UNIVERSITY OF ILLINOIS
URBANA

AERONOMY REPORT NO. 96

A ROCKET-BORNE MICROPROCESSOR-BASED EXPERIMENT FOR INVESTIGATION OF ENERGETIC PARTICLES IN THE D AND E REGIONS

by
F. M. Braswell
L. G. Smith

May 1, 1981

Library of Congress ISSN 0568-0581

Aeronomy Laboratory
Department of Electrical Engineering
University of Illinois
Urbana, Illinois

Supported by
National Aeronautics and Space Administration

UILU-ENG-81 2502

AERONOMY REPORT

NO. 96

A ROCKET-BORNE MICROPROCESSOR-BASED EXPERIMENT FOR INVESTIGATION OF ENERGETIC PARTICLES IN THE *D* AND *E* REGIONS

by

F. M. Braswell
L. G. Smith

Supported by
National Aeronautics
and Space Administration
Grant NGR 14-005-181

Aeronomy Laboratory
Department of Electrical Engineering
University of Illinois
Urbana, Illinois

ABSTRACT

This report describes the realization of an energetic particle experiment using the Z80 family of microcomputer components. Data collected from the experiment allows fast and efficient postprocessing, yielding both energy-spectrum and pitch-angle distribution of energetic particles in the *D* and *E* regions. Advanced microprocessor system architecture and software concepts are used in the design to cope with the large amount of data being processed. This requires the Z80 system to operate at over 80% of its total capacity.

The microprocessor system was included in the payloads of three rockets launched during the Energy Budget Campaign at ESRANGE, Kiruna, Sweden in November 1980. Based on a preliminary examination of the data, the performance of the experiment was satisfactory and good data were obtained on the energy spectrum and pitch-angle distribution of the particles.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Study of energetic particles in the *D* and *E* regions require several pieces of information, including particle type, energy spectrum and pitch angle distribution.

The microprocessor experiment described in this report gives good spectrum information between the limits of about 14 to 128 keV. Also available is pitch angle information from four detectors in 32 sectors of rocket azimuth. The types of particles can be deduced from the comparison of flux in pairs of detectors.

Section 2.1 demonstrates the importance of energetic particle information in relation to the SAR arc phenomena occurring at middle latitudes. Data from the microprocessor experiment not only holds the key to the SAR arc, but also to auroral events.

The first-generation microprocessor experiment was actually launched into a SAR arc (see Section 2.1.4), while the second-generation system, described in this report, was launched into auroral activity near Kiruna, Sweden, and showed very high count rates (see Chapter 6).

The first-generation microprocessor experiment is briefly discussed in Section 2.2 leading up to a general discussion of the second-generation system in Section 2.3. A detailed account of the system is provided in Chapters 3 and 4.

Chapter 5 outlines the use of various diagnostic tools used in the development of the microprocessor system.

A discussion of the testing and calibration of the experiment can be found in Chapter 6 together with preliminary results from Taurus Orion 33.009. The flight data lead to the conclusion that the experiment was successful in providing good spectrum and pitch angle information (Chapter 7).

The microprocessor system can be used in future flights with almost no changes, however the continuing advancements in VLSI technology and computer architecture will no doubt result in a third-generation microprocessor system. Two interesting possibilities are given in Chapter 7.

2. ENERGETIC PARTICLES IN THE *D* AND *E* REGIONS

2.1 *Introduction*

Some of the most magnificent atmospheric displays are the polar auroras which have been observed for many years, yet just beyond the visual intensity threshold lies perhaps an ever more spectacular display. The Stable Auroral Red (SAR) arc or M arc, which occurs at middle latitudes, has an amazing spectral purity (630 nm), and may persist through an entire night in contrast with the more transient polar auroras.

The SAR arc was first observed by Barbier in 1956 from Haute Provence in southern France [*Barbier*, 1958] and its origin has puzzled scientists for many years.

2.1.1 *Observed features.* SAR arcs generally occur at middle latitudes and are distinct from their polar counterparts which occur several degrees of latitude northward [*Hoch and Clark*, 1970]. Their position corresponds to geomagnetic *L*-shell values between 2 and 4.

The arcs are usually stable, homogeneous, and extend several hundred kilometers in the north-south direction occurring in the region from about 300 to 700 km in height.

Current observations show the arcs extend at least around the night side of the earth and probably encircle the earth. The OGO 4 satellite showed that the arc of September 1967 was globe encircling [*Reed and Blamont*, 1968].

The arcs are almost always accompanied by poleward aurora; however there is not enough evidence that they always occur together [*Hoch and Clark*, 1970].

The signature of every SAR arc is its characteristic spectral emission of the atomic oxygen transition $O\ I(^3P - {}^1D_2)$ at 630 nm (see Figure 2.1). The absence of any other spectral lines, most importantly the 557.7 nm line of oxygen, implies a low-energy excitation of the oxygen atom since only 1.97 eV are required for the ^1D transition. Spectroscopic studies of the March 1970 SAR arc show the 630 nm line by far the dominant radiation with slight traces of (*OI*) 557.7 nm and (*NI*) 520 nm [*Hernandez*, 1972].

The 630 nm emission rate ranges from barely detectable above the airglow level to several kilorayleighs with the most intense SAR arc recorded at 18 kR in August 1972 [*Shephard et al.*, 1976]. The mean intensity during the last solar cycle was about 6 kR [*Roach and*

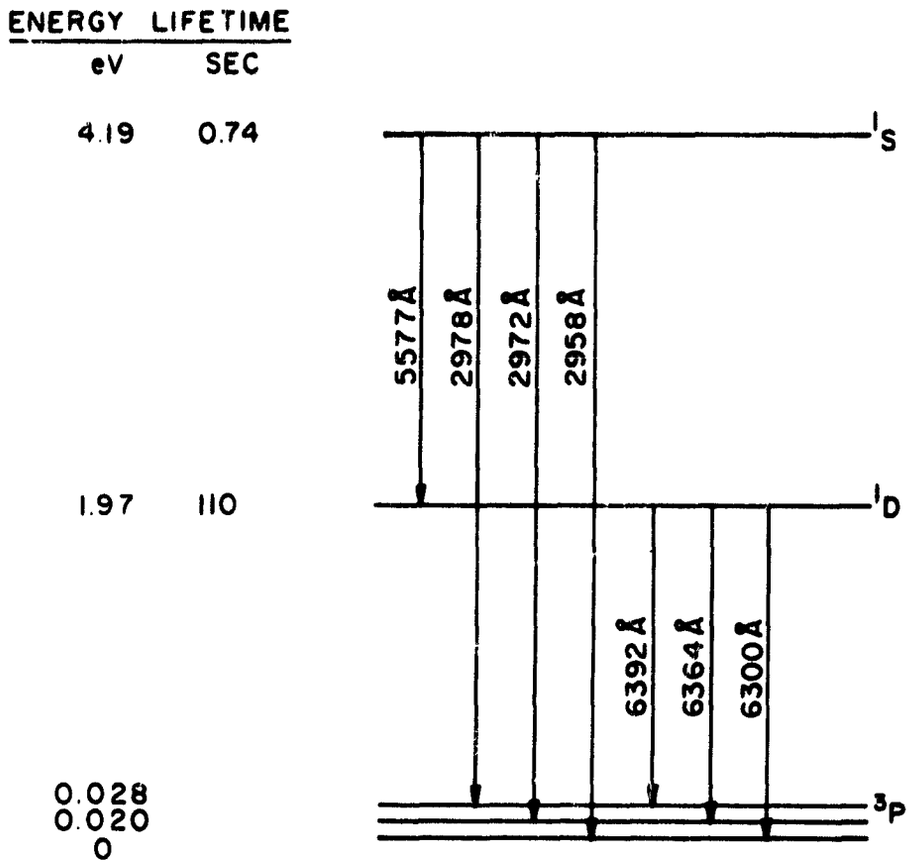


Figure 2.1 Energy level diagram of the low-lying states of atomic oxygen.

Roach, 1963], however, the mean intensity is much lower this cycle.

The fact that SAR arcs are only observed during times of magnetic storms indicates a strong correlation between SAR arc occurrence, solar activity, and the solar cycle [*Rees and Akasofu*, 1963].

SAR arcs generally persist throughout the night and the mean lifetime has been set at about 10-12 hours [*Roach and Roach*, 1963].

Concurrent sightings of SAR arcs in both Northern and Southern Hemispheres at about the same geomagnetic L-values would indicate magnetic conjugacy. The M bands in Figure 2.2 indicate the regions of occurrence of SAR arcs.

Increased electron temperatures in the F region were found during the geomagnetic storms of June, August, and September of 1965 using the incoherent Thomson scatter technique [*Evans*, 1970]. Satellite measurements have shown that the electron temperatures inside the arc are enhanced relative to surrounding regions (see Figure 2.3).

The electron density in the region of the SAR arc, at the $F2$ peak, is observed to be considerably reduced compared to densities outside the arc [*Rees and Roble*, 1975].

Ion temperatures within SAR arcs have always been measured lower than the electron temperatures while neutral temperatures show very little enhancement.

An ac electric field was measured by the OGO 6 satellite during the SAR arc of August 8-9, 1970. The magnitude of the field was about 100 mV m^{-1} greater within the SAR arc than outside the region of the arc. No dc field enhancements were detected [*Nagy et al.*, 1972].

The position and brightness of the March 8-9, 1970 SAR arc were correlated with changes in the H and D components of the ground-based geomagnetic field deviations [*Okuda et al.*, 1971].

2.1.2 *Proposed hypotheses.* Prior to observations of high electron temperatures in the region of the SAR arc exothermic reactions of oxygen were thought to be the source of energy driving the arc.

The first hypothesis was put forth by *King and Roach* [1961] and involved the pair of reactions:

$$O^+ + N_2 \rightarrow NO^+ + N \qquad (2.1)$$

and

$$NO^+ + e \rightarrow N + O + 2.72 \text{ eV} \qquad (2.2)$$

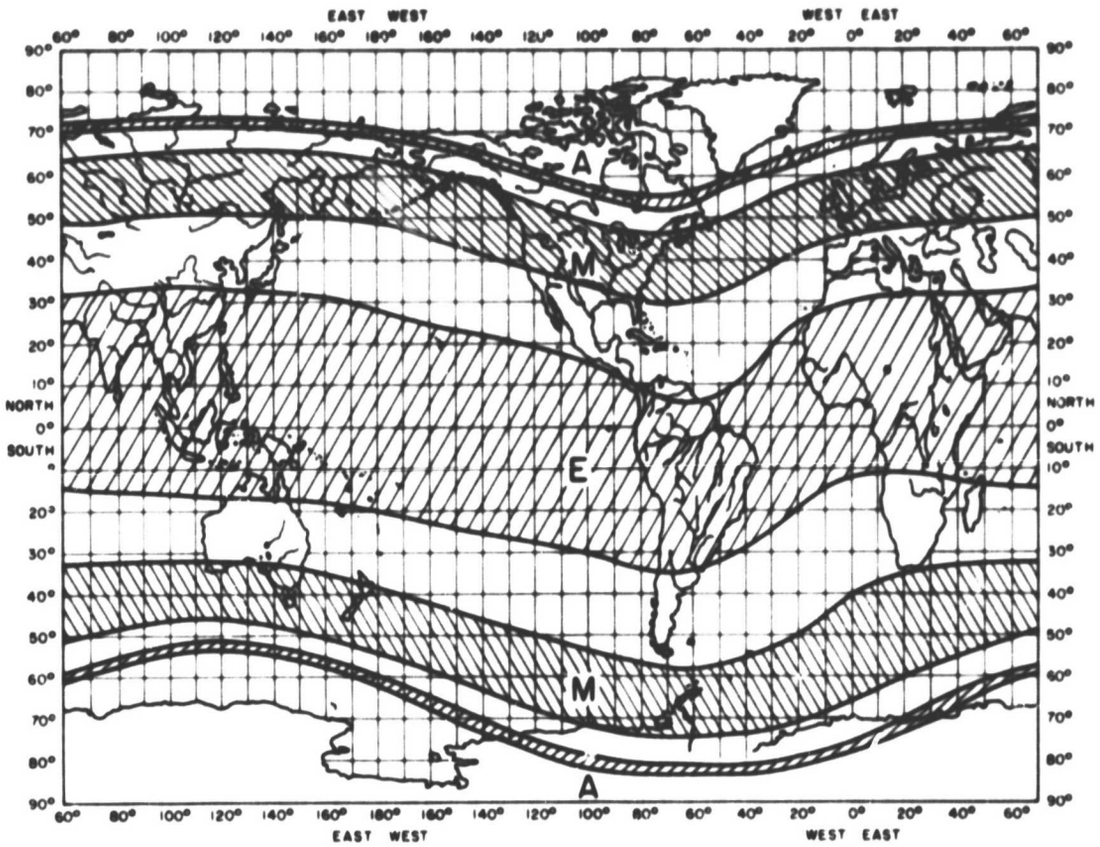


Figure 2.2 Zones of 630 nm emission enhancements: E, equatorial arcs; M, SAR arcs; A, aurora [*Roach and Roach*, 1963].

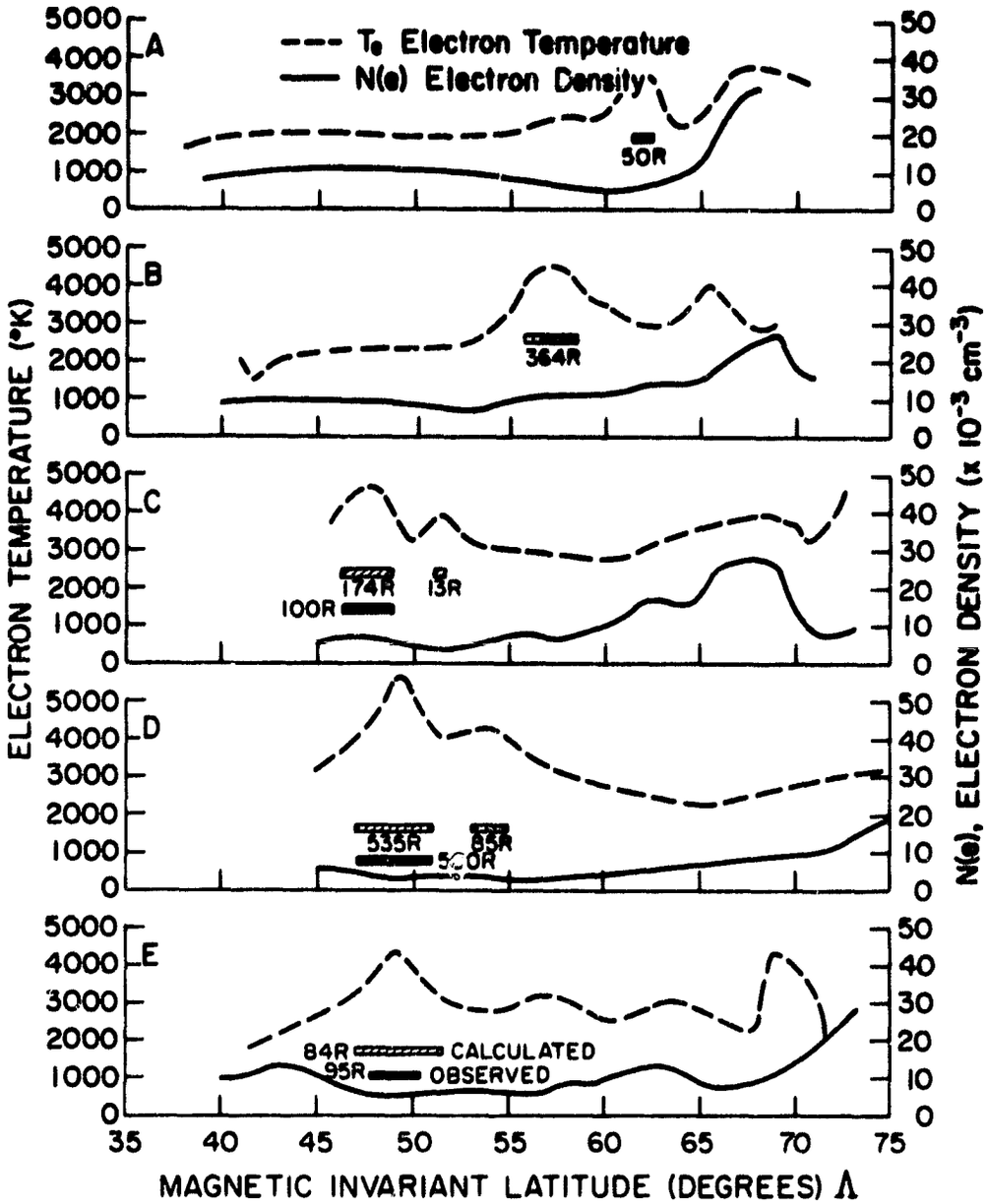


Figure 2.3 The electron temperature and electron density measured by the Alouette 2 satellite as a function of the magnetic invariant latitude. Curves A-E were measured at different times during the geomagnetic storm of October 29 - November 2, 1968. The hatched areas indicate the calculated 630 nmemission rate in rayleighs, and the solid areas indicate the 630 nm emission rate measured from airglow observatories [*Roble et al.*, 1971].

The second reaction will leave enough energy to excite oxygen to the $O(^1D)$ state, but not the $O(^1S)$ state thus explaining the 630 nm emission line. It was later pointed out by *Dalgarno and Walker* [1964] that the second equation has a very low reaction rate and could not account for the production of the 630 nm wavelength.

The following reactions were also proposed:

$$O^+ + O_2 \rightarrow O_2^+ + O \tag{2.3}$$

$$O_2^+ + e \rightarrow O + O + 6.9 \text{ eV} \tag{2.4}$$

but were rejected since 6.9 eV is sufficient to produce the 557.7 nm line which is not observed.

Rees [1961] and others proposed that an electric field acting perpendicular to the geomagnetic field lines would provide energy to the ambient electrons and ions sufficient to excite the oxygen atoms to the $O(^1D)$ state. A field on the order of 100 mV m^{-1} could excite the arc [*Megill and Carleton*, 1964]. It was later shown that this hypothesis is inconsistent with satellite and ground observations.

Cole [1965] proposed a heat conduction mechanism in which *F*-region electron temperature is maintained by conduction of heat from the magnetosphere along geomagnetic field lines into the ionosphere. The SAR arc is then excited by energetic electron impact with atomic oxygen. This hypothesis is somewhat supported by the observations of *Evans* [1970].

Coulomb dissipation alone may not be sufficient to supply SAR arc heating rates and therefore another energy transfer mechanism between energetic ring current protons and plasmapause electrons is proposed.

During the main phase of a magnetic storm the ring current is enhanced by protons injected from the magnetospheric tail. During the recovery the plasmapause expands slowly outward eroding the symmetric ring current along the inner edge.

It has been argued that ring current protons dissipate most of their energy into ion-cyclotron wave turbulence instead of Coulomb interactions when ring current instabilities occur [*Cornwall et al.*, 1971]. Particle precipitation will also accompany due to pitch-angle scattering.

It has been postulated that the ion-cyclotron wave turbulence is an important source of heat for the SAR arc electrons.

There are several steps in which the energy is transferred: ring current protons just inside the plasmapause give energy to ion cyclotron waves and some are precipitated by pitch-angle scattering; these waves are then Landau absorbed by thermal electrons, and the heat is then thermally conducted to the ionosphere. Proton Coulomb dissipation is still important, however, for warming plasmaspheric electrons [*Cornwall et al.*, 1971].

Near the boundary of the plasmapause and ring current instabilities occur and proton Coulomb dissipation begins electron heating. Landau resonant energy exchange will not become dominant until the electron energy exceeds 0.6 eV [*Cornwall et al.*, 1971]. The electron temperature and heat flux will continue to increase until it is balanced by strong diffusion heat conduction to the ionosphere.

According to *Rees and Roble* [1975] electron heating was not needed as a primer for Landau resonant energy exchange to take place at the plasmapause.

It was argued that proton Coulomb interactions were not a sufficient source of heat for magnetospheric electrons based on measurements by *Smith and Hoffman* [1973] and *Williams et al.* [1973] during the December 17-19, 1971, geomagnetic storm and the ion-cyclotron wave hypothesis was further developed as the principal source of electron heating

It was again pointed out that Coulomb collisions of ring current particles may be a sufficient heat source after all. Calculations by *Rees and Roble* [1975] estimated heat exchange only for conditions near the equatorial plane, where as *Cole's* [1965] estimate was made all along the tube of force. It is shown that ring current particles exchange energy with background electrons at a greater rate the further they are from the equatorial plane, thus increasing the heat available by Coulomb collisions. It was suggested that if ion-cyclotron radiation be involved, it would be an additional energy source but not the major transmitter of energy [*Cole*, 1975].

The main problem with the previous theories of *Cole* [1975] and *Cornwall et al.* [1971] is that electrons heated in the equatorial region of the magnetopause must transfer their heat down to the ionosphere.

Anomalous transport produced by kinetic Alfven wave turbulence was proposed as a solution to this problem.

A kinetic Alfven wave is a shear Alfven wave which has certain kinetic properties and meets some special requirements. This shear Alfven wave has

a finite perpendicular wavelength on the order of the ion gyroradius and can cause anomalous transport through Cerenkov resonance [*Hasegawa and Chen*, 1975].

There are two mechanisms available for excitation of kinetic Alfven waves. It can be excited by a magneto-hydrodynamic surface wave through a resonant mode conversion process and also by drift wave instability in a plasma with $\beta(=2\mu_0 n_0 T/B_0^2)$ larger than the electron-to-ion mass ratio m_e/m_i [*Mikhailovskii*, 1967].

It is shown that these waves are universally excited and can produce localized enhancement of electron temperature along with being selective in L value.

The kinetic Alfven wave has a wavelength comparable to the length of the geomagnetic field line and is accompanied by a parallel electric field. Resonant heating of electrons occurs at the plasmapause and the particles are easily transported down the field lines [*Hasegawa and Mima*, 1978] thus explaining SAR arc phenomena.

2.1.3 *Comparisons of the various hypotheses.* Figures 2.4 and 2.5 give different perspectives of SAR arc formation in relation to ring current instabilities and illustrate the hypotheses of *Cornwall et al.* [1971] and *Rees and Roble* [1975].

The similarities and differences between the current hypotheses of *Cornwall et al.* [1971], *Rees and Roble* [1975], *Cole* [1975] and *Hasegawa and Mima* [1978] can easily be pointed out using the block diagrams of Figures 2.6, 2.7, and 2.8.

All agree on the necessity of geomagnetic disturbance, electron energies of approximately 2 eV in the upper F region, collisional excitation of atomic oxygen to the $O(^1D)$ state, and 630 nm emission in the 300 to 700 km region. Of the four hypotheses Hasegawa's is most different (Figure 2.8).

Figure 2.6 illustrates the hypotheses of both *Cole* [1975] and *Rees and Roble* [1975]. While the figure is basically the same for both, their hypotheses are quite different. The disagreement occurs over the importance of ion-cyclotron radiation with Landau damping and Coulomb interaction between ring current protons and plasmapause electrons. Cole claims that Coulomb interaction is by far the dominant heat source for electrons in the plasmapause while Rees and Roble argue that ion-cyclotron radiation with Landau damping is dominant.

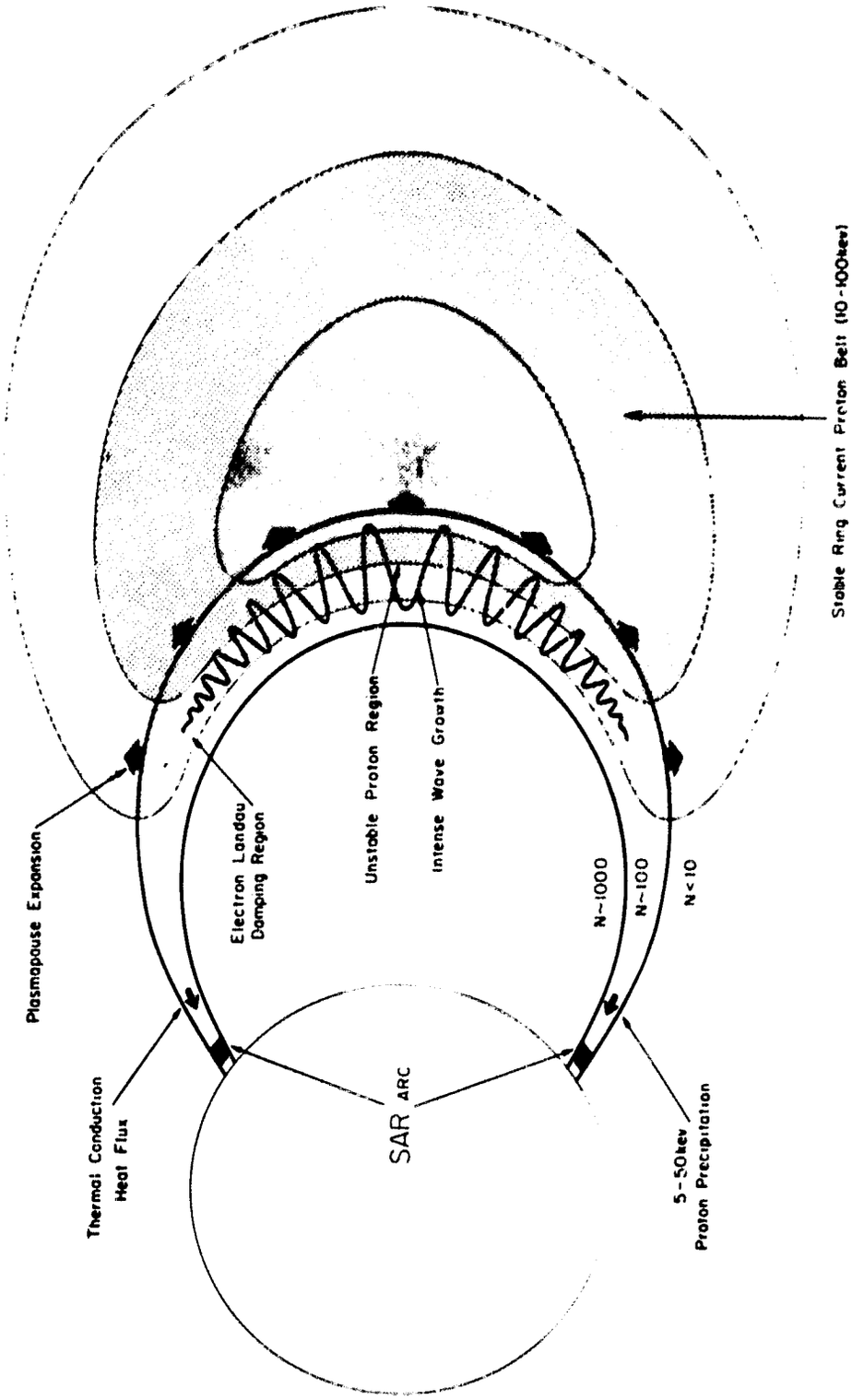


Figure 2.4 A synopsis of the energy transfer processes responsible for SAR arc formation at the plasmapause [*Cornwall et al.*, 1971].

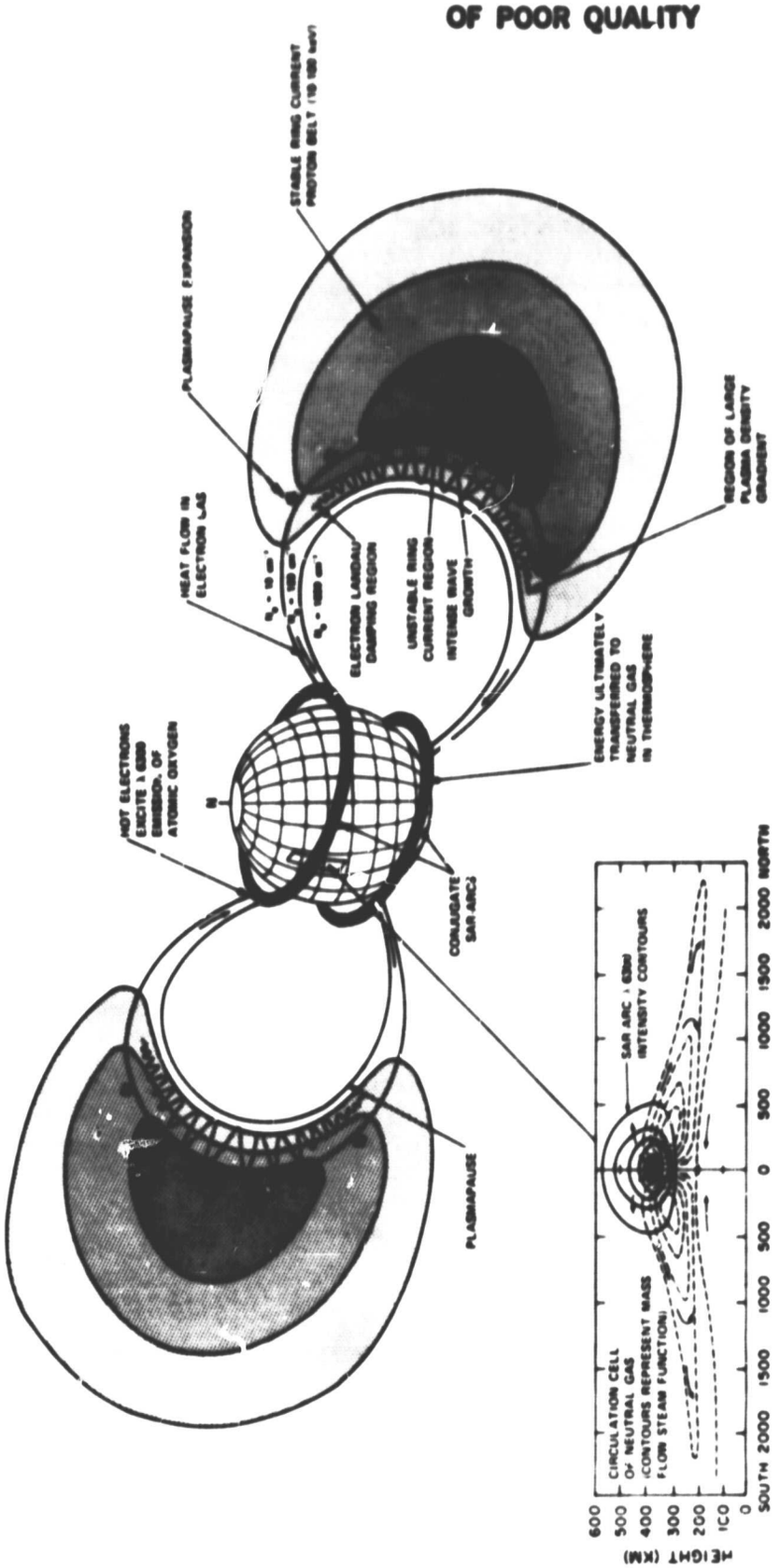


Figure 2.5 Schematic diagram of the processes acting within a SAR arc [*Rees and Roble*, 1975].

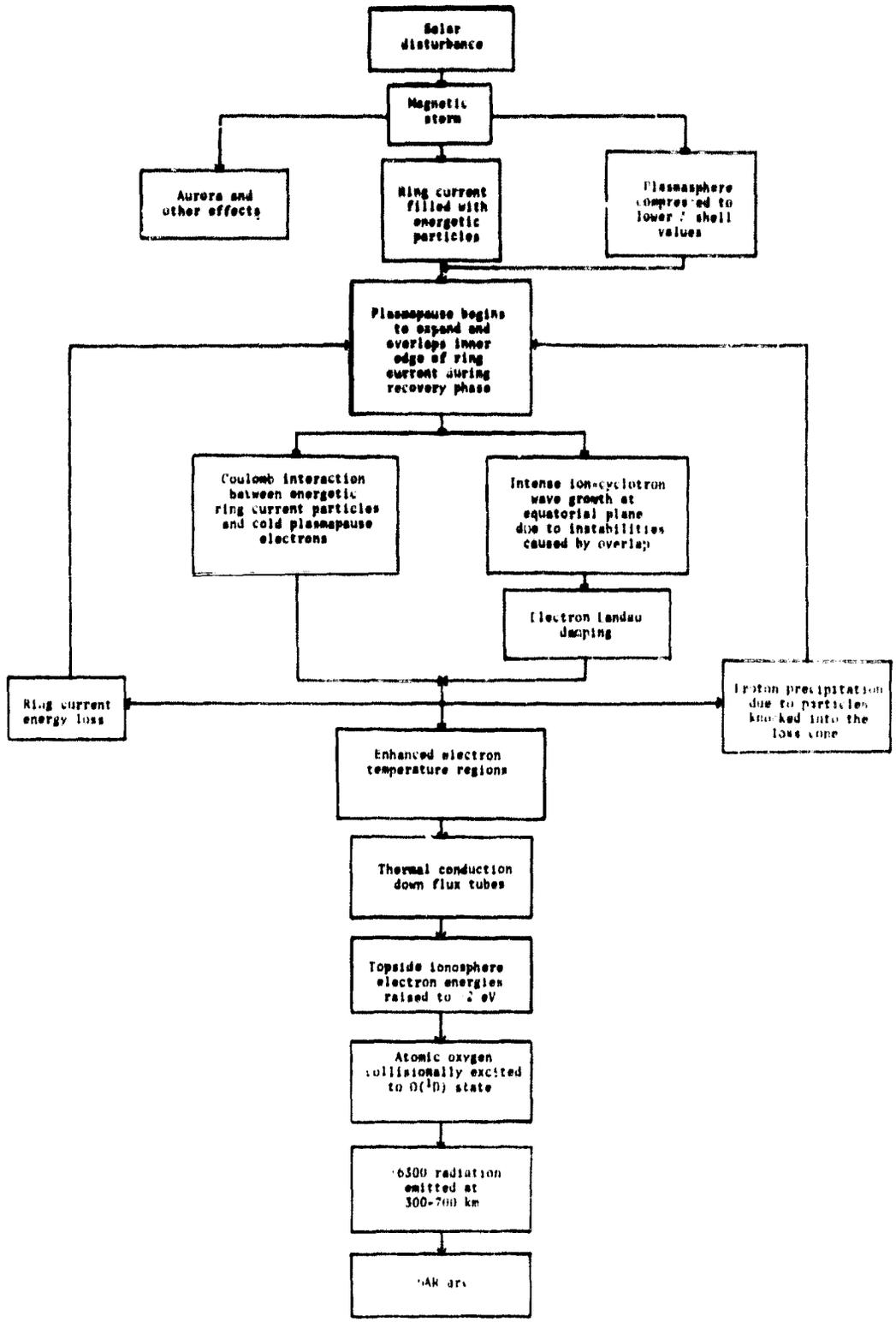


Figure 2.6 A representation of the hypotheses of *Rees and Roble* [1975] and of *Cole* [1975].

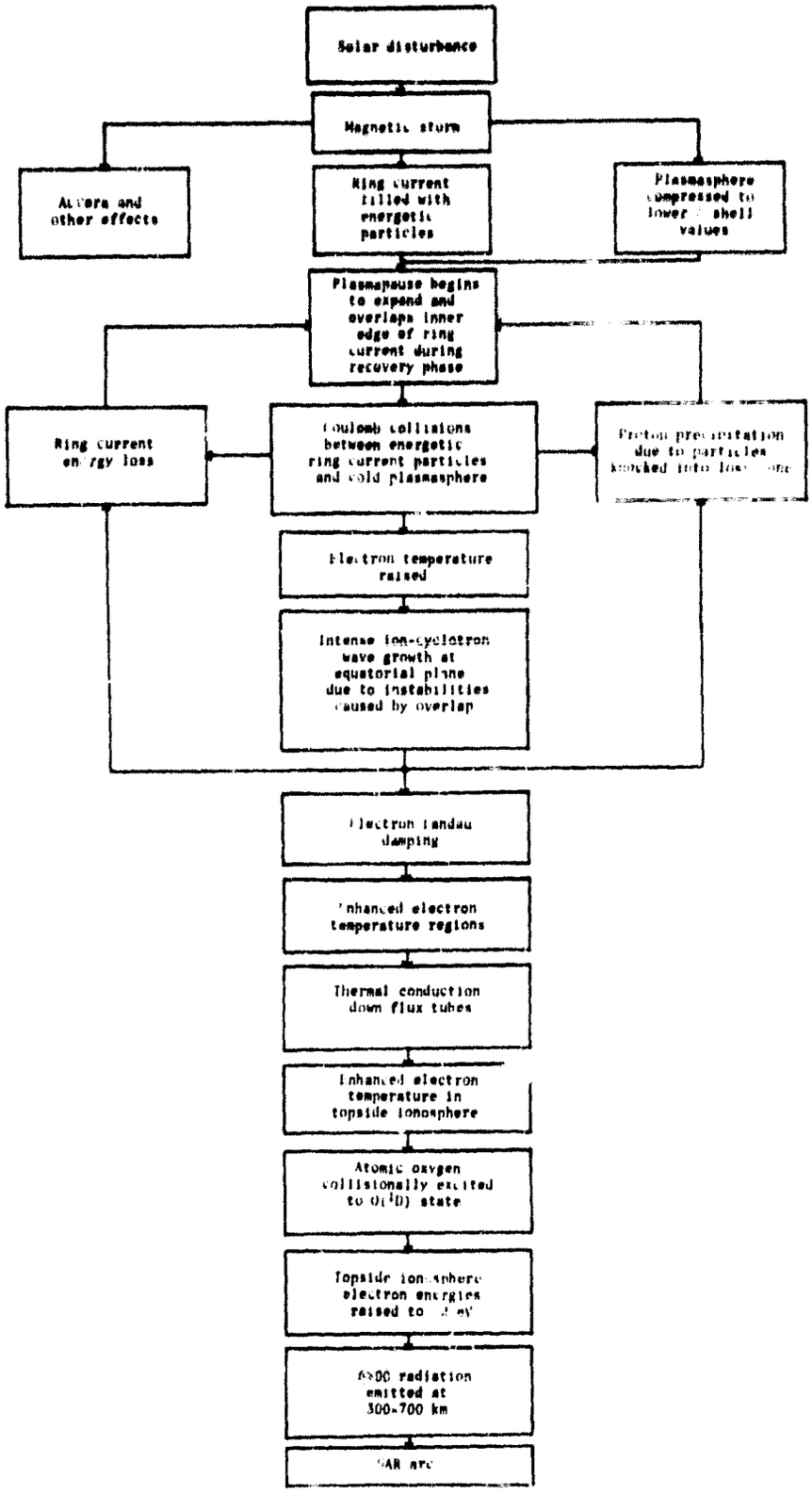


Figure 2.7 A representation of the hypothesis of *Cornwall et al.* [1971].

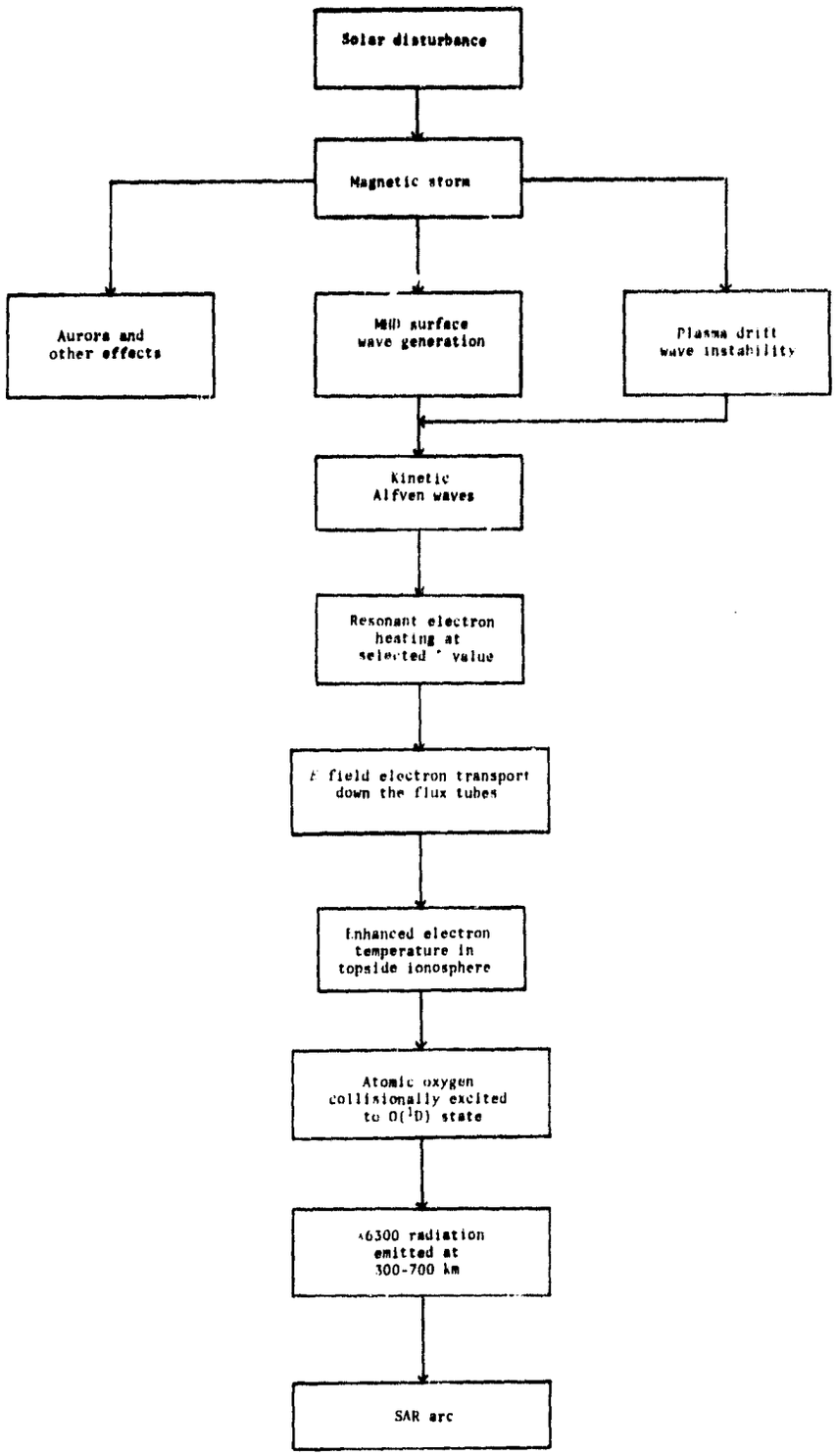


Figure 2.8 A representation of the hypothesis of *Hasegawa and Mima* [1978].

Cornwall's hypothesis (see Figure 2.7) makes use of Coulomb interaction, ion-cyclotron radiation, and Landau damping but places different emphasis on the three phenomena. Coulomb interactions are first needed to heat plasmapause electrons to a point where ion-cyclotron waves can be generated. Landau damping can then occur which further heats the electrons.

The hypotheses illustrated by Figures 2.6 and 2.7 require thermal conduction of heated electrons down the magnetic flux tubes to the region where the SAR arc occurs.

Hasegawa on the other hand proposes an entirely different method of transporting thermal electrons via kinetic Alfven waves and resonant electron heating as can be seen in Figure 2.8.

2.1.4 *Observations supporting the various mechanisms.* It has been observed that during the recovery phase of the December 17, 1971, storm there was more than enough energy loss from the ring current to drive a SAR arc [*Williams et al.*, 1976]. Also observed is an injection of ring current particles following the onset of a magnetic storm along with a drastic reduction in the size of the plasmasphere. SAR arcs were observed at the L-value associated with the position of the plasmapause during the recovery phase of the storm [*Chappell et al.*, 1971].

One very important aspect of these theories that has not been discussed yet is particle precipitation. The proposals which include Coulomb heating and ion-cyclotron radiation also predict pitch-angle scattering of ring current protons which implies that more of these particles will be scattered into the loss cone and precipitated into the lower to middle F region. This was in fact observed by *Hultqvist et al.* [1976] and *Lundblad and Soraas* [1978] during March 1969 SAR arc observations. On the other hand the kinetic Alfven wave hypothesis would predict that electrons would be precipitated along the magnetic lines of force by an electric field parallel to these lines. Enhanced fluxes of precipitating electrons were observed during the storm of August 1972 in which unusual SAR arcs were observed [*Shepherd et al.*, 1976].

