "DSP CONTROL PROGRAM", contains the source code for the digital signal processor. No Network II simulation was performed on the DSP.

CHAPTER 8

CONCLUSION

This thesis presented a hardware and software characterization of an adaptable controller for a multi-channel millimeter wave radiometer. The design goals were to construct a simple to use, yet powerful, system with the following guidelines:

1. symmetrical hardware and software designs,
2. adaptable hardware and software,
3. user friendliness,
4. automatic setup and operation.

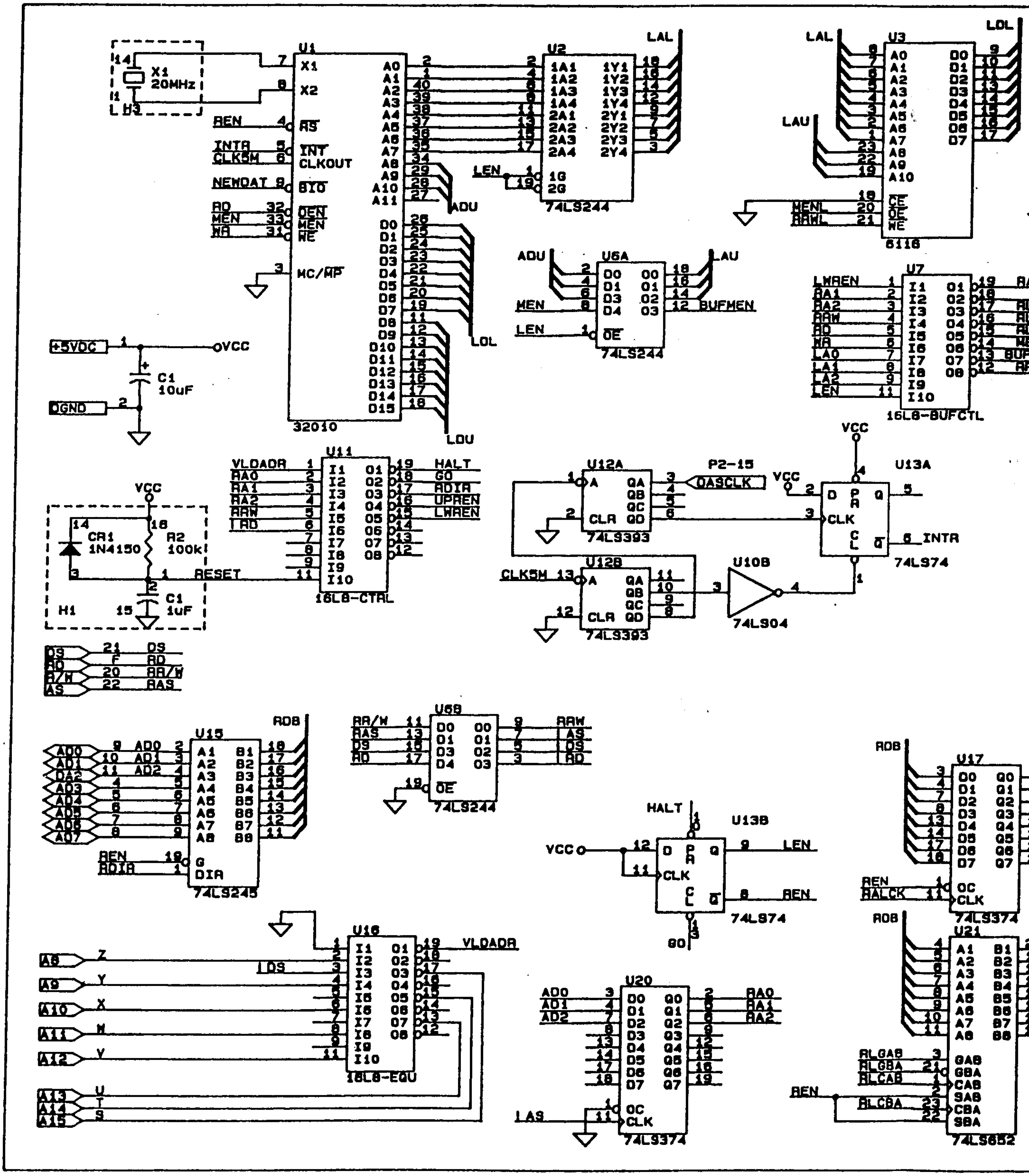
These goals were attained in all parts of the system. The host can configure the entire system under user control. Once the system is setup properly, the parameters are stored in a configuration file, which will be read each time the program is executed. Further, additional analysis tasks may be freely added to the system without the need to re-compile and re-build the control program. All that is required is that their name or names be added to the configuration file.

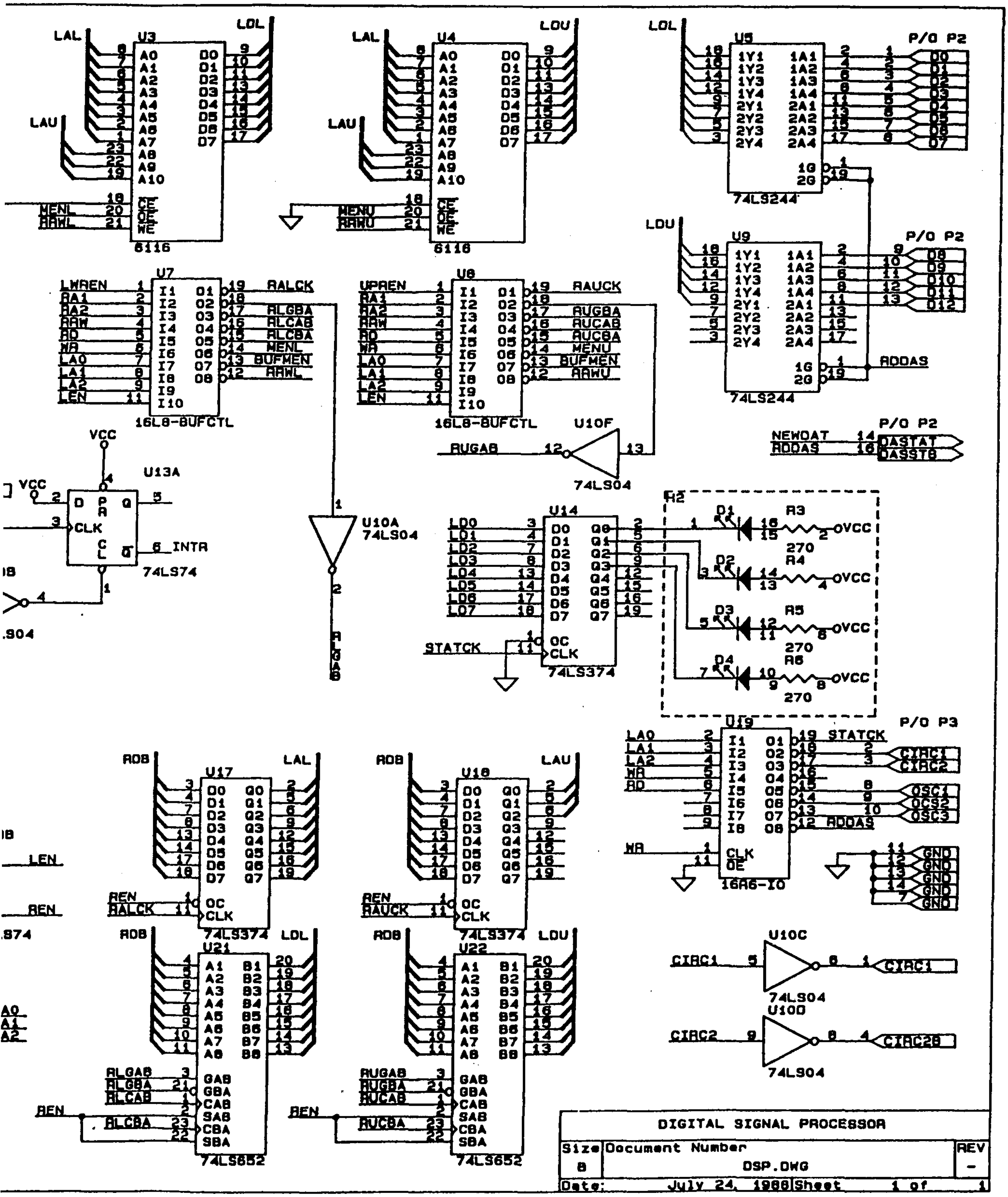
The intermediate data collector, which is comprised of four circuit cards, is identical in all four of the radiometers. The four single board computers execute the same BASIC program, the JAIL interpreter. Differences in program requirements between the systems are handled in the personality module, that executes under the JAIL interpreter. The digital signal processing hardware, in each of the radiome-ters, is also identical. The program that executes in the digital signal processor is similar in the four radiometers. The difference in each of the programs is in the number of receiver frequencies that the particular radiometer has. The filter algorithms and subroutines for the digital signal processors are otherwise identical.

As experience is gained in using the system, and as knowledge is extracted from the data base, this system will change. It is hoped that the hardware and software framework presented here, and built for this thesis, will be able to keep up with the challenge of, as of yet, unforeseen future demands.

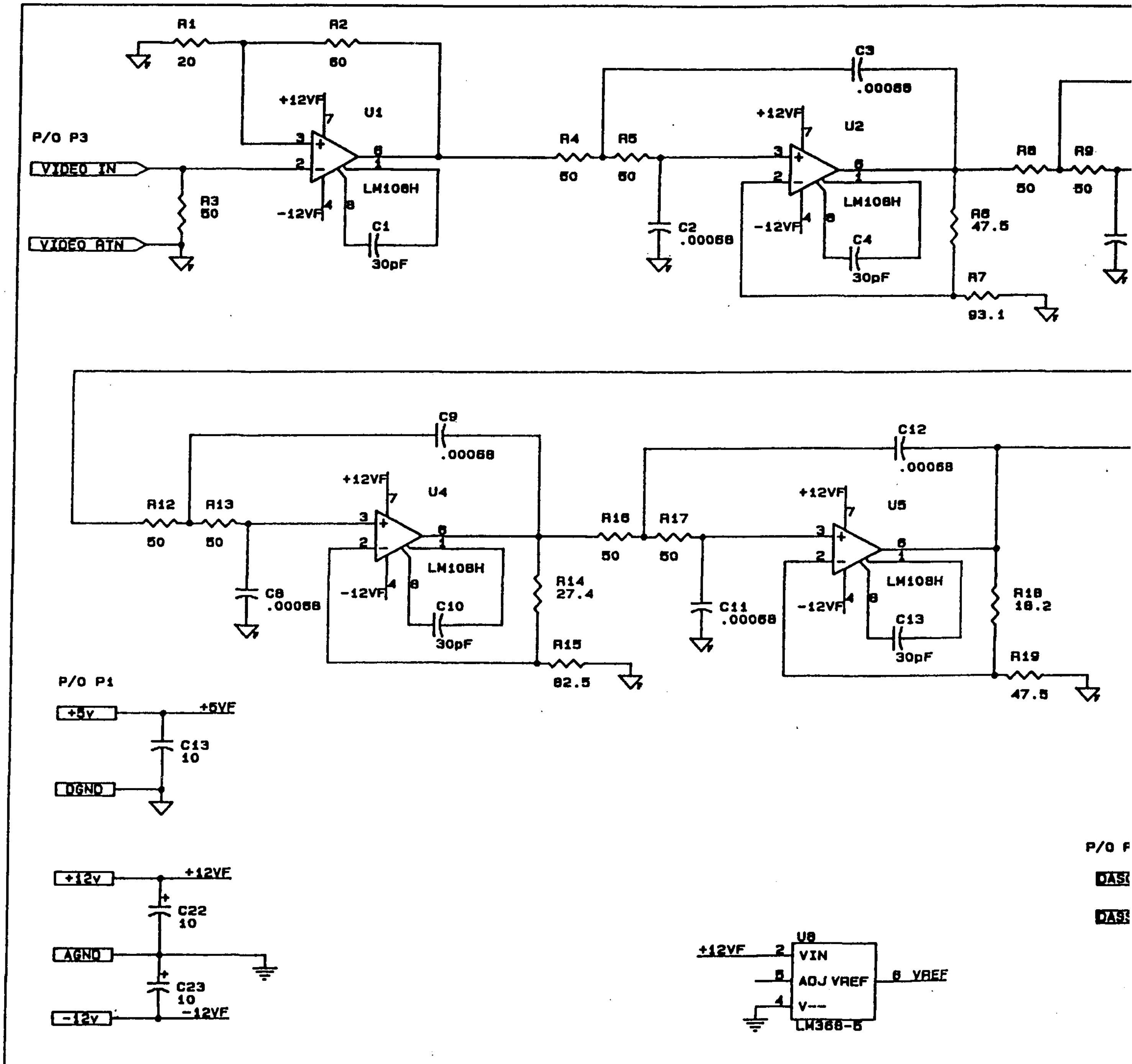
APPENDIX A

DIAGRAMS


X1 20MHz
H3
U1
32010
X1
X2
RS
INT
CLKOUT
BIO
DEN
MEN
WE
MC/MP
REN
INTR
CLK5M
NEWDAT
RD
MEN
WR
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 A11
D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15
ADU
LDL
LDU
U2
74LS244
1A1 1A2 1A3 1A4 2A1 2A2 2A3 2A4
1Y1 1Y2 1Y3 1Y4 2Y1 2Y2 2Y3 2Y4
1G 2G
LEN
LAL
U3
6116
A0 A1 A2 A3 A4 A5 A6 A7 A8 A9 A10
D0 D1 D2 D3 D4 D5 D6 D7
CE OE WE
LAL
LAU
LDL
MENL
RAWL
U6A
74LS244
ADU
LAU
MEN
LEN
BUFMEN
U7
16L8-BUFCTL
LWREN
RA1 RA2 RAW RD WR LA0 LA1 LA2 LEN
+5VDC
VCC
C1 10uF
DGND
U11
16L8-CTRL
VLDADR
RA0 RA1 RA2 RAW RD
HALT
GO
RDIR
UPREN
LWREN
VCC
CR1 1N4150
R2 100k
C1 1uF
H1
RESET
U12A
74LS393
P2-15
DASCLK
U12B
74LS393
CLK5M
U10B
74LS04
U13A
74LS74
VCC
INTR
DS
RD
R/W
AS
RR/W
RAS
U15
74LS245
RDB
AD0 AD1 AD2 AD3 AD4 AD5 AD6 AD7
REN
RDIR
U6B
74LS244
RAW
AS
DS
RD
U13B
74LS74
HALT
VCC
LEN
REN
GO
U17
74LS374
RDB
REN
RALCK
U16
16L8-EQU
A8 A9 A10 A11 A12 A13 A14 A15
Z Y X W V U T S
LDS
VLDADR
U20
74LS374
AD0 AD1 AD2
RA0 RA1 RA2
LAS
U21
74LS652
RDB
RLGAB
RLGBA
RLCAB
REN
RLCBA
GAB GBA CAB SAB CBA SBA