Data from Nike Apache rockets launched from Wallops Island give a different perspective. Two flights may be compared: one flight was made into mildly disturbed conditions while the second is believed to have been launched into a SAR arc.

The electron-density profile in Figure 2.9 for Nike Apache 14.534 is

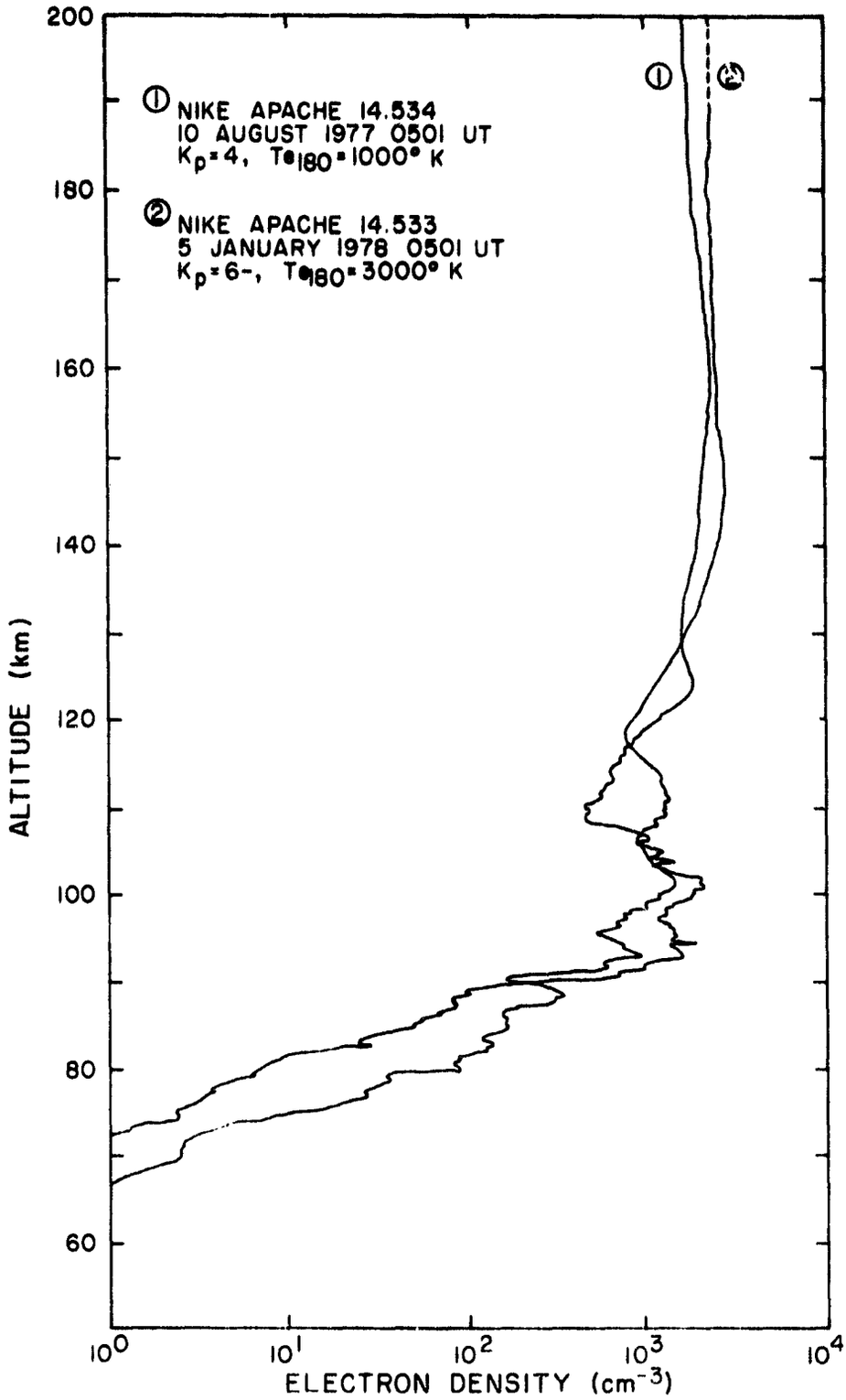


Figure 2.9 Electron-density profiles.

typical of mildly disturbed conditions [see also *Voss and Smith*, 1979]; the electron-density profile for Nike Apache 14.533 is similar in the E region.

More interesting are the electron temperature profiles in Figures 2.10 and 2.11. Figure 2.10 shows a typical measurement while Figure 2.11 shows a tremendous enhancement in electron temperature.

Figure 2.12 shows that there is definitely energetic proton precipitation along the magnetic field lines. The tendency for the energetic particles to mirror is indicated by the two peaks close to 90 degrees.

Energetic proton precipitation is predicted by *Cornwall et al.* [1971] and is in agreement with rocket data. High electron temperatures in the lower *F* region are expected due to thermal conduction and are again verified by rocket data.

2.1.5 *Conclusion.* In view of the facts, it is believed that Hasegawa's hypothesis plays the major role in the production of SAR arcs, however, Cole's and Cornwall's hypotheses cannot be completely ignored.

Along with Hasegawa, the arcs are seen as a large-scale resonance phenomenon of the geomagnetic *L*-shells during disturbed conditions. It is interesting that a correlation has been made between D and H components of the earth's geomagnetic field and SAR arc position and intensity. Additional work needs to be done in this area. The conclusion by *Shepherd et al.* [1976] that plasma heating occurs over widely separated regions in the magnetosphere would again indicate large-scale resonance and an ac electric field enhancement measured within the arc of August 1970 is easily explained by Hasegawa.

Energetic proton precipitation as observed by Nike Apache 14.533 on January 5, 1978, can be explained by the plasmapause and ring current interactions proposed by *Cole* [1975] and *Cornwall et al.* [1971]. Coulomb interaction probably plays a larger role in particle precipitation than ion-cyclotron radiation since there has never been direct verification of the ion-cyclotron radiation, while we can be fairly certain of Coulomb interactions. However, as far as their contribution to SAR arc phenomena these two interactions are probably minor.

Further examination of these hypotheses is possible since there is such a massive amount of data yet to be analyzed. We need even more detailed measurements in the future of magnetic and electric field variations, energetic particle precipitation, electron densities and electron

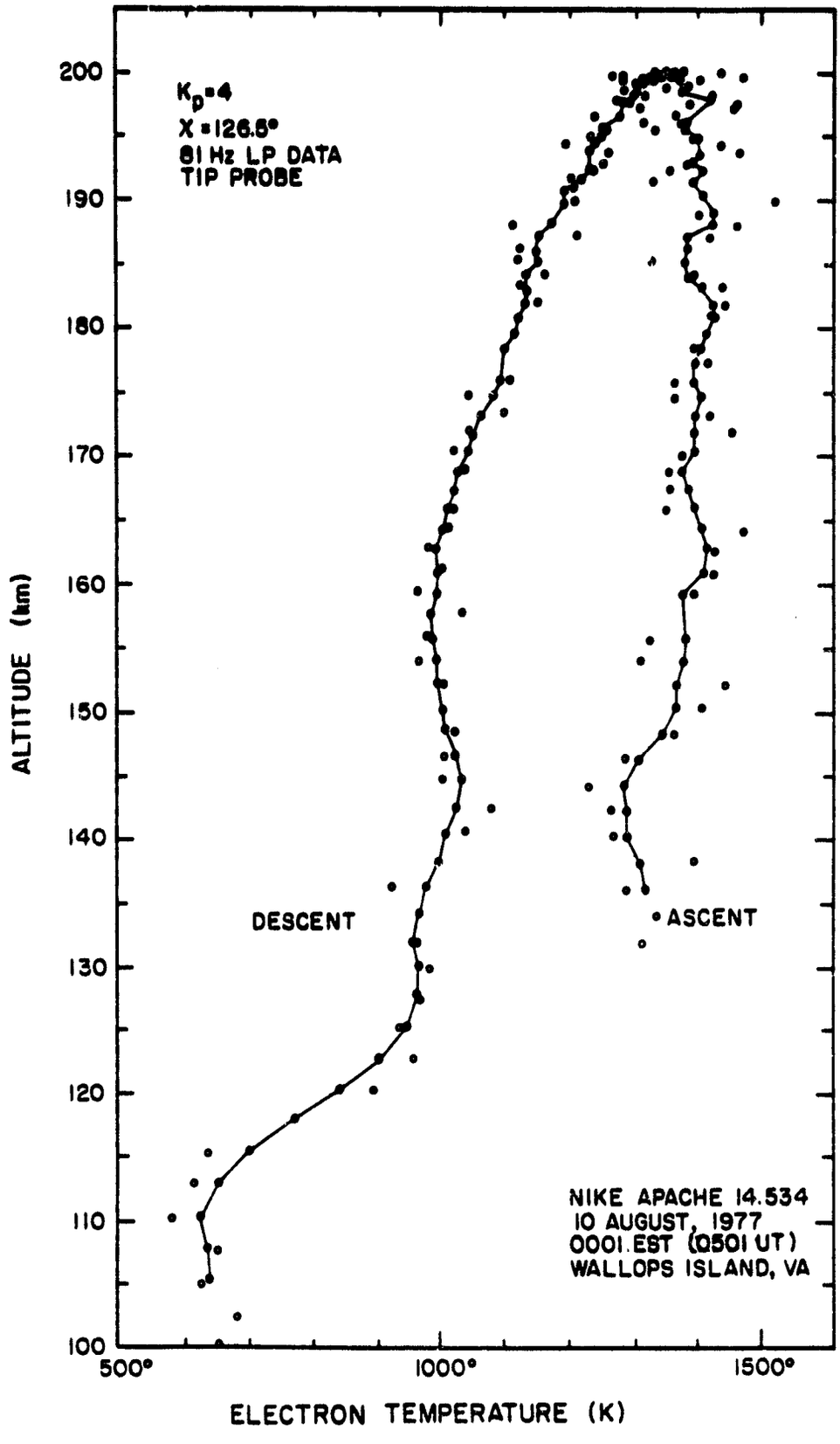


Figure 2.10 The electron-temperature profile for Nike Apache 14.534. The points connected by the broken line are five-point average values [*Zimmerman and Smith*, 1980].

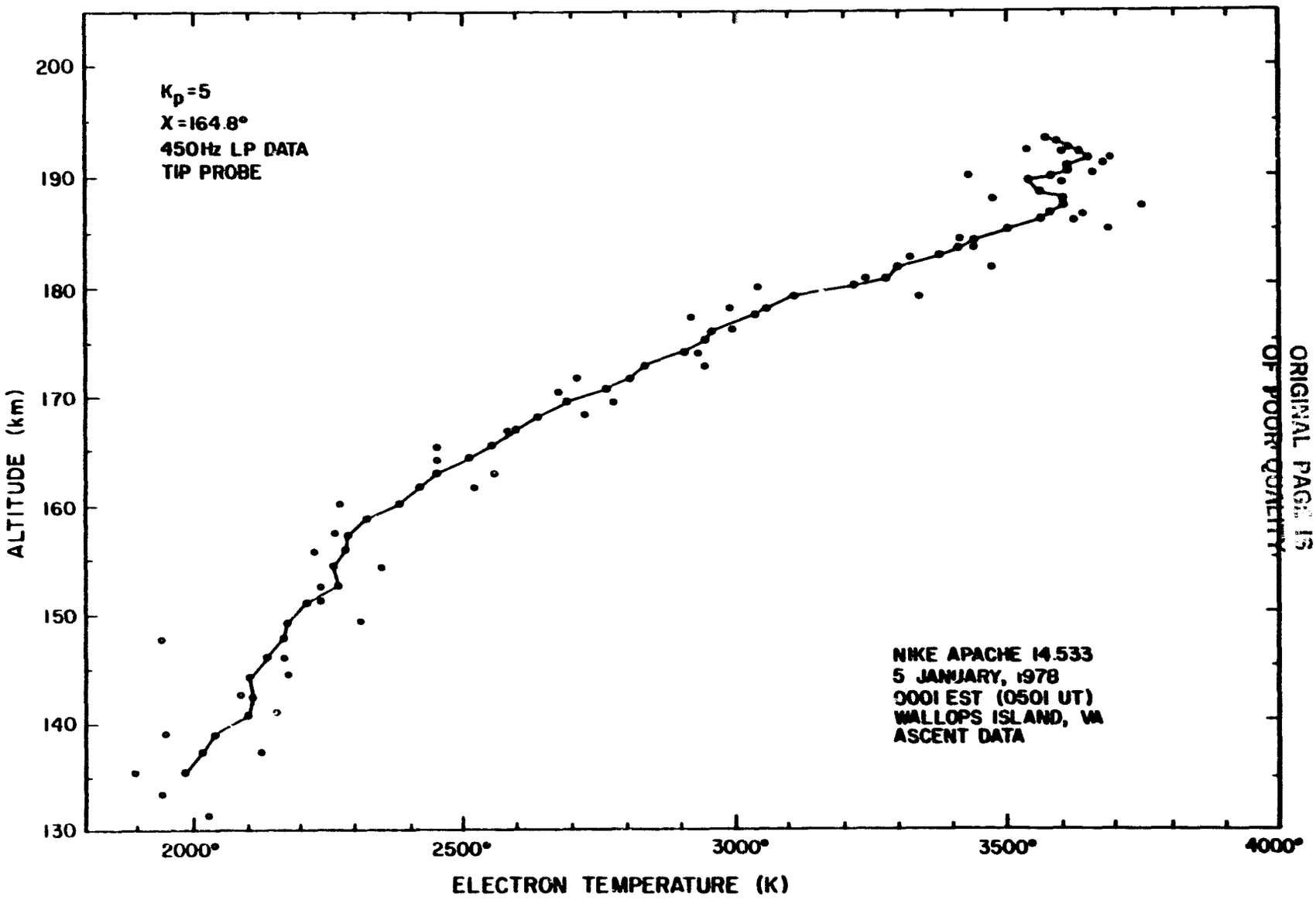


Figure 2.11 The electron-temperature profile for Nike Apache 14.533. Points connected by the broken line are five-point average values. The extremely high temperature observed during this flight cannot be explained in terms of normal nighttime heating processes [*Zimmerman and Smith*, 1980].

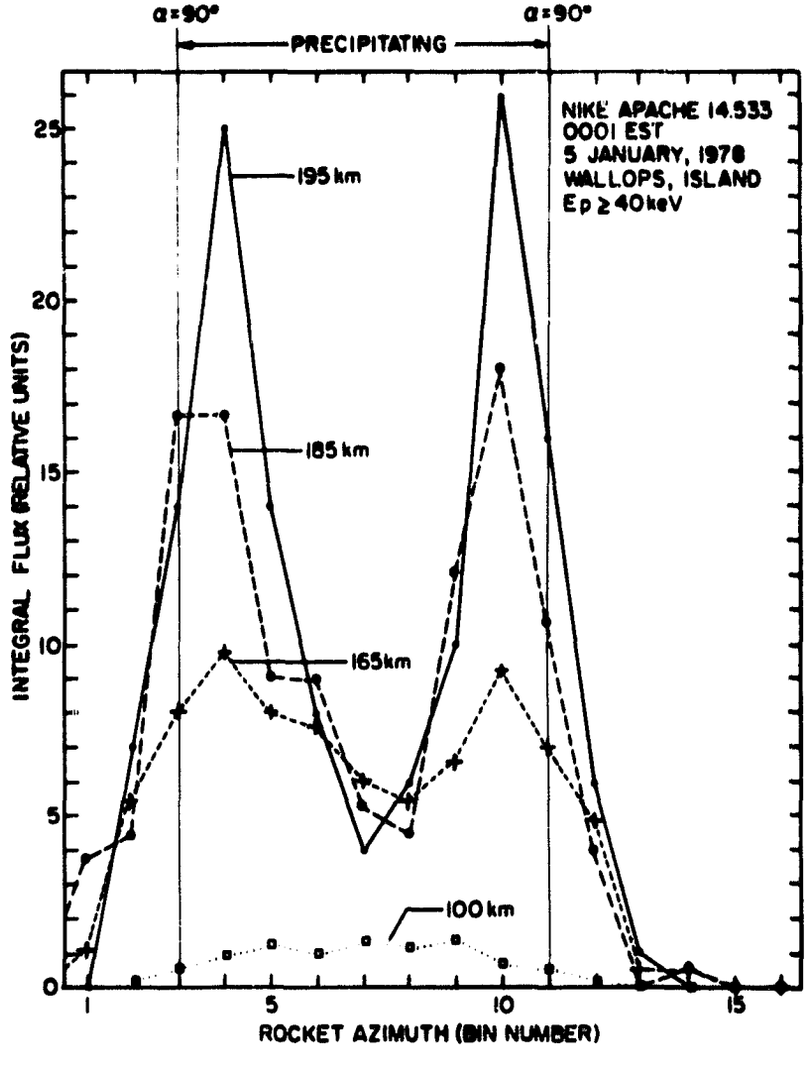


Figure 2.12 SAR arc pitch-angle plot showing energetic proton precipitation near 90° [*Voss and Smith*, 1979].

temperatures, plus direct evidence of ion-cyclotron waves in the magnetopause during magnetic storm recovery.

Barbier discovered the SAR arc over 23 years ago, yet today in spite of a wealth of data, its origin remains a mystery and a challenge to scientists.

2.2 *Experiment Used in the JASPIC Program*

The fundamental parts of the JASPIC system are shown in Figure 2.13.

The magnetometer signal digitizer supplies the microprocessor with a 4-bit sector number (16 sectors) indicating the rocket azimuth bin. This sector number can be monitored by PIO #2 port A.

Six bits of pulse-height and detector-number information are available at PIO #1 port A. The A/D converter supplies 4 bits of pulse-height information (16 energy bins) connected with 2 bits of detector information (4 detectors). This information is input to the PIO as it is available.

Data output synchronization is accomplished with the help of the CTC. Each time the CTC channel 1 decrements to zero it generates an interrupt at which time one byte of data is output through PIO #1 port B. The data is converted to an analog signal in the range 0 to 5V.

Energetic particle data is cataloged and stored in 2K of RAM and the operating system resides in 2K of EPROM.

A detailed discussion of the original microprocessor system can be found in *Davis et al.* [1979].

The experiment described in this report has the same input and output facilities at its disposal, however the data organization of the microprocessor system and software are quite different.

The main hardware differences are in the magnetometer signal processing and the telemetry output. *Davies et al.* [1979] used a PIO port to monitor the magnetometer signal in digital form. Another PIO port was used to send digital output to a D/A converter which was connected to the telemetry output. Only 2K of RAM was used.

The current system eliminates the 2 PIO ports mentioned above and increased the amount of RAM to 8K. The new system added an SIO chip for communication, replacing the D/A converter.

The most radical change is in the software: *Davies et al.* [1979] used the mode 1 interrupt structure; the new system uses the mode 2 vectored interrupt.

The sector is calculated off board in Mr. Davis' system while the current version used the CTC to calculate the sector interval and number.

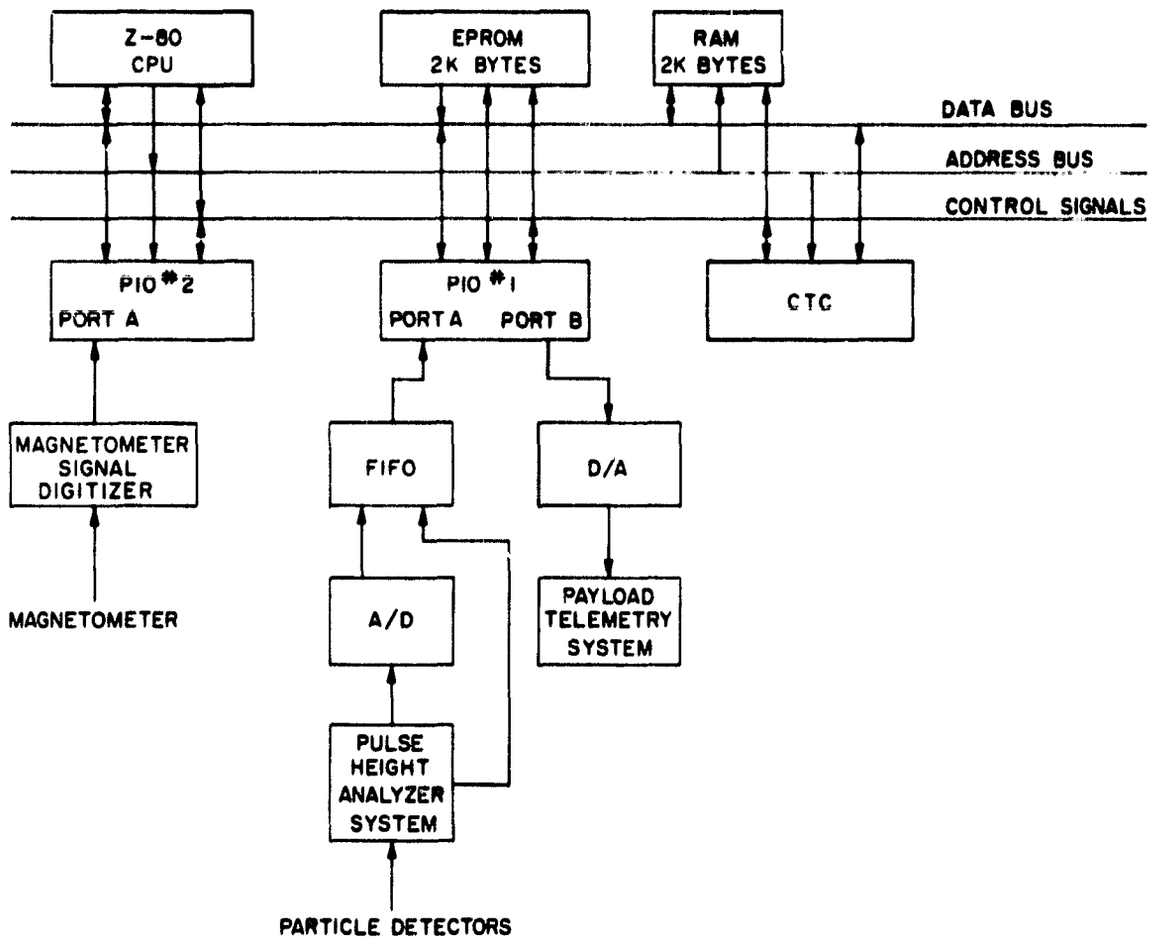


Figure 2.13 The microprocessor system of *Davies et al.* [1979].

Doubling the number of azimuth bins has increased the memory requirement from 2K to 4K bytes of data.

2.3 *General Description of the Improved System*

The rocket-borne microprocessor experiment allows real-time processing of energetic particle data in the lower ionosphere (70 km to 180 km).

The data processed by the microprocessor experiment holds energy-spectrum (Figure 2.14) and pitch-angle information (Figure 2.15), which is the particles angle relative to the earth's magnetic field. Further analysis will allow identification of the kinds of particles (electrons, protons, etc.) encountered by the detectors.

Figure 2.14 and 2.15 are results of data analyzed from Nike Apache 14.542 which carried the first-generation version of the microprocessor experiment [*Davis et al.*, 1979]. Hardware and software problems with the first generation experiment demonstrate the need for self-checking features to be incorporated into the design of later versions of the microprocessor experiment.

The second generation of the microprocessor experiment reflects changes in architecture due to new VLSI chips on the market and past experience with the first generation experiment. Better program development support has led to sophisticated software which more fully exercises the capabilities of the Z80 microprocessor: in particular the mode 2 vectored interrupt response feature.

Figure 2.16 shows how the microprocessor fits into the energetic particle experiment. The solid-state detectors, amplifiers, threshold detectors and staircase generators are described in *Voss and Smith*, 1977, and have been used on several past flights. The other boxes, except for the magnetometer, represent new hardware currently under development.

When an energetic particle strikes one of the four solid-state particle detectors a small current pulse is formed due to electron-hole pair generation. This pulse is amplified and ends up as charge stored on a capacitor which is at the input of an analog-to-digital converter on the pulse-height analyzer (PHA) board. The output of the analog-to-digital converter (8 bits) is fed into an EPROM which contains a log look-up table (Table 2.1) which packs the pulse-height data into its logarithmic representation (4 bits). At this point the respective detector code (2 bits for four detectors) is appended to the pulse-height information and the resulting 6 bits of information are fed into a

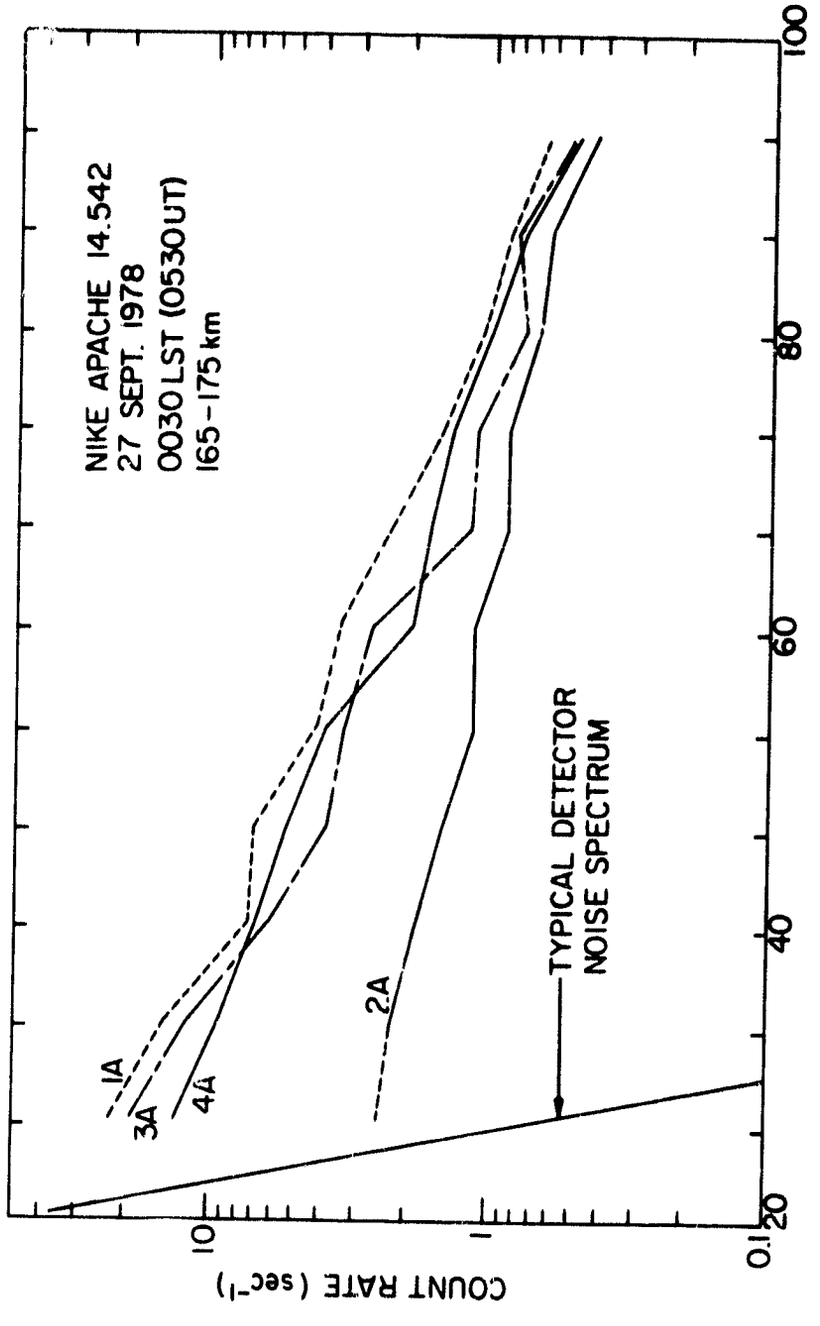


Figure 2.14 Differential energy spectrums, not corrected for dead-zone loss. A typical detector noise spectrum is also shown [*Voss et al.*, 1979].

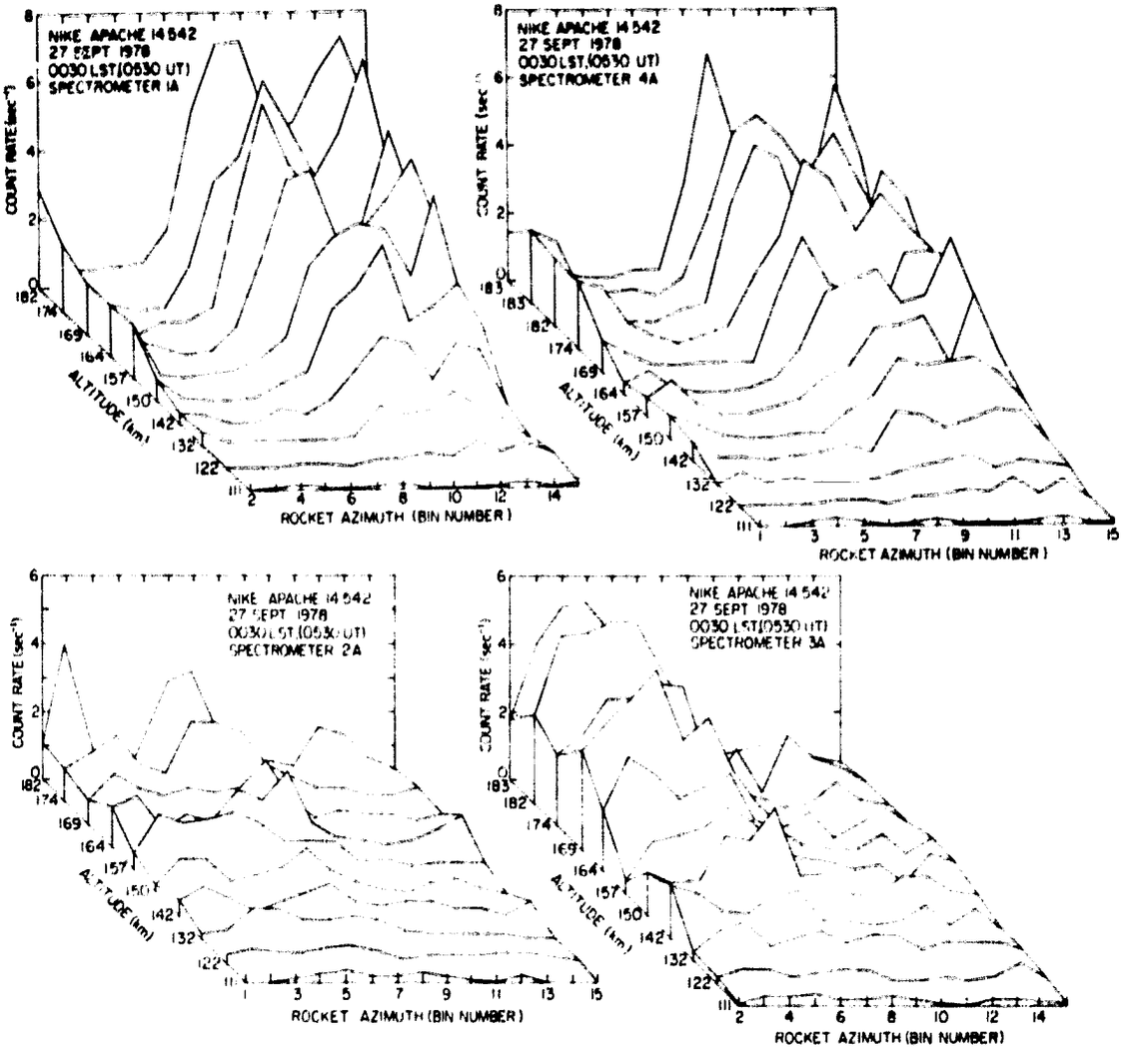


Figure 2.15 Pitch-angle distribution represented by rocket azimuthal variation of flux [*Voss et al.*, 1979].

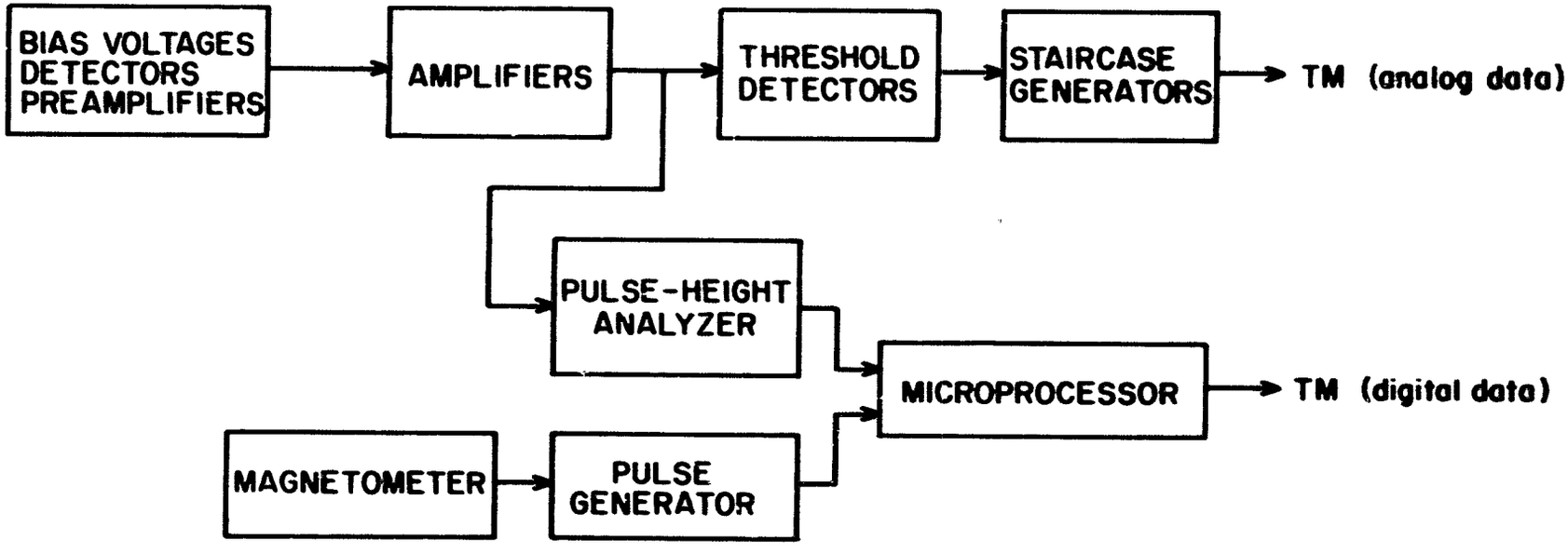


Figure 2.16 General arrangement of the circuits of the energetic-particle experiment for the Energy Budget Campaign.

Table 2.1 Comparison of the energy ranges obtained without and with logarithmic compression of the pulse-height data. The input is in units of keV [*Smith in Edwards*, 1979].

OUTPUT DATA	INPUT (LINEAR)[1]	INPUT (LOGARITHMIC)[2]
0	0 - 14	0 - 14
1	15 - 22	15 - 16
2	23 - 30	17 - 19
3	31 - 38	20 - 22
4	39 - 46	23 - 26
5	47 - 54	27 - 31
6	55 - 62	32 - 36
7	63 - 70	37 - 42
8	71 - 78	43 - 49
9	79 - 86	50 - 57
10	87 - 94	58 - 67
11	95 - 102	68 - 79
12	103 - 110	80 - 92
13	111 - 118	93 - 108
14	119 - 126	109 - 126
15	127 - ∞	127 - ∞

[1]Without logarithmic compression

[2]With logarithmic compression

first-in-first-out (FIFO) buffer which is connected to the parallel-input port of the microprocessor.

As the rocket spins the magnetic aspect sensor (magnetometer) indicates the rocket position relative to the earth's magnetic field. A typical signal is shown in Figure 2.17. One revolution of the rocket corresponds to one cycle of the sine wave, and at each negative-going crossing of the signal, a pulse is generated (the pulse-generator box in Figure 2.16) which toggles the non-maskable interrupt line of the microprocessor. Two counter-timer channels are used to keep track of which one of 32 sectors the rocket is in.

Channel 0 decrements once every 1.638 ms and by counting the number of decrements the time per revolution can be determined. Channel 1 decrements once every 51.2 μs which is 32 times faster than Channel 0 and by loading the time per revolution determined from Channel 0 into Channel 1, there will be 32 interrupts per revolution from Channel 1 as shown in Figure 2.17.

At this point there is sector, detector, and pulse-height information ready to be processed. The sector number has been calculated by the CPU and when the first-in-first-out buffer indicates detector and pulse-height data are available the parallel I/O port generates an interrupt. When the data are fetched they are concatenated with sector information to form a memory address as shown in Figure 2.18. By setting higher order bits the 2K byte blocks of data can be placed anywhere in memory space. After the address is formed, the memory location is incremented by one (one particle count).

The memory map is detailed in Figure 2.19. The 2K byte regions indicated are used for data accumulation and data transmission. While one region is being used for accumulation the other region is used for transmitting previously accumulated data. After an entire block of data has been transmitted the two regions are switched according to the sequence shown in Figure 2.20.

Since the amount of time needed to transmit 2K bytes of data corresponds to approximately 20 to 25 spins of the rocket, the end of a block will fall in the middle of a revolution. In order to keep an integral number of revolutions associated with the 2K bytes of data accumulated, the processor waits until the beginning of the next revolution before it begins transmitting the next block of data.

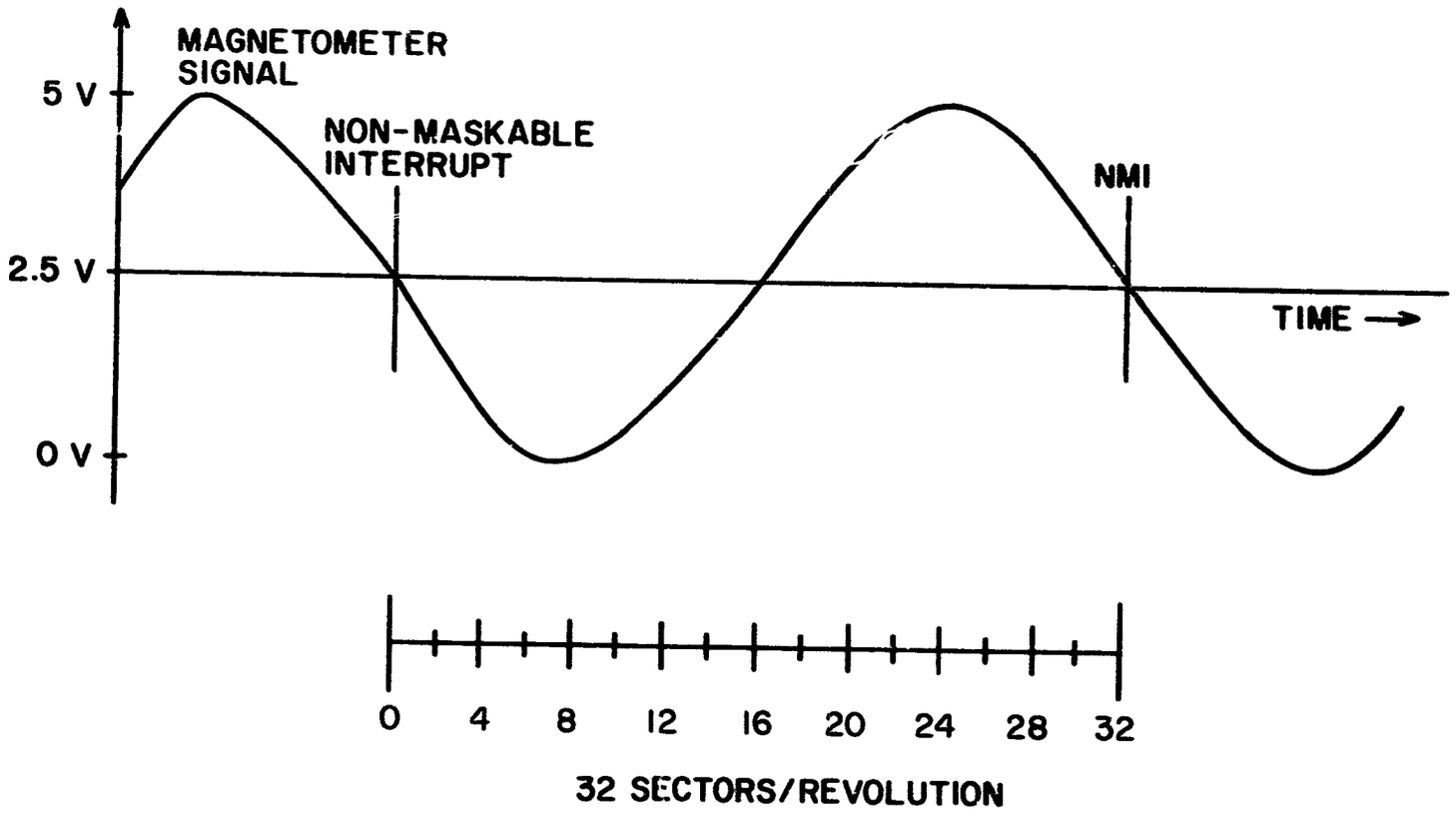


Figure 2.17 Sector algorithm. CTC channel 0 used to calculate the time between NMIs. CTC channel 1 is used to divide time between NMIs into 32 divisions.

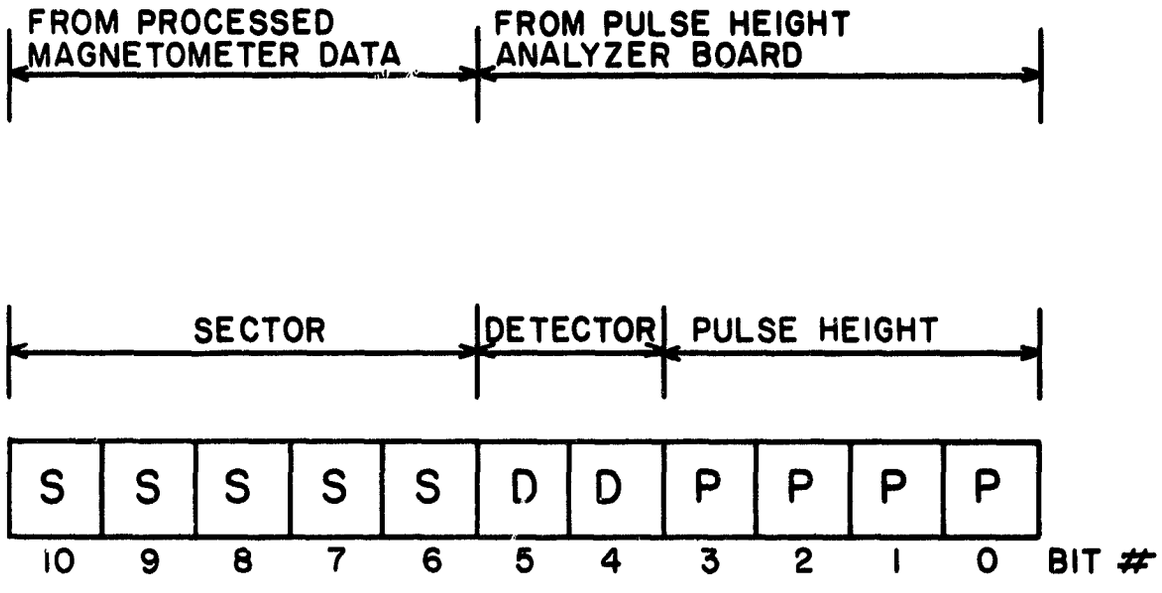


Figure 2.18 Address calculation.