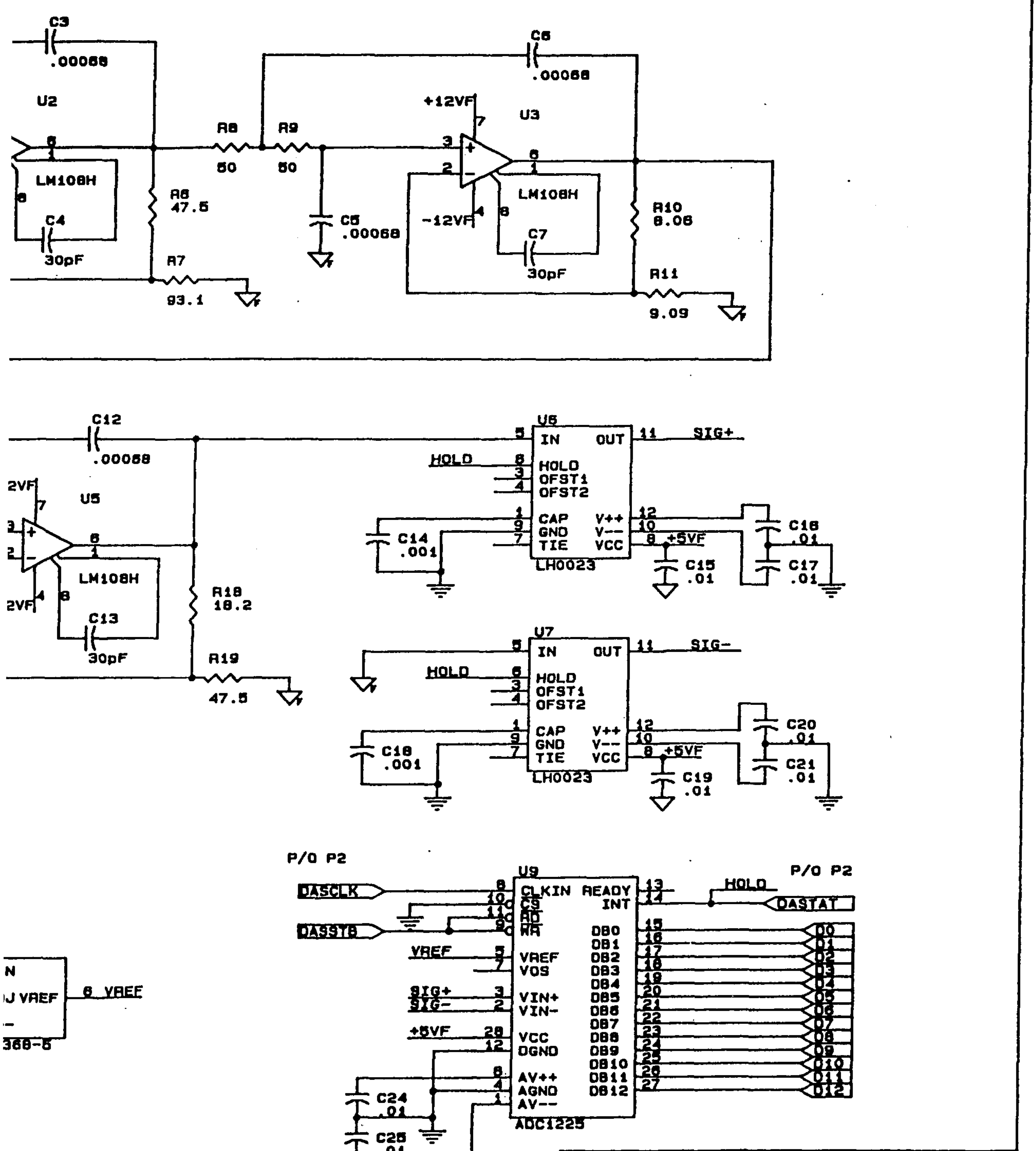


DIGITAL SIGNAL PROCESSOR		
Size	Document Number	REV
B	DSP.DWG	-
Date: July 24, 1988	Sheet 1 of 1	



NOTES:

1. UNLESS OTHERWISE SPECIFIED ALL RESISTANCE IN K OHMS AND CAPACITANCE IN uF.



RADIOMETER INTERFACE

Size	Document Number	REV
B	RIO.DWG	–
Date: July 25, 1988	Sheet 1 of 1	

APPENDIX B

CUSTOM HARDWARE

The following sections of Appendix B contain the pin definitions and the state equations necessary to configure the programmable logic devices, PLDs, used on the digital signal processor board. The basic logic functions, and-ing, or-ing and inverting, are denoted by the '&', '#', and "!" symbols respectively. Comments are formed using the '" 'symbol.

EQU PLD

```
"PLD EQU is a 16L8 device

"       INPUT PIN DEFINITIONS
P0      PIN 2;          "P data bus, bits 0-7
P1      PIN 4;
P2      PIN 6;
P3      PIN 8;
P4      PIN 11;
P5      PIN 13;
P6      PIN 15;
P7      PIN 17;
SEL     PIN 1;          "page address select
DS      PIN 3;          "device enabled when low

"       OUTPUT PIN DEFINITION
VALAD   PIN 19;         "low when page number is correct

"       VARIABLE DEFINITION
PBUS = [P6,P5,P4,P3,P2,P1,P0];

"       STATE EQUATION for EQU
!VALAD  = ( ((PBUS == ^hB8) & !SEL & !DS ) #
            ((PBUS == ^hB9) &  SEL & !DS ) );
```

IO PLD

```
"PLD IO is a 16R6 device

"       INPUT PIN DEFINITIONS

STRB    PIN 1;          "Write Latch
LA0     PIN 2;          "Address LSB
LA1     PIN 3;          "Address 2-LSB
LA2     PIN 4;          "Address 3-LSB
WR      PIN 5;          "Write Enable
RD      PIN 6;          "Read Enable
OE      PIN 11;         "Output Enable

"        OUTPUT PIN DEFINITIONS

STATCK  PIN 19;
CIRC1   PIN 18;
CIRC2   PIN 17;
OSC1    PIN 15;
OSC2    PIN 14;
OSC3    PIN 13;
RDDAS   PIN 12;

"        VARIABLE DEFINITION

addr      = [LA2, LA1, LA0];

"        STATE EQUATIONS for IO

!STATCK   = ( !WR & (addr == 0) );

!RDDAS    = ( !RD & (addr == 0) );

!OSC1    := ( (!WR & (addr == 5)) #
              (!WR & !OSC1 & (addr == 0) ) #
              (!WR & !OSC1 & (addr == 1) ) #
              (!WR & !OSC1 & (addr == 2) ) #
              (!WR & !OSC1 & (addr == 3) ) #
              (!WR & !OSC1 & (addr == 4) ) );

!OSC2    := ( (!WR & (addr == 6)) #
              (!WR & !OSC2 & (addr == 0) ) #
              (!WR & !OSC2 & (addr == 1) ) #
              (!WR & !OSC2 & (addr == 2) ) #
              (!WR & !OSC2 & (addr == 3) ) #
              (!WR & !OSC2 & (addr == 4) ) );
```

```
!OSC3    := ( (!WR & (addr == 7)) #
              (!WR & !OSC3 & (addr == 0) ) #
              (!WR & !OSC3 & (addr == 1) ) #
              (!WR & !OSC3 & (addr == 2) ) #
              (!WR & !OSC3 & (addr == 3) ) #
              (!WR & !OSC3 & (addr == 4) ) );

!CIRC1   := ( (!WR & (addr == 3)) #
              (!WR & !CIRC1 & (addr == 0)) #
              (!WR & !CIRC1 & (addr == 1)) #
              (!WR & !CIRC1 & (addr == 5)) #
              (!WR & !CIRC1 & (addr == 6)) #
              (!WR & !CIRC1 & (addr == 7)) );

!CIRC2   := ( (!WR & (addr == 2)) #
              (!WR & (addr == 3)) #
              (!WR & !CIRC2 & (addr == 0)) #
              (!WR & !CIRC2 & (addr == 1)) #
              (!WR & !CIRC2 & (addr == 5)) #
              (!WR & !CIRC2 & (addr == 6)) #
              (!WR & !CIRC2 & (addr == 7)) );
```