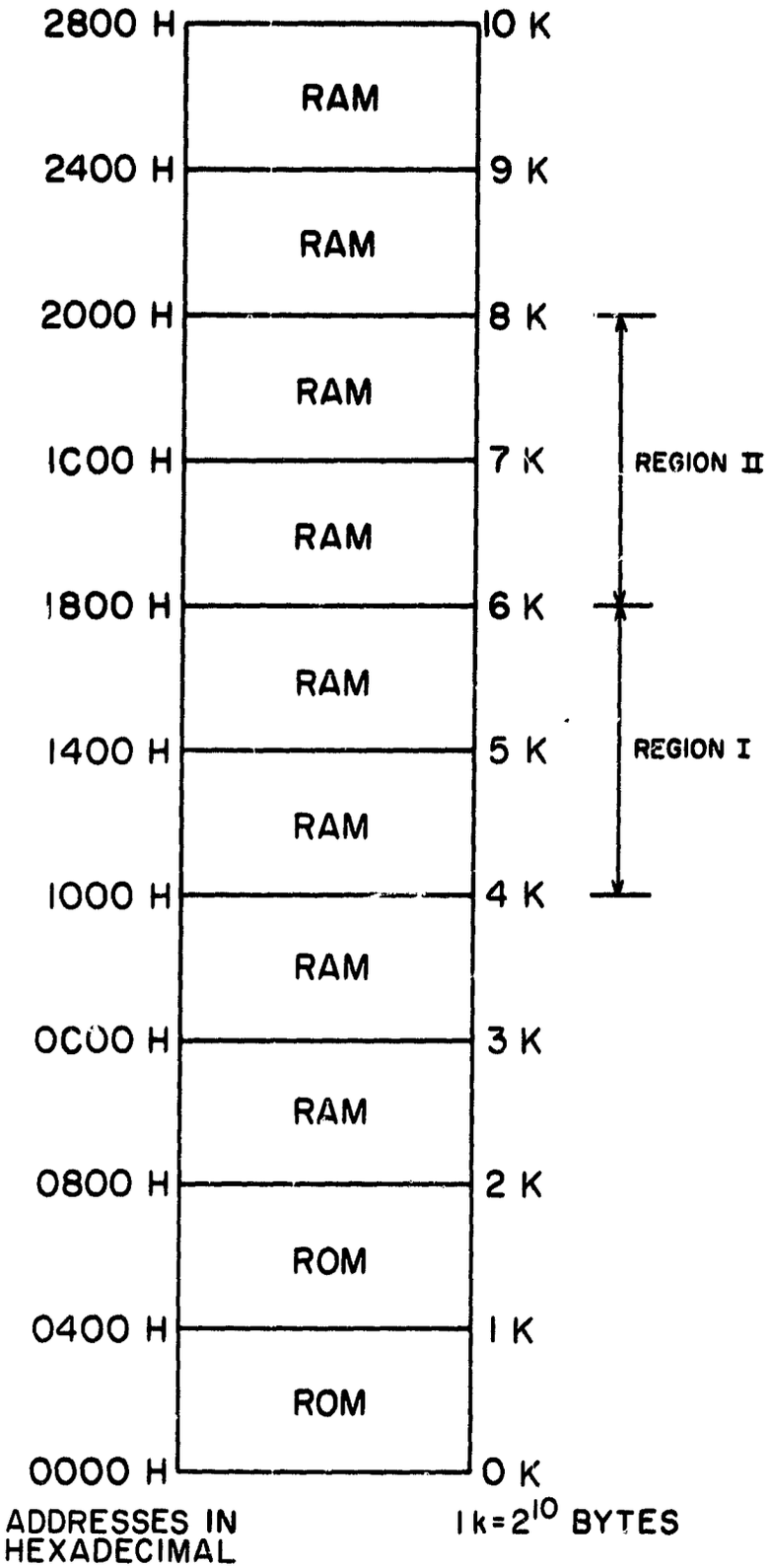


Figure 2.19 Microprocessor memory map.

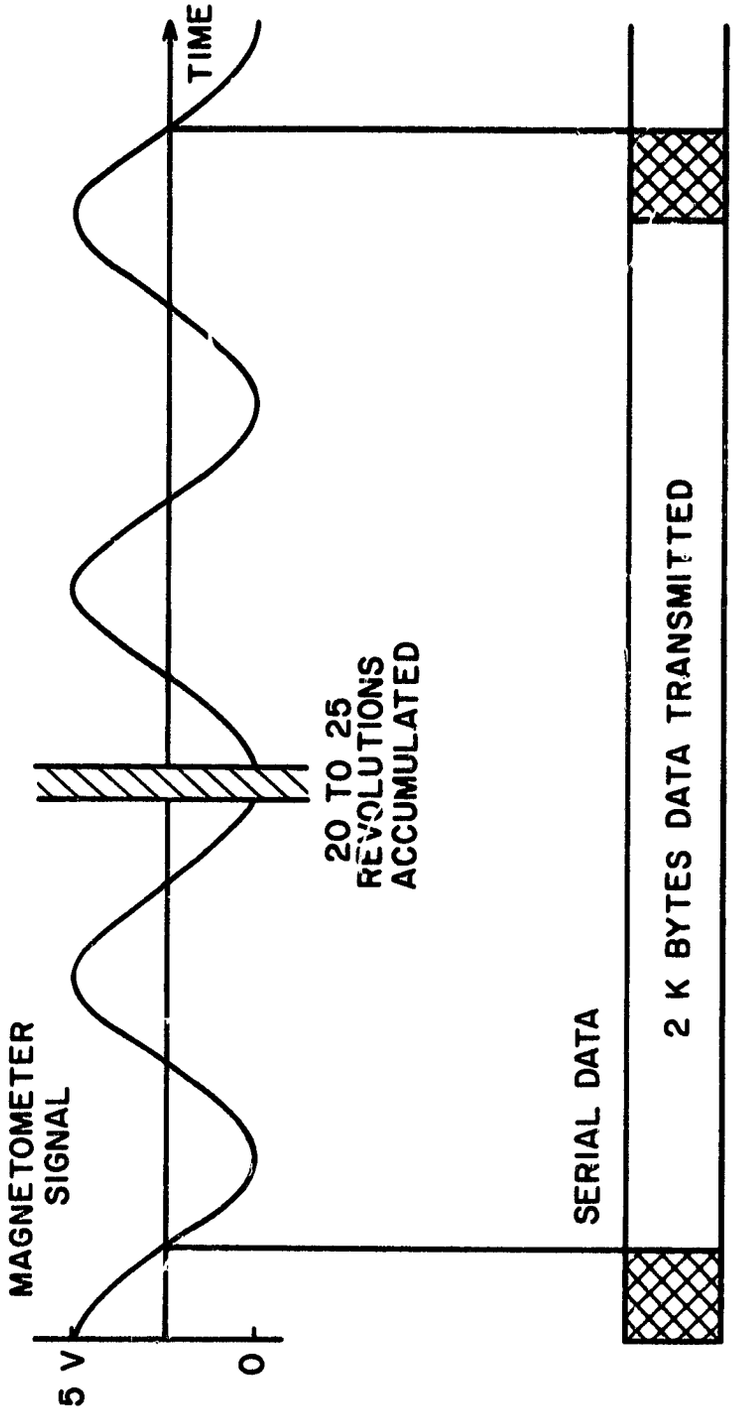


Figure 2.20 Accumulation-transmission algorithm.

Figure 2.21 shows what happens if the serial data are plotted against time. Each sequence of 8 bits correspond to a value on the ordinate from 0 to 255 (2^8) and are seen as steps. Each step gives the number of counts at a given pulse height, for a given detector, in a particular sector all accumulated over 20 to 25 revolutions during a specific time interval.

Each one of the 16 detector bytes corresponds to a pulse height; however, pulse-height bin (number) zero always contains noise from the detector which will, for all practical purposes, be 255 in every case. Figure 2.21 shows an identification pulse at the beginning of each detector sequence. It was decided that each pulse-height bin 0 should contain the following information:

Detector #	Information in bin 0
1	Marker Pulse (255)
2	Record Number (0-255)
3	Sector Number (0-31)
4	Record XOR Sector

The marker pulse in detector 1 flags the beginning of a sector of data. The record number indicates which block of data is currently being transmitted and recycles every 256 blocks of data. The sector number indicates which of the 32 sectors the data was accumulated in and the exclusive OR of the record and sector is an error check.

The system also features an automatic reset circuit which, if not addressed at least once every 50 ms, will cause a system reset. A power-up circuit causes a system reset several milliseconds after power is applied to the system. The delay allows any transients due to an instant power-on to have dissipated before the system is reset and begins operation.

The system diagram (Figure 2.22) illustrates the interconnection of the various components of the microprocessor. The system is built around the Mostek Z80 family which include the MK3880 CPU, MK3881 PIO, MK3882 CTC, and the MK3884 SIO. The 8K bytes of RAM is composed of sixteen 1K-by-4-bit low-power 2114 static RAM chips and the operating system ROM is a 2K byte 2716 EPROM. Other devices pictured include a 4-to-16 decoder (74LS154), 4-bit binary counters (74LS393), retriggerable monostable multivibrators (74LS123), Schmitt-triggered NAND gates (74LS132), inverting buffers (74LS04), AND gates (74LS08), a 2.5 MHz clock (Motorola K115A) and a comparator (MC3302P).

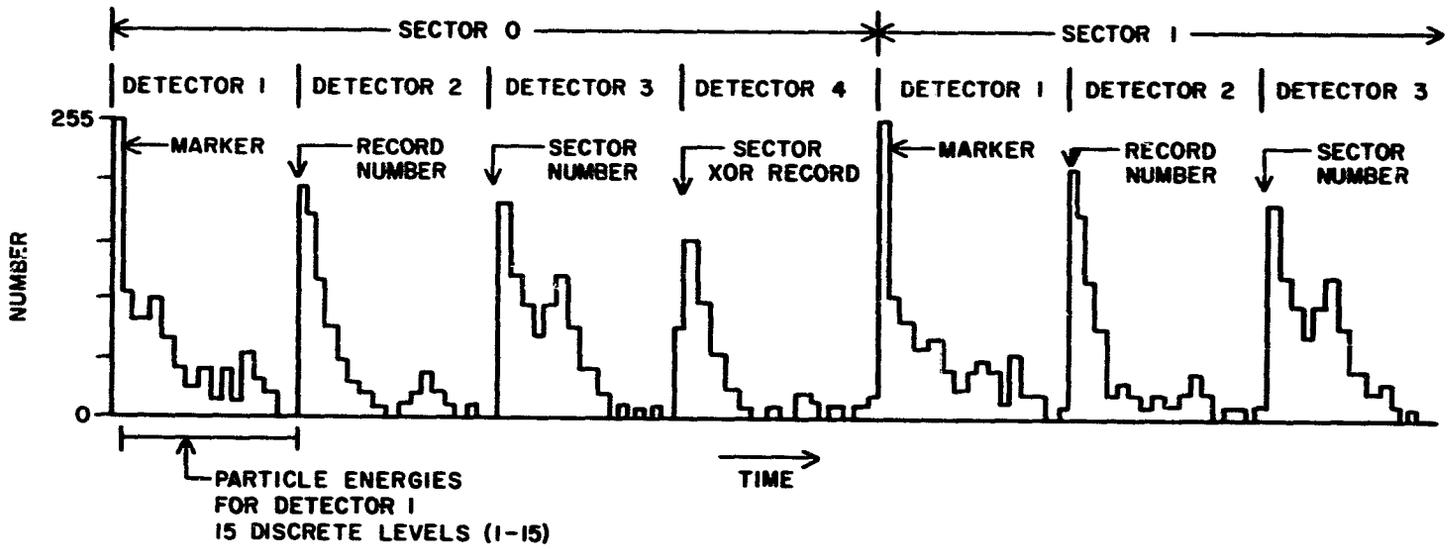


Figure 2.21 Microprocessor serial data stream. Each of the steps displays one byte of data which represents the number of counts of a given particle energy on a detector in a particular sector.

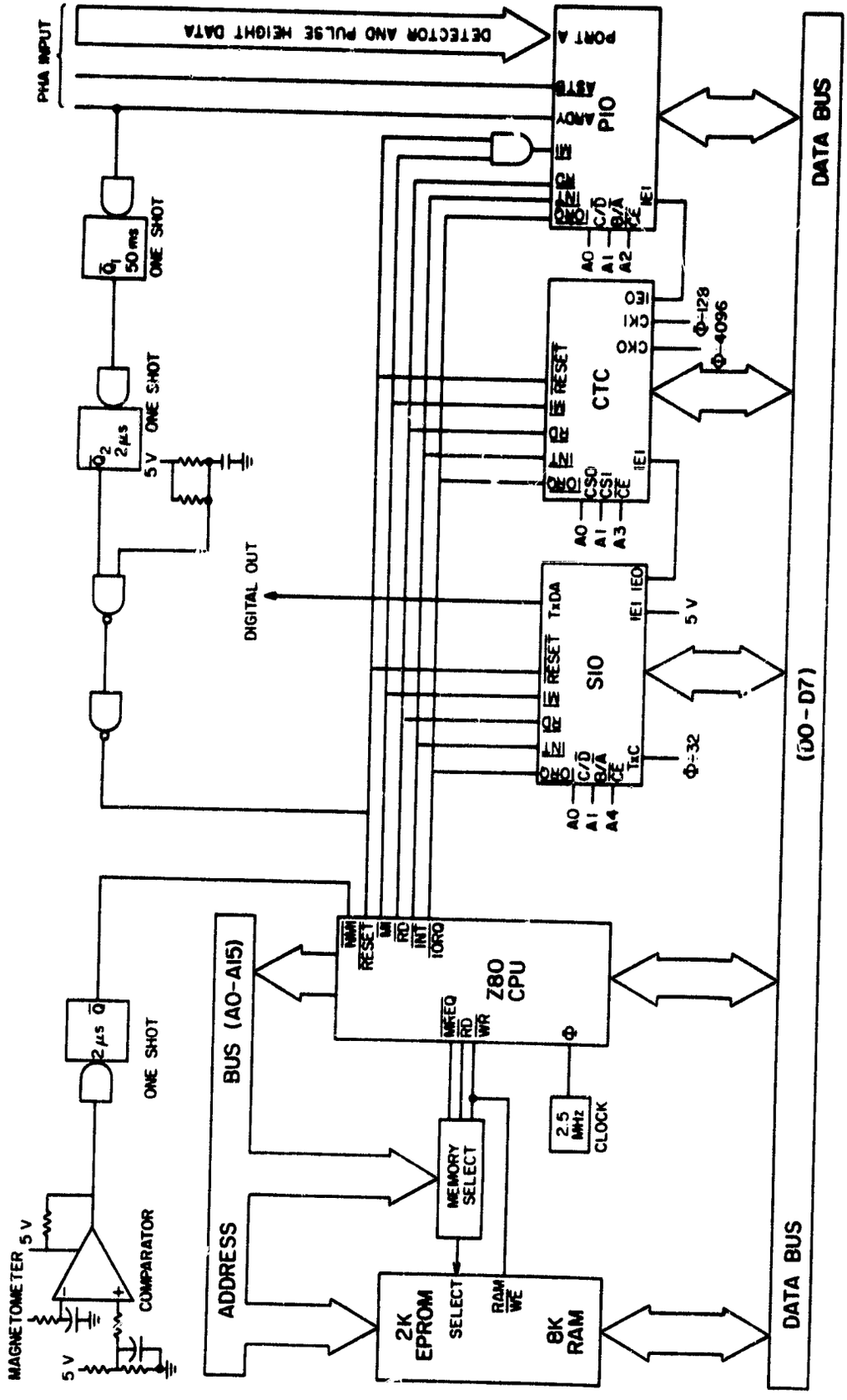


Figure 2.22 Microprocessor system diagram.

The magnetometer pulse generator is made up of the comparator and monostable (in the upper left of Figure 2.22) and generates a CPU non-maskable interrupt by toggling the $\overline{NMI}$ line once during each spin of the rocket.

The automatic reset circuit is made up of the two monostables in the upper right of the diagram. The reset circuit timing diagram (Figure 2.23) illustrates how constant retriggering of the monostable using the $\overline{ARDY}$ line will prevent signal $\overline{Q}_1$ from triggering $\overline{Q}_2$ which will reset the system. The function of the automatic reset is to prevent the operating system from getting hung up. Data must be strobed onto the PIO port and the CPU must respond within any given 50 ms period.

The mode 2 vectored interrupt response of the Z80 (see Appendix IV), allows implementation of sophisticated software techniques. The interrupt-vector table (Figure 2.24) resides at the top of RAM and is set up at initialization to point to each of the subroutines which service their respective devices. The ability to alter the table under program control allows the routine servicing a device to be switched so that any number of routines can service a single device. This concept was used to service the SIO as it output its 2K byte data blocks. As explained earlier each detector has its respective bin 0 information, and, instead of using counters to distinguish detectors, four service routines were written which were linked together through the interrupt table. When the detector 1 routine had finished outputting its 16 bytes it changed the interrupt table to point to the detector 2 routine which would link to the detector 3 routine, etc. (Figure 2.25a). Refer to section 4.2 for more detail.

The general operating system diagram (Figure 2.25b) illustrates how the regions are switched and how the service routines are linked to the main program by the interrupt vector table. Each of the region boxes is continually testing to see if the last byte of a 2K block of data has been transmitted and if the rocket has finished the revolution in which that byte was transmitted. When the conditions are met the main program jumps to the next region box, changes the necessary flags and then begins its own testing loop.

The most critical service routine is that of the PIO since data is input to the FIFO approximately once every 50 μs. This implies that the PIO service routine cannot be longer than 50 μs otherwise the FIFO would fill up. The current execution time of the PIO service routine is about 40 μs which

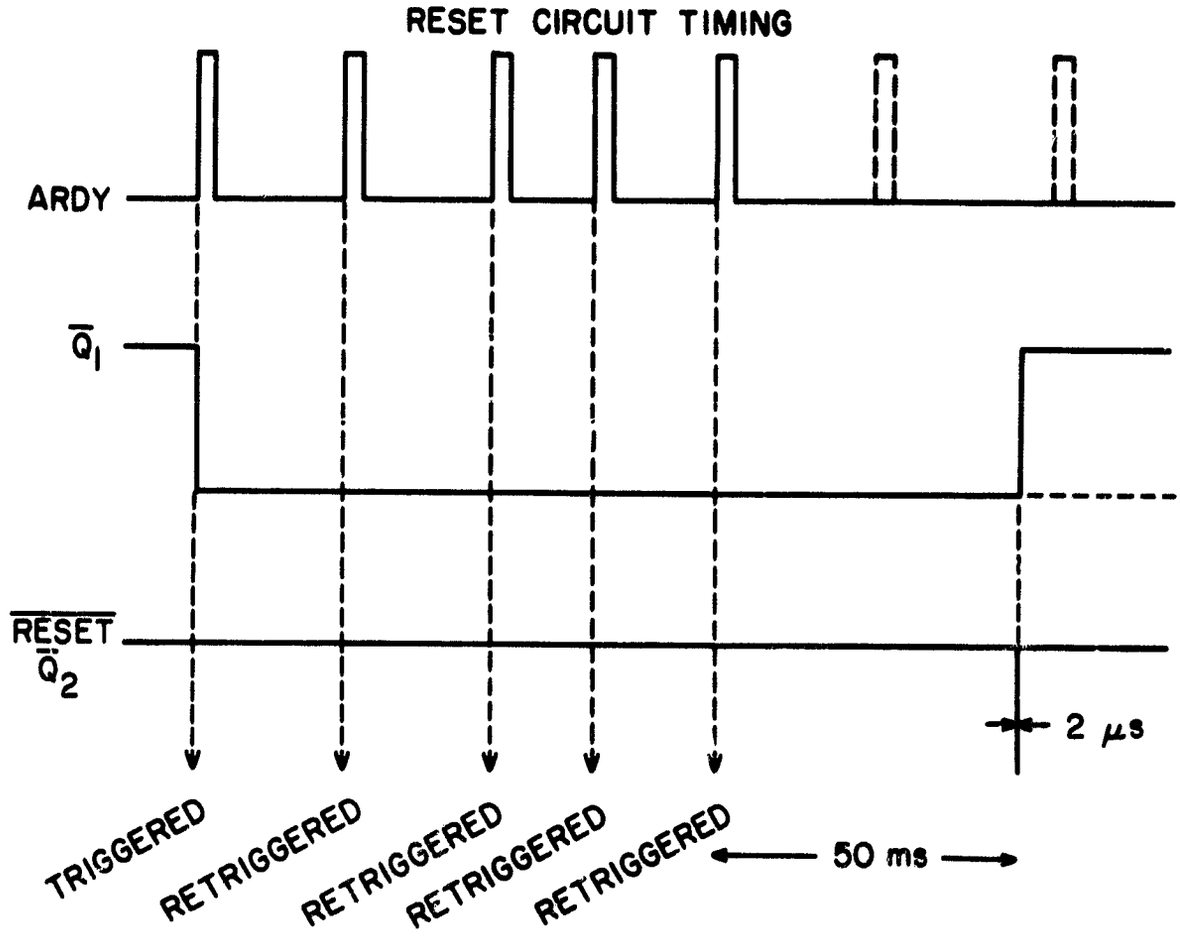


Figure 2.23 This figure illustrates how the automatic reset circuit causes a system reset if not addressed at least once every 50 ms. The time scale of the ARDY signal has been exagerated to illustrate the functioning of the reset circuit. ARDY should be active approximately once every 50 μs.

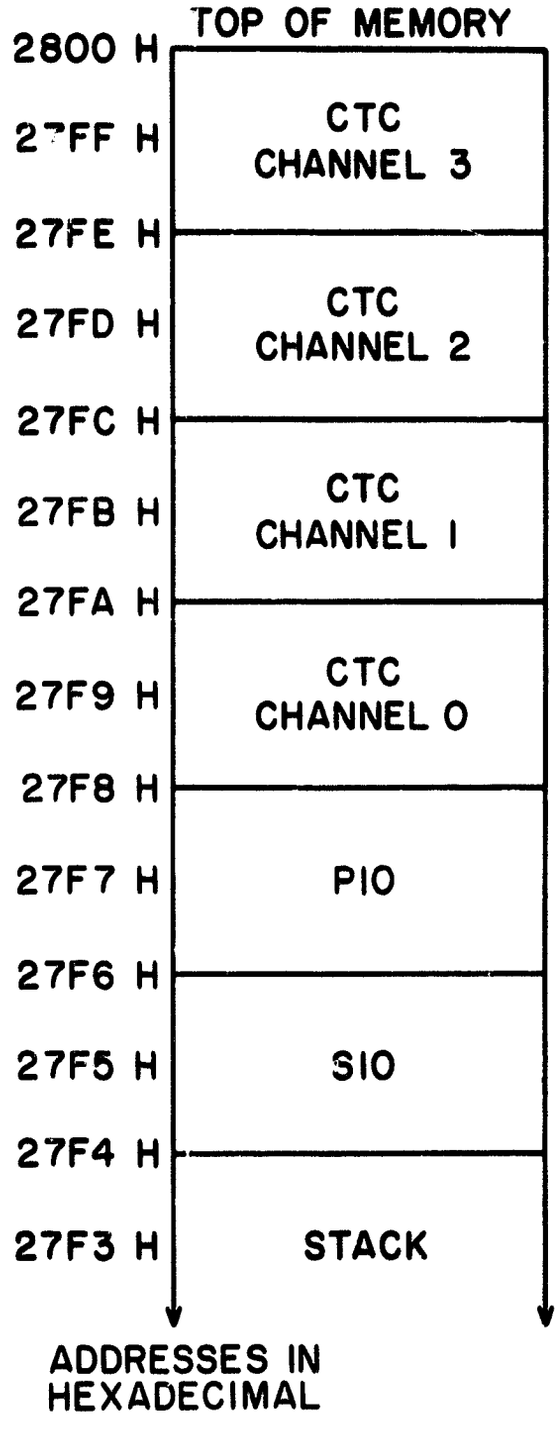


Figure 2.24 Interrupt vector table addresses.

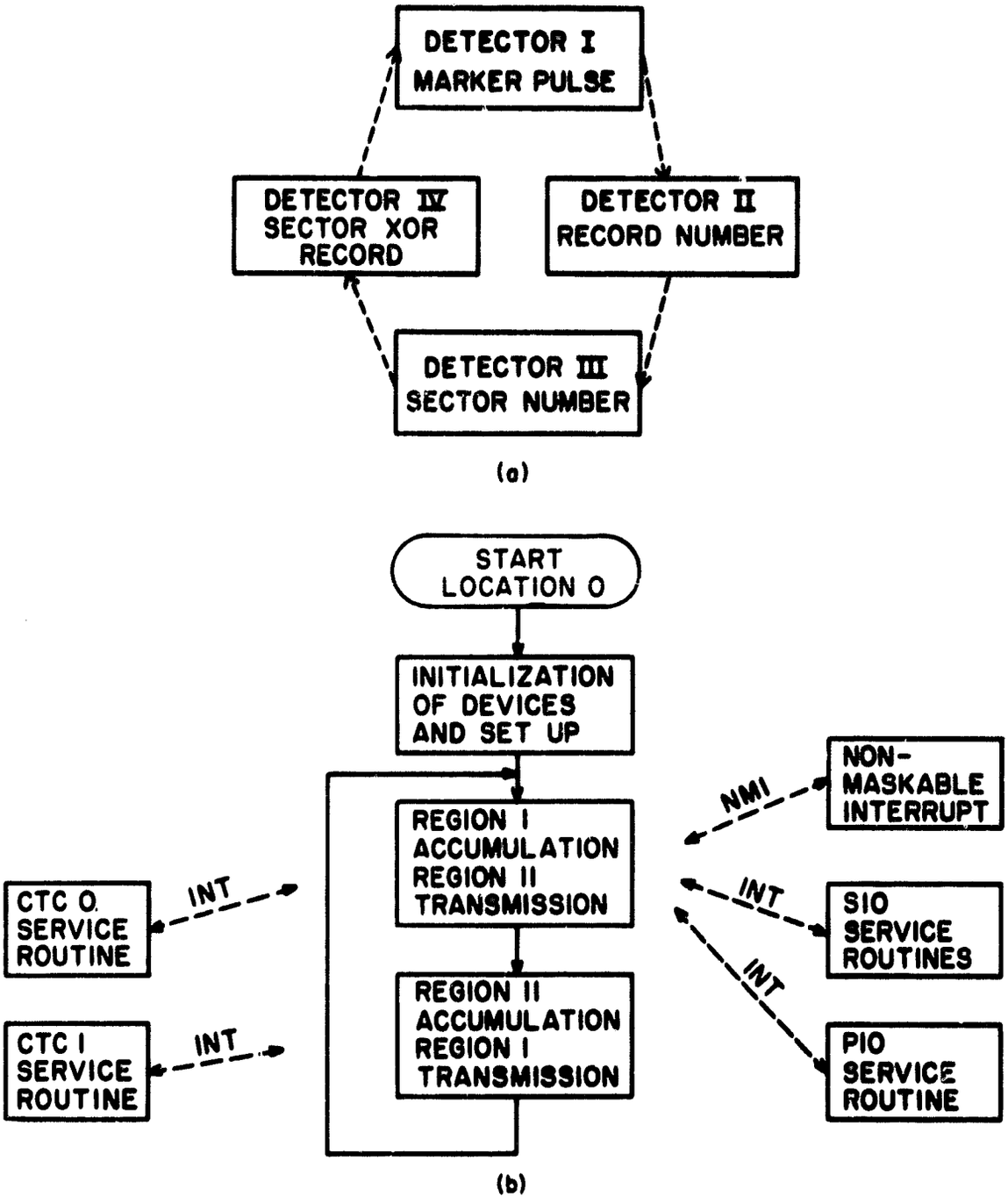


Figure 2.25 (a) SIO service routine linkage using the interrupt table; (b) operating system diagram showing the linkage of the various service routines via the interrupt structure.

includes the 19 clock cycles used to service the interrupt. This means that servicing the PIO accounts for at least 80% of the total processing capability of the microprocessor system.

If all the other parts of the operating system are included it is clear that the system is near capacity and that the incoming data rate cannot be increased without exceeding the capability of the system.

Chapter 6 will discuss why the data rate is important and Chapter 7 will discuss ways to improve the speed with which the data are processed.

3. ROCKET-BORNE SYSTEM ARCHITECTURE

3.1 *Central Processing Unit (CPU)*

The Z80 central processing unit (CPU) is the heart of the rocket-borne experiment. It is here that the programmed sequence of instructions in memory are executed and control signals for the peripheral devices originate.

The Z80 CPU is not a stand-alone device, but depends on its connection to memory and peripheral devices to define its personality and function, making it very versatile.

The rocket-borne application requires 2K bytes of read-only memory, 8K bytes of randon-access memory, a serial I/O port, a parallel I/O port, a counter timer circuit, and a magnetometer-interrupt processor. These are described in the following sections.

The Z80 CPU coordinates the communication between the pulse-height analyzer board and itself through the parallel I/O port. Once the data are available, they are organized and accumulated in memory and then transmitted via the serial I/O port.

The Z80 CPU, powerful as it is, would be quickly rendered ineffective if it had to concern itself with all the details of timing and communication. This explains the need for special peripheral devices to handle these functions. These devices are programmed by the Z80 CPU to define various aspects of their function and, once programmed, run by themselves, interrupting the Z80 CPU only when necessary to transfer data or mark certain events. This leaves the Z80 CPU free to concentrate on data processing.

3.2 *Memory*

The microprocessor memory map (Figure 2.19) shows the memory layout in 1K byte blocks.

The lower 2K (address 0 to 07FFH) is read-only memory (ROM), in which the operating system resides. The ROM was placed here because of the Z80 hardware reset feature which branches to location 0 when executed. The hardware reset is used upon power-up and also if the automatic reset circuit detects a fatal system error.

Everything from 2K to 10K (address 0800H to 27FFH) is random-access memory (RAM). For data organization purposes the region 4K to 8K (address 1000H to 1FFFH) is divided into region 1 and region 2. The upper 2K (address 2000H to 27FFH) is used for the interrupt-vector table and the stack. The 2K

to 4K (address 0800H to 0FFFH) region is not used.

The ROM resides in an Intel 2716 EPROM which is 2K by 8 bits. The RAM consists of 16 Hitachi HM472114-4 low power memory chips, each chip organized as 1K by 4 bits.

In order to fit 8K of RAM on a 5 inch by 5 inch printed circuit board (see Figure 3.1) advantage was taken of the similar pinout of the chips which allowed them to be stacked 2 high. Only the $\overline{CS}$ (pin 8) was different on the stacked chips so it was jumpered directly to the printed circuit board from the chip on top.

The question immediately arises of the heat dissipation qualities of the 2114 RAM chips in this configuration. If the chips exceed their heat specifications (0 - 70°C) they may fail.

A study was made of this problem using a simple circuit to test the RAM's while their temperatures were monitored with thermistors (see Appendix I).

The study concludes that the specifications are not exceeded in several different environments and subsequent testing and use of the rocket-borne microprocessor system has also shown reliable operation of the RAM's.

Appendix V.3 shows the organization of the 2114 RAM chips. The data bus is split into high- and low-order bits since the 2114 RAM's are only 4 bits wide. This means that each pair of 2114 RAM chips constitute 1K by 8 bits of memory.

The 4 address lines input to the decoder select which 1K block of memory is to be enabled during a memory request by the CPU. Only one select line is active during a memory request enabling only one pair of RAM chips or the EPROM thus avoiding memory conflicts.

3.3 *Parallel Input/Output (PIO)*

The parallel input-output port (PIO) is used to input the energetic particle data from the pulse-height analyzer board. Only 6 of the 8 bits are actually used and contain the detector information (2 bits) and pulse-height information (4 bits). The remaining 2 bits are set to zeros.

Only port A is used and it is addressed by the CPU as shown in the PIO hardware addressing table (Table 3.1). Port A is used in the input mode which is set by the operating mode vector shown in the PIO software control table (Table 3.2). Other options selected by the microprocessor are also indicated.

Communication between the PIO and the PHA board is coordinated by the $\overline{ASTB}$ and ARDY handshake lines. The $\overline{ASTB}$ is pulsed every 50 μs and

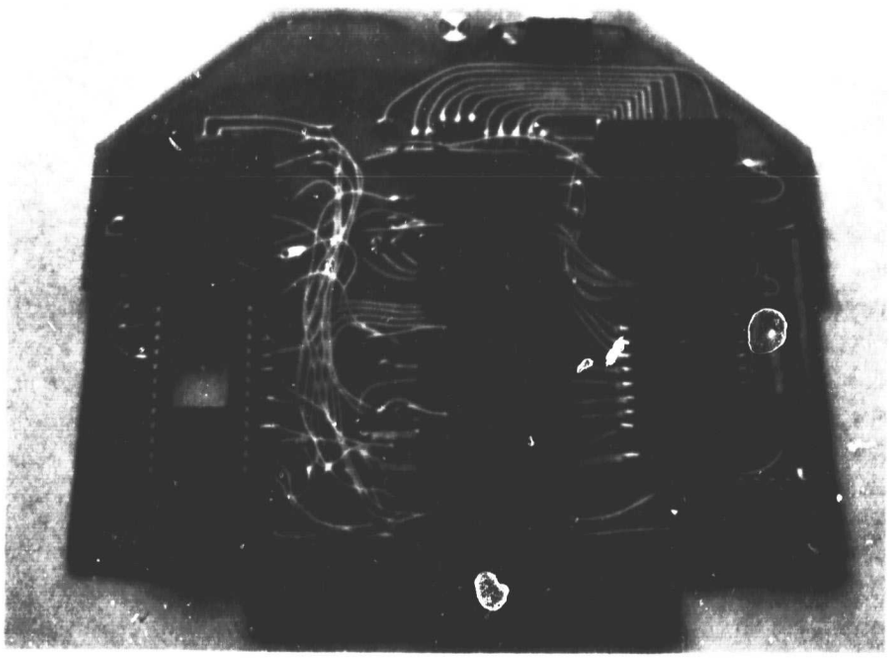


Figure 3.1 Microprocessor memory board. Notice the 16 RAM chips stacked 2 high. The EPROM is on the left and the decoder is on the right.

Table 3.1 This table shows how the PIO chip must be addressed to avoid conflicts with the other peripheral devices. For example, the OUT (19H),A instruction will send the contents of register A to PIO port A control and will not conflict with the SIO or CTC control or data registers.

PIO Hardware Addressing

Address lines	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Hex Value	Function
PIO pin	nc	nc	nc	nc	nc	$\overline{CE}$	Port A/B select	Control/Data select		
Values	x	x	x	1	1	0	0	0	18	Port A data
	x	x	x	1	1	0	0	1	19	Port A control
	x	x	x	1	1	0	1	0	1A	Port B data
	x	x	x	1	1	0	1	1	1B	Port B control

x = don't care

Table 3.2 This table shows the settings of the PIO control vectors which determine the functional characteristics of the PIO. See Appendix II.1 flight program lines 25-31, 101-103, and 144-146 for the runtime example.

PIO Software Control

Control vector	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Comments
Interrupt vector	V_7	V_6	V_5	V_4	V_3	V_2	V_1	0	Pio vector resides at 27F6H
Settings	1	1	1	1	0	1	1	0	
Operating mode vector	M_1	M_0	x	x	1	1	1	1	Input mode $M_1 = 0$, $M_0 = 1$
Settings	0	1	0	0	1	1	1	1	
Interrupt control vector	EI	And/Or	High/Low	Mask Follows	0	1	1	1	
Settings	1	0	0	0	0	1	1	1	Enable interrupts

x = don't care

loads the data from the PHA into the port A input register which causes a CPU interrupt to be generated. As soon as the CPU is able to service the interrupt and fetch the data from the port, the ARDY line becomes active and the port is ready for the next data word.

The time it takes the microprocessor to completely service the interrupt must be less than the incoming data rate of 50 µs per input byte or the CPU would be overrun with interrupt requests. The PIO interrupt service routine (see Table 3.3) shows that the length of time needed to handle PIO input data is 39.2 µs which leaves 10.8 µs out of every 50 µs for the CPU to service its other peripherals. Since the PIO is buffered by a first-in-first-out (FIFO) memory which contains 40 nine-bit registers the data will stack up if the CPU is unable to immediately service the PIO interrupt. As soon as the CPU is available to service the interrupt request, the FIFO will be quickly emptied.

3.4 *Serial Input/Output (SIO)*

The Mostek 3884 serial I/O (SIO) chip is an extremely versatile communications device with 2 ports and many options which are software selectable.

Channel A is used to transmit all of the microprocessor data and is completely interrupt driven. The hardware addressing of the SIO is shown in Table 3.4.

The transmitter contains 2 data buffers: the output shift register and the transmitter buffer which holds the next data word to be output. When the transmitter buffer is empty the SIO informs the CPU through an interrupt that it needs data during which time the shift register is busy shifting out the former data word. Since the transmitter buffer is loaded long before a data word is shifted out the data flow is continuous.

Notice that the CPU never has to poll the SIO to see if it needs data (as in many other systems). This adds to the overall system efficiency.

The SIO contains 8 control registers (0-7) which select the many options available. Write register 0 is used for several commands in addition to pointing to any of the other 7 write registers. For example, if write register 5 is to be set, write register 0 is first output with a value of 5 (points to write register 5) and then the value of write register 5 is output.

Using this method the interrupt vector is loaded into write register 2. Write register 4 is used to select the clock rate Φ×16 (the bit rate divided

Table 3.3 Listing of the approximate execution times of system subroutines. The most critical routine is PIOINT which must execute in less than 50 μs which is determined by the incoming data rate.

Name	Function	Approximate Execution Time
NMINT	non-maskable interrupt service routine	88.8 μs
PIOINT	parallel I/O interrupt service routine	39.2 μs
SIOMK	serial I/O marker byte interrupt service routine	96.0 μs
SIOBK	SIO record (block) number byte interrupt service routine	96.0 μs
SIOSR	SIO sector number byte interrupt service routine	96.0 μs
SIOCK	SIO error check byte interrupt service routine	96.0 μs
CTCRG1	CTC memory region 1 interrupt service routine	54.8 μs
CTCRG2	CTC memory region 2 interrupt service routine	54.8 μs
REGIN1	memory region 1 setup subroutine	58.8 μs
REGIN2	memory region 2 setup subroutine	58.8 μs

Table 3.4 This table shows how the SIO is connected to the rest of the system and how it must be addressed to avoid device conflicts. Port addresses 0C, 0D, 0E, and 0F will enable SIO channels only.

SIO Hardware Addressing

Address lines	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Hex Value	Function
SIO pin	nc	nc	nc	$\overline{CE}$	nc	nc	B select	C/D select		
Values	x	x	x	0	1	1	0	0	0C	Channel A data
	x	x	x	0	1	1	0	1	0D	Channel A control
	x	x	x	0	1	1	1	0	0E	Channel B data
	x	x	x	0	1	1	1	1	0F	Channel B control

x = don't care

by 16, 2 stop bits and disable the parity bit. Write register 5 is used to select 8 bits per transmitted character and enable the transmitter. Finally write register 1 is used to enable the transmitter interrupts. See Table 3.5 for the actual settings of the control registers.

Write register 0 is used by itself to reset channel A and reset transmitter with interrupts pending. The reset channel command is used whenever the system is initialized before any other commands are sent to the SIO. The reset transmitter with interrupts pending command is more interesting. This command disables the SIO from interrupting the CPU until the next data word is loaded into the transmitter buffer even if the transmitter buffer is empty which would normally cause an interrupt. This feature is used between the end of a block of data and the next NMI to avoid generating unnecessary interrupts. When the next NMI occurs, the SIO is started again by loading it with the first byte of the new block of data (a marker byte of 256).

Thus, there are 2 stop bits, 1 start bit and 8 data bits, for a total of 11 bits for every byte of data transmitted.

The clock frequency at the T×CA input (pin 14) is 78.125 KHz (Φ×16) (see Table 3.6) thus the bit rate is (78.125 KHz)/16 = 4.8828 KHz. Each group of 11 bits is transmitted at a rate of (4.8828 KHz)/11 = 443.89 Hz, or 1 byte of data every 2.25 ms. Thus one block of 2K (2048) bytes of data will be 4.6137 s long.

3.5 *Counter Timer Circuit (CTC)*

The counter timer circuit (CTC), with the non-maskable interrupt (NMI), is used to divide the rocket's spin into 32 equal intervals (azimuth bins). The CTC is addressed as shown in the hardware addressing Table 3.7 and the characteristics are set as shown in the software control Table 3.8.

An NMI is generated once every time the rocket completes a revolution and the CTC takes the time between NMI's and divides that time into 32 intervals.

Two of the four CTC channels are used in the counter mode which is programmed to count on the negative edge of a user-supplied clock signal. Channel 0 is decremented every 1.6384 ms and channel 1 is decremented every 51.2 μs which is 32 times faster than channel 0 (see Table 3.6).

During flight channel 0 is set to 256 counts at the beginning of each spin and at the end of the spin the channel is interrogated to determine how many times it was decremented during that spin. The actual algorithm is

Table 3.5 The many SIO options are detailed here. The runtime example can be found in Appendix II.1, flight program lines 33-61, 105-113, and 147-155.

SIO Software Control

Control vector	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Comments
Write register 0	CRC 1	CRC 0	CMD 2	CMD 1	CMD 0	Point 2	Point 1	Point 0	R's point to register 0-7, C's are commands
Settings	0	0	C	C	C	R	R	R	
Write register 2	V_7	V_6	V_5	V_4	V_3	V_2	V_1	0	Interrupt vector located at 27F4H
Interrupt vector setting	1	1	1	1	0	1	0	0	
Write register 4	Clock Rate 1	Clock Rate 0	Sync Mode 1	Sync Mode 0	Stop Bits 1	Stop Bits 0	Parity Even/ Odd	Parity	
Settings	0	1	0	0	1	1	0	0	
Write register 5	DTR	Tx Bits 1	Tx Bits 0	Send/ Break	Tx Enable	SDLC/ CRC 16	RTS	Tx CRC Enable	
Settings	0	1	1	0	1	0	0	0	
Write register 1	Wait/ Ready Enable	Wait FN/ Ready FN	Wait/ Ready on R/T	Receive Interrupt 1	Receive Interrupt 0	Status Affects Vector	Tx Interrupt Enable	External Interrupt Enable	
Settings	1	0	0	0	0	0	1	0	

Table 3.6 The time constants for the various devices are indicated in this table.

Division of ϕ	Frequency	Period	Comments
1	2.50 MHz	0.40 µs	
2	1.25 MHz	0.80 µs	
4	625.00 kHz	1.60 µs	
8	312.50 kHz	3.20 µs	
16	156.25 kHz	6.40 µs	
32	78.12 kHz	12.80 µs	SIO Ch A TxRx (pins 13, 14)
64	39.06 kHz	25.60 µs	
128	19.53 kHz	51.20 µs	CTC Ch 1 (pin 22)
256	9.76 kHz	102.40 µs	
512	4.88 kHz	204.80 µs	
1024	2.44 kHz	409.60 µs	
2048	1.22 kHz	819.20 µs	
4096	610.35 Hz	1.63 ms	CTC Ch 0 (pin 23)
8192	305.17 Hz	3.27 ms	
16384	152.58 Hz	6.55 ms	
32768	76.29 Hz	13.10 ms	
65536	38.14 Hz	26.21 ms	

Table 3.7 To avoid device conflicts the CTC channel addresses used are 14, 15, 16, and 17.

CTC Hardware Addressing

Address lines	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	Hex value	Channel addressed
PIO pin	nc	nc	nc	nc	$\overline{CE}$	nc	CSI	CSO		
Values	x	x	x	1	0	1	0	0	14	Ch 0
	x	x	x	1	0	1	0	1	15	Ch 1
	x	x	x	1	0	1	1	0	16	Ch 2
	x	x	x	1	0	1	1	1	17	Ch 3

x = don't care

Table 3.8 CTC channel 0 and channel 1 options are indicated below. The runtime example is in Appendix II.1, flight program lines 63-73, 114-116, and 156-158.

CTC Software Control

Control vector	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Comments
Interrupt vector	V_7	V_6	V_5	V_4	V_3	x	x	0	Interrupt vector located at 27F8H
Settings	1	1	1	1	1	0	0	0	
Channel control register	EI	Mode	Range	Slope	Trigger	Load Time Constant	Reset	1	
Settings	1	1	0	0	0	1	1	1	
Time constant register	TC_7	TC_6	TC_5	TC_4	TC_3	TC_2	TC_1	TC_0	

x = don't care

explained below.

Upon receipt of the NMI signal the CPU reads the contents of the CTC channel 0 down-count register. During the rocket's spin, this register has been decremented every 1.6384 ms starting at 256; therefore the spin time is equal to 256 minus the value of the down-count register multiplied by 1.6384 ms. The counts versus spin rate table (see Table 3.9) shows the range of allowable spin rates.