BUFCTL PLD

```
"       INPUT PINS

XEN     PIN 1;          "active low device enable
RA1     PIN 2;          "SBC address bit 2
RA2     PIN 3;          "SBC address bit 3
RRW     PIN 4;          "SBC R/!W
RD      PIN 5;          "DSP read request
WR      PIN 6;          "DSP write request
LA0     PIN 7;          "DSP A0
LA1     PIN 8;          "DSP A1
LA2     PIN 9;          "DSP A2
LENB    PIN 11;         "!Local address bus enable
BUFMEN  PIN 13;         "MEN from uP

"       OUTPUT PINS

RAXCK       PIN 19;     "address latch strobe
GAB         PIN 18;     "ls652 buffer control
GBA         PIN 17;          "
CAB         PIN 16;          "
CBA         PIN 15;          "
MENX        PIN 14;     "RAM !OE control
RRWX        PIN 12;     "RAM R/!W line
```

```
"       VARIABLES

RX = [X,X];
LX = [X,X,X];

Saddr = [RA2,RA1];
Daddr = [LA2,LA1,LA0];


"       STATE EQUATIONS for BUFCTL

EQUATIONS

!RAXCK = ( !RRW & RA1 & !RA2 & !XEN );

!MENX = ( (!LENB & !BUFMEN) #
          (RRW & !RA1 & RA2 & LENB & !XEN) );

!RRWX = (LENB & !RRW & !XEN & RA2 & !RA1);


!GAB = ( (!RRW & LENB & RA2 & !RA1 & !XEN) #
         (!LENB & !RD & (Daddr == 1)) );

!GBA = ( (RRW & LENB & RA2 & !RA1 & !XEN) #
         (RRW & !RA2 & !RA1 & !XEN));

!CAB = (!RRW & !RA1 & !RA2 & !XEN);

!CBA = ( !LENB & !WR & (Daddr == 1) );
```

CTRL PLD

```
"PLD CTRL is a 16L8 device

"       INPUT PIN DEFINITIONS

VALAD   PIN 1;          "valid page address from SBC (active low)
RA0     PIN 2;          "SBC address bit 0
RA1     PIN 3;          "SBC address bit 1
RA2     PIN 4;          "SBC address bit 2
RRW     PIN 5;          "SBC r/!w line
RESET   PIN 11;         "reset into the system
```

```
"       OUTPUT PIN DEFINITIONS

HALT          PIN 19;       "disables the DSP & enables SBC control
GO            PIN 18;       "enables the DSP & disables SBC
RDIR          PIN 17;       "controls direction of remote data bus
UPREN         PIN 16;       "!enable bits 8-15 of DSP via the SBC (RA0=1)
LWREN         PIN 15;       "!enable bits 0-7 of DSP via the SBC  (RA0=0)

"       VARIABLES

RX = [X,X,X];

Saddr = [RA2,RA1,RA0];

"       STATE EQUATIONS for CTRL

!HALT  =  ( (!VALAD & !RDB & (Saddr == 2)) #
            (!RESET) );

!GO    =  ( !VALAD & !RDB & (Saddr == 3) );

!RDIR  =  ( !VALAD & RRW );

!UPREN =  ( !VALAD & RA0 );

!LWREN =  ( !VALAD & !RA0 );
```

APPENDIX C

JAIL USERS GUIDE AND LANGUAGE REFERENCE

This document will describe the use of JAIL to operate the microprocessor controlled radiometer. The full command set will be described and examples will be presented. The commands will be presented by category.

JAIL is an acronym for Just Another Interface Language. JAIL is the language that the single board computer, SBC, in the radiometer understands. Data collection, operational control and communications are accomplished through the use of JAIL. The source file is assumed to have an extension of RAD. This default extension may be overridden by specifying a different one. The output file will have the same name as the source but with an extension of JL.

Any value or values required by expressions, may be the actual numeric value(s) or a symbolic name(s). If a symbolic name is used, it must evaluate to the type of value required by the expresion using it. The following nomenclature will be used when defining commandin JAIL:

'i' integer constant,
'r' real constant,
'l' a label.

Labels are defined by placing a colon before the name. The name must be less than 10 characters long and cannot begin with a number. There is room for 50 labels.

Symbols are defined by placing a hash, '#', sign before the name. The name must be less than 10 characters long and cannot begin with a number. There is room for 100 symbols.

COMMENTS may and should be used freely. Comments can come after any command, symbol or label as long as at least one space separates them. A comment may be place on a line by starting the line with a semi-colon. Blank lines may be used to improve readability.

The following describes the commands of the JAIL interpreter:

HOST	establishes communications with the host computer.
CHANNEL i	sets the A to D converter to the specified channel.
READ	causes the A to D converter to be read at the previously defined channel.
FILTER	similar to READ, however the data is filtered. See also PARAMETERS and IC below.
PARAMETERS r1, r2, r3, r4	sets up the parameters for the digital filter. 'r1' is the sampling interval for reading the data and the remaining three parameters define the filter according to : $h(s) = \frac{r2}{r3\ s + r4}$ A separate filter may be defined for each of the sixteen analog channels.
IC r1, r2	sets the initial conditions for the voltage and charge of the digital filter. 'r1' is the initial value for the voltage and 'r2' is the initial value of dv/dt.
OSC i	determines what frequency the radiometer operates at. When: i = 0, lowest frequency; i = 1, middle frequency; i = 2, highest frequency.

SENSOR i	determines the path of the circulators. When: i = 0, the sky; i = 1, warm load; i = 2, hot load.
FAN ON	turns the fan on.
FAN OFF	turns the fan off.
NOP	No OPeration.
Z	signals the end of the program. No lines following this statement will be compiled.
LOOP i . . ENDLOOP	any legal statements may be places inside the loop. The loop is executed i times or until a 'break' is found.
BREAK	aborts the loop.
IF > r . . ENDIF	tests weather the current channel is greater than r. If so, the commands will be executed upto the ENDIF.
IF < r . . ENDIF	same as above, except less than r.
WHILE TEST . . ENDWHILE	executes the statements within the loop as long TEST is True.
TEST TRUE	sets TEST to true.
TEST FALSE	sets TEST to false.
GOTO 1	program execution jumps to line '1'.