A normal spin rate would be 5 or 6 revolutions per second, which is in range of the CTC. However if the spin rate falls below 2.384 revolutions per second the rocket would probably have experienced a catastrophic failure. The data could still be recovered, but with great difficulty. Conversely, if the spin rate exceeds about 8 revolutions per second, structural damage will result, namely the booms will break off and 2 of the 4 detectors will be lost. The data from the remaining 2 detectors can still be used and would be processed in the normal fashion.

The number of decrements per spin from channel 0 is loaded into the channel 1 time-constant register which will be automatically loaded into the down-count register every time the down-count register reaches zero or channel 1 is reset. Since channel 1 is running 32 times faster than channel 0, the channel 1 down-count register will reach zero and generate interrupts 32 times per spin using the above scheme.

By counting the number of interrupts on channel 1 the CPU can keep track of which of the 32 sectors the rocket is in.

As long as the magnetometer signal is generating NMI's channel 0 should never time out. However, if it does, channel 0 is then used to simulate the rocket's spin. There are at least two cases where the NMI would not be generated. The first case would include various types of hardware failures and the second case is the absence of the rocket's spin before flight.

If this happens, CTC channel 0 takes the place of the NMI and is loaded with a time constant of 121 counts (which will look like approximately 5 revolutions per second; see Table 3.9). This will continue until another NMI is received at which time the NMI's will determine the spin rate.

3.6 *Magnetometer Interrupt Processor*

The magnetometer interrupt processor is used to generate non-maskable interrupts (NMI's) each time the rocket completes one revolution (see

Table 3.9 This table shows the range at allowable spin rates for the microprocessor system. Spin rates outside of this range would indicate catastrophic vehicle failure. Normal spin rates for the Taurus Orions are around 5 or 6 rps.

CTC Channel 0
counts vs spin rate

Rocket spin (rps)	Period (sec)	Number of counts at 1.6384 ms/count	
1	1.0000	610.35	
2	0.5000	305.17	
2.38	0.4194	256.00	
3	0.3333	203.45	
4	0.2500	152.58	
5	0.2000	122.07	
6	0.1666	101.72	in range of CTC
7	0.1428	87.16	
8	0.1250	76.79	
9	0.1111	67.81	
10	0.1000	61.03	
20	0.0500	30.50	
30	0.0333	20.34	
40	0.0250	15.26	
50	0.0200	12.20	
60	0.0166	10.17	
70	0.0142	8.72	
80	0.0125	7.63	
90	0.0111	6.78	
100	0.0100	6.10	
610.35	0.0016	1.00	

Figure 3.2). As the rocket spins in the earth's magnetic field, the magnetometer generates a sinusoidal signal, the frequency of which corresponds to the spin rate of the rocket.

The magnetometer interrupt processor is used to find the beginning and end of each spin of the rocket. This means there must be a signal generated for each cycle of the magnetometer signal.

The negative-going zero-crossing of the magnetometer is used as a reference so that each crossing generates an NMI.

A comparator (MC3302) is used to convert the magnetometer signal to a square wave of the same frequency (see Appendix V.2). The comparators reference (+ input) voltage is set at approximately 2.5 V to match the DC offset of the magnetometer signal. The output of the comparator is fed into a one shot (1/2 74LS123) which is triggered on the rising edge of the comparator signal.

3.7 *System Self-Checking*

Certain guarantees that the system is operating correctly are always desirable.

During testing it was observed that, at times, the PIO port would stop receiving data. Everything else functioned correctly but it was quickly observed that no valid data was being transmitted by the SIO since all the energy bins showed zero counts.

This error occurred most often when the payload was switched from external to internal power and vice versa.

In order to correct this problem, a reset inhibit circuit was used. If the reset inhibit circuit is not addressed by the microprocessor during a given length of time, a system reset will be activated causing the microprocessor to reinitialize all the peripherals and start over.

The ARDY signal of PIO port A was chosen to address the reset inhibit circuit. If the ARDY signal becomes stuck and fails to change over a 50 ms time period the system will be reset. ARDY is important since it not only tells if the PIO is receiving data, it also indicates that the microprocessor is responding to the FIFO. Since data are being received by the microprocessor every 50 μs, the 50 ms requirement of the reset inhibit circuit is easily satisfied. Figure 2.23 shows the timing diagram and Appendix V.1 shows the hardware involved.

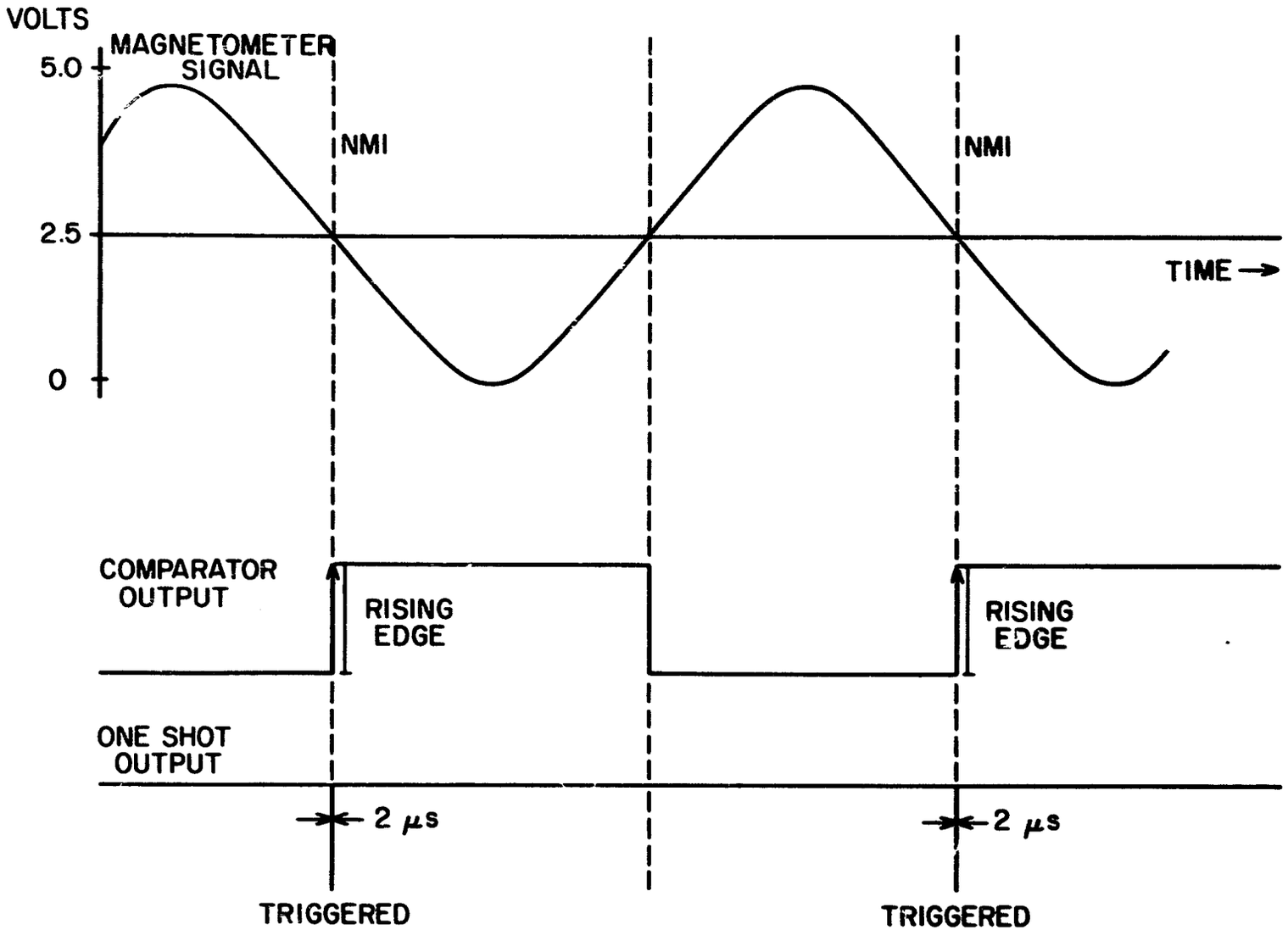


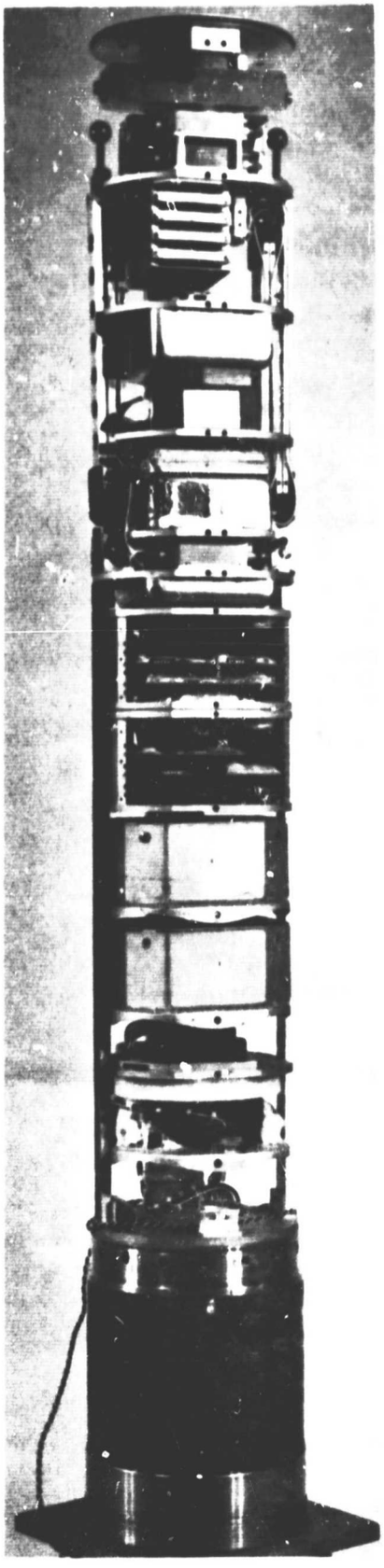
Figure 3.2 Magnetometer interrupt timing. The comparator is set to trigger at 2.5 V producing a square wave from the magnetometer signal. Once every cycle the rising edge of the comparator causes a one shot to trigger giving a 2 μs NMI pulse.

3.8 *Physical Organization*

The microprocessor system is physically split into 3 sections (see Appendix V.1, V.2 and V.3). Each section is contained on a 5" 5" printed circuit board all of which are mounted in a special box and connected together via a common backplane.

This arrangement was chosen to allow the experiment to be included in payloads of 6.5" outside diameter (used on Nike Apache and Nike Orion rockets) as well as in payloads of 12" diameter (for the Taurus Orion rockets). Figure 3.3 shows the microprocessor experiment in the payload of Nike Orion 31.014, scheduled for launch in the summer of 1981.

For testing purposes the box can be removed from the payload as shown in Figure 3.4. Special extender cards were made to give easy access to the circuit under test.



Microprocessor
Experiment

Figure 3.3 Nike Orion payload.

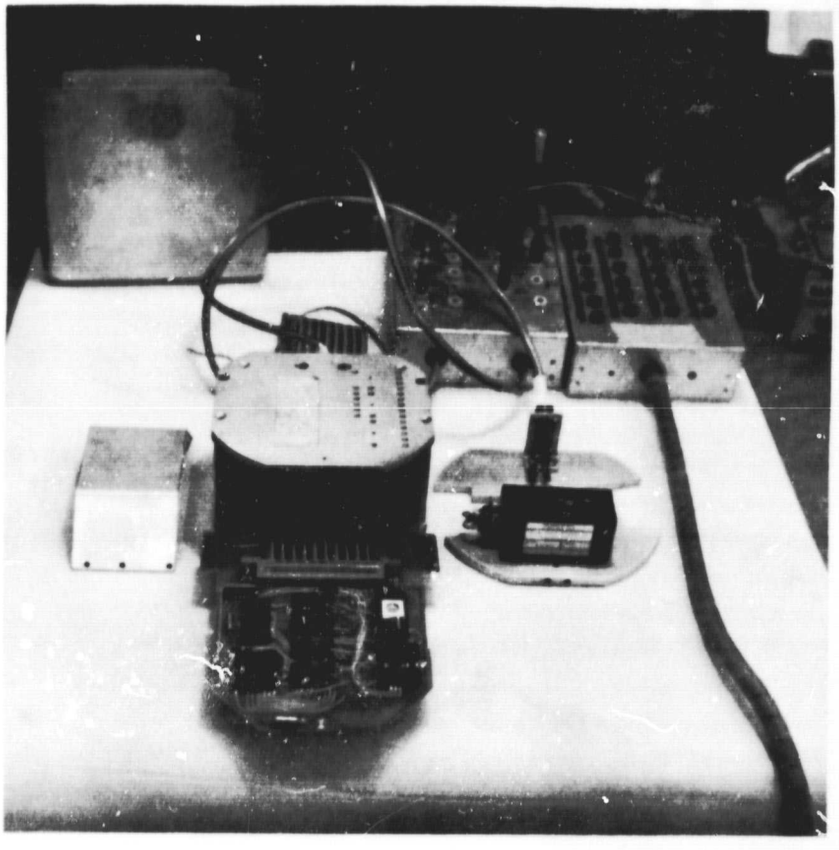


Figure 3.4 The microprocessor system can be easily tested on the bench by removing the box from the payload and using extender cards to access the circuit.

4. ROCKET-BORNE SYSTEM SOFTWARE

4.1 *Introduction*

The data processing algorithm is simple in nature. Each energetic particle is defined by detector number, pulse height, and sector which maps into a unique memory location. If several energetic particles accumulate in the same memory location during a given period of time, then the count rate (number of energetic particles per time) of energetic particles with particular characteristics can be established.

The implementation of the data processing algorithm can be followed from the program listing (Appendix II.1) with the help of system flowcharts (Appendix III) and the register layout in Table 4.1.

4.2 *Data Processing*

While the algorithm is simple in concept the implementation is difficult. Time constraints and interrupt synchronization are major obstacles.

The Z80 has many powerful software features at the assembly language level which can be used to great advantage in applications such as this. However trying to write Z80 code similar in nature to FORTRAN code written for a mainframe computer will defeat the purpose of the Z80's advanced architecture.

The vectored interrupt mode (see Appendix IV) was used extensively to link subroutines together with the main program. Normally a subroutine is called explicitly from a main program, however the Z80 allows subroutines to be called through actions initiated by an external device. The subroutine is called only when needed as opposed to the CPU constantly polling the device. In fact, the CPU never issues an explicit call to any of the peripheral devices.

For example, assume that the PIO receives one byte of data from the FIFO on the PHA board. With this data now available for the CPU the PIO issues an interrupt request. Using the I register and one byte of information from the PIO, an address pointer is formed which directs the CPU to a memory location containing the address of the routine which will service the PIO. The CPU fetches the address and branches to the service routine.

When the routine is finished control can be returned to the main program.

The interrupt vector table (see Figure 2.24) shows that the address of the PIO vector is 27F6H, therefore, upon initialization of the system, the I

Table 4.1 This table shows which system variables are assigned to the various Z80 registers.

A	General purpose
B	Sector time constant
C	Bit 1,C: ready for region change, Bit 7,C: may go ahead and change region
D, E	Contains region and sector bits
H, L	Combined with DE and PIO data to form data address
A'	General purpose
B', C'	Same as BC
D'	Sector number
E'	Record number
H', L'	Holds address of byte to be output by SIO
IX	Interrupt table address
IY	Not used
SP	Stack pointer

register is set to 27H and the PIO interrupt vector pointer is set to F6H (see Appendix II.1, lines 29 and 76).

The interrupt vector table is stored at the top of RAM memory which allows it to be changed at any time. As long as the I register and device interrupt vector pointer are not changed, the device will point to the same memory location which means if that memory location is changed to point to another routine, it is possible to branch to any one of several routines.

Setting the locations 27F4H and 27F5H to the SIOMK routine (line 236, address 0097H in the flight program listing, Appendix II.1) will cause the SIO to branch to SIOMK every time it issues an interrupt. If the table is changed at some point to SIOBK (line 290, address 00D1H) then the SIO will branch to SIOBK instead.

The runtime stack is another important feature of the Z80. A stack is like a last-in-first-out buffer accessed by PUSH and POP instructions which store and retrieve data from the stack respectively. The address of the top of the stack is held in the stack pointer register.

The stack is used mainly to store the machine status during interrupt servicing by PUSHing the AF register pair at the beginning of a subroutine and POPping the AF pair at the end. This avoids interference of the main program by interrupting routines. There are many other uses of a stack, though not needed in this application, such as dynamic memory allocation and parameter passing among subroutines.

The Z80 has two separate register sets: the primed and unprimed (which are identical), but only one set can be accessed at any given time. The EXX and EX AF, AF' instructions are used to change between register sets.

The exchange feature is useful for keeping track of system parameters since most of the information was held in the Z80 machine registers (see Table 4.1).

Indexed addressing is accomplished using the IX and IY index registers. Indexed addressing allows easy access to tables of information by allowing a displacement off of the address held in the index register. The instruction LD(IX + 6), L (see line 434) will load the contents of register L into the memory location which is 6 bytes after the address in the IX register. This feature is used to access the interrupt vector table at the top of memory.

Processor interrupts may be enabled and disabled with the EI and DI instructions respectively, preventing certain critical operations from being disturbed. Interrupt service routines are entered with interrupts disabled so that no other interrupt may occur until they are explicitly enabled by the program. Thus the programmer can force the interrupting devices to wait their turn while others are being serviced. Interrupts are explicitly disabled (lines 429 through 446) during the resetting of devices and initialization of system parameters, for example.

4.2.1 *Assembler.* Several features of the SDB 80 assembler greatly aided program development.

The most important feature is the ability to define and use symbolic labels, in fact the first two pages of the flight program (Appendix II.1) are used to define each label and comment on its use. If a label is encountered in the program code, a glance at the first two pages will explain its use. The label itself is usually enough of a hint as to its use in the program thus making the code self documenting.

For example, consider the following code from lines 200 through 203.

```
LD      A,CTCENA
OUT     (CTC0),A
LD      A,CTC256
OUT     (CTC0),A
```

Now consider the identical code without labels.

```
LD      A,0C7H
OUT     (14H),A
LD      A,256
OUT     (14H),A
```

One glance at the labeled code tells that something is happening to CTC channel 0 and, with a few comments, the code is quickly and easily understood.

Labels also allow easy program modification and consistency. If it is found that CTC0 in the above code should have been a 15H instead of 14H one change to the label definition would cause the assembler to automatically change every occurrence throughout the program instead of the programmer having to hunt every occurrence of 14H.

Comments are used liberally to aid in program development and debugging.

Other features such as address, object code and statement number columns are standard features of any assembler. Assembler error messages quickly catch any syntax errors or undefined labels.

5. LABORATORY SUPPORTING SYSTEMS

5.1 *Introduction*

The operating system of the rocket-borne microprocessor, which resides in the EPROM, was developed on the MOSTEK software development board (SDB 80), Figure 5.1. The SDB 80 work station is pictured in Figure 5.2.

The SDB 80 is used to transform the Z80 source code into machine code which is then transferred to an EPROM to become the operating system of the rocket-borne system.

The source code is the human readable version of the Z80 instructions (see Appendix II) which make up a program. An assembler and linking loader takes this source code and translate it into machine code which is executable by the Z80 central processor.

The editor, assembler, and linking loader of the SDB 80 were adequate for this application. However, the memory requirements of the flight program were barely satisfied.

5.2 *EPROM Programmer*

The EPROM programmer, as described by *Davis et al.* [1979], worked reliably with the exception of the failure of one of its counter chips which was replaced.

The 2716 EPROM chips with their 2K×8 bit capacity prove to be more than enough memory required by the approximately 475 byte flight program.

Of several different brands of EPROMs used, the INTEL 2716 was found to be the easiest to erase and reburn.

5.3 *Cassette Tape Transport*

The BETA-1 cassette transport (manufactured by MECA) is a valuable tool in the development of the system.

Prior to the addition of the BETA-1 to the SDB 80, the only means of storage was paper tape generated on an old teletype machine. Simple programs would take 30 minutes or more to punch or read which made program development very tedious. The BETA-1 cut program access to less than 1 minute allowing longer programs to be easily developed.

The SDB 80 is connected to the BETA-1 through its parallel interface #1. Port A is used for receiving data from the BETA-1 while port B transmits data. A copy of a page of the SDB 80 manual shows how socket U16 is strapped to give the correct polarity to the handshaking signals (Figure 5.3).

The BETA-1 was a bit more difficult to modify since it was designed to be

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

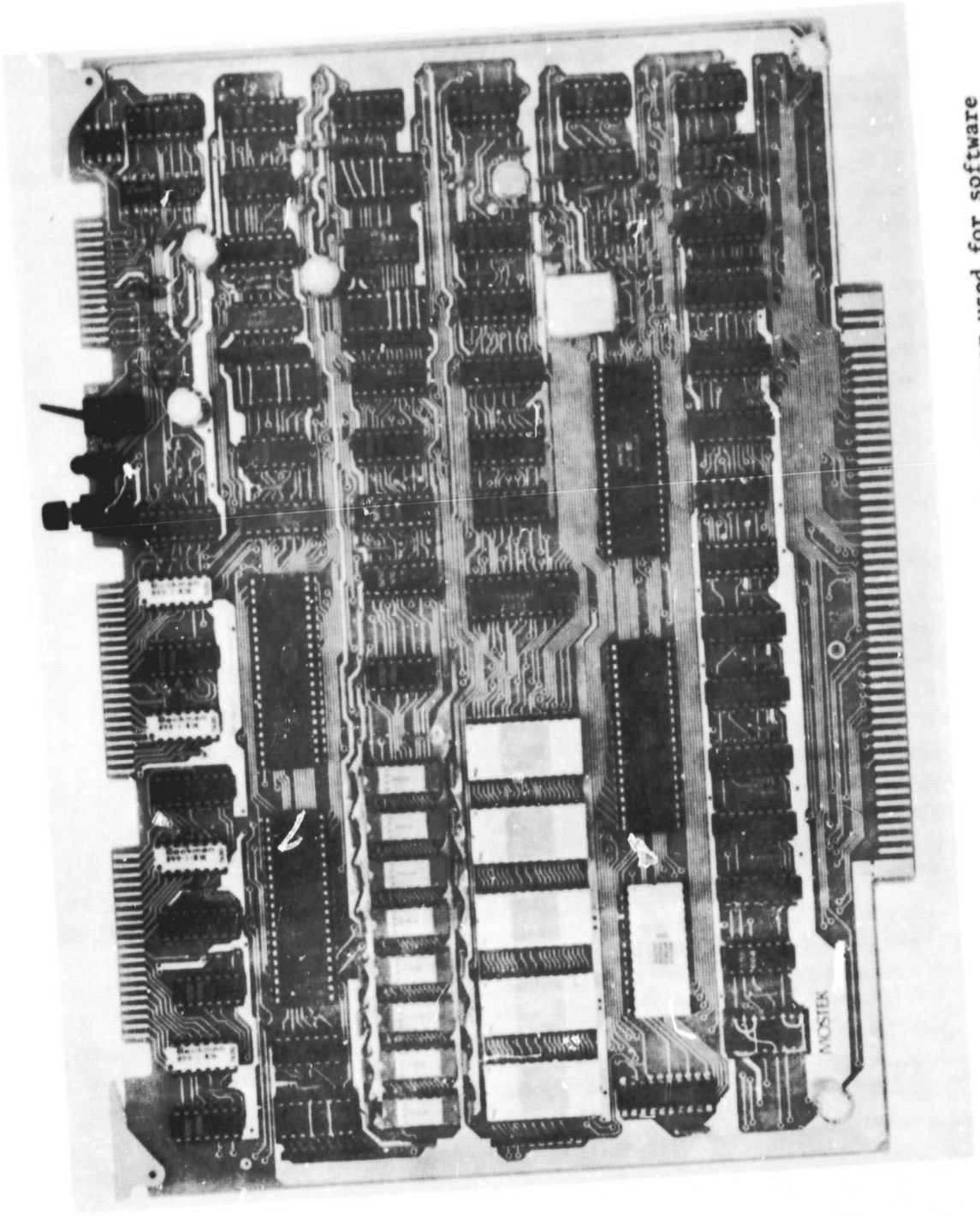


Figure 5.1 Mostek SDB 80 single board microprocessor used for software development.

Figure 5.2 SDB 80 work station. The SDB 80 is housed in a card cage to the left of the CRT. In front of the card cage is an EPROM programmer and eraser. On top of the CRT is a BETA 1 cassette tape transport used for mass storage. The teletype 33 on the right is used for program listings.

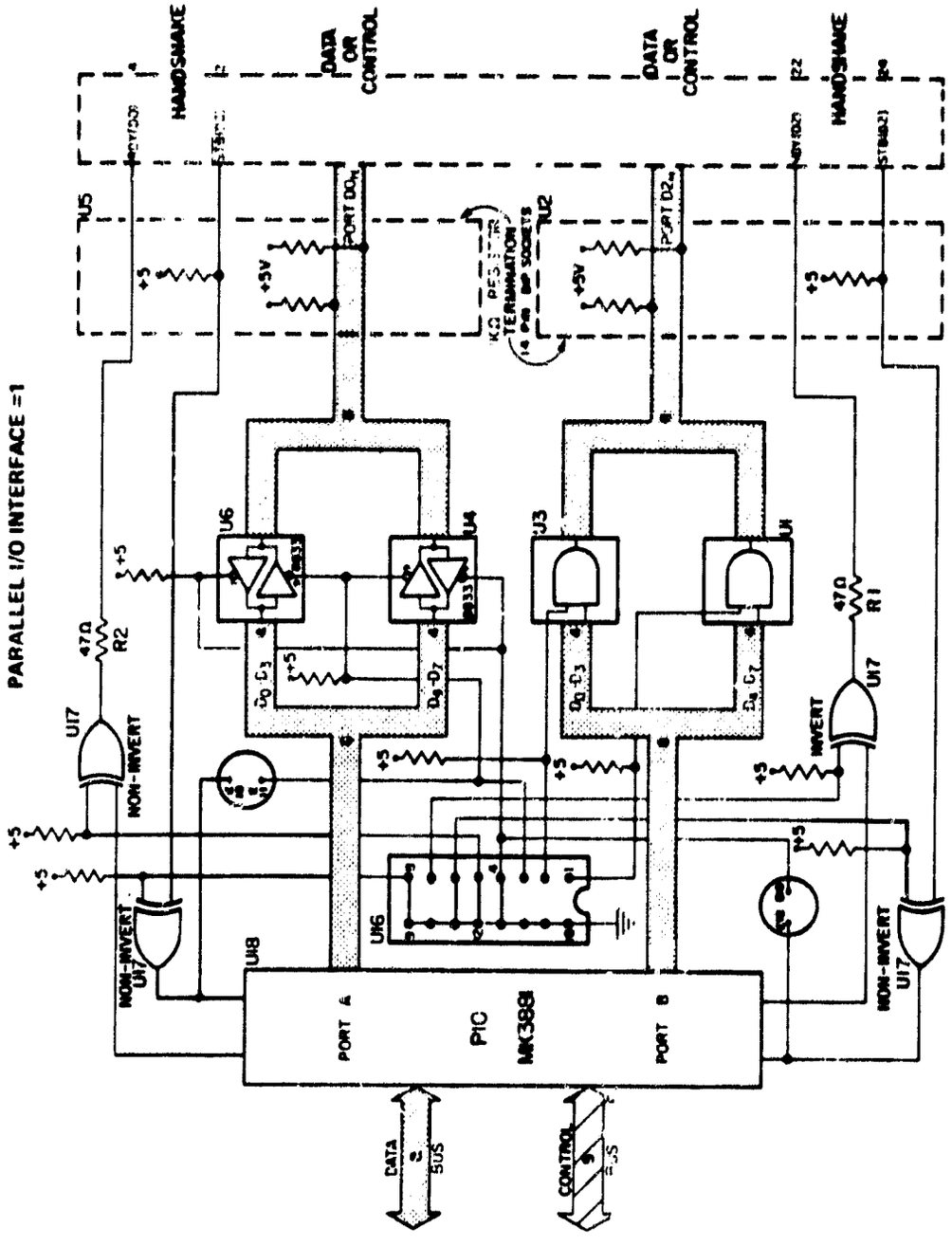


Figure 5.3 The polarities of the handshake signals are determined by the jumper options indicated on U16.

interfaced to any number of parallel I/O configurations.

As can be seen from Figure 5.4, there are many jumper options to choose from, allowing selection of handshake signal timing and polarity. The options selected for the BETA-1 are indicated.

The hardware interface would have been easy had software been supplied that was functional. However, uncertainty in both areas proved to make the interfacing difficult.

Once the programs were working, they were burned into an EPROM so they could be easily accessed by the SDB 80.

The write program (Appendix II.2) was placed in EPROM at location 4370H and the read program at 44C0H (Appendix II.3). In order to access these routines from the text editor location, :OO is set to 43D5H and :SI is set to 44C0H.

These routines allow manual operation of the BETA-1 in order to open files and to perform directory operations. These operations are described in the BETA-1 manual.

When a read (↑R) or write (↑W) operation is requested from the BETA-1 the respective data transfer is done through the special routines.

5.4 *Serial-To-Analog Converter*

In order to monitor the output of the rocket-borne system, a special serial-to-analog converter was built. In essence each data word of 8 bits was converted to a signal between 0 and 10 volts.

The circuit, Appendix V.4, is quite simple but very effective in debugging the microprocessor system. By observing the analog signal on an oscilloscope, one can immediately tell if the microprocessor is operating correctly.

The analog data stream looks exactly like that in Figure 2.21 and allows observations of each energy bin and the detector delimitors. Oscilloscope photographs of data from one detector, Figure 5.5, show that a low resolution energy spectrum can be obtained (since only 16 energy bins are available) when a radioactive source is held in front of the detector.

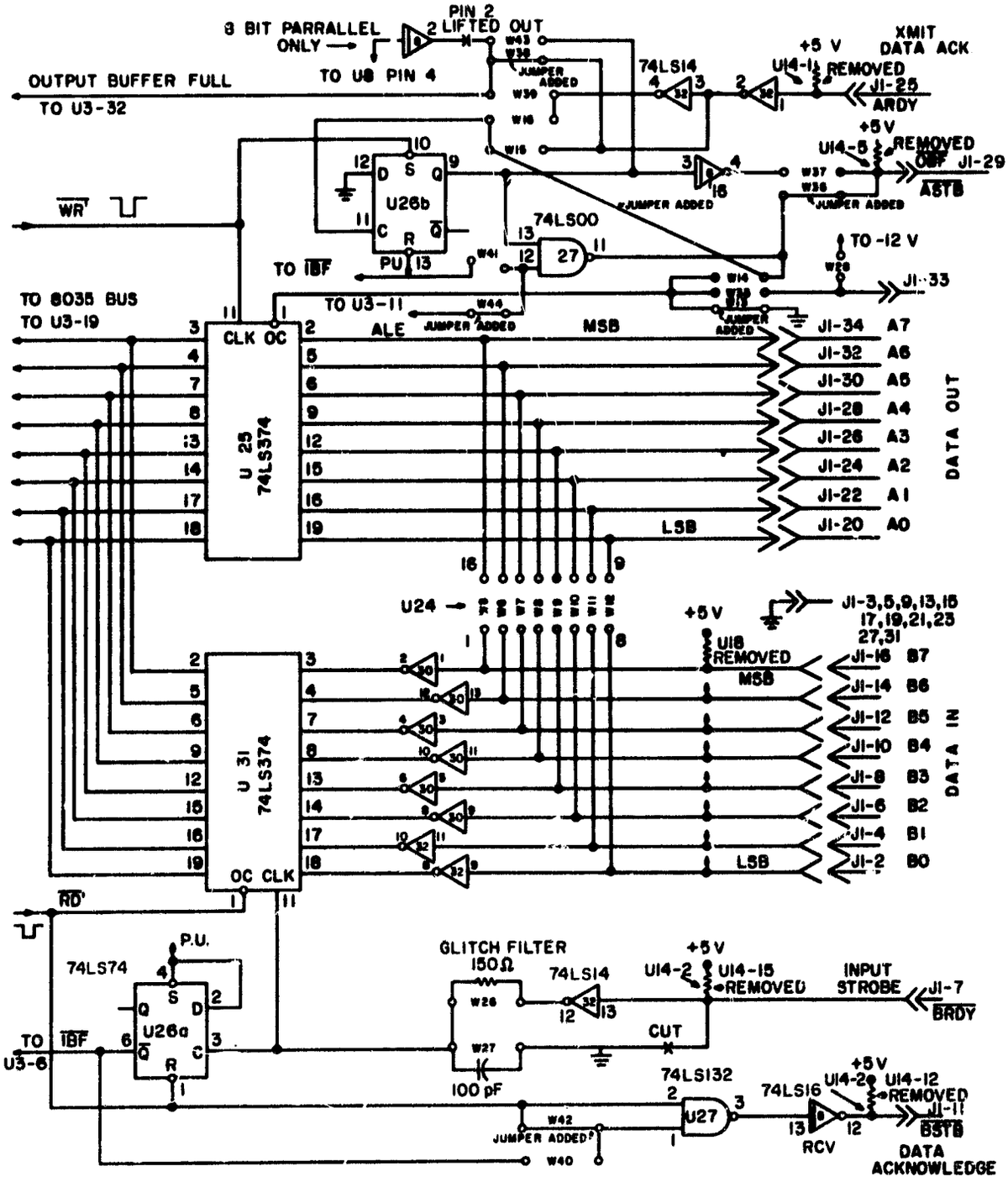


Figure 5.4 Jumper options required for SDB 80 communication are indicated. More detail can be found in the BETA-1 manual

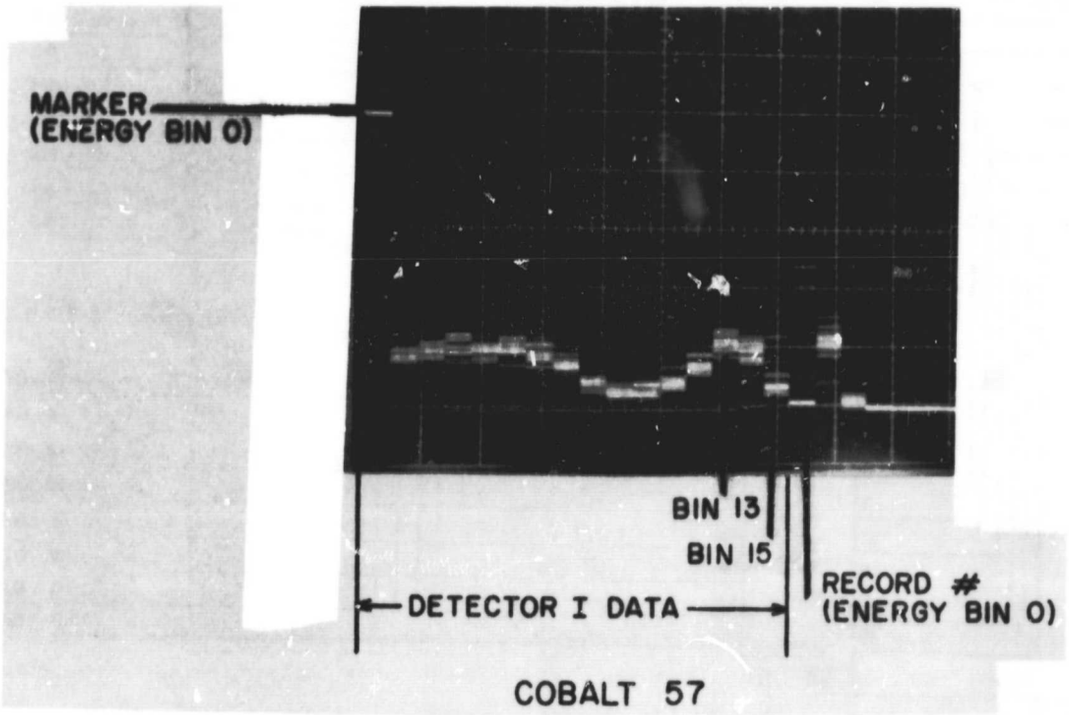


Figure 5.5 A time-exposure photograph of an oscilloscope displaying the energy spectrum from the microprocessor.

6. TESTING AND FLIGHT PERFORMANCE OF THE ROCKET-BORNE EXPERIMENT

6.1 *Calibration*

Calibration of the microprocessor system is necessary to establish the energy range (in keV) for each of the 16 bins. The primary reference used for calibration is a 1 μC radioactive source of Am 241, which has a very strong emission line at 60 keV. It provides a simpler method than the use of a particle accellerator, described in *Voss and Smith* [1974, 1977].

Only half of the -10 to 10 V range of the 8-bit A/D converter is used in the system, in particular the 128 steps over the -10 to 0 V range. Adjustment of the system's sensitivity is such that each step is nominally 1 keV: the slope is therefore 0.078 V/keV.

Assume initially that there exists a linear relationship between the output of the detector amplifiers and the particle energies as shown by the straight line on Figure 6.1. The ordinate of this graph is the peak detector (sample-and-hold) voltage at the input to the A/D converter.

The 16 energy bins are assigned to various energy ranges as shown by the ideal values in Table 6.1. The bin assignments are almost identical to these of Table 2.1: for example, a 43 keV particle would fall into bin 7. The discrepancy at the higher energies will be explained later.

Since the A/D converter uses an offset binary encoding scheme it outputs a 00H for -10 V input. Taking this into consideration a memory map, Figure 6.2, is implemented in an EPROM which for a number as input yields the correct bin number as output. Since the bin number requires 4 bits either the high or low order 4 bits can be used since they are programmed identically (explaining the double entries in Figure 6.2). A 43 keV particle, for example, will appear at the input of the A/D converter as -3.43 V and be converted to 54H. Location 54H of the EPROM contains 77, thus the bin number is 7 in agreement with the example of the previous paragraph.

Calibration of the experiment involves making sure that the straight line on Figure 6.1 matches as closely as possible the experimental results. As soon as the various amplifier gains prior to the A/D converter input are adjusted correctly, a picture similar to Figure 6.3 can be obtained.

Once the correct energy bin is obtained a pulse generator (Ortec 448) can be substituted for the detector at a special input on the preamplifier. The pulser is calibrated to select bin 10 for a 60 keV input.

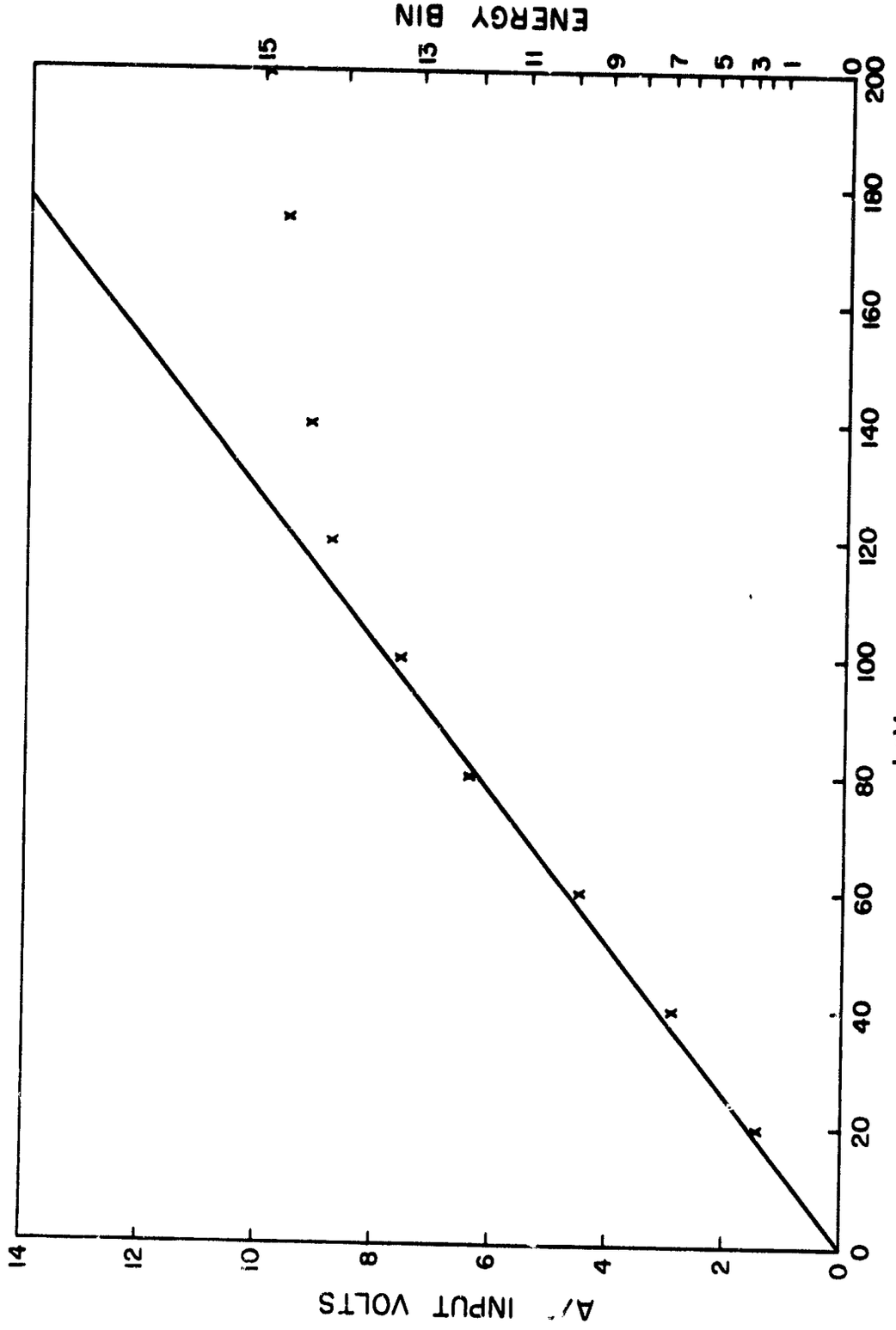


Figure 6.1 keV input to the system vs A/D input voltage.

Table 6.1 Mapping of keV into energy bin numbers. System non-linearities cause the actual energy range values to differ from the ideal energy range values at high energies.

Ideal Energy Range (keV)	Actual Energy Range (keV)	Bin Number	Sample-and-Hold Voltage (input to A/D)	EPROM Location
>127	>200	15	-10.00	00H
109 - 126	117 - 200	14	-8.59	12H
93 - 108	93 - 116	13	-7.34	22H
80 - 92	80 - 92	12	-6.32	2FH
69 - 79	69 - 79	11	-5.46	3AH
59 - 68	59 - 68	10	-4.68	44H
51 - 58	51 - 58	9	-4.06	4CH
44 - 50	44 - 50	8	-3.51	53H
38 - 43	38 - 43	7	-3.04	59H
33 - 37	33 - 37	6	-2.65	5EH
28 - 32	28 - 32	5	-2.26	63H
24 - 27	24 - 27	4	-1.95	67H
20 - 23	20 - 23	3	-1.64	6BH
17 - 19	17 - 19	2	-1.40	6EH
14 - 16	14 - 16	1	-1.17	71H
0 - 13	0 - 13	0	0.00	80H

location	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	FF	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE	EE
0010	EE	EE	EE	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD	DD
0020	DD	DD	DD	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC	CC
0030	BB	BB	BB	BB	BB	BB	BB	BB	BB	BB	BB	AA	AA	AA	AA	AA
0040	AA	AA	AA	AA	AA	99	99	99	99	99	99	99	99	88	88	88
0050	88	88	88	88	77	77	77	77	77	77	66	66	66	66	66	55
0060	55	55	55	55	44	44	44	44	33	33	33	33	22	22	22	11
0070	11	11	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Figure 6.2 Log EPROM memory map showing the locations of the 16 energy bins. The EPROM past location 72H is filled with zeros. For example, suppose the input to the EPROM is 54H then the output will be 77H.