APPENDIX D

HOST-SBC NETWORK II SIMULATION

```
* HOST - SBC Simulation

***** PROCESSING ELEMENTS - SYS.PE.SET
HARDWARE TYPE = PROCESSING
  NAME = HOST
    BASIC CYCLE TIME =      1.000000 MICROSEC
    INPUT CONTROLLER = YES
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = WRITE
        NAME ; STORE DATA
          STORAGE DEVICE TO ACCESS ; HARD DISK
          FILE ACCESSED ; RADIOMETER DATA
          NUMBER OF BITS TO TRANSMIT ; 6400
          REPLACE FLAG ; NO
          ALLOWABLE BUSSES ;
            BUS
      INSTRUCTION TYPE = PROCESSING
        NAME ; ACCEPT 1
          TIME ; 100 CYCLES
        NAME ; ACCEPT 2
          TIME ; 100 CYCLES
        NAME ; ACCEPT 3
          TIME ; 100 CYCLES
        NAME ; ACCEPT 4
          TIME ; 100 CYCLES
  NAME = SBC 1
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; SEND 1
          MESSAGE ; SBC 1 DATA
          LENGTH ; 500 BITS
          DESTINATION PROCESSOR ; HOST
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            COM 1
      INSTRUCTION TYPE = PROCESSING
        NAME ; PROCESS 1
          TIME ; 600 CYCLES
  NAME = SBC 2
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
```

```
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; SEND 2
          MESSAGE ; SBC 2 DATA
          LENGTH ; 600 BITS
          DESTINATION PROCESSOR ; HOST
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            COM 2
      INSTRUCTION TYPE = PROCESSING
        NAME ; PROCESS 2
          TIME ; 600 CYCLES
  NAME = SBC 3
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; SEND 3
          MESSAGE ; SBC 3 DATA
          LENGTH ; 500 BITS
          DESTINATION PROCESSOR ; HOST
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            COM 3
      INSTRUCTION TYPE = PROCESSING
        NAME ; PROCESS 3
          TIME ; 600 CYCLES
  NAME = SBC 4
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; SEND 4
          MESSAGE ; SBC 4 DATA
          LENGTH ; 500 BITS
          DESTINATION PROCESSOR ; HOST
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            COM 4
      INSTRUCTION TYPE = PROCESSING
        NAME ; PROCESS 4
          TIME ; 600 CYCLES

***** BUSSES - SYS.BUS.SET
HARDWARE TYPE = DATA TRANSFER
  NAME = BUS
    CYCLE TIME =      1.00 MICROSEC
    BITS PER CYCLE =      16
    CYCLES PER WORD =      1
    WORDS PER BLOCK =      1
    WORD OVERHEAD TIME =      0.   MICROSEC
    BLOCK OVERHEAD TIME =      0.   MICROSEC
```

```
    PROTOCOL = FIRST COME FIRST SERVED
    BUS CONNECTIONS =
      HOST
      HARD DISK
  NAME = COM 1
    CYCLE TIME =    105.00 MICROSEC
    BITS PER CYCLE =      10
    CYCLES PER WORD =       1
    WORDS PER BLOCK =       1
    WORD OVERHEAD TIME =      0.   MICROSEC
    BLOCK OVERHEAD TIME =      0.   MICROSEC
    PROTOCOL = FIRST COME FIRST SERVED
    BUS CONNECTIONS =
      SBC 1
      HOST
  NAME = COM 2
    CYCLE TIME =    105.00 MICROSEC
    BITS PER CYCLE =      10
    CYCLES PER WORD =       1
    WORDS PER BLOCK =       1
    WORD OVERHEAD TIME =      0.   MICROSEC
    BLOCK OVERHEAD TIME =      0.   MICROSEC
    PROTOCOL = FIRST COME FIRST SERVED
    BUS CONNECTIONS =
      SBC 2
      HOST
  NAME = COM 3
    CYCLE TIME =    105.00 MICROSEC
    BITS PER CYCLE =      10
    CYCLES PER WORD =       1
    WORDS PER BLOCK =       1
    WORD OVERHEAD TIME =      0.   MICROSEC
    BLOCK OVERHEAD TIME =      0.   MICROSEC
    PROTOCOL = FIRST COME FIRST SERVED
    BUS CONNECTIONS =
      SBC 3
      HOST
  NAME = COM 4
    CYCLE TIME =    105.00 MICROSEC
    BITS PER CYCLE =      10
    CYCLES PER WORD =       1
    WORDS PER BLOCK =       1
    WORD OVERHEAD TIME =      0.   MICROSEC
    BLOCK OVERHEAD TIME =      0.   MICROSEC
    PROTOCOL = FIRST COME FIRST SERVED
    BUS CONNECTIONS =
      SBC 4
      HOST

***** STORAGE.DEVICES - SYS.SD.SET
HARDWARE TYPE = STORAGE
  NAME = HARD DISK
```

```
    WORD ACCESS TIME = 30000. MICROSEC
    BITS PER WORD =      16
    WORDS PER BLOCK =     512
    OVERHEAD TIME PER BLOCK ACCESS = 0. MICROSEC
    CAPACITY =         24000000. BITS
    NUMBER OF PORTS =     1

***** MODULES - SYS.MODULE.SET
SOFTWARE TYPE = MODULE
  NAME = ACCEPT 1
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ALLOWED PROCESSORS =
      HOST
    REQUIRED MESSAGES =
      SBC 1 DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 ACCEPT 1
    ANDED SUCCESSORS =
      CHAIN TO ; STORE DATA
      WITH ITERATIONS THEN CHAIN COUNT OF ;      18
  NAME = ACCEPT 2
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ALLOWED PROCESSORS =
      HOST
    REQUIRED MESSAGES =
      SBC 2 DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 ACCEPT 2
    ANDED SUCCESSORS =
      CHAIN TO ; STORE DATA
      WITH ITERATIONS THEN CHAIN COUNT OF ;      18
  NAME = ACCEPT 3
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ALLOWED PROCESSORS =
      HOST
    REQUIRED MESSAGES =
      SBC 3 DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 ACCEPT 3
    ANDED SUCCESSORS =
      CHAIN TO ; STORE DATA
      WITH ITERATIONS THEN CHAIN COUNT OF ;      18
  NAME = ACCEPT 4
    PRIORITY =     0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
```

```
    ALLOWED PROCESSORS =
      HOST
    REQUIRED MESSAGES =
      SBC 4 DATA
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 ACCEPT 4
    ANDED SUCCESSORS =
      CHAIN TO ; STORE DATA
      WITH ITERATIONS THEN CHAIN COUNT OF ;      18
  NAME = STORE DATA
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ORED PREDECESSOR LIST =
      ACCEPT 1
      ACCEPT 2
      ACCEPT 3
      ACCEPT 4
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 STORE DATA
  NAME = PROCESS 1
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ITERATION PERIOD = +9.0000000E+07 MICROSEC
    ALLOWED PROCESSORS =
      SBC 1
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 18 PROCESS 1
      EXECUTE A TOTAL OF ; 1 SEND 1
  NAME = PROCESS 2
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ITERATION PERIOD = +9.0000000E+07 MICROSEC
    ALLOWED PROCESSORS =
      SBC 2
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 18 PROCESS 2
      EXECUTE A TOTAL OF ; 1 SEND 2
  NAME = PROCESS 3
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ITERATION PERIOD = +9.0000000E+07 MICROSEC
    ALLOWED PROCESSORS =
      SBC 3
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 18 PROCESS 3
      EXECUTE A TOTAL OF ; 1 SEND 3
  NAME = PROCESS 4
    PRIORITY =      0
```

```
      INTERRUPTABILITY FLAG = NO
      CONCURRENT EXECUTION = NO
      ITERATION PERIOD = +9.0000000E+07 MICROSEC
      ALLOWED PROCESSORS =
        SBC 4
      INSTRUCTION LIST =
        EXECUTE A TOTAL OF ; 18 PROCESS 4
        EXECUTE A TOTAL OF ; 1 SEND 4

***** FILES - SYS.FILE.SET
SOFTWARE TYPE = FILE
  NAME = RADIOMETER DATA
      NUMBER OF BITS =            0.
      INITIAL RESIDENCY =
           HARD DISK
      READ ONLY FLAG = NO
```

```
                    PROCESSING ELEMENT UTILIZATION STATISTICS

                         FROM    0.   TO   7000. SECONDS

                       (ALL TIMES REPORTED IN MICROSECONDS)

PROCESSING ELEMENT NAME          HOST      SBC 1      SBC 2



STORAGE REQUESTS GRANTED          16         0          0
INTERRUPTED REQUESTS               0         0          0
AVERAGE WAIT TIME                  0.        0.         0.
MAXIMUM WAIT TIME                  0.        0.         0.
STD DEV WAIT TIME                  0.        0.         0.

GEN STORAGE REQUESTS               0         0          0

FILE REQUESTS GRANTED             16         0          0
INTERRUPTED REQUESTS               0         0          0
AVERAGE WAIT TIME                  0.        0.         0.
MAXIMUM WAIT TIME                  0.        0.         0.
STD DEV WAIT TIME                  0.        0.         0.

TRANSFER REQUESTS GRANTED         16        77         78
INTERRUPTED REQUESTS               0         0          0
AVERAGE WAIT TIME                  0.        0.         0.
MAXIMUM WAIT TIME                  0.        0.         0.
STD DEV WAIT TIME                  0.        0.         0.

INPUT CONTROLLER REQUESTS          0        77         78

DEST PE REQUESTS GRANTED           0         0          0
AVERAGE WAIT TIME                  0.        0.         0.
MAXIMUM WAIT TIME                  0.        0.         0.
STD DEV WAIT TIME                  0.        0.         0.

RESTARTED INTERRUPTS               0         0          0
AVG TIME PER INTERRUPT             0.        0.         0.
MAX TIME PER INTERRUPT             0.        0.         0.
STD DEV INTERRUPT TIME             0.        0.         0.
MAX INTERRUPT QUEUE SIZE           0         0          0
AVG INTERRUPT QUEUE SIZE           0.        0.         0.
STD DEV INTERRUPT QUEUE            0.        0.         0.

MAX MODULE QUEUE SIZE              1         1          1
AVG MODULE QUEUE SIZE              0.010     0.         0.
STD DEV MODULE QUEUE               0.100     0.         0.

PER CENT PE UTILIZATION            2.743    23.766     24.076
```

```
                    PROCESSING ELEMENT UTILIZATION STATISTICS

                          FROM    0.  TO  7000. SECONDS

                      (ALL TIMES REPORTED IN MICROSECONDS)

PROCESSING ELEMENT NAME        SBC 3              SBC 4



STORAGE REQUESTS GRANTED         0                  0
INTERRUPTED REQUESTS             0                  0
AVERAGE WAIT TIME                0.                 0.
MAXIMUM WAIT TIME                0.                 0.
STD DEV WAIT TIME                0.                 0.

GEN STORAGE REQUESTS             0                  0

FILE REQUESTS GRANTED            0                  0