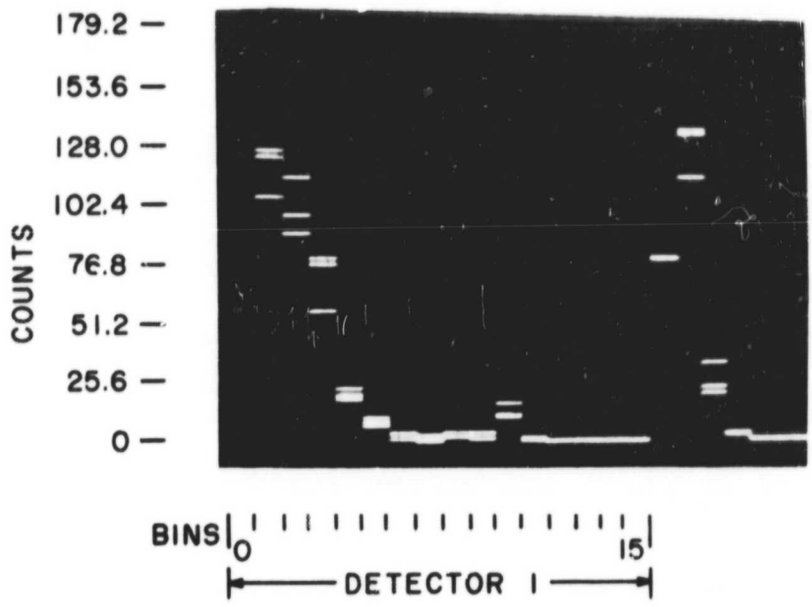


Figure 6.3 The 60 keV line of Am 241 appearing in energy bin 10 indicates correct calibration of the experiment.

Once the pulser is calibrated each energy bin can then be selected by dialing the appropriate keV level on the pulser. A plot of pulser output (in keV) versus A/D input voltage is shown in Figure 6.1 as crosses. As the higher keV levels are approached the amplifiers begin to depart from their linear range as shown by the deviation from the straight line. This means that the energy ranges of the three highest bins differ from the ideal. Each detector has its own characteristic curve.

The curve of Figure 6.1 is subject to a slight error due to the calibration of the pulser and human error in reading the A/D input voltage from the oscilloscope. This means that the actual energies of Table 6.1 are accurate to within a few keV.

The peak at 116 keV of a 0.1 μC radioactive source of Co 57 is used to check the higher energy bins. As can be seen in Figure 6.4 bin 13 is selected for 116 keV particles matching closely the actual energy range of bin 13 predicted from Table 6.1.

6.2 *Particle Sampling*

Sampling of the energetic particles encountered by the detectors is implicit in the experiment and needs to be discussed to interpret some of the flight data.

6.2.1 *Sampling procedure.* The peak detector (sample-and-hold) circuit is made up of a capacitor and an analog switch which is used to discharge the capacitor. The pulse created by an energetic particle striking a detector causes a charge to be stored on the capacitor.

The sampling sequence is shown in Figure 6.5. A detector is sampled for approximately 150 μs and, just prior to the end of this time, the voltage on the capacitor is digitized. The capacitor is then discharged and held at 0 V until the next sampling period starts 50 μs later.

6.2.2 *Sampling error.* If a particle enters a detector during the discharge time then it will not be seen at all; only if it strikes during the sampling period will it be processed by the microprocessor. The first effect of sampling, then, is that only 75% of the particles are recorded. This is true for low particle fluxes: at higher fluxes other factors must be considered and the percentage is reduced below 75% and becomes a function of energy.

Consider the case where two particles of different energies strike the

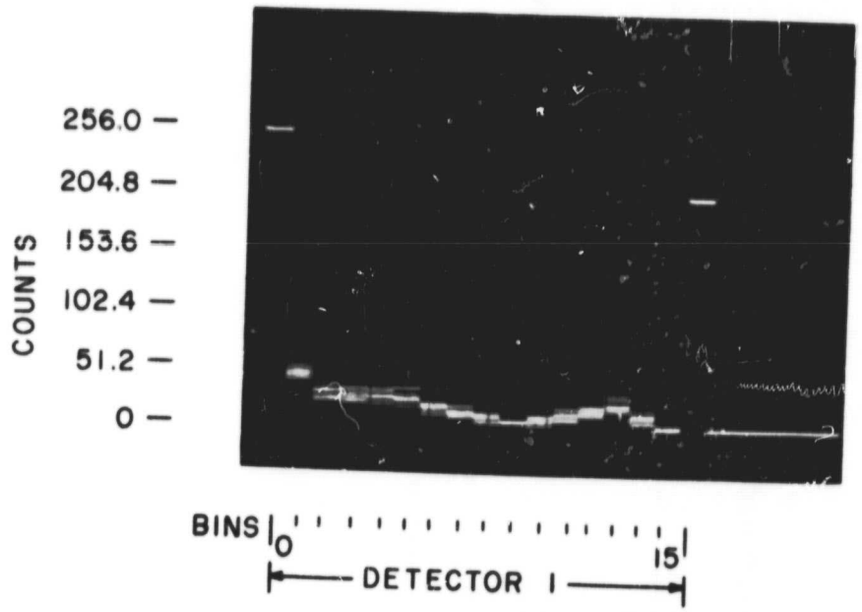


Figure 6.4 The 116 keV line of Co 57 appears in energy bin 13 helping to calibrate the higher energy bins.

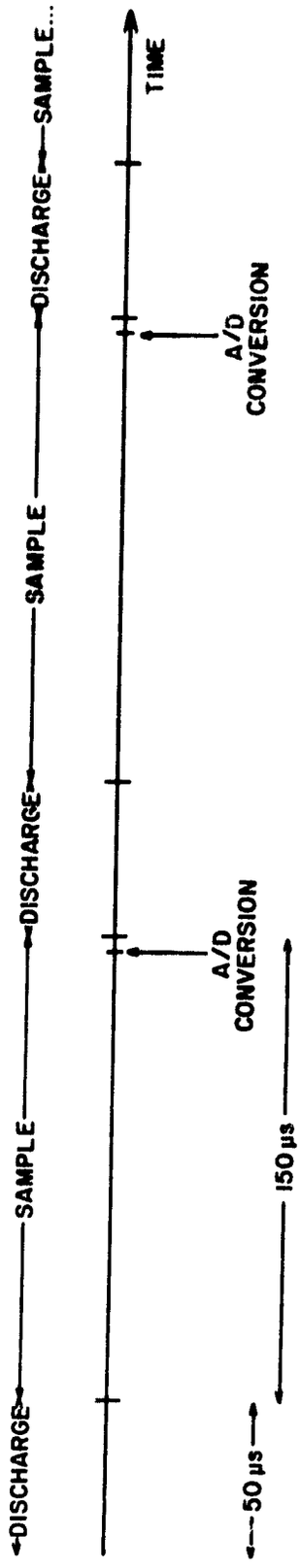


Figure 6.5 Particle sampling sequence.

detector during the same 150 μs sampling period. If this happens, only the more energetic of the two particles is seen by the A/D causing a sampling error.

If the sampling error occurs infrequently it will have no noticeable effect on the particle spectrum. Figure 6.4 is an example of a normal spectrum. This is the case if the particle flux is no greater than 1 particle every 150 μs.

As the particle flux exceeds 1 particle per 150 μs the sampling error will reduce the apparent flux of low energy particles, biasing the spectrum to the higher energy bins. Figures 6.4 and 6.6 were both produced by a 0.1 μC Co 57 radioactive source except that the source was held closer to the detector in Figure 6.6 to increase the particle flux. The preference of the higher energy particles in Figure 6.6 is shown by the higher energy bins having more counts than the lower energy bins, while the opposite is true in Figure 6.4.

If the particle flux is much higher than 1 particle every 150 μs then high energy particles may occur in every sampling period and the low energy bins would be completely empty. A strong radioactive source held away from the detector produces the display shown in Figure 6.7. If the same source is held close to the detector Figure 6.8 results. Particles in the range of bins 10 and 11 occur so frequently that they dominate any of the particles of lower energies. Energy bins 1 through 5 have zero counts because every time a particle with energy in this range is sampled a particle of higher energy has also been detected during the same sampling interval.

6.2.3. *Solution to sampling problem.* There are two ways in which the sampling problem can be solved. The first solution is to sample at a much faster rate. This presents considerable circuit difficulties and, of course, is not possible for the flights that have already taken place.

The second solution is to correct for the sampling error in the post-flight processing of the data. To do this the behavior of the experiment is carefully recorded for different particle fluxes and energies so that this information can be applied to the actual flight data. This solution reduces to an exercise in the statistics of particle counting and has not yet been worked out in detail.

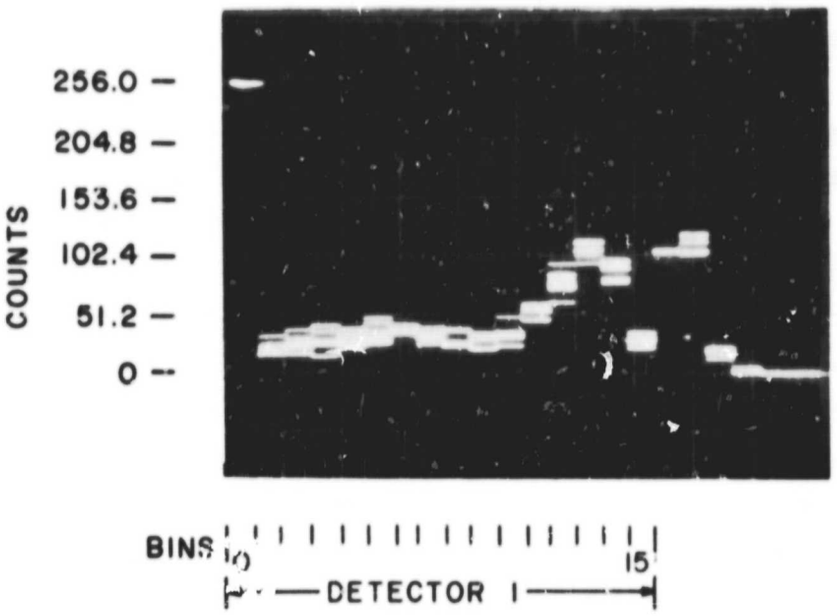


Figure 6.6 The presence of sampling error is indicated by more counts in the high energy bins than in the low energy bins.

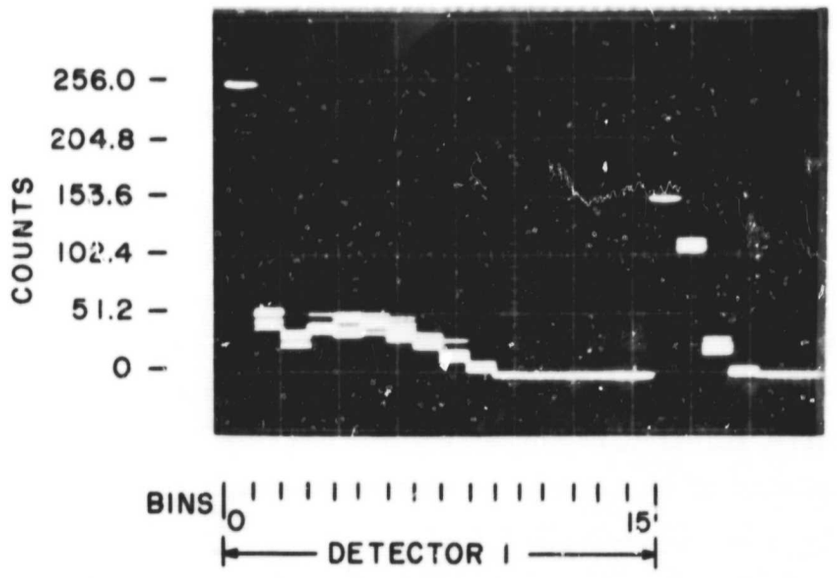


Figure 6.7 A Ni source is held far enough from the detector to prevent sampling error.

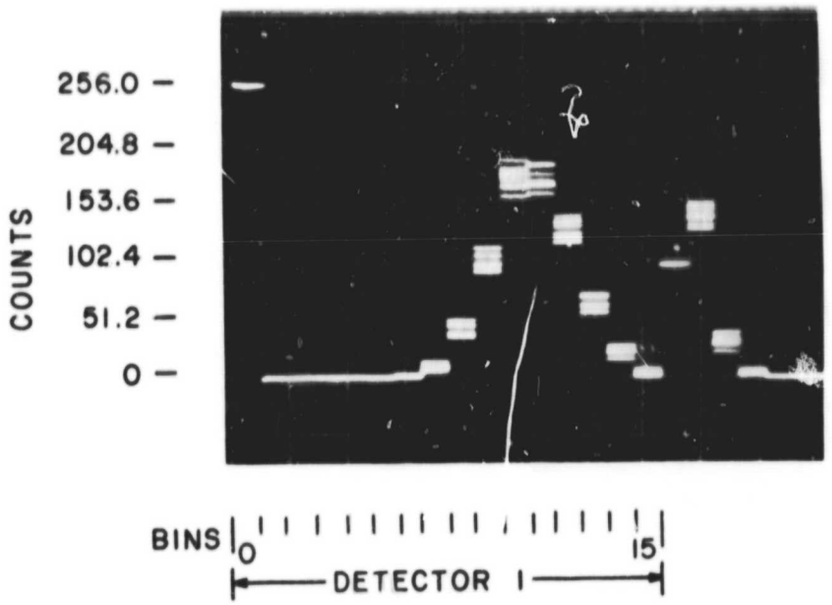


Figure 6.8 The same Ni source held close to the detector dramatically illustrates sampling error.

6.3 *Flight Data From The Energy Budget Campaign*

Figure 6.9 shows data at 134 km from Taurus Orion 33.009 launched at 23:44:30 UT on November 30, 1980. One complete sector of data is marked off by detector and energy bin numbers. Bin 0 of detector 1 contains the marker pulse of 255. Bin 0 of detector 2 indicates the block number being output is 51 (refer to section 2.3) and bin 0 of detector 3 indicates the sector is 3.

All the detectors indicate that sampling error is present since the lower energy bins are not showing as many counts as the higher energy bins. This is especially evident in detector 1 where the first 9 bins show zero counts.

The sequence of photographs in Figures 6.10 to 6.25 show data from Taurus Orion 33.009 from before launch up to apogee, demonstrating the ability of the microprocessor system to function under flight conditions.

Figure 6.10 at 10 s before launch shows the noise spectrum of each of the detectors; only the first 2 or 3 energy bins show any counts. Ten seconds after launch at 10 km (Figure 6.11) the affects of launch have caused the A/D to fill the high energy bins with counts. By 15 km (Figure 6.12) the experiment had recovered and again shows the noise spectrum. At 39 km and 48 km (Figures 6.13 and 6.14) the booms have not yet extended and the noise spectrum is still present, as it should be. At 72 km (Figure 6.15) the booms have extended and particle counts are beginning to show on detectors 1 and 2. More counts are appearing on detectors 1 and 2 at 90 km (Figure 6.16) while detectors 3 and 4 are still quiet. At 99 km (Figure 6.17) detectors 3 and 4 are beginning to show counts in the higher energy bins. At 106 km, only 5 km later, (Figure 6.18) the flux has increased considerably on detectors 1 and 2 and at 108 km (Figure 6.19) it can be seen that the sampling error has become dominant on detectors 1 and 2. Detectors 3 and 4 are showing a steady rise in counts. At 113 km (Figure 6.20) the spectrum has shifted even more toward the high energies. The sampling error has not yet affected detectors 3 and 4. At 119 km (Figure 6.21) all detectors are affected by the sampling error as the spectrum shifts even more toward the high energies. At 128 km (Figure 6.22) detectors 2 and 3 show increases in the number of counts in all their high energy bins. Complete dominance of high energy particles in bin 15 is shown in Figure 6.23 (137 km) for detector 1. Notice that the counts in bin 15 are almost at 255 which is the maximum possible number of

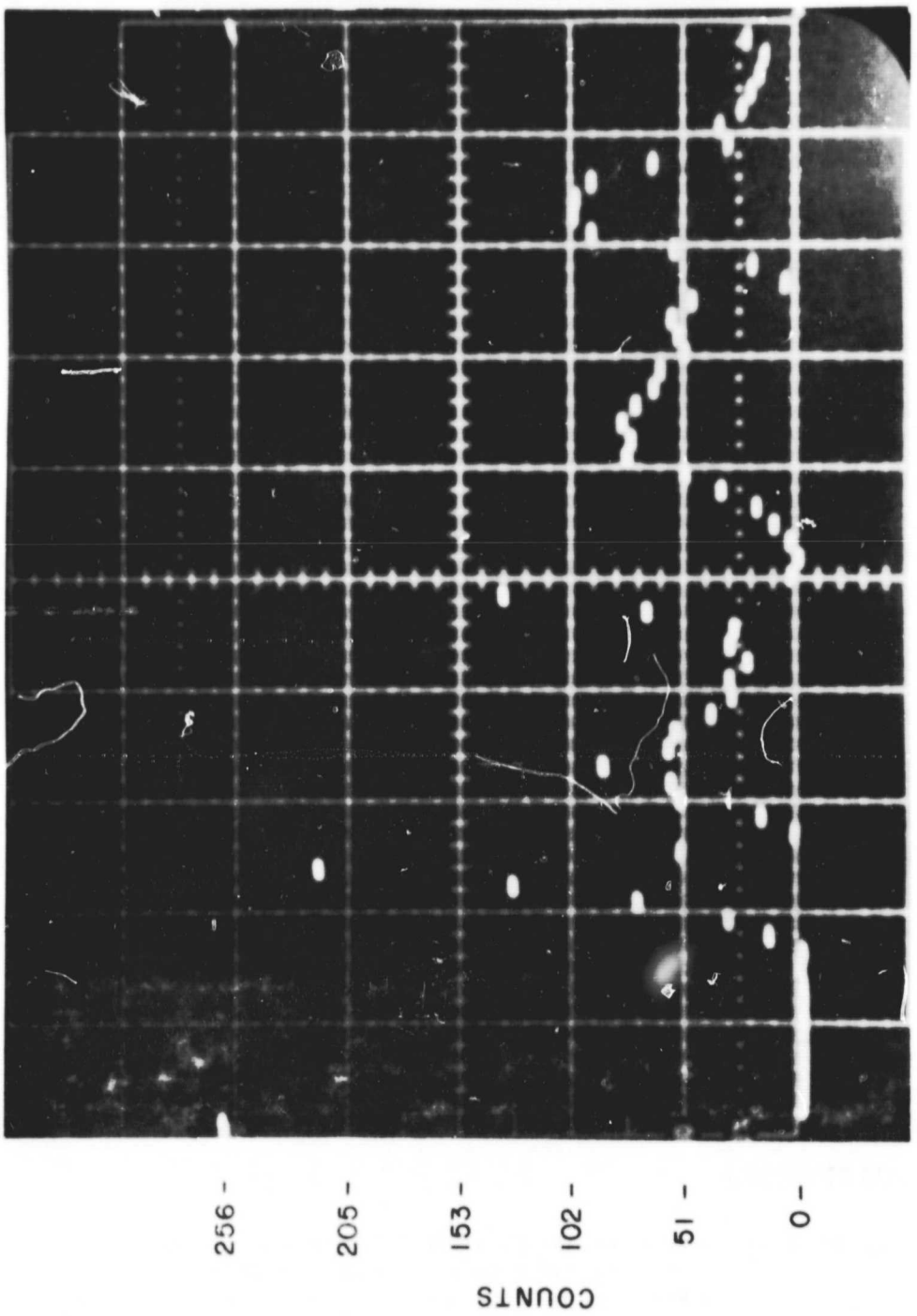


Figure 6.9 Enlargement of one sectors data from Taurus Orion 23.009 showing the individual energy bin.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

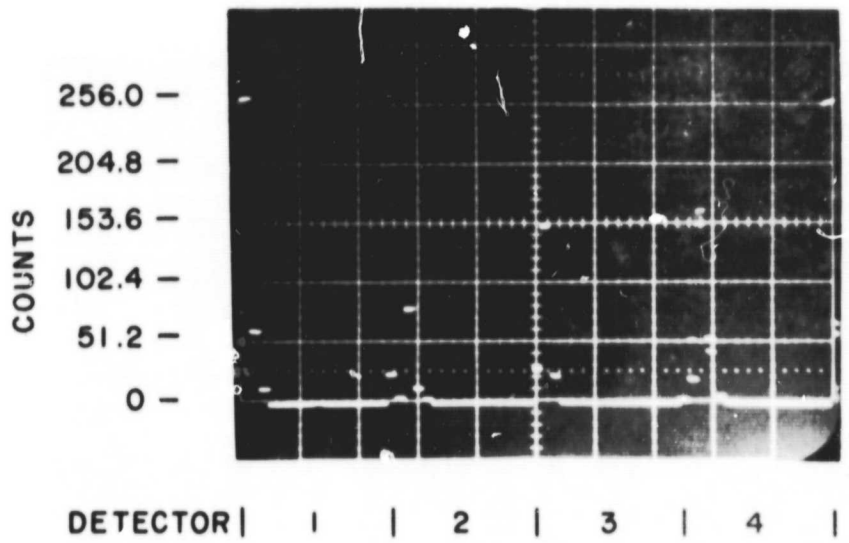


Figure 6 10 Pre-launch data shows typical detector noise spectrum (0 km).

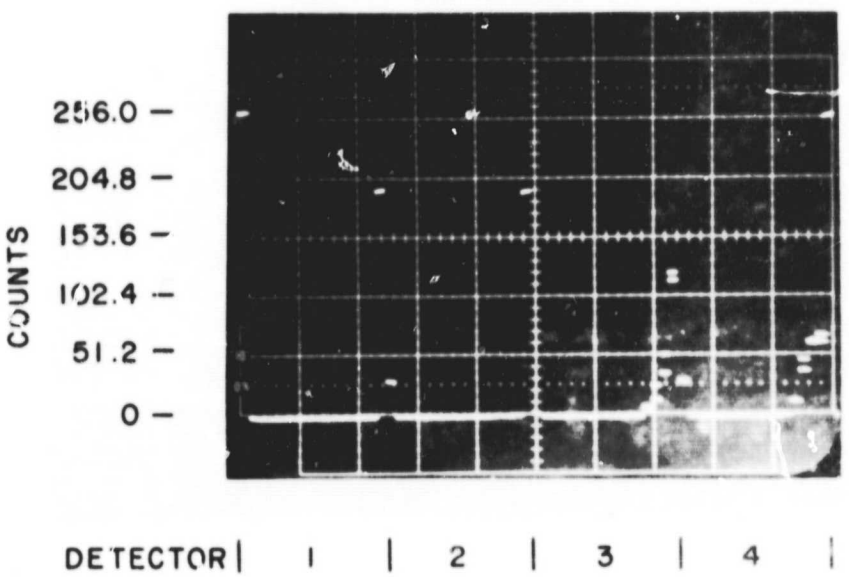


Figure 6.11 The effects of launch seems to force all the data into the high energy bins (10 km).

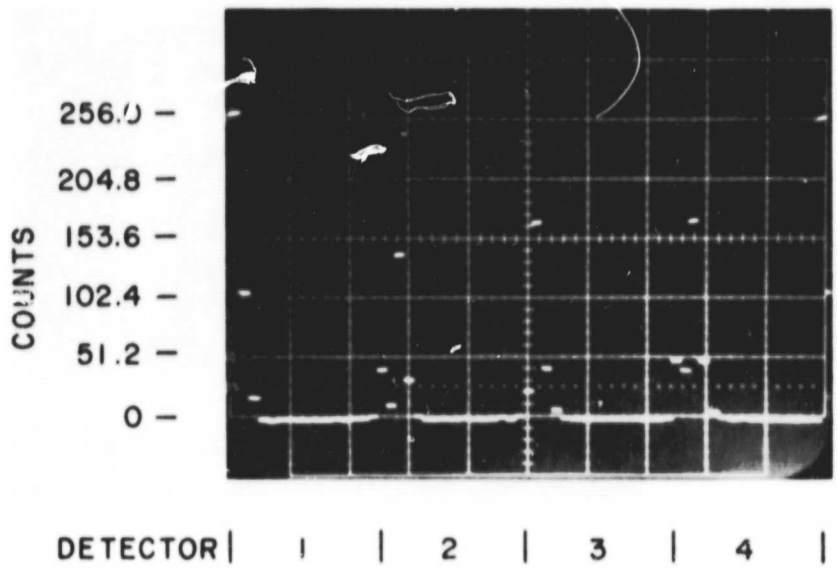


Figure 6.12 The experiment has recovered from the launch shock and again shows a typical noise spectrum. The booms are not yet extended (15 km).

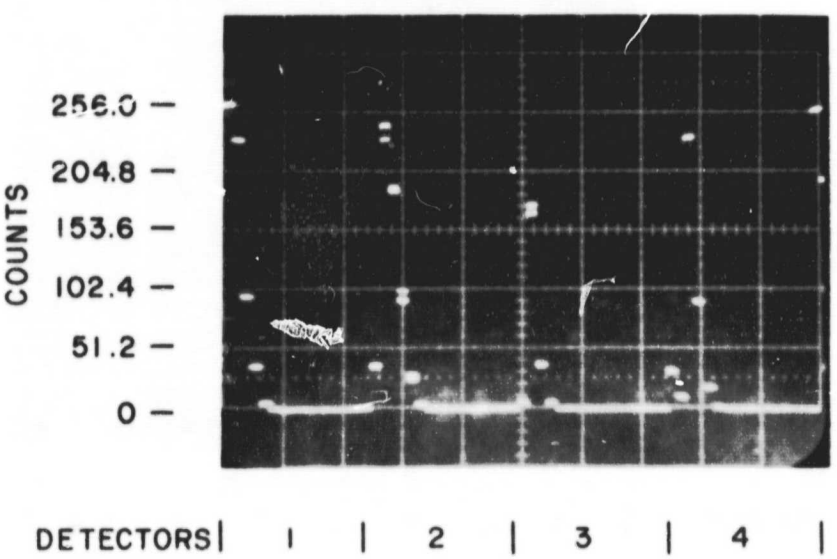


Figure 6.13 Typical noise spectrum (39 km).

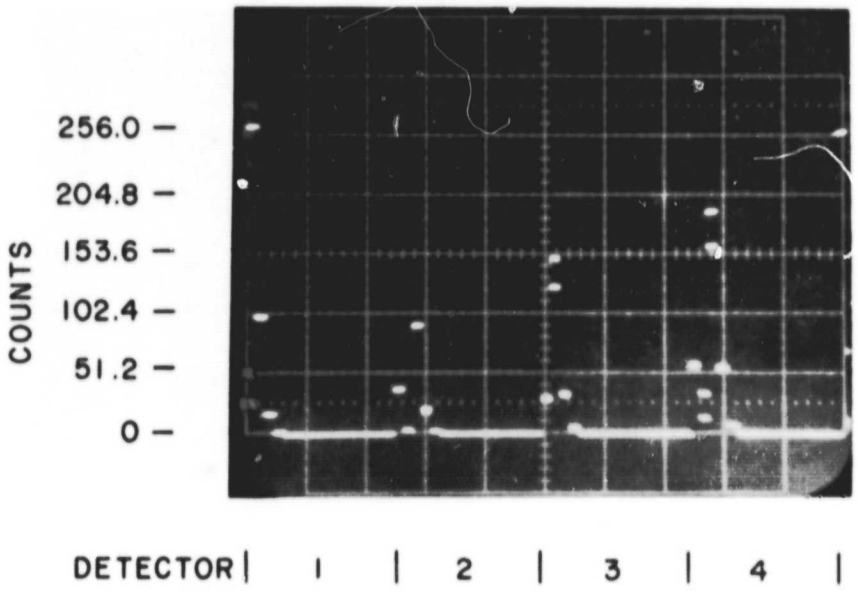


Figure 6.14 Typical noise spectrum; booms not yet extended (48 km).

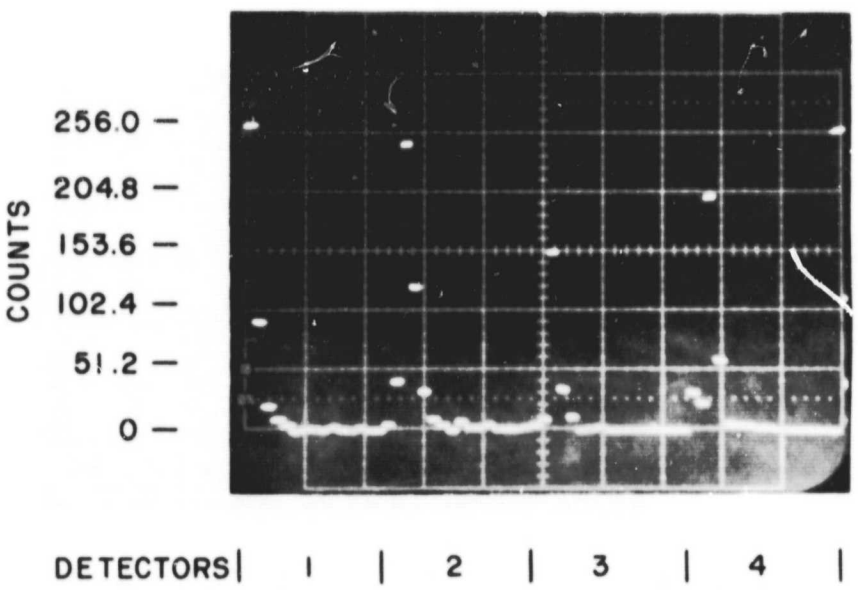


Figure 6.15 The booms have been extended and counts are beginning to show in the higher energy bins of detectors 1 and 2 (72 km).

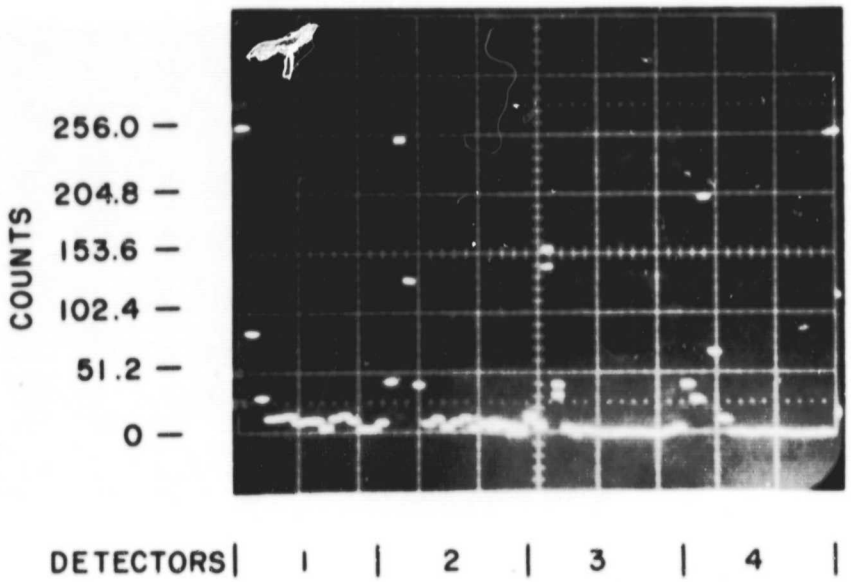


Figure 6.16 More counts are showing in detectors 1 and 2, while 3 and 4 are still quiet (90 km).

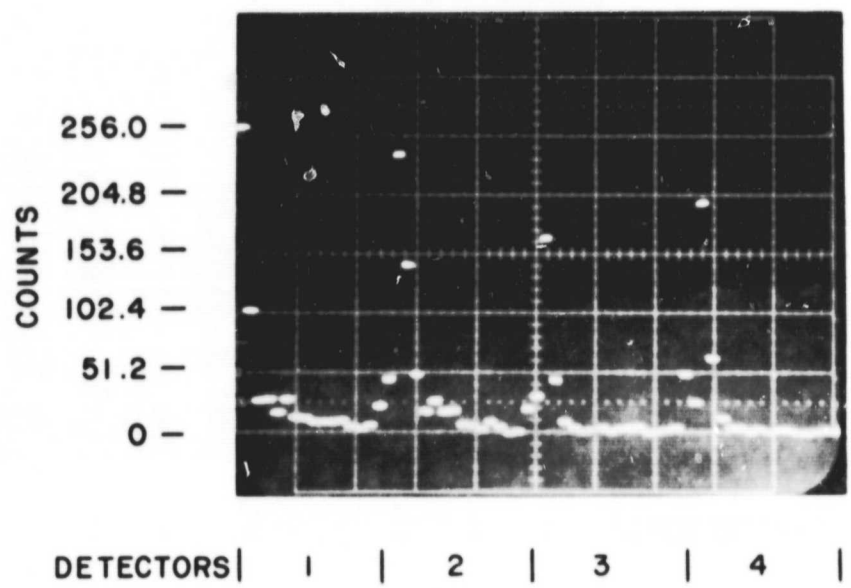


Figure 6.17 Detectors 3 and 4 are now beginning to show counts (99 km).

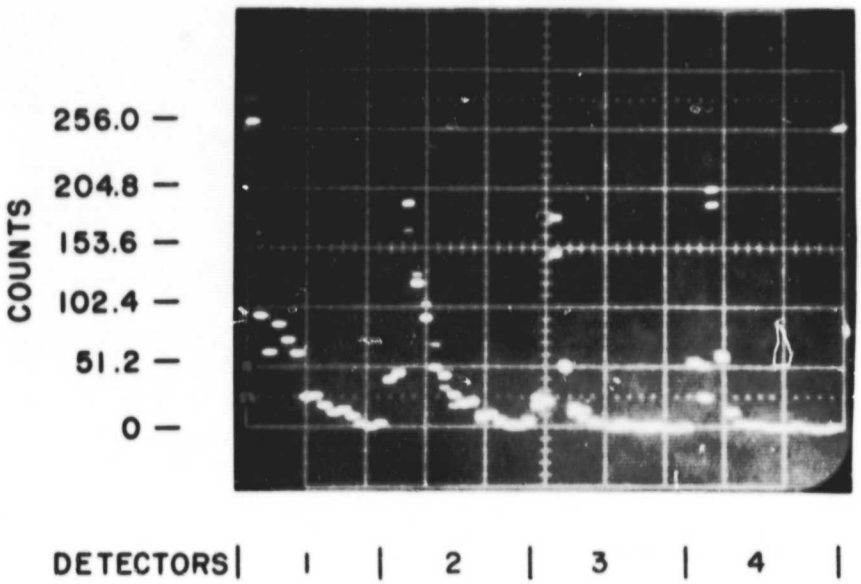


Figure 6.18 The count rate continues to build on all detectors (104 km).

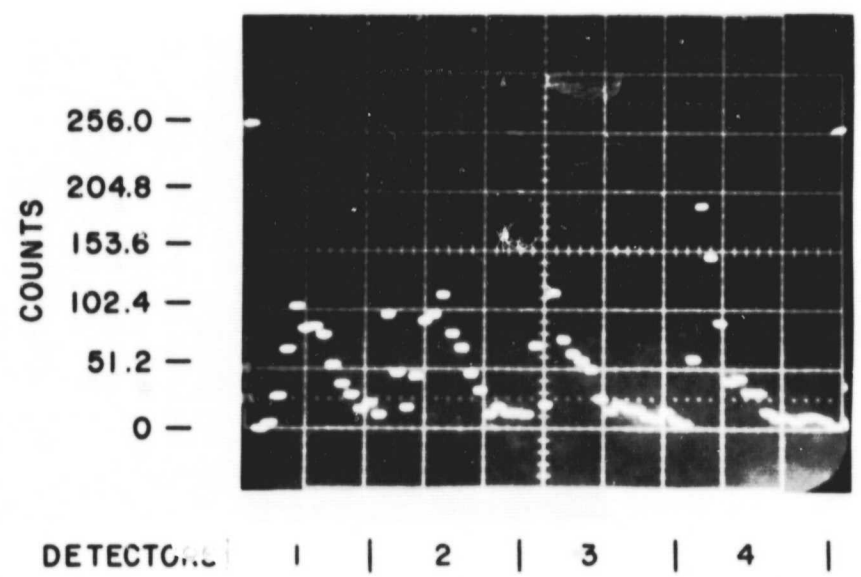


Figure 6.19 Sampling error is beginning to appear on detectors 1 and 2 (108 km).

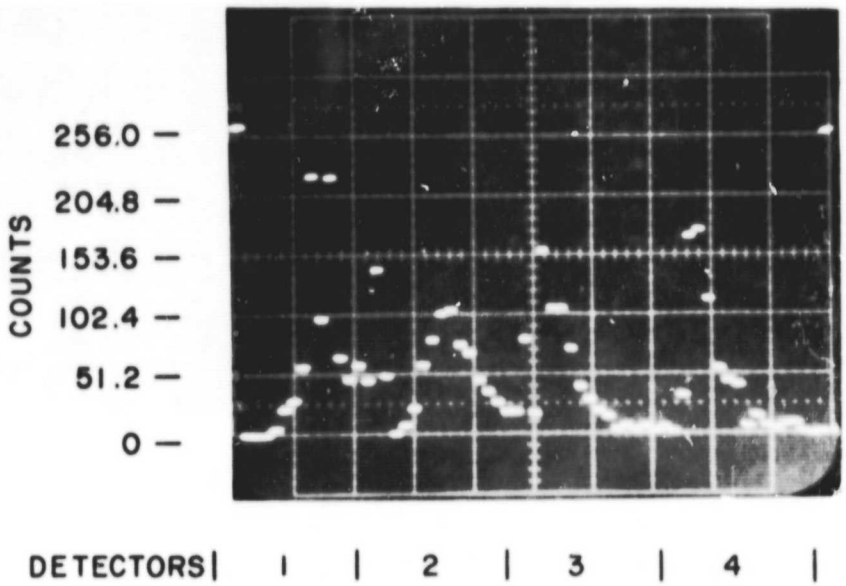


Figure 6.20 Sampling error becomes more dominant on detectors 1 and 2 (113 km).

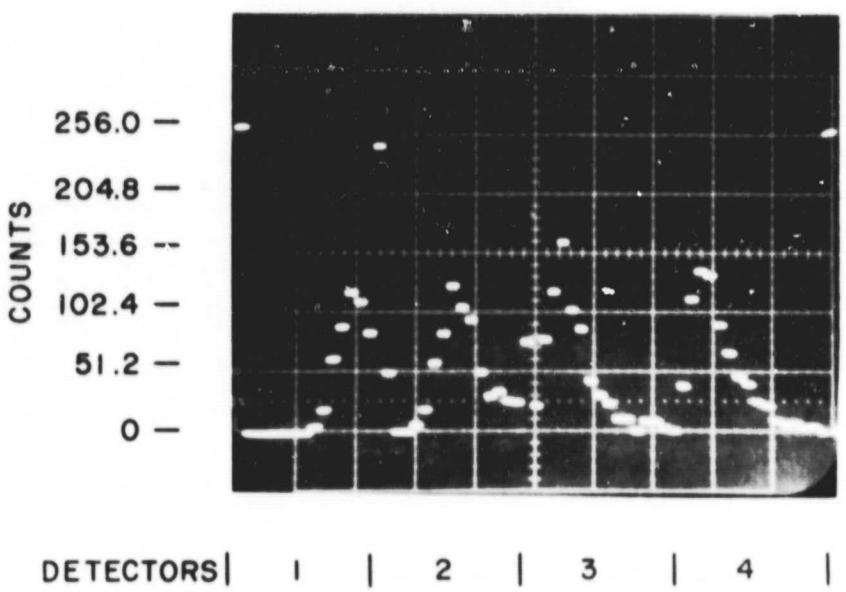


Figure 6.21 Detectors 3 and 4 are just beginning to show sampling error as the particle flux continues to increase (119 km).

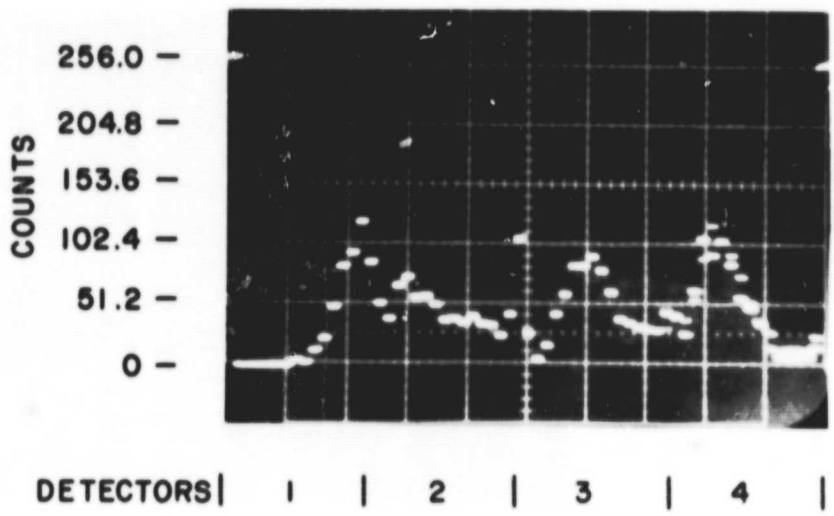


Figure 6.22 All detectors are now showing sampling error indicating the dominance of high energy particles upon the sampling (128 km).

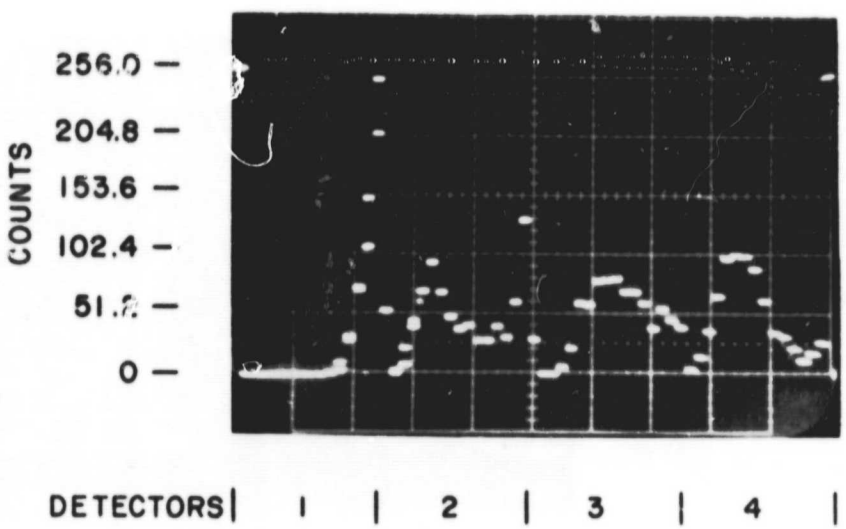


Figure 6.23 Sampling error approaches its limit as only the last few energy bins show any counts on detector 1 (137 km).

counts that can appear in a bin.

If the number of counts in a bin exceeds 255 the counts will recycle to 0 and continue from there. In Figure 6.24 (150 km) bin 15 of detector 1 is now showing fewer counts than bin 14 and judging from the preceding few pictures in the sequence it would be safe to say bin 15 has recycled. Notice that bin 15 of detector 2 is steadily increasing showing the dominance of very high energy particles.

Figure 6.25 shows data at rocket apogee (171 km). The particle flux has increased substantially in detectors 3 and 4 as shown by the increase of counts in every energy bin.

The sampling error and bin recycling problem make the post-processing difficult; however, the data are still recoverable for most of the energy range.

The data indicates very high particle fluxes. The post-processing will reveal the actual particle energy spectrum, together with the pitch-angle distribution

Finally it should be noted that, in addition to the particle data from the microprocessor-based experiment there are other particle data from these payloads (to be described in a separate report). Preliminary examination confirms the high count rate and also indicates a strong pitch-angle variation for angles near 90 degrees.

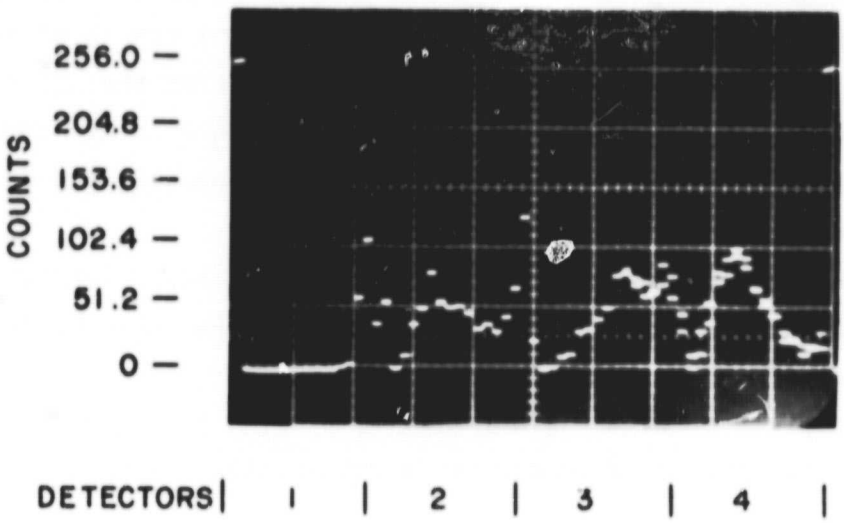


Figure 6.24 Recycling of energy bin 15 on detector 1 occurs since more than 255 counts are recorded in that bin (150 km).

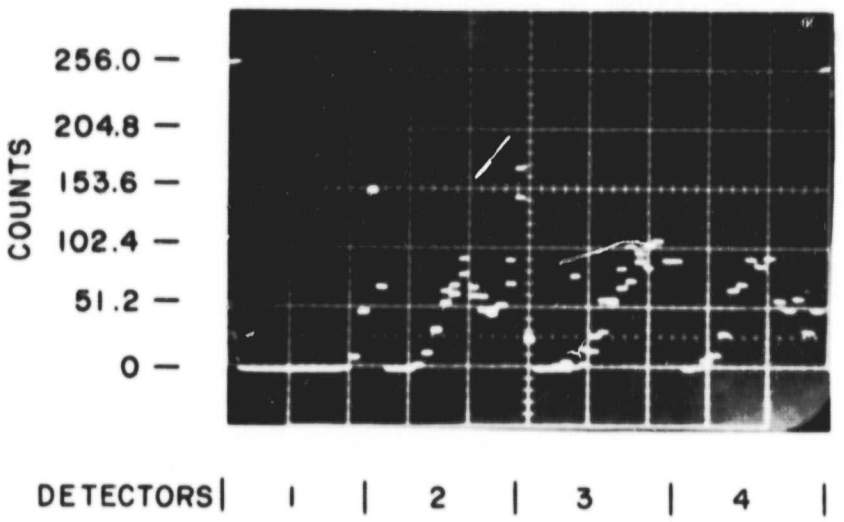


Figure 6.25 Particle flux is at a maximum since all the detectors show higher counts in every energy bin (171 km; apogee).

7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

The successful flights, in the Energy Budget Campaign, of Taurus Orions 33.009, 33.010, and 33.011 demonstrate the ability of the microprocessor system to function correctly under actual flight conditions.

The particle sampling problem (discussed in Chapter 6) needs further consideration in future versions of the microprocessor system. The solution is difficult because of the need to allow for a large range of count rates and by limitations on the data rate.

The speed of the present system can be increased from 2.5 MHz to 4.0 MHz by changing to the faster Z80 family and memories that can operate at that speed. This would allow for a slight increase in the data input rate. However, to handle count rates as great as those experienced during the Energy Budget Campaign the data input rate must be much greater. This implies an even more advanced system architecture than is used at present.

Now that single chip microprocessors are available parallel processing could be used to increase the data flow. The redundancy would also increase the system's reliability. Figure 7.1 shows a possible configuration. With the new high density memories the entire microprocessor system would take very little space: it would eliminate the SIO, CTC, PIO and many of the memory chips currently used.

The Intel 8751, for example, contains 4096 bytes of on-chip EPROM, four 8-bit parallel ports, a high speed serial port, two 16-bit counter/timers and other features which make it more powerful by itself than the current system composed of several printed circuit boards.

Another interesting possibility would be to use an I/O processor such as the Intel 8089 to coordinate the data collection leaving the CPU free to do the data processing. The complete experiment would then be as shown in Figure 7.2. The 8089 would coordinate the collection of data from the detectors and the magnetometer signal and then deliver it directly to the microprocessor memory for processing. The 8089 would also be programmed to do some error checking of the various peripheral devices to increase the system's reliability.

The use of a 16-bit processor such as the Intel 8086 or the Motorola 68000 would yield more efficient memory usage. The 16-bit processors also have a more advanced architecture and instruction set allowing easier, more

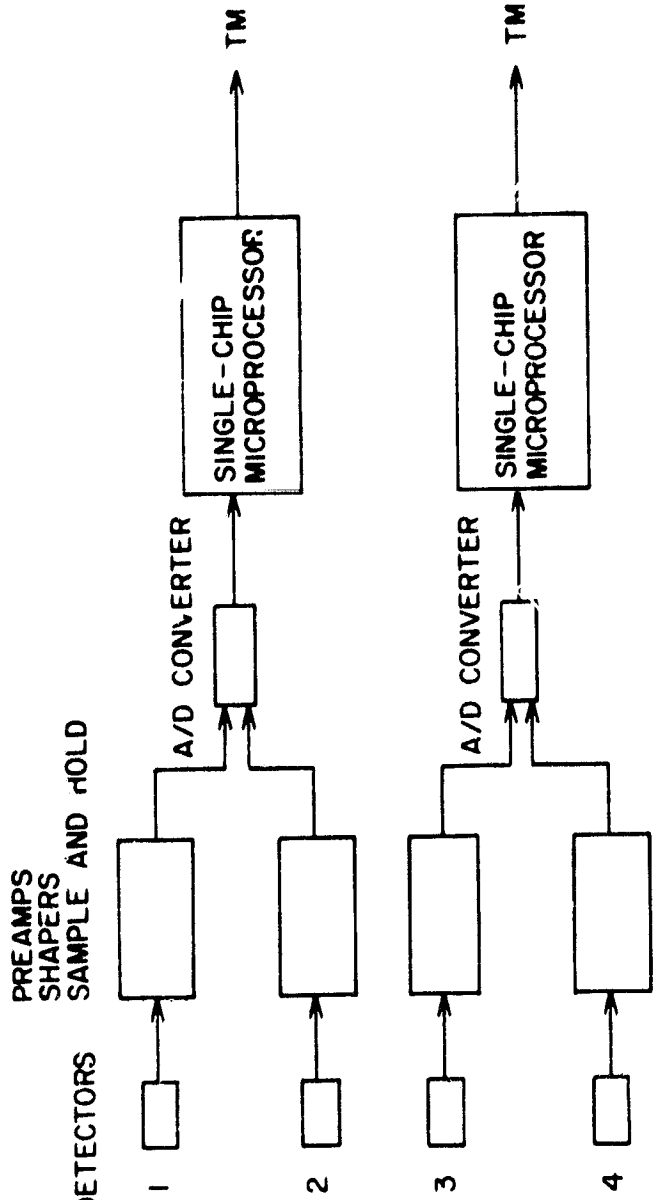


Figure 7.1 Parallel processing scheme using single chip microprocessors improves data flow and reliability through redundancy.

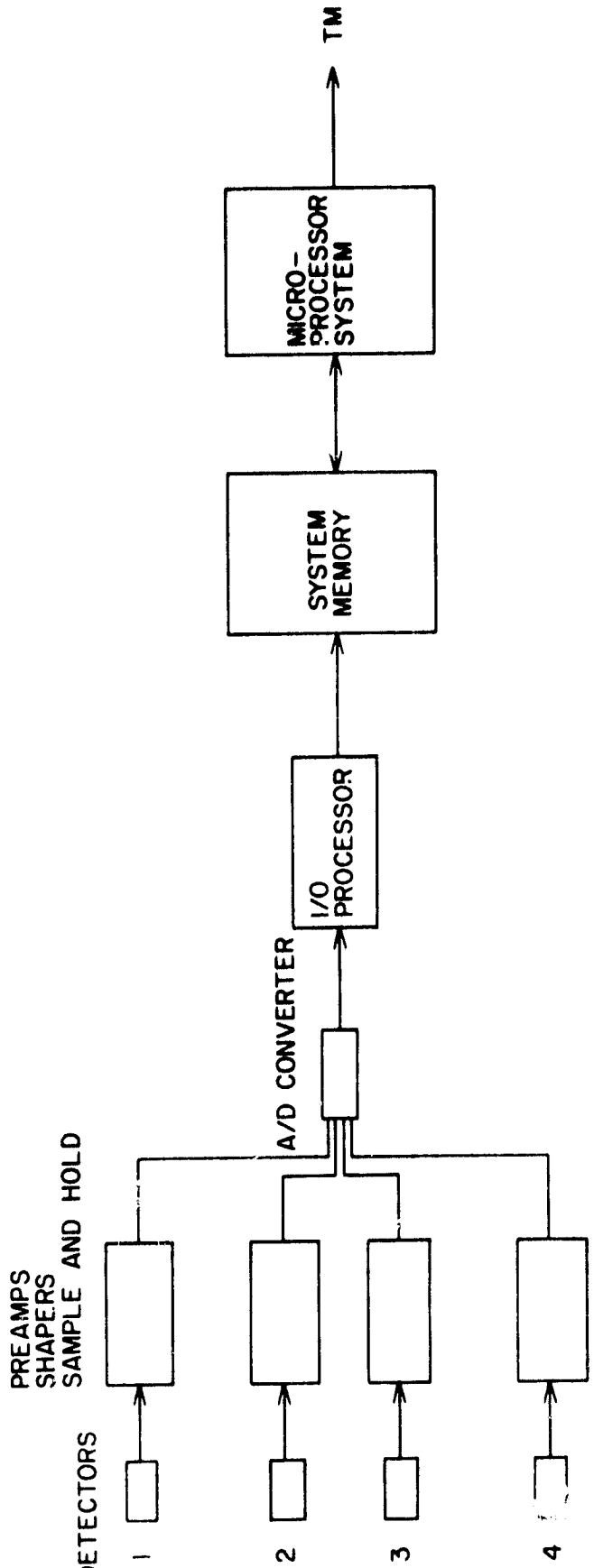


Figure 7.2 An I/O proces·. can be used to increase system efficiency and speed, in addition to error checking for greater reliability.

sophisticated implementation.

The present system for on-board processing of data in the energetic particle experiment has made effective use of existing microprocessor technology. The field is developing rapidly and, as new components become generally available, these will be incorporated in the experiment to improve its performance.

APPENDIX I. Thermal testing of the random access memory

I.1 *Introduction*

The purpose of this project is to test the 2114 RAM chips under extreme conditions of temperature and altitude.

The test circuit is designed to drive the RAM chips while continuously checking for errors at each memory location.

The RAM chips are to be used in an unusual configuration. Due to limited PC board space the chips will be stacked two high and it turns out that all the pin connections are identical except for the 8 pin (chip select) when they are mounted on the PC board. This is further reason for testing the chips. It must be established that these two-chip packages, without heat sinking, can safely dissipate enough heat to stay in their safe operating range.

I.2 *Test Circuit*

The test circuit (Figure I.1) is designed to continuously cycle through every memory location checking to see that none of the bits are stuck at one or stuck at zero. Should an error occur the RAM chips are immediately disconnected from the circuit and an LED is illuminated to indicate the error condition. The RAM chip isolation is accomplished by means of three octal tri-state buffers (74LS244) and a relay switch. The relay switch physically disconnects the RAM (pin 18) from the 5 V supply to prevent the chips from overheating and the buffers are put into their high impedance state. If necessary the circuit may be reset by toggleing the power supply off-on.

The temperature of each two-chip package is monitored by a thermistor imbedded in the sockets underneath the chips. When the chips are pressed into the socket the thermistor bead touches the bottom of the epoxy casing.

The 2.5 MHz clock defines eight states, as indicated in Figure I.2, for each memory location. These eight states thoroughly test each memory location for bits stuck at one or zero.

The eight states are:

I Increment address by one position and put zeros on the data lines

II Write zeros into the memory location specified

III Read the memory location

IV Check to see that there are all zeros on the data lines

V Put ones on the data lines

VI Write ones into the memory location specified

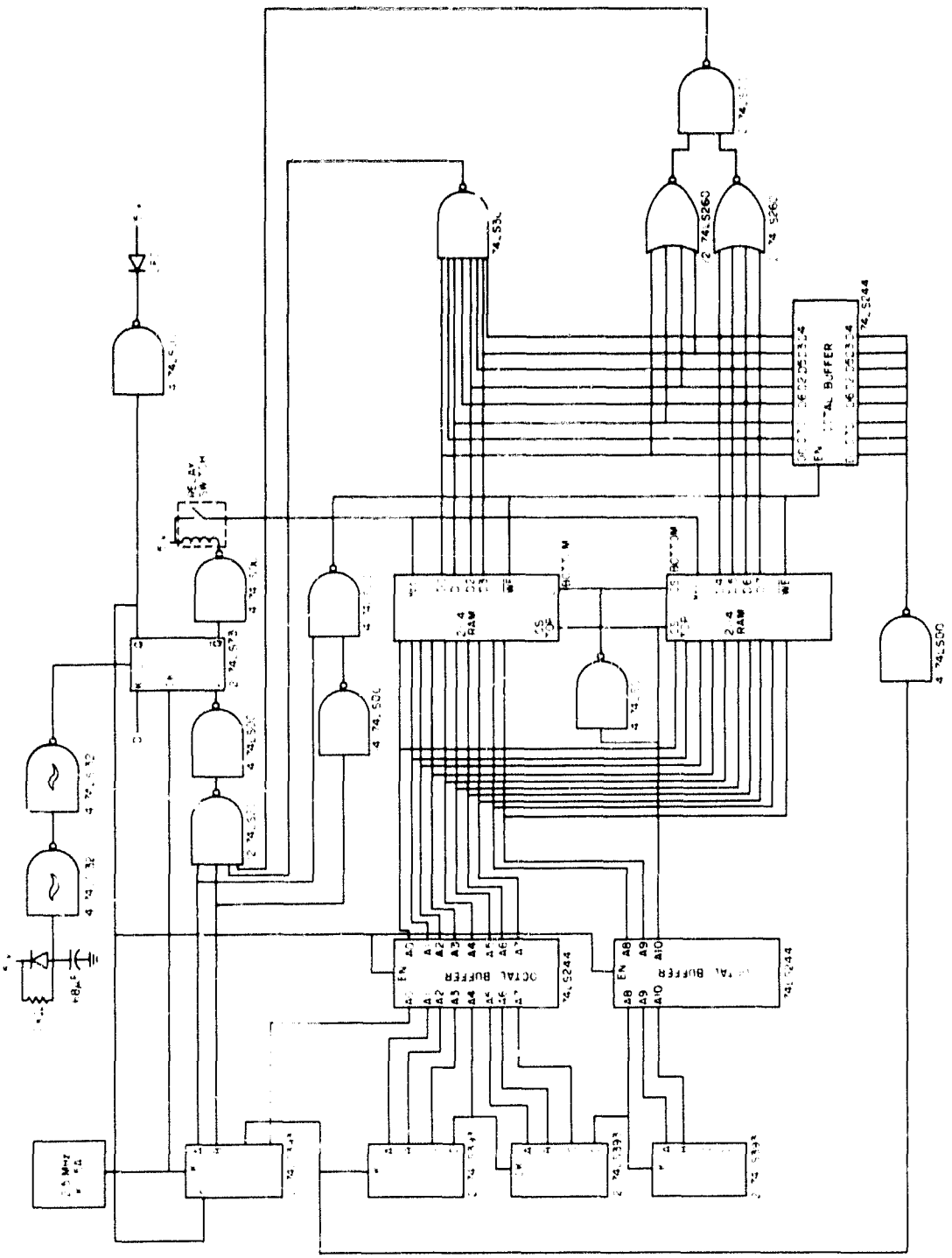


Figure I.1 Schematic of RAM tester.

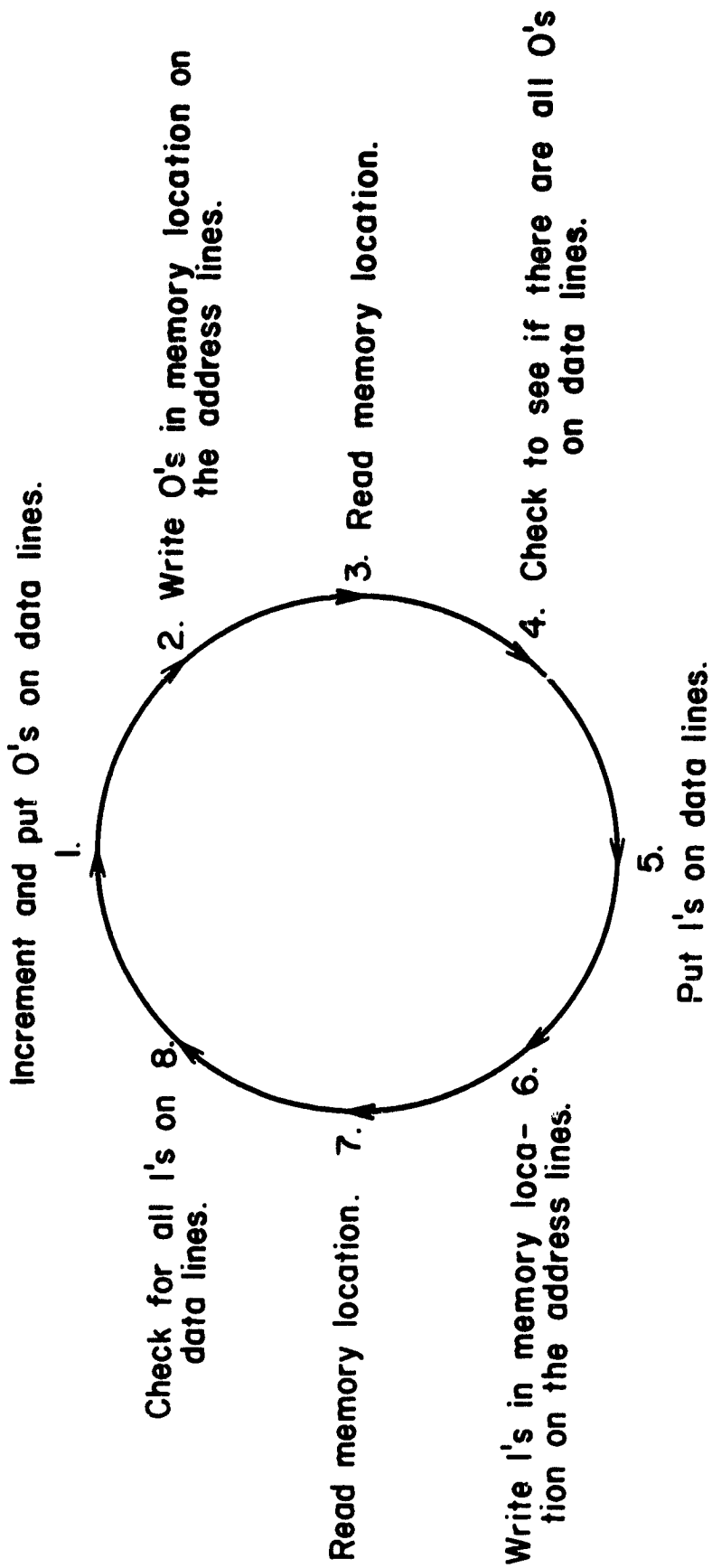


Figure I.2 State diagram of the RAM tester.

VII Read the memory location

VIII Check to see that there are all ones on the data lines

The timing diagram (Figure I.3) shows the relationship between the clock pulses and the eight states. Figure I.4 shows the actual signals. In order to achieve eight states, three flip flops are needed for this synchronous sequential machine. They are conveniently provided in the 74LS393 dual 4-bit binary counter chips. As you look on the circuit diagram (Figure I.1) you will see the clock pulse is fed into the clock input of one of the 4-bit counters. The A, B and C outputs then go into some combinational logic for controlling the circuit. Signals A, B and C are also shown on the timing diagram (Figure I.3). The $\overline{WE}$ signal tells the RAM chips when to write zeros or ones into the specified memory location (see Figure I.3) and is active low. $\overline{WE}$ is derived from the signals A and B using two NAND gates (see Figure I.1). The $\overline{WE}$ signal also controls the data line buffer chip (74LS244) on lower right and tells it when to place data on the data lines. Signals A and B along with the output from the 74LS260 and 74LS30 gates check the data read from a specific memory location. The C line places either zeros or ones on the data lines and also increments the address by one on its falling edge.

Zeros or ones are detected using the 74LS260 and 74LS30 gates respectively. If all the data lines are ones the 74LS30 output will be zero and if all are zeros the 74LS260 outputs will be ones which are fed into a NAND gate whose output will be zero. This means that if all zeros or all ones are on the data lines then the outputs of the 74LS30 and 74LS00 NAND gates will be either one and zero or zero and one. Also notice that if one of the data lines is stuck high or stuck low that both outputs will be ones at the same time. These signals coupled with signals A and B are fed into a 74LS20 NAND gate, and as indicated above, when AB is one, a check is performed. It is now obvious that the only time the gate will give a zero output is when there is an error and all four signals are ones. The output signal is then inverted so now a zero output means no error and a one signals an error condition.

This signal goes to the J input of a clocked JK flip flop (74LS73) whose K input is held at zero. The flip flop is initialized into the zero state ($Q, \overline{Q} = 0,1$) by its clear line (the initialization is explained later).

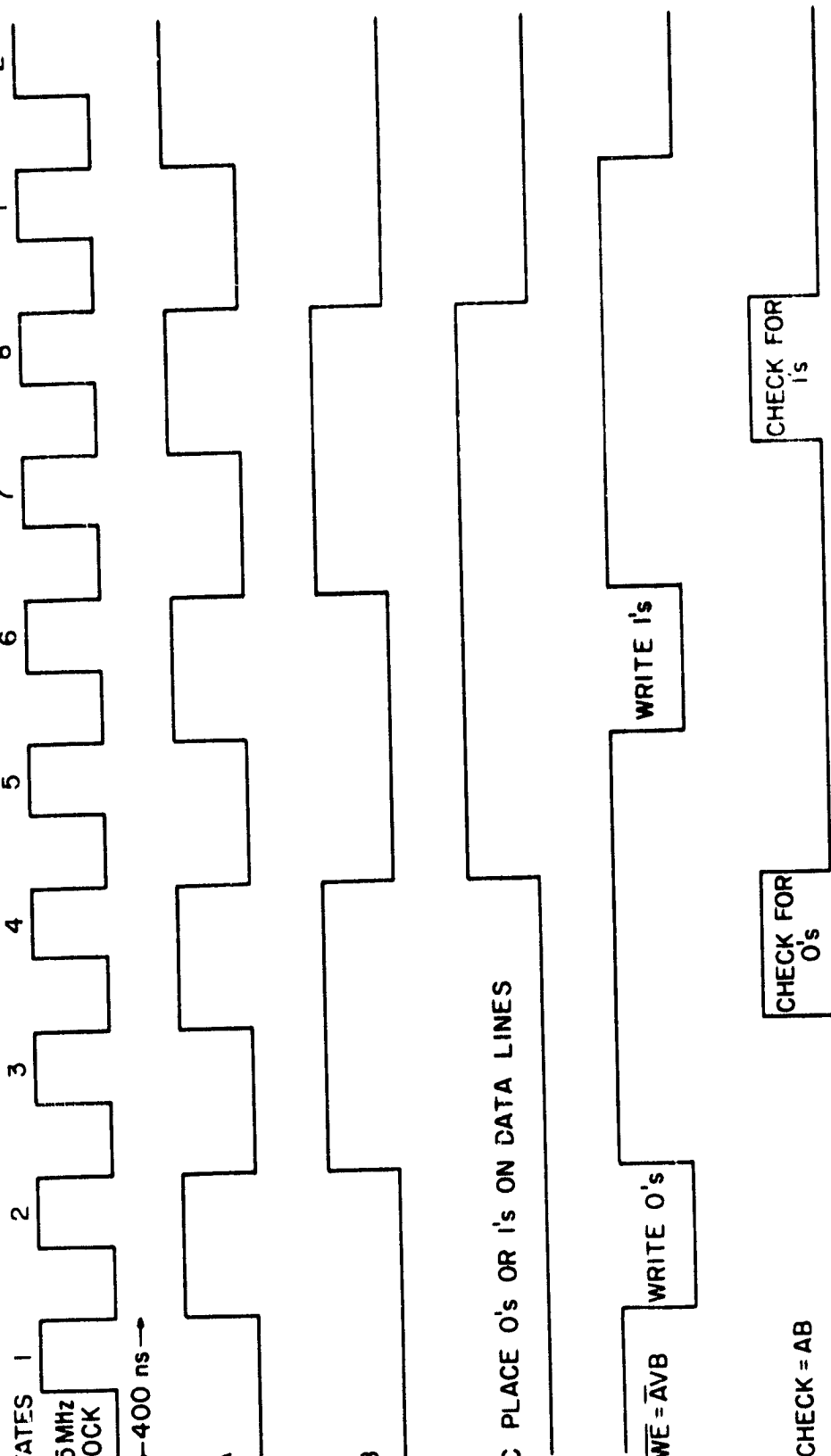


Figure I.3 Timing diagram.

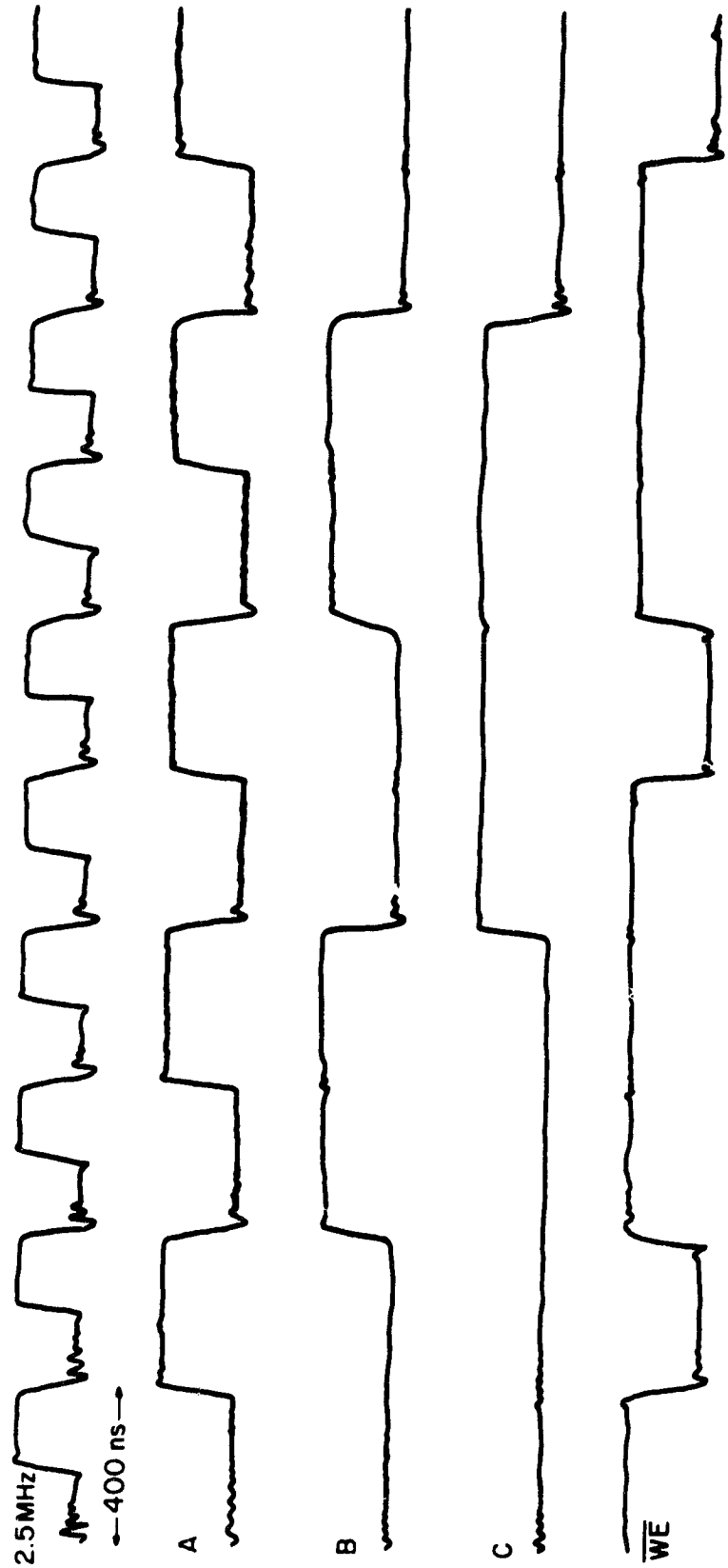


Figure I.4 Actual signals observed.

Once initialized and running J and K are 0,0 as long as no errors are encountered and the flip flop will stay in its Q, $\overline{Q}$ = 0,1 state. Should an error occur J will become one and Q, $\overline{Q}$ will change to 1,0. Once the change has occurred J can be either zero or one in the error state and not affect the state of the flip flop.

When an error occurs Q changes from zero to one. This signal goes to the address buffers (74LS244's) and puts them into a high impedance state. The signal also drives a NAND gate which turns on the LED to visually signal the error condition. The NAND gate is used as a buffer between the output of the flip flop and the LED since it was unable to drive the LED plus four other gates by itself.

Conversely $\overline{Q}$ changes from one to zero. This signal is fed into a NAND gate which drives a coil activating a relay switch. The other end of the coil is held at 5V and works as follows:

Signal $\overline{Q}$	Output of NAND	Relay position	Condition
1	0	closed	no error
0	1	open	error

The relay physically disconnects the RAM chips from the 5V power supply to prevent the chips from overheating.

The Schmitt triggered NAND gates (74LS132) were used to set the JK flip flop into its zero state when the power is applied to the circuit. The first NAND gate is triggered by the resistor, capacitor, and diode circuit as follows.

With power suddenly applied to the circuit (assuming the capacitor is fully discharged) the diode will be back biased and the capacitor will charge with a time constant of 680 ms. After about 0.26 s the voltage across the capacitor will reach 1.6 V which is the triggering level of the NAND gate so there has been a 0.26 s delay between power on and the setting of the flip flop. This delay allows the rest of the circuit to be in operation long before the flip flop is set and makes certain the flip flop is set for normal operation 0.26 s after power is applied.

The reason for this elaborate initialization scheme is because the vacuum chamber only allows four lines to be brought out. The four wires

brought out were the P (purple) and G (gray) thermistor leads, the 5V supply and ground, to which the common thermistor lead was tied. Had the flip flop clear line been brought out there would have been no way to monitor the temperature of both RAM chip packages simultaneously.

During operation the ground and common lines of the power supply and of two V-Ω-meters were tied together to the ground line of the circuit. The 5 V line of the power supply was applied to the 5 V line of the circuit and the P and G thermistor leads were connected to each one of the two V-Ω-meters.

The circuit was tested by either grounding or holding at 5 V any one of the eight data lines which immediately illuminated the LED as expected. Each signal was monitored and is shown in Figure I.4. The check signal is not included since it is not actually present as the signal AB, but is ANDed together with two other signals in the 74LS20.

I.3 *Experimental Results*

The first step in the experimental part of this project was to calibrate the thermistors. This was done by placing the circuit (with no power applied) in a thermal chamber and watching the change in resistance of each thermistor with respect to temperature. The results in Figure I.5 and I.6 show that the two thermistors are not identical and differ greatly at low temperatures.

The first data run was made in the lab at room temperature (23°C). The circuit was placed on the lab bench in open surroundings and the power was applied. Figures I.7 and I.8 are the results showing temperature (resistance) plotted against time. The RAM chips reach their peak operating temperature of about 39°C in about 15 minutes and appeared to stay constant for the next 35 minutes. The apparent downward trend in temperature toward the end of the experiment was perhaps due to the circulation of air in the room from the cooling system of the building, and it is probably not related to any physical characteristics of the device.

Figures I.9 and I.10 show that the RAM chips take approximately 20 minutes to return to room temperature, which is about 5 minutes more than it takes them to heat up.

Figure I.11 through I.14 were the results of data taken in the thermal chamber. This special chamber allows the temperature to rise and fall slightly above and below the desired settings. Also it has a small fan to circulate the air.

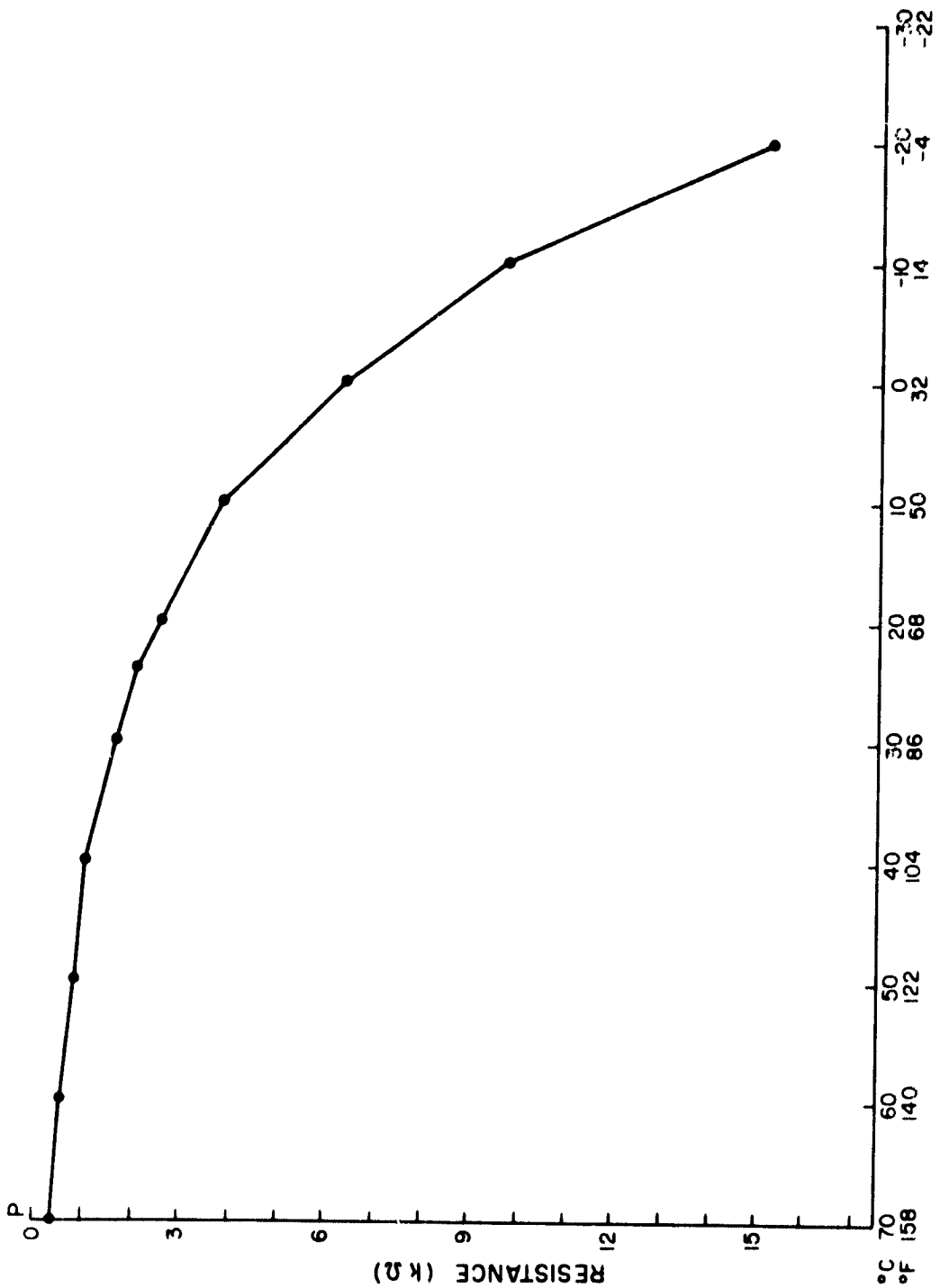


Figure I.5 R-T calibration curve for the purple-colored thermistor.

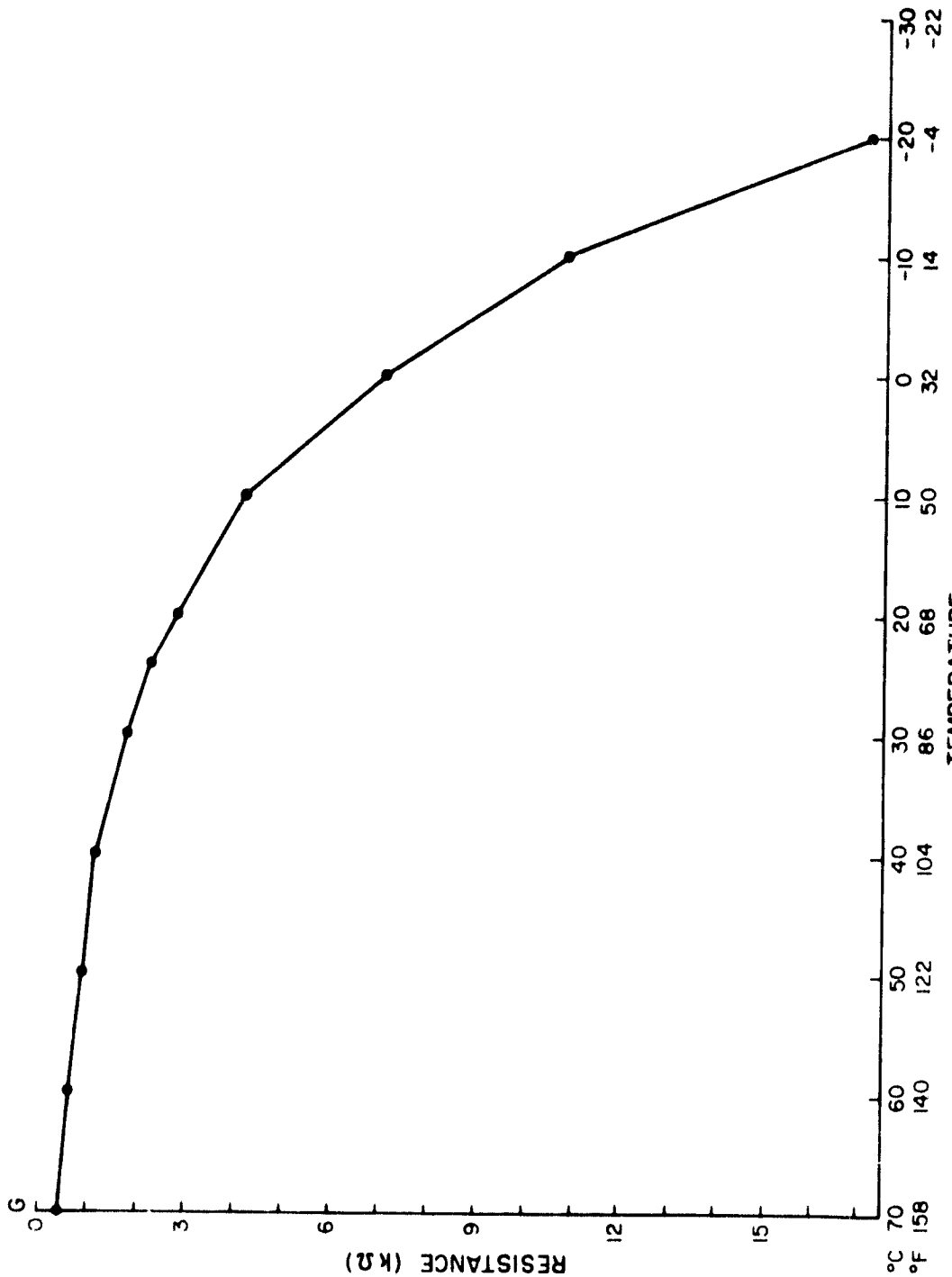


Figure I.6 R-T calibration curve for the gray-colored thermistor.

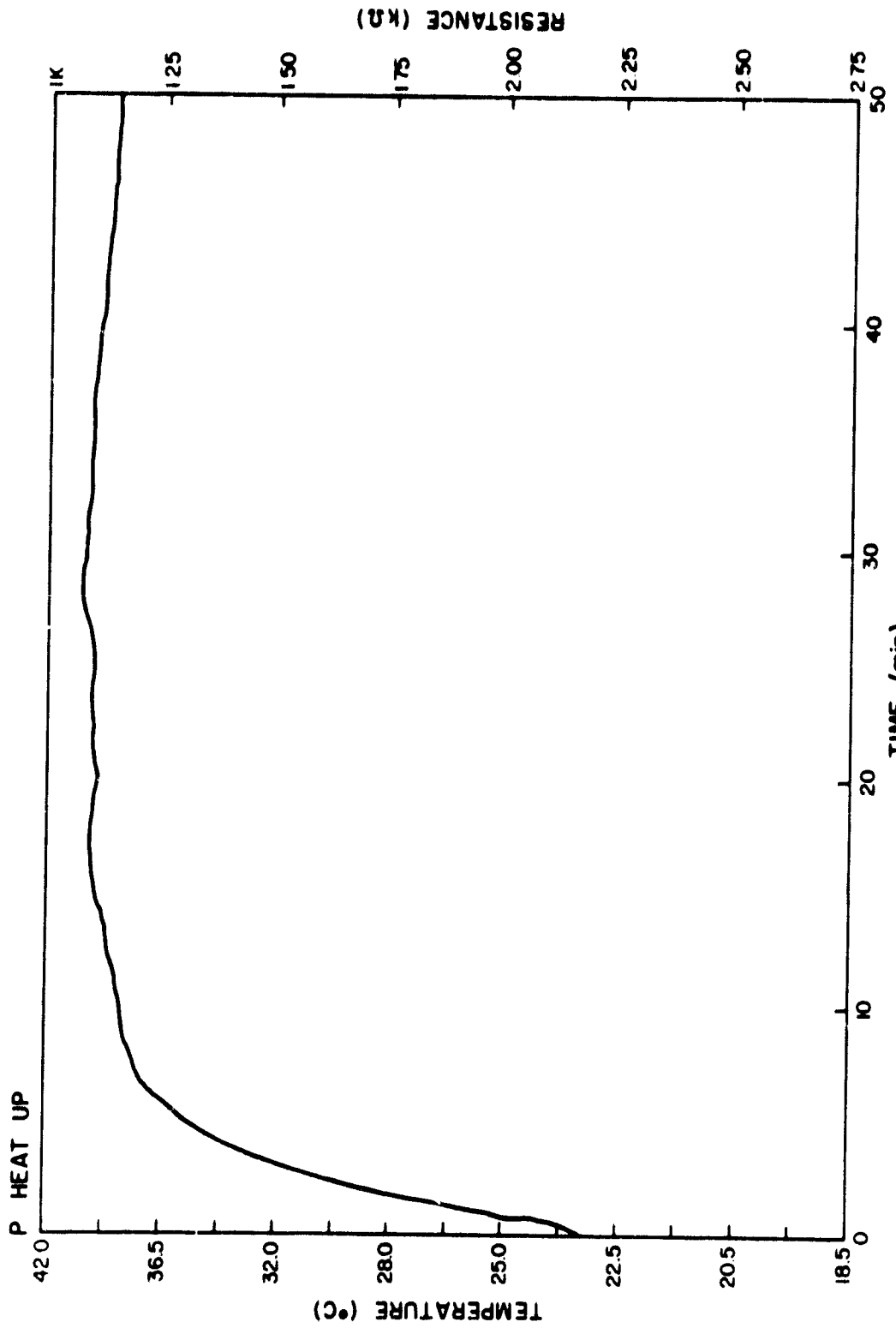


Figure I.7 Purple thermistor heating up in open surroundings.

Figure I.8 Gray thermistor heating up in open surroundings.