INTERRUPTED REQUESTS             0                  0
AVERAGE WAIT TIME                0.                 0.
MAXIMUM WAIT TIME                0.                 0.
STD DEV WAIT TIME                0.                 0.

TRANSFER REQUESTS GRANTED       78                 77
INTERRUPTED REQUESTS             0                  0
AVERAGE WAIT TIME                0.                 0.
MAXIMUM WAIT TIME                0.                 0.
STD DEV WAIT TIME                0.                 0.

INPUT CONTROLLER REQUESTS       78                 77

DEST PE REQUESTS GRANTED         0                  0
AVERAGE WAIT TIME                0.                 0.
MAXIMUM WAIT TIME                0.                 0.
STD DEV WAIT TIME                0.                 0.

RESTARTED INTERRUPTS             0                  0
AVG TIME PER INTERRUPT           0.                 0.
MAX TIME PER INTERRUPT           0.                 0.
STD DEV INTERRUPT TIME           0.                 0.
MAX INTERRUPT QUEUE SIZE         0                  0
AVG INTERRUPT QUEUE SIZE         0.                 0.
STD DEV INTERRUPT QUEUE          0.                 0.

MAX MODULE QUEUE SIZE            1                  1
AVG MODULE QUEUE SIZE            0.                 0.
STD DEV MODULE QUEUE             0.                 0.

PER CENT PE UTILIZATION         24.074             23.766
```

```
                       COMPLETED INSTRUCTION REPORT

                       FROM   0.  TO  7000. SECONDS

  INSTRUCTION NAME     COUNT     INSTRUCTION NAME     COUNT

HOST
  STORE DATA              16     ACCEPT 1                77
  ACCEPT 2                78     ACCEPT 3                78
  ACCEPT 4                77

SBC 1
  SEND 1                  77     PROCESS 1             1386

SBC 2
  SEND 2                  78     PROCESS 2             1404

SBC 3
  SEND 3                  78     PROCESS 3             1404

SBC 4
  SEND 4                  77     PROCESS 4             1386
```

```
                        STORAGE DEVICE UTILIZATION STATISTICS

                            FROM    0.   TO   7000. SECONDS

                        (ALL TIMES REPORTED IN MICROSECONDS)


STORAGE DEVICE NAME          HARD DISK




STORAGE REQUESTS GRANTED          16
INTERRUPTED REQUESTS               0
AVG REQUEST DELAY                  0.
MAX REQUEST DELAY                  0.
STD DEV REQUEST DELAY              0.

GEN STORAGE REQUESTS               0

FILE REQUESTS GRANTED             16
INTERRUPTED REQUESTS               0
AVG FILE DELAY                     0.
MAXIMUM FILE DELAY                 0.
STD DEV FILE DELAY                 0.

INTERRUPTED ACCESSES               0

COMPLETED ACCESSES                16
AVERAGE USAGE TIME          12000000.000
MAXIMUM USAGE TIME          12000000.000
STD DEV USAGE TIME                 0.

BITS NOW IN USE               102400.
PER CENT NOW USED                  0.427

AVERAGE BITS USED              40027.706
MAXIMUM BITS USED             102400.
STD DEV BITS USED              29808.937

PER CENT OF TIME BUSY              2.743

                               HOST - SBC Simulation


                       TRANSFER DEVICE UTILIZATION STATISTICS

                            FROM    0.   TO   7000. SECONDS
```

```
                    (ALL TIMES REPORTED IN MICROSECONDS)


TRANSFER DEVICE NAME              BUS        COM 1      COM 2


TRANSFER REQUESTS GRANTED          16         77         78
INTERRUPTED REQUESTS                0          0          0
AVG REQUEST DELAY                   0.         0.         0.
MAX REQUEST DELAY                   0.         0.         0.
STD DEV REQUEST DELAY               0.         0.         0.

INTERRUPTED TRANSFERS               0          0          0
COLLISIONS

COMPLETED TRANSFERS                16         77         78
AVG USAGE TIME            12000000.000 5250.000   6300.000
MAX USAGE TIME            12000000.000 5250.000   6300.000
STD DEV USAGE TIME                  0.         0.         0.

AVG QUEUE SIZE                      0.         0.         0.
MAX QUEUE SIZE                      1.000      1.000      1.000
STD DEV QUEUE SIZE                  0.         0.         0.

PER CENT OF TIME BUSY               2.743      0.006      0.007
```

```
                         TRANSFER DEVICE UTILIZATION STATISTICS

                              FROM   0.  TO  7000. SECONDS

                          (ALL TIMES REPORTED IN MICROSECONDS)


TRANSFER DEVICE NAME           COM 3              COM 4



TRANSFER REQUESTS GRANTED        78                 77
INTERRUPTED REQUESTS              0                  0
AVG REQUEST DELAY                 0.                 0.
MAX REQUEST DELAY                 0.                 0.
STD DEV REQUEST DELAY             0.                 0.

INTERRUPTED TRANSFERS             0                  0
COLLISIONS

COMPLETED TRANSFERS              78                 77
AVG USAGE TIME                5250.000           5250.000
MAX USAGE TIME                5250.000           5250.000
STD DEV USAGE TIME                0.                 0.

AVG QUEUE SIZE                    0.                 0.
MAX QUEUE SIZE                    1.000              1.000
STD DEV QUEUE SIZE                0.                 0.

PER CENT OF TIME BUSY             0.006              0.006
```

```
                      COMPLETED MODULE STATISTICS

                      FROM    0.  TO  7000. SECONDS

                   (ALL TIMES REPORTED IN MICROSECONDS)


MODULE NAME              ACCEPT 1     STORE DATA      ACCEPT 2



HOST PE                    HOST          HOST           HOST

 COMPLETED EXECUTIONS       77            16             78

CANCELLATIONS DUE TO
  ITERATION PERIOD          0             0              0
  RUN UNTIL SEMAPHORES      0             0              0
  MESSAGE REQUIREMENTS      0             0              0
  SUCCESSOR ACTIVATION      0             0              0

NUM PRECONDITION TIME       77            16             78
AVG PRECONDITION TIME       0.            0.        394773.461
MAX PRECONDITION TIME       0.            0.       7698082.499
MIN PRECONDITION TIME       0.            0.             0.
STD DEV PRECOND TIME        0.            0.       1697984.861

AVG EXECUTION TIME        100.000   12000000.000       100.000
MAX EXECUTION TIME        100.000   12000000.000       100.000
MIN EXECUTION TIME        100.000   12000000.000       100.000
STD DEV EXECUTION TIME      0.            0.             0.

RESTARTED INTERRUPTS        0             0              0
AVG TIME PER INTERRUPT      0.            0.             0.
MAX TIME INTERRUPTED        0.            0.             0.
STD DEV INTERRUPT TIME      0.            0.             0.
```

```
                         COMPLETED MODULE STATISTICS

                         FROM   0.  TO  7000. SECONDS

                       (ALL TIMES REPORTED IN MICROSECONDS)


MODULE NAME              ACCEPT 3        ACCEPT 4   PROCESS 1


HOST PE                     HOST            HOST       SBC 1

COMPLETED EXECUTIONS         78              77          77

CANCELLATIONS DUE TO
  ITERATION PERIOD            0               0           0
  RUN UNTIL SEMAPHORES        0               0           0
  MESSAGE REQUIREMENTS        0               0           0
  SUCCESSOR ACTIVATION        0               0           0

NUM PRECONDITION TIME        78              77          77
AVG PRECONDITION TIME    194693.724      317145.814       0.
MAX PRECONDITION TIME   3796527.609     6105056.915       0.
MIN PRECONDITION TIME         0.              0.          0.
STD DEV PRECOND TIME     837409.369     1354847.510       0.

AVG EXECUTION TIME          100.0           100.0  21605250.000
MAX EXECUTION TIME          100.0           100.0  21605250.000
MIN EXECUTION TIME          100.0           100.0  21605250.000
STD DEV EXECUTION TIME        0.              0.          0.

RESTARTED INTERRUPTS          0               0           0
AVG TIME PER INTERRUPT        0.              0.          0.
MAX TIME INTERRUPTED          0.              0.          0.
STD DEV INTERRUPT TIME        0.              0.          0.
```

```
                            COMPLETED MODULE STATISTICS

                            FROM   0.  TO  7000. SECONDS

                          (ALL TIMES REPORTED IN MICROSECONDS)


MODULE NAME              PROCESS 2      PROCESS 3      PROCESS 4


HOST PE                      SBC 2          SBC 3          SBC 4

COMPLETED EXECUTIONS            78             78             77

CANCELLATIONS DUE TO
  ITERATION PERIOD               0                             0
  RUN UNTIL SEMAPHORES           0              0              0
  MESSAGE REQUIREMENTS           0              0              0
  SUCCESSOR ACTIVATION           0              0              0

NUM PRECONDITION TIME           78             78             77
AVG PRECONDITION TIME           0.             0.             0.
MAX PRECONDITION TIME           0.                            0.
MIN PRECONDITION TIME           0.             0.             0.
STD DEV PRECOND TIME            0.             0.             0.

AVG EXECUTION TIME     21606300.000   21605250.000   21605250.000
MAX EXECUTION TIME     21606300.000   21605250.000   21605250.000
MIN EXECUTION TIME     21606300.000   21605250.000   21605250.000
STD DEV EXECUTION TIME          0.             0.             0.

RESTARTED INTERRUPTS             0              0              0
AVG TIME PER INTERRUPT          0.             0.             0.
MAX TIME INTERRUPTED            0.             0.             0.
STD DEV INTERRUPT TIME          0.             0.             0.
```