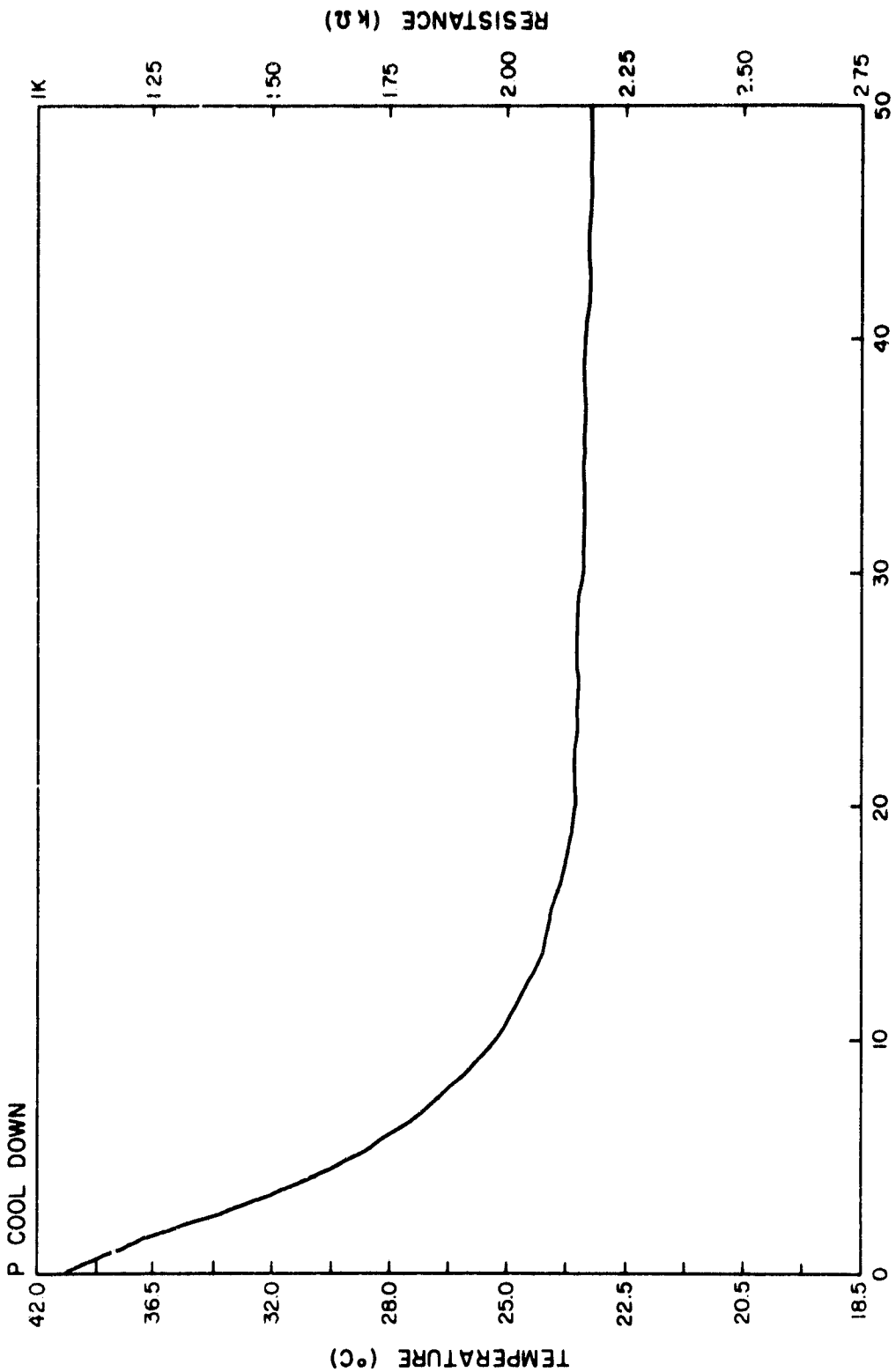


Figure I.9 Purple thermistor cooling off in open surroundings.

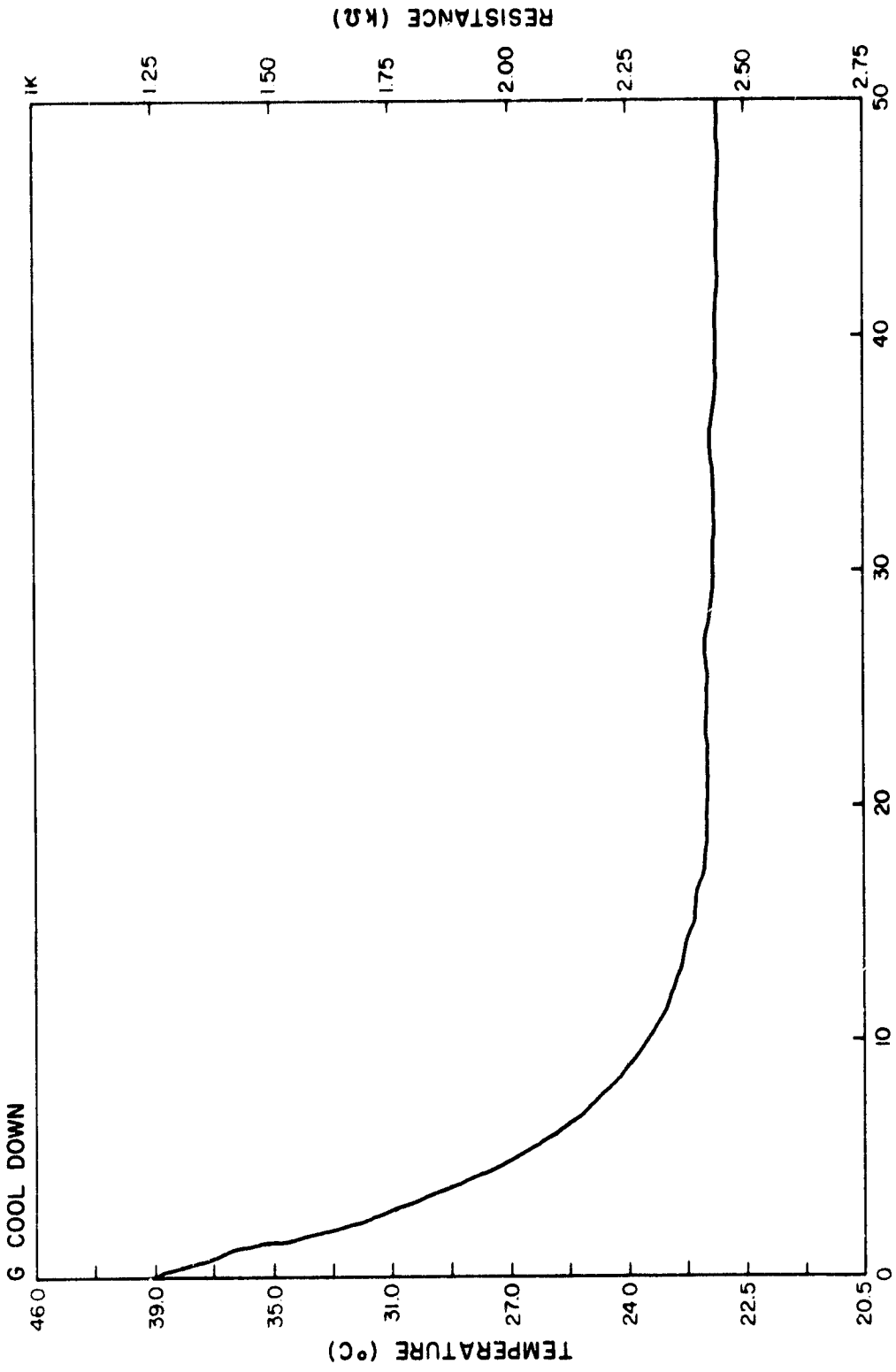


Figure I.10 Gray thermistor cooling cff in open surroundings.

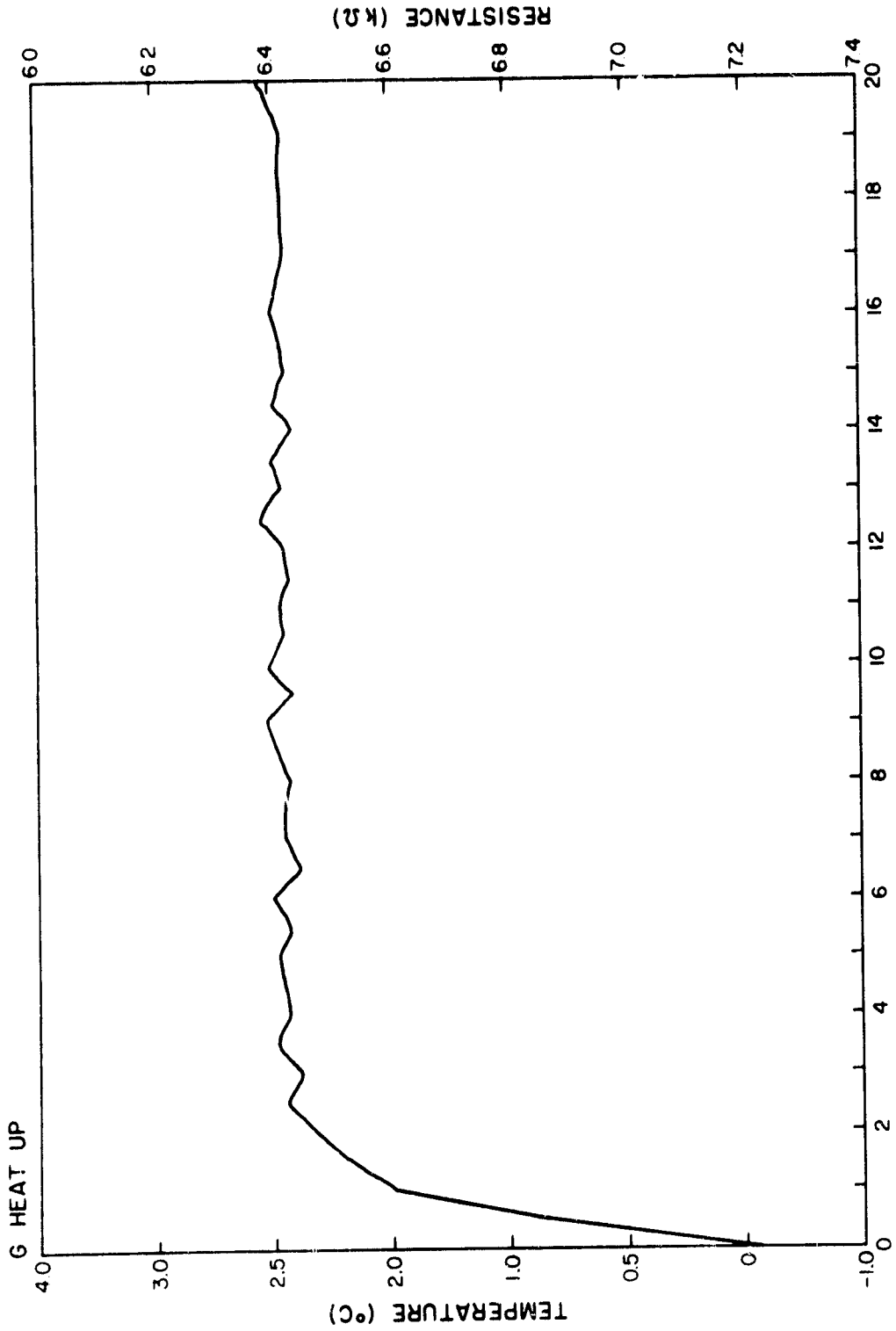


Figure I.11 Gray thermistor in a 0°C thermal chamber.

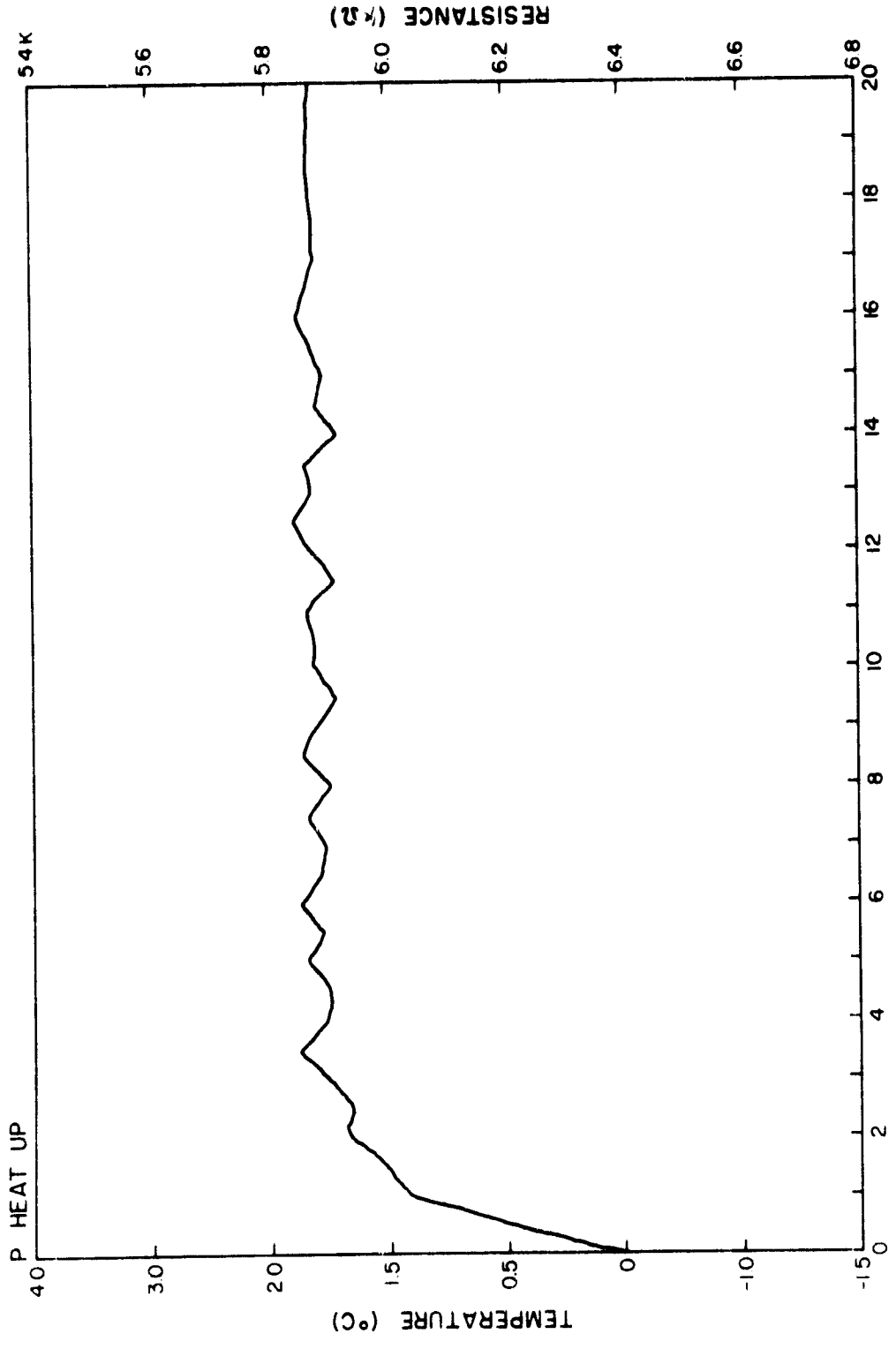


Figure I.12 Purple thermistor in a 0°C thermal chamber.

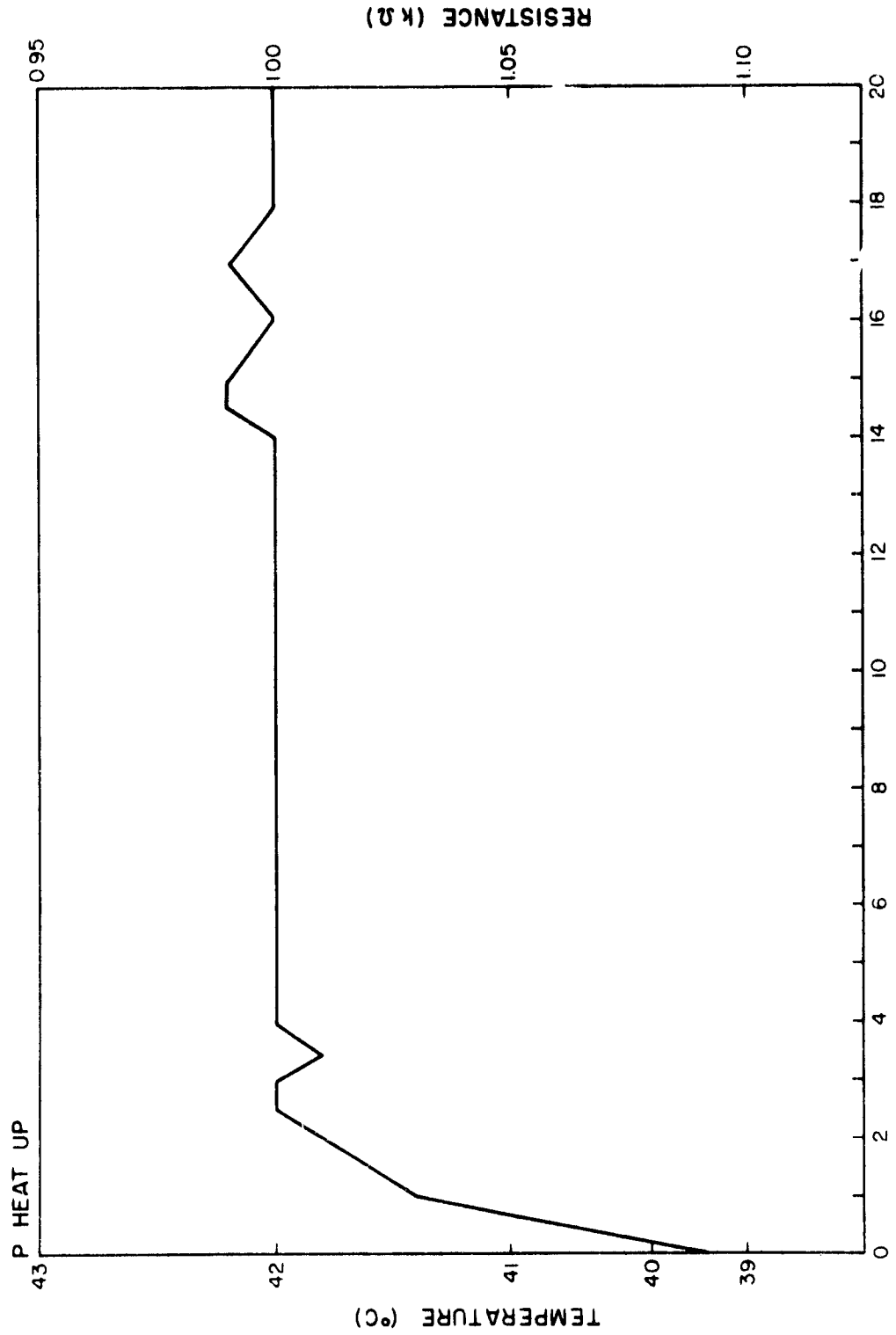


Figure I.13 Purple thermistor in a 39°C thermal chamber.

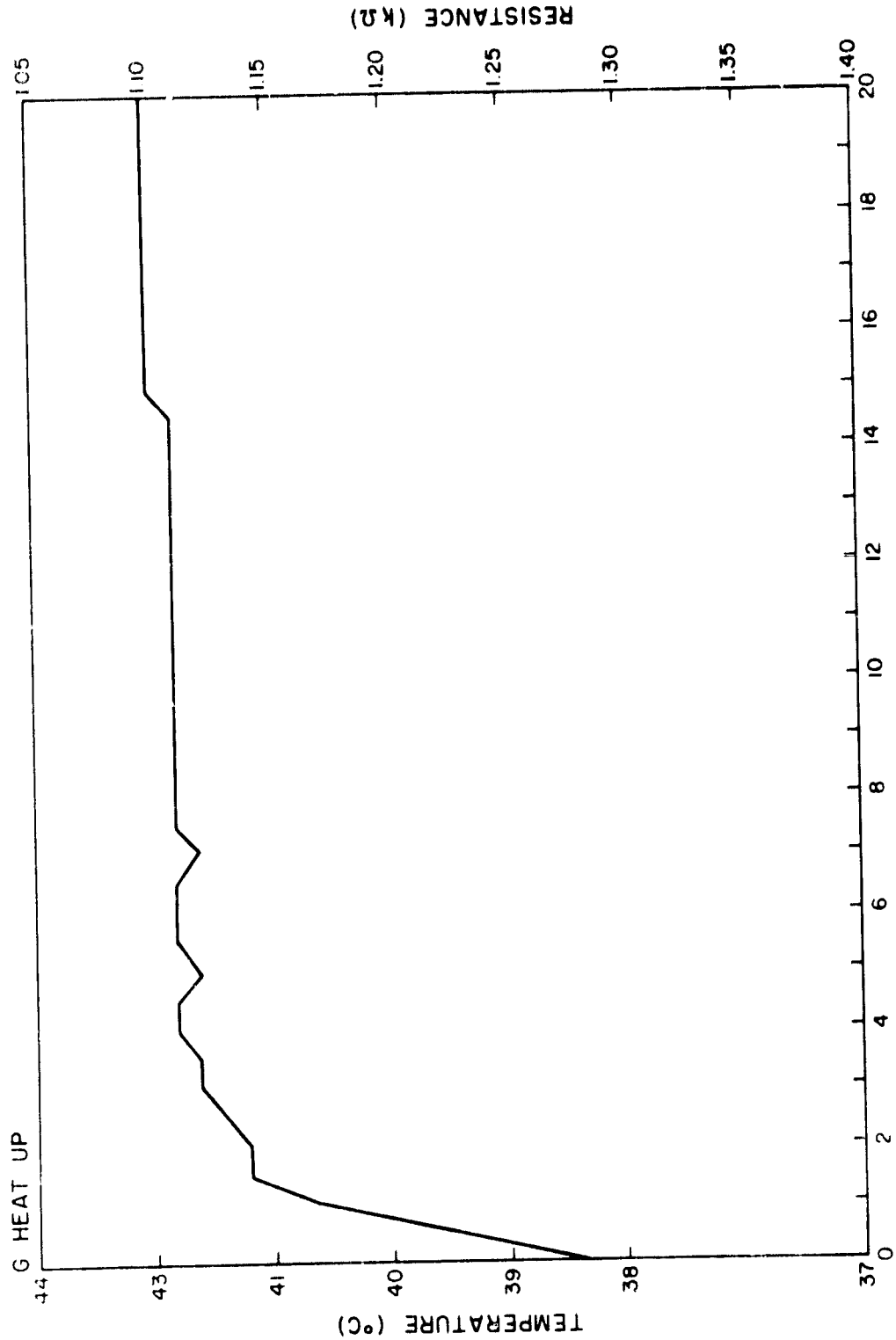


Figure I.14 Gray thermistor in a 39°C thermal chamber.

The jagged graphs shown in Figures I.11 and I.12 would appear to be due to properties of the thermal chamber and not the RAM chips. It would also appear that the chamber is acting as a heat sink for the RAM chips since there is such a small increase in their temperature. Any heat given off by the chips is quickly carried off by the atmosphere of the chamber which is held at 0°C.

The high temperature runs shown in Figures I.13 and I.14 show the same behavior, in fact they only show a 2.5°C rise in temperature over the temperature of the chamber, which is exactly the same rise shown in the low temperature runs (Figures I.11 and I.12).

There are a few problems with Figures I.11 and I.14 however, which need to be mentioned before any firm conclusions can be drawn. The first problem is that the R-T curves (Figures I.5 and I.6) are very difficult to read if only a three-degree temperature difference is to be observed. This introduces human error in the temperature readings of Figures I.11 through I.14. The second problem is noise in the system itself. In other words, how much of that small change in resistance is due to noise in the electronics of the ohm-meter or the inherent noise of the thermistors. Observe that in Figures I.13 and I.14 there is only a 100Ω total change in the thermistor readings.

The curves of Figures I.15 and I.16 are of greatest interest. The circuit was placed in a bell jar which was evacuated, equivalent to an altitude of 50 to 60 km. The reason for the altitude variation is because the heat given off by the circuit forced the altitude down several kilometers, which is to be expected. When power was applied the curves of Figures I.15 and I.16 results. Observe that the temperature rise is 35°C but that the temperature definitely becomes stable after about 25 minutes at high altitudes. The operating range of the 2114 chips in the epoxy package is from 0° to 70°C so there is still about 10°C leeway at high altitude before their range is exceeded.

One final problem needs to be mentioned. Figure I.4 shows the actual signals observed; these are quite noisy. Noise generated by the circuit affected the readings of the thermistors since the ground of the circuit was connected to the common lead of the two thermistors. When power was applied to the circuit the thermistor readings typically jumped several k ohms. This problem was solved by switching the power supply on and off (after the initial

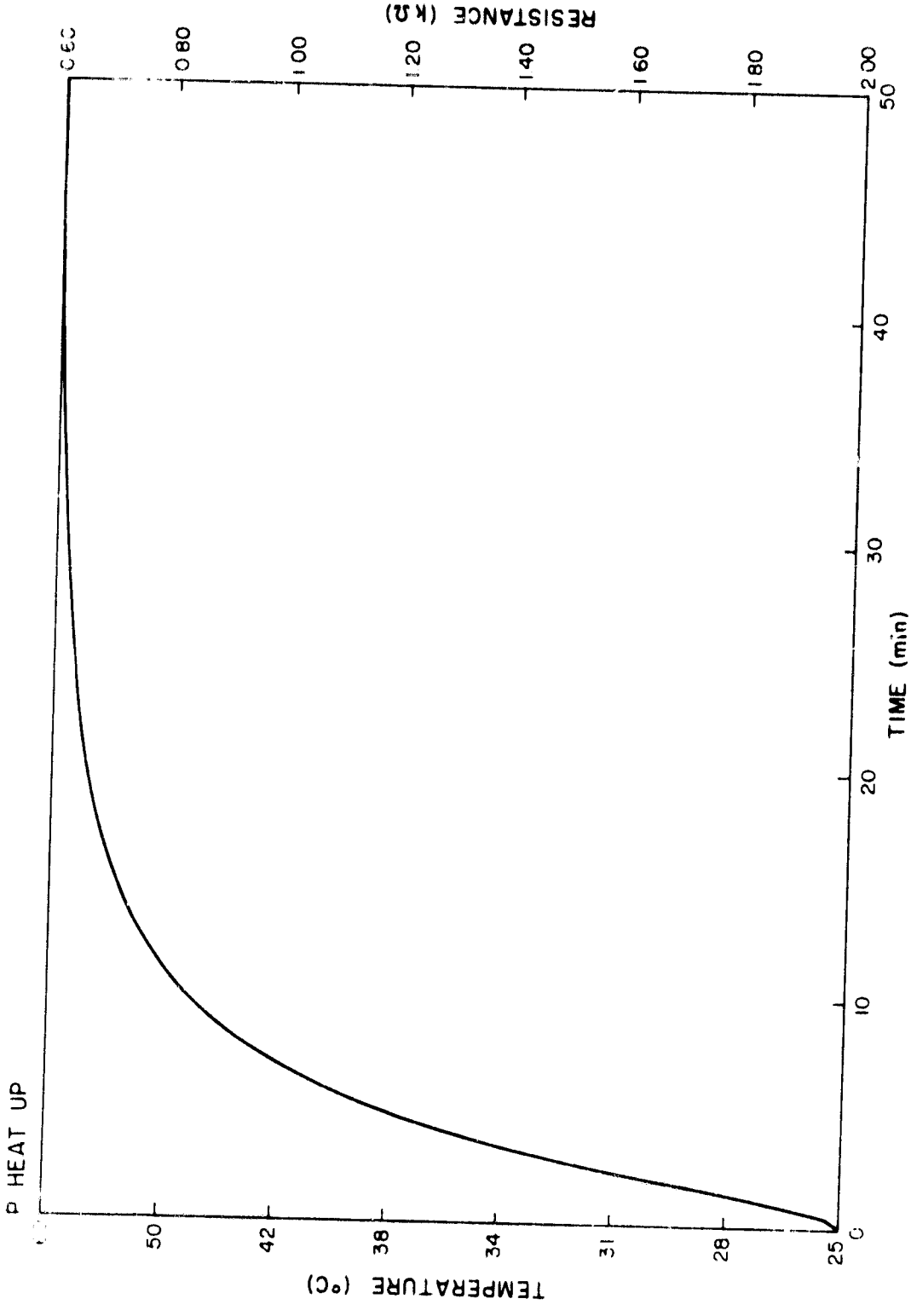


Figure I.15 Purple thermistor in a bell jar evacuated to about 52 km.

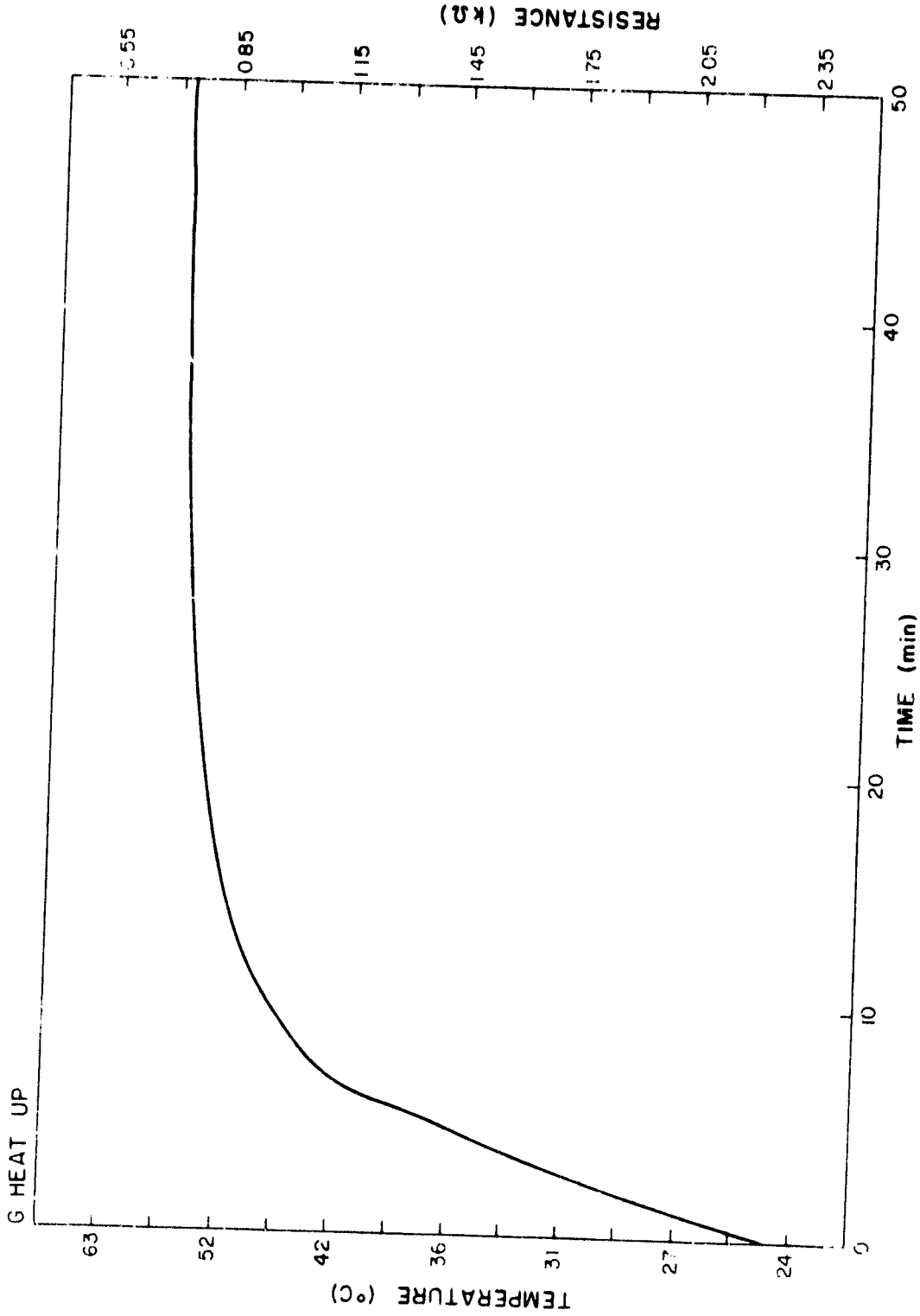


Figure I.16 Gray thermistor in a bell jar evacuated to about 52 km.

data run) at different resistances which gave true readings at each resistance taken as data.

I.4 *Conclusion*

The experiment showed very clearly that the 2114 RAM chips in the epoxy package will function under extremes of temperature and altitude and will not overheat or malfunction even if they are stacked on top of each other in pairs. The test circuit never detected an error and at no time did the RAM chips exhibit unusual or unpredictable temperature behavior.