APPENDIX E

SBC-DSP NETWORK II SIMULATION

```
* DSP - SBC   5 Second IP Simulation

***** PROCESSING ELEMENTS - SYS.PE.SET
HARDWARE TYPE = PROCESSING
  NAME = DSP
    BASIC CYCLE TIME =      0.200000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; DATA READY
          MESSAGE ; DATA AVAILABLE
          LENGTH ; 16 BITS
          DESTINATION PROCESSOR ; SBC
          QUEUE FLAG ; NO
          ALLOWABLE BUSSES ;
            PARALLEL
      INSTRUCTION TYPE = PROCESSING
        NAME ; FILTER
          TIME ; 80 CYCLES
  NAME = SBC
    BASIC CYCLE TIME =   2000.000000 MICROSEC
    INPUT CONTROLLER = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = PROCESSING
        NAME ; READ SENSOR
          TIME ; 160 CYCLES
        NAME ; PROCESS SENSOR DATA
          TIME ; 200 CYCLES
        NAME ; STORE DATA
          TIME ; 20 CYCLES
        NAME ; GET FILTERED VIDEO
          TIME ; 5 CYCLES
        NAME ; PROCESS VIDEO
          TIME ; 200 CYCLES

***** BUSSES - SYS.BUS.SET
HARDWARE TYPE = DATA TRANSFER
  NAME = PARALLEL
    CYCLE TIME =      0.20 MICROSEC
    BITS PER CYCLE =      16
    CYCLES PER WORD =       1
```

```
      WORDS PER BLOCK =        1
      WORD OVERHEAD TIME =      0.   MICROSEC
      BLOCK OVERHEAD TIME =      0.   MICROSEC
      PROTOCOL = FIRST COME FIRST SERVED
      BUS CONNECTIONS =
        DSP
        SBC

***** MODULES - SYS.MODULE.SET
SOFTWARE TYPE = MODULE
  NAME = DSP MODULE
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ITERATION PERIOD = +5.0000000E+06 MICROSEC
    ALLOWED PROCESSORS =
      DSP
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 FILTER
      EXECUTE A TOTAL OF ; 1 DATA READY
  NAME = SBC MODULE
    PRIORITY =      0
    INTERRUPTABILITY FLAG = NO
    CONCURRENT EXECUTION = NO
    ALLOWED PROCESSORS =
      SBC
    REQUIRED MESSAGES =
      DATA AVAILABLE
    INSTRUCTION LIST =
      EXECUTE A TOTAL OF ; 1 READ SENSOR
      EXECUTE A TOTAL OF ; 1 STORE DATA
      EXECUTE A TOTAL OF ; 1 PROCESS SENSOR DATA
      EXECUTE A TOTAL OF ; 1 GET FILTERED VIDEO
      EXECUTE A TOTAL OF ; 1 PROCESS VIDEO
```

```
                  PROCESSING ELEMENT UTILIZATION STATISTICS

PROCESSING ELEMENT NAME        DSP              SBC

STORAGE REQUESTS GRANTED        0                0
INTERRUPTED REQUESTS            0                0
AVERAGE WAIT TIME               0.               0.
MAXIMUM WAIT TIME               0.               0.
STD DEV WAIT TIME               0.               0.

GEN STORAGE REQUESTS            0                0

FILE REQUESTS GRANTED           0                0
INTERRUPTED REQUESTS            0                0
AVERAGE WAIT TIME               0.               0.
MAXIMUM WAIT TIME               0.               0.
STD DEV WAIT TIME               0.               0.

TRANSFER REQUESTS GRANTED       5                0
INTERRUPTED REQUESTS            0                0
AVERAGE WAIT TIME               0.               0.
MAXIMUM WAIT TIME               0.               0.
STD DEV WAIT TIME               0.               0.

DEST PE REQUESTS GRANTED        5                0
AVERAGE WAIT TIME               0.               0.
MAXIMUM WAIT TIME               0.               0.
STD DEV WAIT TIME               0.               0.

RESTARTED INTERRUPTS            0                0
AVG TIME PER INTERRUPT          0.               0.
MAX TIME PER INTERRUPT          0.               0.
STD DEV INTERRUPT TIME          0.               0.
MAX INTERRUPT QUEUE SIZE        0                0
AVG INTERRUPT QUEUE SIZE        0.               0.
STD DEV INTERRUPT QUEUE         0.               0.

MAX MODULE QUEUE SIZE           1                1
AVG MODULE QUEUE SIZE           0.               0.
STD DEV MODULE QUEUE            0.               0.

PER CENT PE UTILIZATION         0.000           22.500
                          COMPLETED INSTRUCTION REPORT

  INSTRUCTION NAME      COUNT    INSTRUCTION NAME     COUNT

DSP
  DATA READY                5    FILTER                   5

SBC
  READ SENSOR               5    PROCESS SENSOR DATA      5
```

```
  STORE DATA                  5     GET FILTERED VIDEO       5
  PROCESS VIDEO               5


                         TRANSFER DEVICE UTILIZATION STATISTICS

TRANSFER DEVICE NAME          PARALLEL

TRANSFER REQUESTS GRANTED        5
INTERRUPTED REQUESTS             0
AVG REQUEST DELAY                0.
MAX REQUEST DELAY                0.
STD DEV REQUEST DELAY            0.

INTERRUPTED TRANSFERS            0
COLLISIONS

COMPLETED TRANSFERS              5
AVG USAGE TIME                   0.200
MAX USAGE TIME                   0.200
STD DEV USAGE TIME               0.

AVG QUEUE SIZE                   0.
MAX QUEUE SIZE                   1.000
STD DEV QUEUE SIZE               0.

PER CENT OF TIME BUSY            0.000
```

```
                       COMPLETED MODULE STATISTICS


MODULE NAME                       DSP MODULE        SBC MODULE


HOST PE                              DSP                SBC

COMPLETED EXECUTIONS                  5                  5

CANCELLATIONS DUE TO
  ITERATION PERIOD                    0                  0
  RUN UNTIL SEMAPHORES                0                  0
  MESSAGE REQUIREMENTS                0                  0
  SUCCESSOR ACTIVATION                0                  0

NUM PRECONDITION TIME                 5                  5
AVG PRECONDITION TIME                 0.                 0.
MAX PRECONDITION TIME                 0.                 0.
MIN PRECONDITION TIME                 0.                 0.
STD DEV PRECOND TIME                  0.                 0.

AVG EXECUTION TIME                   16.200         1170000.000
MAX EXECUTION TIME                   16.200         1170000.000
MIN EXECUTION TIME                   16.200         1170000.000
STD DEV EXECUTION TIME                0.                 0.

RESTARTED INTERRUPTS                  0                  0
AVG TIME PER INTERRUPT                0.                 0.
MAX TIME INTERRUPTED                  0.                 0.
STD DEV INTERRUPT TIME                0.                 0.
```

REFERENCES

Artwick, Bruce A. *Microcomputer Interfacing*. Englewood Cliffs: Prentice-Hall, Inc., 1980.

Blaskovic, M., Box, T.C., and Rogers, R. R. "Radiometric Observations During CASP," *Stormy Weather Group Scientific Report*, McGill University (July 1987).

Camp, R. C., Smay, T. A., and Triska, C. J. *Microcomputer Systems Principles*. Portland: Matrix Publishers, Inc., 1978.

DATEL (1979), Data Acquisition and Conversion Handbook, Manefield, Massachusetts.

Duncan, Ray *Advanced MS-DOS*. Bellevue: Microsoft Press, 1986.

Elachi, Charles *Introduction to the Physics and Techniques of Remote Sensing*. New York: John Wiley and Sons, Inc., 1987.

Evans, G., and McLeish, C. W. *RF Radiometer Handbook*. Dedham: Artech House, Inc., 1977.

Hayes-Roth, Frederick, Waterman, Donald A., and Lenant, Douglas B. *Building Expert Systems*. London: Addison-Wesley Publishing Company, Inc., 1983.

Hobbs, L.C., Theis, D.J, Trimble, Joel, Titus, Harold, and Highberg, Ivar *Parallel Processor Systems, Technologies, and Applications*. London: Macmillan and Company, Ltd., 1970.

Huelsman, Lawrence P. *Active Filters: Lumped, Distributed, Integrated, Digital, and Parametric*. New York: McGraw-Hill Book Company, 1970.

Huelsman, Lawrence P. *Theory and Design of Active RC Circuits*. New York: McGraw-Hill Book Company, 1968.

Huelsman, L.P., and Allen, P.E. *Introduction to the Theory and Design of Active Filters*. New York: McGraw-Hill Book Company, 1980.

Hufault, John R. *OP AMP Network Design Manual*. Tucson: Marketpul Corporation, 1982.

Intel Corporation. (1985), MCS BASIC-52 Users Manual, Santa Clara, California.

Kernighan, Brian W., and Ritchie, Dennis M. The C Programming Language. Englewood Cliffs: Prentice-Hall, Inc., 1978.

Lemone, Karen A. Assembly Language and Systems Programming for the IBM PC and Compatibles. Boston: Little, Brown and Company, 1985.

Liebowitz, Burt H., and Carson, John H. Multiple Processor Systems for Real-Time Applications. Englewood Cliffs: Prentice-Hall, Inc., 1985.

Manassewitsch, Vadim Frequency Synthesizers - Theory and Design. New York: John Wiley and Sons, Inc., 1980.

The Micromint, Inc. (1985), BCC52 BASIC Computer/Controller Users Manual, Rev. 2.0, Vernon, Connecticut.

The Micromint, Inc. (1986), BCC30 12 Bit - 16 Channel A/D Converter Users Manual, Rev. 1.0, Vernon, Connecticut.

Monolithic Memories, Inc. (1983), Programmable Array Logic Handbook, Third Edition, Santa Clara, California.

Norton, Peter Programmer's Guide to the IBM PC. Bellevue: Microsoft Press, 1985.

Oppenheim, Alan V., and Schafer, Ronald W. Digital Signal Processing. Englewood Cliffs: Prentice-Hall, Inc., 1975.

Rector, Russel, and Alexy, George THE 8086 Book. Berkeley: Osborne/McGraw-Hill 1980.

Schustack, Steve Variations in C. Bellevue: Microsoft Press, 1985.

Texas Instruments, Inc. (1986), Digital Signal Processing Applications with the TMS320 Family, Houston, Texas.

Texas Instruments, Inc. (1984), Implementation of FIR/IIR Filters, Houston, Texas.

Texas Instruments, Inc. (1984), TMS32010 Assembly Language Programmes's Guide, Houston, Texas.

Texas Instruments, Inc. (1985), TMS32010 User's Guide, Houston, Texas.

Thomson, D.W. "New Perspectives on Atmospheric Structure and Dynamics", Vol. 57, No.1 (1987/88), Pennsylvania State University, College of Earth and Mineral Sciences.

Tobey, Gene E., Graeme, Jerald G., and Huelsman, Lawrence P. Operational Amplifiers. New York: McGraw-Hill Book Company, 1971.

Ulaby, Fawwaz T., Moore, Richard K., and Fung, Adrian K. Microwave Remote Sensing - Active and Passive, Vol. I. Reading: Addison-Wesley Publishing Co., 1981.

Ward, Robert Debugging C. Indianapolis: QUE Corporation, 1986.