APPENDIX II. Program listings

II.1 *Flight Program*

```
    FLIGHT PROGRAM
ADDR  OBJECT     ST #

                 0002          PSECT   ABS
                 0003 ;SIO DATA AND CONTROL CHANNELS
>000C            0004 SIOCAD: EQU     0CH      ;CHANNEL A DATA
>000D            0005 SIOCAC: EQU     0DH      ;CHANNEL A CONTROL
>000E            0006 SIOCBD: EQU     0EH      ;CHANNEL B DATA
>000F            0007 SIOCBC: EQU     0FH      ;CHANNEL B CONTROL
                 0008 ;
                 0009 ;PIO DATA AND CONTROL CHANNELS
>0018            0010 PIOPAD: EQU     18H      ;PORT A DATA
>0019            0011 PIOPAC: EQU     19H      ;PORT A CONTROL
>001A            0012 PIOPBD: EQU     1AH      ;PORT B DATA
>001B            0013 PIOPBC: EQU     1BH      ;PORT B CONTROL
                 0014 ;
                 0015 ;CTC CHANNELS
>0014            0016 CTC0:   EQU     14H      ;CHANNEL 0
>0015            0017 CTC1:   EQU     15H      ;CHANNEL 1
>0016            0018 CTC2:   EQU     16H      ;CHANNEL 2
>0017            0019 CTC3:   EQU     17H      ;CHANNEL 3
                 0020 ;
                 0021 ;THE FOLLOWING CONTROL VECTORS WILL BE
                 0022 ;REFERRED TO THROUGHOUT THE PROGRAM AND
                 0023 ;ARE EXPLAINED IN DETAIL HERE ONLY
                 0024 ;
                 0025 ;PIO CONTROL VECTORS
                 0026 ;       SET UP FOR INPUT MODE
>004F            0027 PIOV1:  EQU     01001111B
                 0028 ;       INTERRUPT VECTOR
>00F6            0029 PIOV2:  EQU     11110110B
                 0030 ;       ENABLE INTERRUPTS
>0087            0031 PIOV3:  EQU     10000111B
                 0032 ;
                 0033 ;SIO CONTROL VECTORS
                 0034 ;       WR0,WR1,...,WR7 - WRITE REG 0,..
                 0035 ;       ..,WRITE REG 7
                 0036 ;       WR0 POINTS TO WR2
>0002            0037 SIOV1:  EQU     00000010B
                 0038 ;       WR2     INTERRUPT VECTOR
>00F4            0039 SIOV2:  EQU     11110100B
                 0040 ;       WR0 POINTS TO WR4
>0004            0041 SIOV3:  EQU     00000100B
                 0042 ;       WR4 - NO PARITY, 2 STOP BITS,
                 0043 ;       TX CLOCK TIMES 16
>004C            0044 SIOV4:  EQU     01001100B
                 0045 ;       WR0 POINTS TO WR5
>0005            0046 SIOV5:  EQU     00000101B
                 0047 ;       WR5 - 8 BITS PER CHAR,
                 0048 ;       ENABLE TX
>0068            0049 SIOV6:  EQU     01101000B
                 0050 ;       WR0 POINTS TO WR1
                 0051 ;       CMDS RESET TX AS IF LAST CHAR
                 0052 ;       OF BLOCK SO NO INTERRUPTS
                 0053 ;       OCCUR UNTIL NEXT CHAR LOADED
                 0054 ;       INTO TX BUFFER
>0001            0055 SIOV7:  EQU     00000001B
                 0056 ;       WR1 - ENABLE INTERRUPTS
>0082            0057 SIOV8:  EQU     10000010B
                 0058 ;       CHAN RESET VECTOR
>0018            0059 SIOVR:  EQU     00011000B
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

                0060 ;        STOP INTS UNTIL NEXT CHAR
>0028           0061 SIOHLT:  EQU     00101000B
                0062 ;
                0063 ;CTC CONTROL VECTORS
                0064 ;        INTERRUPT VECTOR
>00F8           0065 CTCV1:   EQU     11111000B
                0066 ;        CHANNEL CONTROL, INTERRUPT ON
                0067 ;        ZERO, USE COUNTER MODE,
                0068 ;        NEXT WORD TIME CONSTANT
>00C7           0069 CTCENA:  EQU     11000111B
                0070 ;        256 COUNTS
>00FF           0071 CTC256:  EQU     11111111B
                0072 ;        121 COUNTS = 5 RPS
>0079           0073 CTC121:  EQU     121
                0074 ;
                0075 ;        UPPER 8 BITS OF INTERRUPT TABLE
>0027           0076 IREG:    EQU     27H
                0077 ;        STACK ADDRESS
>27F3           0078 STAK:    EQU     027F3H
                0079 ;        RESET ERROR TRAP ADDRESS
                0080 ;        A3 LINE MUST GO HIGH
                0081 ;        NOTE ALL PERIFERALS ARE
                0082 ;        DISABLED SO THERE IS NO
                0083 ;        INTERFERANCE
>FFFF           0084 RESET:   EQU     0FFFFH
                0085 ;        INTERRUPT TABLE ADDRESS
>27F4           0086 INTBL:   EQU     027F4H
                0087 ;        ADDR OF REGION 1
>1000           0088 RG1ADR:  EQU     01000H
                0089 ;        ADDR OF REGION 2
>1800           0090 RG2ADR:  EQU     01800H
                0091 ;        SECTOR INCREMENT CONSTABNT
>0040           0092 INCSEC:  EQU     00040H
                0093 ;        MARKER CONSTANT
>00FF           0094 MARKER:  EQU     0FFH
                0095 ;        LOCATION OF VECTORS WHICH WILL
                0096 ;        PROGRAM THE DEVICES
>07E0           0097 CNTVEC:  EQU     07E0H
                0098 ;
                0099 ;CONTROL VECTORS FOR THE DEVICES
                0100          ORG     CNTVEC
07E0  4F        0101          DEFB    PIOV1
07E1  F6        0102          DEFB    PIOV2
07E2  87        0103          DEFB    PIOV3
07E3  18        0104          DEFB    SIOVR
07E4  02        0105          DEFB    SIOV1
07E5  F4        0106          DEFB    SIOV2
07E6  04        0107          DEFB    SIOV3
07E7  4C        0108          DEFB    SIOV4
07E8  01        0109          DEFB    SIOV7
07E9  82        0110          DEFB    SIOV8
07EA  05        0111          DEFB    SIOV5
07EB  68        0112          DEFB    SIOV6
07EC  28        0113          DEFB    SIOHLT
07ED  F8        0114          DEFB    CTCV1
07EE  C7        0115          DEFB    CTCENA
07EF  FF        0116          DEFB    CTC256
                0117 ;
```

```
    FLIGHT PROGRAM
ADDR   OBJECT     ST #

                  0118 ;
                  0119 ;ORGIN 0 IS THE LOCATION OF THE SYSTEM RESET
                  0120          ORG     0
0000   3EFF       0121          LD      A,RESET ;MAKE SURE THERE
0002   D3FF       0122          OUT     (RESET),A  ;IS TIME TO RESET
0004   ED5E       0123          IM      2       ;VECTORED INT MODE
0006   31F327     0124          LD      SP,STAK ;STACK STARTS UNDER
                  0125                          ;INT TABLE
0009   3E27       0126          LD      A,IREG  ;HIGH ORDER INT
000B   ED47       0127          LD      I,A     ;TABLE BYTE
000D   DD21F427   0128          LD      IX,INTBL        ;INT TBL ADDR
                  0129 ;
                  0130 ;SET UP THE INTERRUPT TABLE
0011   219700     0131          LD      HL,SIOMK        ;SIO INT ROUTINE
0014   DD7500     0132          LD      (IX+0),L        ;LOW ORDER BYTE
0017   DD7401     0133          LD      (IX+1),H   ;HIGH ORDER ENTRY
001A   218D00     0134          LD      HL,PIOINT       ;PIO INT ROUTINE
001D   DD7502     0135          LD      (IX+2),L
0020   DD7403     0136          LD      (IX+3),H
0023   21AE01     0137          LD      HL,CTCERR       ;CTC  0 TIMEOUT
0026   DD7504     0138          LD      (IX+4),L        ;THIS IS SYSERR
0029   DD7405     0139          LD      (IX+5),H        ;CONDITION
                  0140 ;
                  0141 ;SET UP THE DEVICES
002C   21E007     0142          LD      HL,CNTVEC       ;LOC OF THE CONT
                  0143                          ;VECTOR TABLE
002F   0E19       0144          LD      C,PIOPAC        ;PIO PORT A CONT
0031   0603       0145          LD      B,3     ;3 VECTORS
0033   EDB3       0146          OTIR            ;INITIALIZE PIO
0035   0E0D       0147          LD      C,SIOCAC        ;RESET CHAN A
0037   EDA3       0148          OUTI
0039   00         0149          NOP             ;GIVE TIME TO RESET
003A   0E0F       0150          LD      C,SIOCBC        ;SIO CHAN B CONT
003C   EDA3       0151          OUTI
003E   EDA3       0152          OUTI            ;SIO INT VECTOR
0040   0E0D       0153          LD      C,SIOCAC        ;SIO CHAN A CONT
0042   0607       0154          LD      B,7     ;7 VECTORS
0044   EDB3       0155          OTIR            ;INITIALIZE SIO
0046   0E14       0156          LD      C,CTC0  ;CTC CHAN 0
0048   0603       0157          LD      B,3     ;3 VECTORS
004A   EDB3       0158          OTIR            ;INITIALIZE CTC CHAN 0
                  0159 ;
                  0160 ;PREPARE TO RUN
004C   DB18       0161          IN      A,(PIOPAD)      ;PIO SET UP
004E   0679       0162          LD      B,CTC121        ;REG B CONTAINS
                  0163                          ;CTC CHAN2 TIME CONST
0050   D9         0164          EXX
0051   1E00       0165          LD      E,0     ;RECORD # FOR SIO
0053   D9         0166          EXX
                  0167 ;
                  0168 ;********MAIN PROGRAM*************
0054   CD9801     0169 LOOP:    CALL    REGIN1
0057   CDBB01     0170          CALL    REGIN2
005A   C35400     0171          JP      LOOP
                  0172 ;
                  0173 ;NON MASKABLE INTERRUPT ROUTINE
                  0174          ORG     66H
                  0175 ;CARE MUST BE TAKEN THAT THE PRIMED REGISTER
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

                0176 ;SET IS NOT MISTAKEN FOR THE UNPRIMED SET
                0177 ;THIS CAN HAPPEN IF THE NMI HAPPENS DURING
                0178 ;AN SIO INTERRUPT.  TO BE SAFE BC MUST BE
                0179 ;COPIED BEFORE AND AFTER NMI SERVICING
0066  C5       0180 NMINT:  PUSH    BC
0067  D9       0181         EXX
0068  C1       0182         POP     BC        ;BC' <= BC
                0183 ;PROTECT THE AF REGS ALSO
0069  F5       0184         PUSH    AF
                0185 ;IF BIT 1,C IS SET THEN GO AHEAD AND SIGNAL
                0186 ;FOR A NEW REGION TO BE OUTPUT, BUT IF IT
                0187 ;IS NOT SET THE CURRENT REGION IS STILL
                0188 ;BEING OUTPUT
006A  CB49       0189         BIT     1,C
006C  CA7100     0190         JP      Z,SKIP1
006F  CBF9       0191         SET     7,C       ;SIGNAL REGION CHANGE
                0192 ;CALCULATE THE SPIN PERIOD BY SUBTRACTING THE
                0193 ;TIME REG VALUE FROM 256
0071  DB14       0194 SKIP1:  IN      A,(CTC0)          ;FETCH TIME
0073  D6FF       0195         SUB     0FFH
0075  ED44       0196         NEG
0077  47         0197         LD      B,A       ;PLACE SPIN PERIOD
                0198                           ;IN B REG
                0199 ;RESET CTC 0 TO 256 COUNTS
0078  3EC7       0200         LD      A,CTCENA          ;RESET CH 0
007A  D314       0201         OUT     (CTC0),A
007C  3EFF       0202         LD      A,CTC256          ;SET FOR 256
007E  D314       0203         OUT     (CTC0),A    ;COUNTS
                0204                           ;RESET CTC 1
0080  3EC7       0205         LD      A,CTCENA
0082  D315       0206         OUT     (CTC1),A
0084  78         0207         LD      A,B
0085  D315       0208         OUT     (CTC1),A
                0209 ;CLEAN UP BEFORE LEAVING
0087  F1         0210         POP     AF
0088  C5         0211         PUSH    BC        ;PROTECT SIO
0089  D9         0212         EXX
008A  C1         0213         POP     BC
008B  ED45       0214         RETN    ;INT FLAGS IFF1 AND IFF2
                0215                           ;AUTOMATICALLY TAKEN
                0216                           ;CARE OF WITHOUT EI
                0217 ;
                0218 ;PIO INTERRUPT ROUTINE
                0219 ;THE PARTICLE ADDR IS FORMED FROM THE DE
                0220 ;REGS AND PLACED IN REGS HL
                0221 ;REG H IS THE SAME DURING ANY GIVEN SECTOR
                0222 ;SO ONLY L NEEDS TO BE SET UP TO GET THE ADDR
008D  F5         0223 PIOINT: PUSH    AF
008E  DB18       0224         IN      A,(PIOPAD)  ;FETCH DETECTOR AND
                0225                           ;ENERGY BIN DATA
0090  AB         0226         XOR     E         ;COMBINE E AND A
0091  6F         0227         LD      L,A       ;FORM ADDR
0092  34         0228         INC     (HL)      ;ADD 1 COUNT
0093  F1         0229         POP     AF
0094  FB         0230         EI
0095  ED4D       0231         RETI
                0232 ;
                0233 ;
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

                0234 ;
                0235 ;SIO MARKER PULSE ROUTINE
0097  D9        0236 SIOMK:  EXX       ;ALL POINTERS RESIDE
0098  08        0237         EX        AF,AF'  ;IN PRIME REGS
                0238 ;FIRST CALL IS CHECKED BY LOOKING AT
                0239 ;THE DETECTOR BITS BEFORE INCREMENTING HL
                0240 ;IF THE BITS ARE STILL 11 THEN IT IS THE
                0241 ;FIRST CALL AND A MARKER SHOULD BE OUTPUT
0099  CB65      0242         BIT       4,L
009B  23        0243         INC       HL        ;DOES NOT AFFECT
                0244                             ;FLAGS!!!
009C  CAB400    0245         JP        Z,NFC1    ;JUMP IF NOT FIRST
                0246                             ;CALL
                0247 ;IF IT IS A FIRST CALL CHECK TO SEE IF AT A
                0248 ;REGION END INDICATED BY THE SECTOR REG = 32
                0249 ;(BIT 5 = 1)
009F  CB6A      0250         BIT       5,D
00A1  CAAF00    0251         JP        Z,NEOB1 ;JUMP IF NOT AT END
                0252 ;AT END OF REGION SWAP TO UNPRIMED REG SET
                0253 ;SET REGION CHANGE FLAG (BIT 1,C) AND TURN OFF
                0254 ;SIO SO IT WONT INTERRUPT AGAIN UNTILL READY TO
                0255 ;SEND NEXT REGION OF DATA
00A4  D9        0256         EXX
00A5  CBC9      0257         SET       1,C       ;REGION CHANGE FLAG
00A7  3E28      0258         LD        A,SIOHLT          ;TURN OFF SIO
00A9  D30D      0259         OUT       (SIOCAC),A
00AB  08        0260         EX        AF,AF'
00AC  FB        0261         EI                  ;INT TABLE LEFT POINTING
00AD  ED4D      0262         RETI                ;TO SIOMK
00AF  3EFF      0263 NEOB1:  LD        A,MARKER
00B1  C3B500    0264         JP        ZERHL1
00B4  7E        0265 NFC1:   LD        A,(HL)  ;GET DATA AT ADDR HL
00B5  3600      0266 ZERHL1: LD        (HL),0  ;ZERO HL
00B7  D30C      0267         OUT       (SIOCAD),A        ;OUTPUT DATA
                0268 ;LOOK FOR THE END OF THE 16 ENERGY BINS
                0269 ;OF THIS DETECTOR BY SENSING A 1111 IN THE
                0270 ;ENERGY BIN BITS
                0271 ;IF AT THE END THEN LINK TO THE NEXT ROUTINE
00B9  7D        0272         LD        A,L
00BA  E60F      0273         AND       00001111B         ;EXPOSE ENERGY
                0274                               ;BIN #
00BC  EE0F      0275         XOR       00001111B         ;CHECK FOR BIN15
00BE  C2CC00    0276         JP        NZ,FINI ;WHICH SETS ZERO FLAG
                0277 ;CHANGE LINKAGE
00C1  E5        0278         PUSH      HL
00C2  21D100    0279         LD        HL,SIOBK          ;BLOCK # ROUTINE
00C5  DD7500    0280         LD        (IX+0),L
00C8  DD7401    0281         LD        (IX+1),H
00CB  E1        0282         POP       HL
00CC  08        0283 FINI:   EX        AF,AF'
00CD  D9        0284         EXX
00CE  FB        0285         EI
00CF  ED4D      0286         RETI
                0287 ;SIO BLOCK # ROUTINE
                0288 ;THIS ROUTINE IS IDENTICAL TO THE MARKER
                0289 ;ROUTINE EXCEPT FOR THE END OF BLOCK CHECK
00D1  D9        0290 SIOBK:  EXX
00D2  08        0291         EX        AF,AF'
```

```
    FLIGHT PROGRAM
ADDR   OBJECT     ST #

00D3   CB65       0292            BIT     4,L
00D5   23         0293            INC     HL
00D6   C2DD00     0294            JP      NZ,NFC2
00D9   7B         0295            LD      A,E ;BLOCK #
00DA   C3DE00     0296            JP      ZERHL2
00DD   7E         0297 NFC2:      LD      A,(HL)
00DE   3600       0298 ZERHL2:    LD      (HL),0
00E0   D30C       0299            OUT     (SIOCAD),A
00E2   7D         0300            LD      A,L
00E3   E60F       0301            AND     00001111B
00E5   EE0F       0302            XOR     00001111B
00E7   C2F500     0303            JP      NZ,FINI2
00EA   E5         0304            PUSH    HL
00EB   21FA00     0305            LD      HL,SIOSR
00EE   DD7500     0306            LD      (IX+0),L
00F1   DD7401     0307            LD      (IX+1),H
00F4   E1         0308            POP     HL
00F5   08         0309 FINI2:     EX      AF,AF'
00F6   D9         0310            EXX
00F7   FB         0311            EI
00F8   ED4D       0312            RETI
                  0313 ;
                  0314 ;SIO SECTOR # ROUTINE
                  0315 ;THIS ROUTINE IS IDENTICAL TO THE MARKER
                  0316 ;ROUTINE EXCEPT FOR THE END OF BLOCK CHECK
00FA   D9         0317 SIOSR:     EXX
00FB   08         0318            EX      AF,AF'
00FC   CB65       0319            BIT     4,L
00FE   23         0320            INC     HL
00FF   CA0601     0321            JP      Z,NFC3
0102   7A         0322            LD      A,D ;SECTOR #
0103   C30701     0323            JP      ZERHL3
0106   7E         0324 NFC3:      LD      A,(HL)
0107   3600       0325 ZERHL3:    LD      (HL),0
0109   D30C       0326            OUT     (SIOCAD),A
010B   7D         0327            LD      A,L
010C   E60F       0328            AND     00001111B
010E   EE0F       0329            XOR     00001111B
0110   C21E01     0330            JP      NZ,FINI3
0113   E5         0331            PUSH    HL
0114   212301     0332            LD      HL,SIOCK
0117   DD7500     0333            LD      (IX+0),L
011A   DD7401     0334            LD      (IX+1),H
011D   E1         0335            POP     HL
011E   08         0336 FINI3:     EX      AF,AF'
011F   D9         0337            EXX
0120   FB         0338            EI
0121   ED4D       0339            RETI
                  0340 ;
                  0341 ;SIO CHECKER # ROUTINE
                  0342 ;THIS ROUTINE IS IDENTICAL TO THE MARKER
                  0343 ;ROUTINE EXCEPT FOR THE END OF BLOCK CHECK
0123   D9         0344 SIOCK:     EXX
0124   08         0345            EX      AF,AF'
0125   CB65       0346            BIT     4,L
0127   23         0347            INC     HL
0128   C23001     0348            JP      NZ,NFC4
012B   7A         0349            LD      A,D ;SECTOR #
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

012C  AB        0350          XOR     E       ;BLOCK #
                0351                          ;A = E XOR D
012D  C33101    0352          JP      ZERHL4
0130  7E        0353 NFC4:    LD      A,(HL)
0131  3600      0354 ZERHL4:  LD      (HL),0
0133  D30C      0355          OUT     (SIOCAD),A
0135  7D        0356          LD      A,L
0136  E60F      0357          AND     00001111B
0138  EE0F      0358          XOR     00001111B
013A  C24901    0359          JP      NZ,FINI4
013D  E5        0360          PUSH    HL
013E  219700    0361          LD      HL,SIOMK
0141  DD7500    0362          LD      (IX+0),L
0144  DD7401    0363          LD      (IX+1),H
0147  E1        0364          POP     HL
0148  14        0365          INC     D
0149  08        0366 FINI4:   EX      AF,AF'
014A  D9        0367          EXX
014B  FB        0368          EI
014C  ED4D      0369          RETI
                0370 ;
                0371 ;
                0372 ;IF CTC CHAN 0 SHOULD EVER INTERRUPT IT
                0373 ;MEANS THE MAGNITOMETER HAS NOT RESET CTC 0;
014E  F5        0374 CTCERR:  PUSH    AF      ;PROTECT AF
                0375 ;SET THE SPIN RATE AT ABOUT 5 RPS OR
                0376 ;121 COUNTS ON CHAN 0
014F  3EC7      0377          LD      A,CTCENA        ;RESET CHAN 0
0151  D314      0378          OUT     (CTC0),A
0153  3E79      0379          LD      A,CTC121        ;121 COUNTS
0155  D314      0380          OUT     (CTC0),A
0157  47        0381          LD      B,A     ;B REG - COUNTS/REV
                0382                          ;RESET CTC 1
0158  3EC7      0383          LD      A,CTCENA
015A  D315      0384          OUT     (CTC1),A
015C  78        0385          LD      A,B
015D  D315      0386          OUT     (CTC1),A
                0387 ;CHECK FOR END OF REGION JUST AS NMI WOULD
015F  CB49      0388          BIT     1,C
0161  CA6601    0389          JP      Z,SKIP10
0164  CBF9      0390          SET     7,C
0166  F1        0391 SKIP10:  POP     AF      ;RESTORE AF
0167  FB        0392          EI
0168  ED4D      0393          RETI
                0394 ;
                0395 ;REGION 1 CTC SECTOR INTERRUPT ROUTINE
016A  F5        0396 CTCRG1:  PUSH    AF
016B  214000    0397          LD      HL,INCSEC       ;INCREMENT CONST
016E  19        0398          ADD     HL,DE   ;NEW SECTOR DATA IN HL
016F  CB5C      0399          BIT     3,H     ;REGION CHANGE?
0171  CA7701    0400          JP      Z,NOCHG1        ;STILL IN REG1
0174  210010    0401          LD      HL,RG1ADR
0177  54        0402 NOCHG1:  LD      D,H     ;HL AND DE ARE THE
0178  5D        0403          LD      E,L     ;SAME ON EXIT
0179  3EFF      0404          LD      A,RESET
017B  D3FF      0405          OUT     (RESET),A
017D  F1        0406          POP     AF
017E  FB        0407          EI              ;REMEMBER THAT THE TIME
```

```
    FLIGHT PROGRAM
ADDR  OBJECT     ST #

017F  ED4D       0408          RETI              ;CONST IS AUTOMATICALLY
                 0409                            ;LOADED IN CTC1
                 0410 ;
                 0411 ;REGION 2 CTC SECTOR INTERRUPT ROUTINE
0181  F5         0412 CTCRG2:  PUSH    AF
0182  214000     0413          LD      HL,INCSEC       ;INCREMENT CONST
0185  19         0414          ADD     HL,DE   ;NEW SECTOR DATA IN HL
0186  CB5C       0415          BIT     3,H     ;REGION CHANGE?
0188  C28E01     0416          JP      NZ,NOCHG2       ;STILL IN REG1
018B  210018     0417          LD      HL,RG2ADR
018E  54         0418 NOCHG2:  LD      D,H     ;HL AND DE ARE THE
018F  5D         0419          LD      E,L     ;SAME ON EXIT
0190  3EFF       0420          LD      A,RESET
0192  D3FF       0421          OUT     (RESET),A
0194  F1         0422          POP     AF
0195  FB         0423          EI                ;REMEMBER THAT THE TIME
0196  ED4D       0424          RETI              ;CONST IS AUTOMATICALLY
                 0425                            ;LOADED IN CTC1
                 0426 ;
                 0427 ;ACCUMULATE DATA IN REGION 1 AND TRANSMIT
                 0428 ;DATA FROM REGION 2
0198  F3         0429 REGIN1:  DI        ;INITIALIZE REGION
0199  0E00       0430          LD      C,0       ;CLEAR END OF
                 0431                            ;BLOCK FLAG
019B  216A01     0432          LD      HL,CTCRG1       ;SET CTC CHAN 1
                 0433                            ;TO REGION 1
019E  DD7506     0434          LD      (IX+6),L        ;INT TABLE
01A1  DD7407     0435          LD      (IX+7),H
01A4  110010     0436          LD      DE,RG1ADR       ;SET DE WITH
                 0437                            ;REGION 1 BOUNDRY
01A7  62         0438          LD      H,D  ;SET UP H FOR THIS SECTOR
01A8  D9         0439          EXX               ;SET UP SIO CONTROL
01A9  210118     0440          LD      HL,RG2ADR+1     ;REGION 2 ADDR
01AC  1600       0441          LD      D,0       ;SECTOR 0
01AE  1C         0442          INC     E         ;INC TO NEXT RECORD
01AF  D9         0443          EXX
01B0  3EFF       0444          LD      A,MARKER        ;TURN ON SIO BY
01B2  D30C       0445          OUT     (SIOCAD),A ;SENDING FIRST MARKER
01B4  FB         0446          EI
                 0447 ;CHECK TO SEE IF THE CHANGE REGION
                 0448 ;BIT HAS BEEN SET
01B5  CB79       0449 WAIT1:   BIT     7,C
01B7  C0         0450          RET     NZ
01B8  C3B501     0451          JP      WAIT1
                 0452 ;
                 0453 ;
                 0454 ;
                 0455 ;ACCUMULATE DATA IN REGION 2 AND TRANSMIT
                 0456 ;DATA FROM REGION 1
01BB  F3         0457 REGIN2:  DI        ;INITIALIZE REGION
01BC  0E00       0458          LD      C,0       ;CLEAR END OF
                 0459                            ;BLOCK FLAG
01BE  218101     0460          LD      HL,CTCRG2       ;SET CTC CHAN 1
                 0461                            ;TO REGION 2
01C1  DD7506     0462          LD      (IX+6),L        ;INT TABLE
01C4  DD7407     0463          LD      (IX+7),H
01C7  110018     0464          LD      DE,RG2ADR       ;SET DE WITH
                 0465                            ;REGION 2 BOUNDRY
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

01CA  62        0466          LD      H,D  ;SET UP H FOR THIS SECTOR
01CB  D9        0467          EXX              ;SET UP SIO CONTROL
01CC  210110    0468          LD      HL,RG1ADR+1      ;REGION 1 ADDR
01CF  1600      0469          LD      D,0      ;SECTOR 0
01D1  1C        0470          INC     E        ;INC TO NEXT RECORD
01D2  D9        0471          EXX
01D3  3EFF      0472          LD      A,MARKER         ;TURN ON SIO BY
01D5  D30C      0473          OUT     (SIOCAD),A ;SENDING FIRST MARKER
01D7  FB        0474          EI
                0475 ;CHECK TO SEE IF THE CHANGE REGION
                0476 ;BIT HAS BEEN SET
01D8  CB79      0477 WAIT2:   BIT     7,C
01DA  C0        0478          RET     NZ
01DB  C3D801    0479          JP      WAIT2
                0480 ;
                0481 ;
                0482          END
```

```
    FLIGHT PROGRAM
ADDR  OBJECT    ST #

CNTVEC      07E0 CTC0       0014 CTC1       0015 CTC121     0079
CTC2        0016 CTC256     00FF CTC3       0017 CTCENA     00C7
CTCERR      014E CTCRG1     016A CTCRG2     0181 CTCV1      00F8
FINI        00CC FINI2      00F5 FINI3      011E FINI4      0149
INCSEC      0040 INTBL      27F4 IREG       0027 LOOP       0054
MARKER      00FF NEOB1      00AF NFC1       00B4 NFC2       00DD
NFC3        0106 NFC4       0130 NMINT      0066 NOCHG1     0177
NOCHG2      018E PIOINT     008D PIOPAC     0019 PIOPAD     0018
PIOPBC      001B PIOPBD     001A PIOV1      004F PIOV2      00F6
PIOV3       0087 REGIN1     0198 REGIN2     01BB RESET      FFFF
RG1ADR      1000 RG2ADR     1800 SIOBK      00D1 SIOCAC     000D
SIOCAD      000C SIOCBC     000F SIOCBD     000E SIOCK      0123
SIOHLT      0028 SIOMK      0097 SIOSR      00FA SIOV1      0002
SIOV2       00F4 SIOV3      0004 SIOV4      004C SIOV5      0005
SIOV6       0068 SIOV7      0001 SIOV8      0082 SIOVR      0018
SKIP1       0071 SKIP10     0166 STAK       27F3 WAIT1      01B5
WAIT2       01D8 ZERHL1     00B5 ZERHL2     00DE ZERHL3     0107
ZERHL4      0131
ERRORS=0000
ERRORS=0000
.
```

II.2 *BETA-1 Punch Program*

```
     BETA1 PUNCH PROG 13
 ADDR  OBJECT    ST #

'0000  00        0002 BETA1:  NOP
'0001  3E00      0003         LD      A,0
'0003  DD7705    0004         LD      (IX+5),A
'0006  1E00      0005 LP:     LD      E,0
'0008  CD22E5    0006         CALL    RDCHR
'000B  D3D2      0007         OUT     (0D2H),A
'000D  76        0008         HALT
'000E  DD7E05    0009         LD      A,(IX+5)
'0011  FEFF      0010         CP      0FFH
'0013  C8        0011         RET     Z
'0014  C30600'    0012         JP      LP
                 0013 ;
'0017  1E00      0014 INPIO:  LD      E,0
'0019  F5        0015         PUSH    AF
'001A  DBD0      0016         IN      A,(0D0H)
'001C  57        0017         LD      D,A
'001D  CD27E5    0018         CALL    WRCHR
'0020  CD2A00'    0019         CALL    STORE
'0023  CD3D00'    0020         CALL    WRTE
'0026  F1        0021         POP     AF
'0027  FB        0022         EI
'0028  ED4D      0023         RETI
                 0024 ;
                 0025 ;STORE SEQUENCE 'W (CR)
'002A  00        0026 STORE:  NOP
                 0027 ;PUT SEQUENCE IN CTC CHAN 2 AND 3
                 0028 ;(NOT USED) LOCATIONS
'002B  D5        0029         PUSH    DE
'002C  DD5603    0030         LD      D,(IX+3)
'002F  DD7204    0031         LD      (IX+4),D
'0032  DD5602    0032         LD      D,(IX+2)
'0035  DD7203    0033         LD      (IX+3),D
'0038  DD7702    0034         LD      (IX+2),A
'003B  D1        0035         POP     DE
'003C  C9        0036         RET
                 0037 ;
'003D  00        0038 WRTE:   NOP
                 0039 ;CHECK FOR 'W (CR)
'003E  3E0D      0040         LD      A,0DH
'0040  DDBE02    0041         CP      (IX+2)  ;A SHOULD HAVE (CR)
'0043  C26400'    0042         JP      NZ,LEAVE
'0046  3E57      0043         LD      A,'W'   ;LOOK FOR W
'0048  DDBE03    0044         CP      (IX+3)
'004B  CA5600'    0045         JP      Z,UPAR
'004E  3E52      0046         LD      A,'R'
'0050  DDBE03    0047         CP      (IX+3)
'0053  C26400'    0048         JP      NZ,LEAVE
'0056  00        0049 UPAR:   NOP
'0057  3E5E      0050         LD      A,'^'   ;LOOK FOR ^
'0059  DDBE04    0051         CP      (IX+4)
'005C  C26400'    0052         JP      NZ,LEAVE
'005F  3EFF      0053         LD      A,0FFH
'0061  DD7705    0054         LD      (IX+5),A
'0064  C9        0055 LEAVE:  RET
  >E522          0056 RDCHR:  EQU     0E522H
  >E527          0057 WRCHR:  EQU     0E527H
  >E59C          0058 CRLF:   EQU     0E59CH
  >E3C7          0059 PTXT:   EQU     0E3C7H
```

```
    BETA1 PUNCH PROG 13
 ADDR  OBJECT    ST #

 >E597           0060 ECHO:   EQU     0E597H
 >E583           0061 ASBIN:  EQU     0E583H
                 0062 ;
                 0063 ;
'0065  00        0064 PUNCH:  NOP
'0066  F3        0065         DI      ;DISABLE EXCH REGS
'0067  08        0066         EX      AF,AF'
'0068  CBE3      0067         SET     4,E     ;SIGNAL REG EXCH
'006A  D5        0068         PUSH    DE
'006B  D9        0069         EXX
'006C  D1        0070         POP     DE      ;DE'<=DE
'006D  FB        0071         EI
'006E  CB5B      0072         BIT     3,E     ;INITIALIZED?
'0070  C49700'   0073         CALL    NZ,SETUP
                 0074 ;RESET CTC
'0073  3ED7      0075         LD      A,11010111B
'0075  D3DB      0076         OUT     (0DBH),A
'0077  3EFF      0077         LD      A,0FFH
'0079  D3DB      0078         OUT     (0DBH),A
'007B  CD8800'   0079         CALL    RTS
'007E  76        0080         HALT
'007F  F3        0081         DI              ;EXCH REGS
'0080  08        0082         EX      AF,AF'
'0081  CBA3      0083         RES     4,E
'0083  D5        0084         PUSH    DE
'0084  D9        0085         EXX
'0085  D1        0086         POP     DE
'0086  FB        0087         EI
'0087  C9        0088         RET
                 0089 ;
                 0090 ;READY TO SEND ROUTINE
'0088  00        0091 RTS:    NOP
'0089  F3        0092         DI
'008A  CB78      0093         BIT     7,B     ;IF ERROR
'008C  C29300'   0094         JP      NZ,EXIT ;DUMP CHAR
'008F  7A        0095         LD      A,D
'0090  4A        0096         LD      C,D     ;SAVE CHAR OUTPUT
'0091  D3D2      0097         OUT     (0D2H),A
'0093  CBBB      0098 EXIT:   RES     7,E     ;IMMED RET
'0095  FB        0099         EI
'0096  C9        0100         RET
                 0101 ;
                 0102 ;INITIALIZATION ROUTINE
'0097  00        0103 SETUP:  NOP
'0098  ED5E      0104         IM      2
'009A  CB9B      0105         RES     3,E     ;INIT BIT
'009C  0640      0106         LD      B,01000000B  ;RTS BIT SET
'009E  0E00      0107         LD      C,0
'00A0  CDE600'   0108         CALL    TABLE   ;TABLE LOC IN IX
                 0109 ;
                 0110 ;SET UP PIO
'00A3  DDE5      0111         PUSH    IX
'00A5  E1        0112         POP     HL      ;HL<=IX
'00A6  7D        0113         LD      A,L
'00A7  D3D1      0114         OUT     (0D1H),A        ;D0 IS INPUT
'00A9  3E7F      0115         LD      A,01111111B     ;INPUT MODE
'00AB  D3D1      0116         OUT     (0D1H),A
'00AD  3E87      0117         LD      A,10000111B     ;EI PIO
```

```
     BETA1 PUNCH PROG 13
 ADDR  OBJECT    ST #

'00AF   D3D1        0118          OUT     (0D1H),A
'00B1   3E00        0119          LD      A,0     ;D2 IS OUTPUT
'00B3   D3D3        0120          OUT     (0D3H),A        ;DUMMY INT VEC
'00B5   3E3F        0121          LD      A,00111111B     ;OUTPUT MODE
'00B7   D3D3        0122          OUT     (0D3H),A
'00B9   3E07        0123          LD      A,00000111B     ;D1 OUTPUT INTS
'00BB   D3D3        0124          OUT     (0D3H),A
                    0125 ;SET UP CTC
'00BD   7D          0126          LD      A,L     ;INT VEC
'00BE   D3D8        0127          OUT     (0D8H),A
'00C0   FB          0128          EI
                    0129 ;SET UP INT TABLE
'00C1   3EFF        0130          LD      A,0FFH  ;SET UP INT VECTORS
'00C3   ED47        0131          LD      I,A
'00C5   211700'     0132          LD      HL,INPIO        ;BETA1 INPUT
'00C8   DD7500      0133          LD      (IX+0),L
'00CB   DD7401      0134          LD      (IX+1),H        ;PIO INT
'00CE   211B01'     0135          LD      HL,INTCTC       ;GO AHEAD AND
'00D1   DD7506      0136          LD      (IX+6),L        ;SET UP CTC
'00D4   DD7407      0137          LD      (IX+7),H        ;CHAN 3
                    0138 ;CALL BETA1 TO SET UP TAPE MANUALLY
'00D7   D5          0139          PUSH    DE
'00D8   CD0000'     0140          CALL    BETA1
'00DB   D1          0141          POP     DE
                    0142 ;PUNCH INT ROUTINE
'00DC   213401'     0143          LD      HL,INTPIO
'00DF   DD7500      0144          LD      (IX+0),L
'00E2   DD7401      0145          LD      (IX+1),H
'00E5   C9          0146          RET
                    0147 ;
                    0148 ;LOCATE INT TABEL ABOVE THE USER
                    0149 ;MNEMONIC TABLE AND BELOW THE STACK
'00E6   00          0150 TABLE:   NOP
'00E7   2132FF      0151          LD      HL,0FF32H
'00EA   23          0152 LOOK:    INC     HL
'00EB   7E          0153          LD      A,(HL)
'00EC   FE80        0154          CP      80H     ;TOP OF TABLE
'00EE   C2EA00'     0155          JP      NZ,LOOK
                    0156 ;SET UP INT TABLE ON CORRECT BYTE
'00F1   23          0157          INC     HL
'00F2   CB85        0158          RES     0,L
'00F4   CB8D        0159          RES     1,L
'00F6   CB95        0160          RES     2,L
'00F8   DD210800    0161          LD      IX,8
'00FC   D5          0162          PUSH    DE
'00FD   54          0163          LD      D,H
'00FE   5D          0164          LD      E,L
'00FF   DD19        0165          ADD     IX,DE
'0101   D1          0166          POP     DE
'0102   C9          0167          RET
                    0168 ;
                    0169 ;INT ROUTINE REG EXCHANGE
'0103   00          0170 EXCH1:   NOP
'0104   CB63        0171          BIT     4,E     ;REGS EXCH'ED?
'0106   C20D01'     0172          JP      NZ,SKIP1
'0109   08          0173          EX      AF,AF'
'010A   D5          0174          PUSH    DE
'010B   D9          0175          EXX
```

```
     BETA1 PUNCH PROG 12
  ADDR  OBJECT    ST #

'010C  D1         0176          POP     DE
'010D  00         0177 SKIP1:   NOP
'010E  C9         0178          RET
                  0179 ;
'010F  00         0180 EXCH2:   NOP
'0110  CB63       0181          BIT     4,E
'0112  C21901'    0182          JP      NZ,SKIP2
'0115  08         0183          EX      AF,AF'
'0116  D5         0184          PUSH    DE
'0117  D9         0185          EXX
'0118  D1         0186          POP     DE
'0119  00         0187 SKIP2:   NOP
'011A  C9         0188          RET
                  0189 ;
                  0190 ;CTC INT ROUTINE
'011B  00         0191 INTCTC:  NOP
'011C  CD0301'    0192          CALL    EXCH1
'011F  CB78       0193          BIT     7,B     ;ERROR?
'0121  C22901'    0194          JP      NZ,EXIT1
'0124  3E03       0195          LD      A,03H   ;ETXT CHAR
'0126  4F         0196          LD      C,A     ;SAVE CHAR OUT.UT
'0127  D3D2       0197          OUT     (0D2H),A
'0129  3E3B       0198 EXIT1:   LD      A,00111011B     ;DISABLE CTC
'012B  F3         0199          DI
'012C  D3DB       0200          OUT     (0DBH),A
'012E  CD0F01'    0201 EXIT2:   CALL    EXCH2
'0131  FB         0202          EI
'0132  ED4D       0203          RETI
                  0204 ;
                  0205 ;PIO INTERRUPT ROUTINE
'0134  00         0206 INTPIO:  NOP
'0135  CD0301'    0207          CALL    EXCH1
'0138  DBD0       0208          IN      A,(0D0H)
'013A  B9         0209          CP      C       ;SEE IF CHAR RCVD
'013B  CA4101'    0210          JP      Z,SKIPA ;IS CHAR SENT
'013E  CBF8       0211          SET     7,B     ;SET ERROR BIT
'0140  AF         0212          XOR     A       ;CLEAR ACC TO AVOID CP
'0141  FE03       0213 SKIPA:   CP      03H     ;SEE IF ETXT
'0143  C24801'    0214          JP      NZ,SKIPB
'0146  D3D2       0215          OUT     (0D2H),A        ;SEND ETXT
'0148  CD0F01'    0216 SKIPB:   CALL    EXCH2
'014B  FB         0217          EI
'014C  ED4D       0218          RETI
                  0219          END
```

```
     BETA1 PUNCH PROG 13
  ADDR  OBJECT    ST #

ASBIN          E583 BETA1          0000 CRLF          E59C ECHO          E597
EXCH1          0103 EXCH2          010F EXIT          0093 EXIT1         0129
EXIT2          012E INPIO          0017 INTCTC        011B INTPIO        0134
LEAVE          0064 LOOK           00EA LP            0006 PTXT          E3C7
PUNCH          0065 RDCHR          E522 RTS           0088 SETUP         0097
SKIP1          010D SKIP2          0119 SKIPA         0141 SKIPB         0148
STORE          002A TABLE          00E6 UPAR          0056 WRCHR         E527
WRTE           003D
ERRORS=0000
ERRORS=0000
```

II.3 *BETA-1 Read Program*

```
     BETA READ PROG 2
 ADDR  OBJECT    ST #

'0000  00          0002 READ:   NOP
'0001  F3          0003         DI
'0002  08          0004         EX      AF,AF'
'0003  CBE3        0005         SET     4,E      ;SIGNAL EXCH REGS
'0005  D5          0006         PUSH    DE
'0006  D9          0007         EXX
'0007  D1          0008         POP     DE       ;DE'<=DE
'0008  FB          0009         EI
'0009  CB5B        0010         BIT     3,E
'000B  CA1A00'     0011         JP      Z,NOSET
'000E  DDE5        0012         PUSH    IX
'0010  E5          0013         PUSH    HL
'0011  CD0744      0014         CALL    SETUP
'0014  CD3000'     0015         CALL    RDVEC
'0017  E1          0016         POP     HL
'0018  DDE1        0017         POP     IX
'001A  00          0018 NOSET:  NOP
                   0019 ;RESET CTC
'001B  3ED7        0020         LD      A,11010111B
'001D  D3DB        0021         OUT     (0DBH),A
'001F  3EFF        0022         LD      A,0FFH
'0021  D3DB        0023         OUT     (0DBH),A
'0023  76          0024         HALT             ;WAIT FOR CHARS
'0024  CBBB        0025         RES     7,E      ;IMMED RET BIT
'0026  F3          0026         DI
'0027  08          0027         EX      AF,AF'
'0028  CBA3        0028         RES     4,E
'002A  D5          0029         PUSH    DE
'002B  D9          0030         EXX
'002C  D1          0031         POP     DE
'002D  7A          0032         LD      A,D      ;A AND D CONTAIN THE
'002E  FB          0033         EI               ;CHAR RECEIVED
'002F  C9          0034         RET
                   0035 ;
'0030  00          0036 RDVEC:  NOP
'0031  214400'     0037         LD      HL,RDCTC         ;CTC INT VEC
'0034  DD7506      0038         LD      (IX+6),L
'0037  DD7407      0039         LD      (IX+7),H
'003A  215500'     0040         LD      HL,RDPIO         ;PIO INT VEC
'003D  DD7500      0041         LD      (IX+0),L
'0040  DD7401      0042         LD      (IX+1),H
'0043  C9          0043         RET
                   0044 ;
'0044  00          0045 RDCTC:  NOP
'0045  CD7344      0046         CALL    EXCH1
'0048  1603        0047         LD      D,03H    ;ETXT
'004A  3E3B        0048         LD      A,00111011B      ;DISABLE CTC
'004C  F3          0049         DI
'004D  D3DB        0050         OUT     (0DBH),A
'004F  CD7F44      0051         CALL    EXCH2
'0052  FB          0052         EI
'0053  ED4D        0053         RETI
                   0054 ;
'0055  00          0055 RDPIO:  NOP
'0056  CD7344      0056         CALL    EXCH1
'0059  DBD0        0057         IN      A,(0D0H)
'005B  57          0058         LD      D,A
'005C  CD7F44      0059         CALL    EXCH2
```

```
     BETA READ PROG 2
  ADDR  OBJECT     ST #

 '005F  FB         0060          EI
 '0060  ED4D       0061          RETI
                   0062 ;
  >4473            0063 EXCH1:   EQU     04473H
  >447F            0064 EXCH2:   EQU     0447FH
  >4407            0065 SETUP:   EQU     04407H
                   0066          END
```

```
     BETA READ PROG 2
  ADDR  OBJECT     ST #

EXCH1          4473 EXCH2          447F NOSET          001A RDCTC          0044
RDPIO          0055 RDVEC          0030 READ           0000 SETUP          4407
ERRORS=0000
ERRORS=0000
.
```

APPENDIX III. Flowcharts for flight programs.

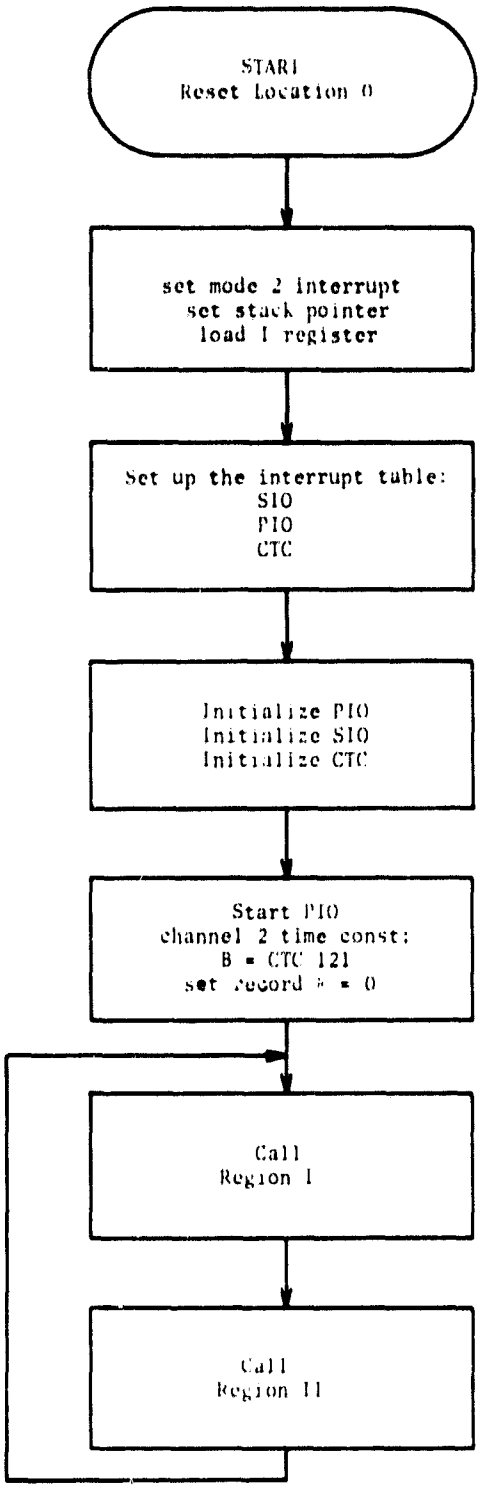


Figure III.1 Flight program lines 120-171 (Appendix II.1). Initialize CPU registers, all I/O devices, and start processing.

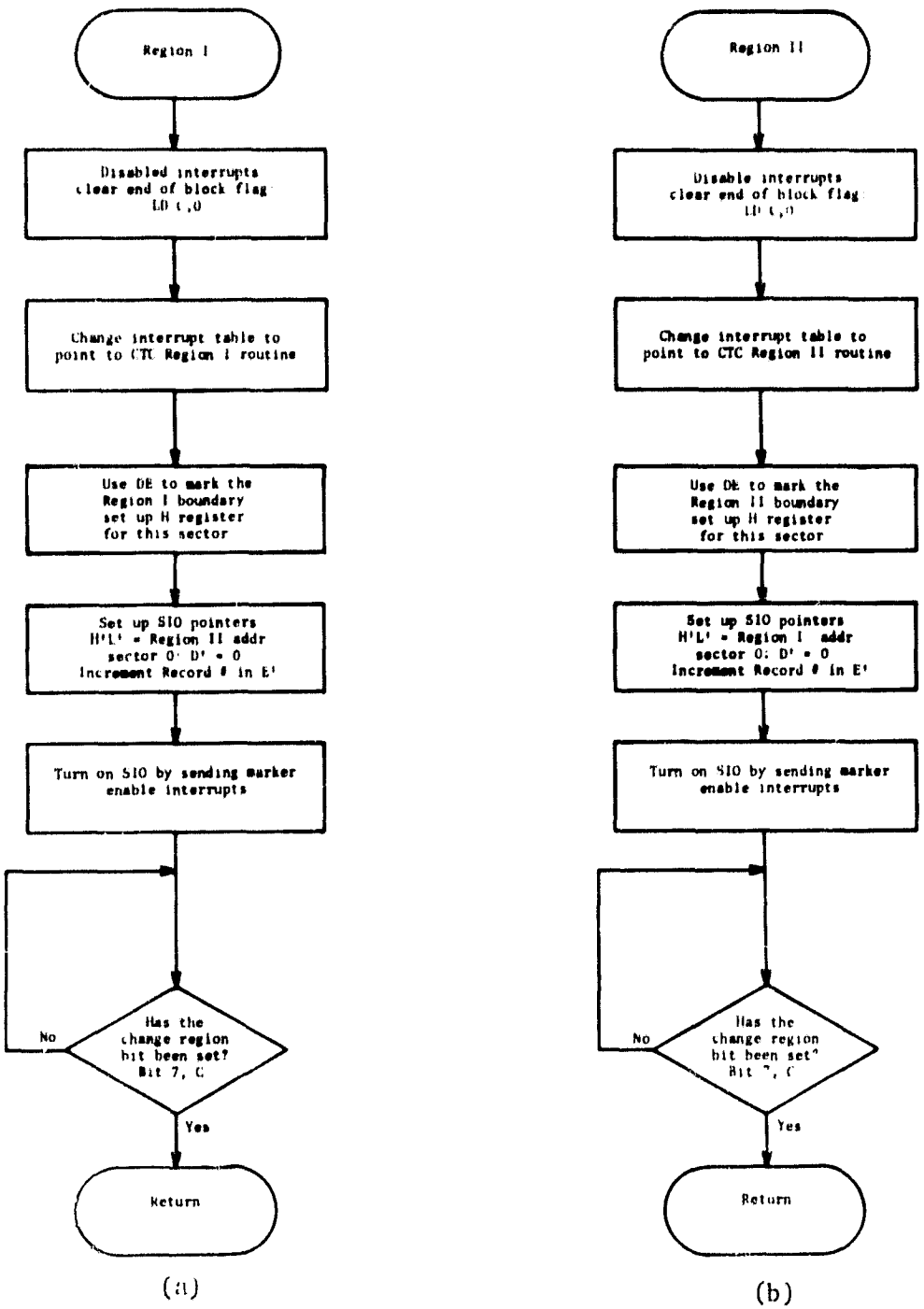


Figure III.2 Flight program lines (a) 427-451; (b) 455-479 (Appendix II.1). Regions I and II initialization and monitoring routines.

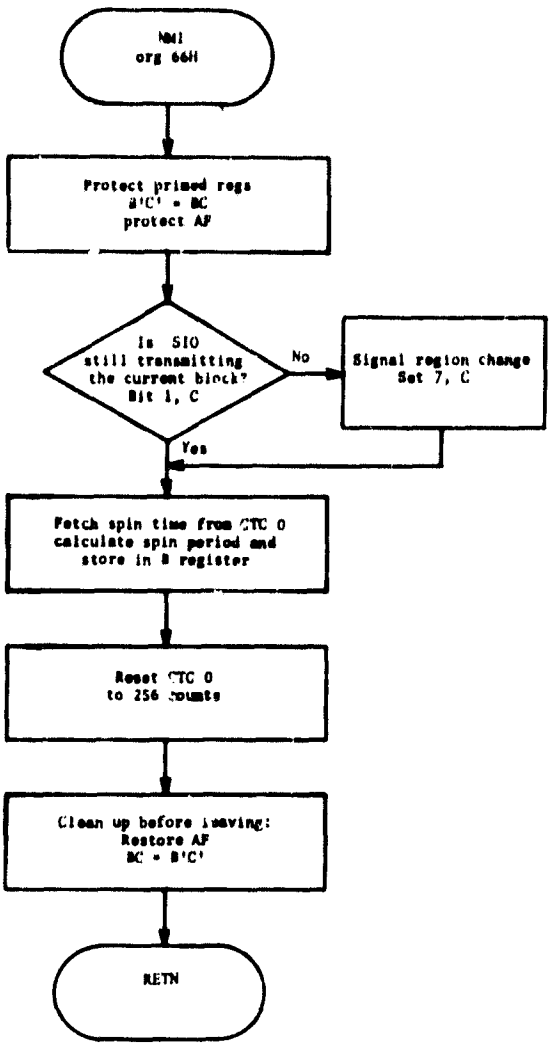


Figure III.3 Flight program lines 174-216 (Appendix II.1). Non-maskable interrupt service routine where the spin period is calculated and the region change is monitored.

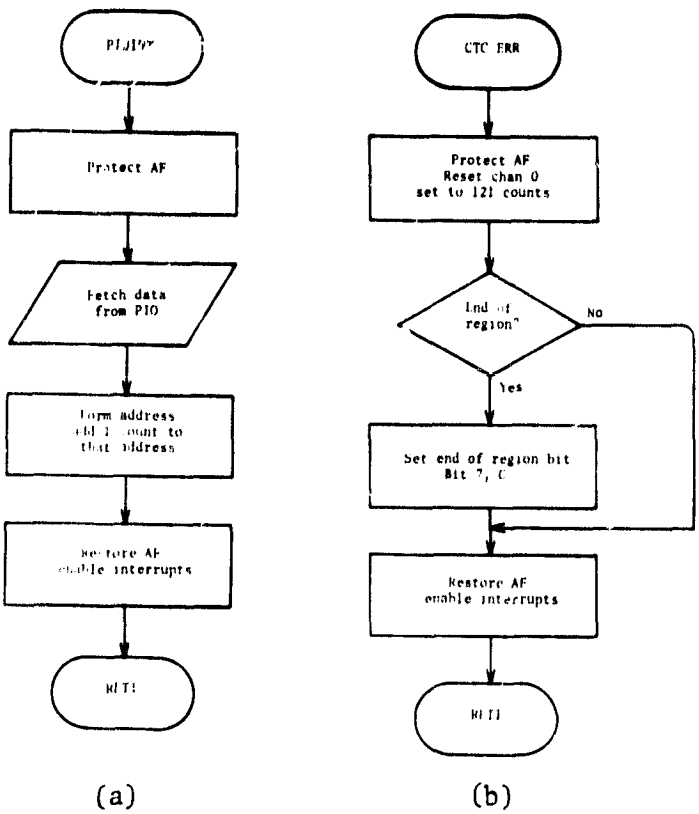
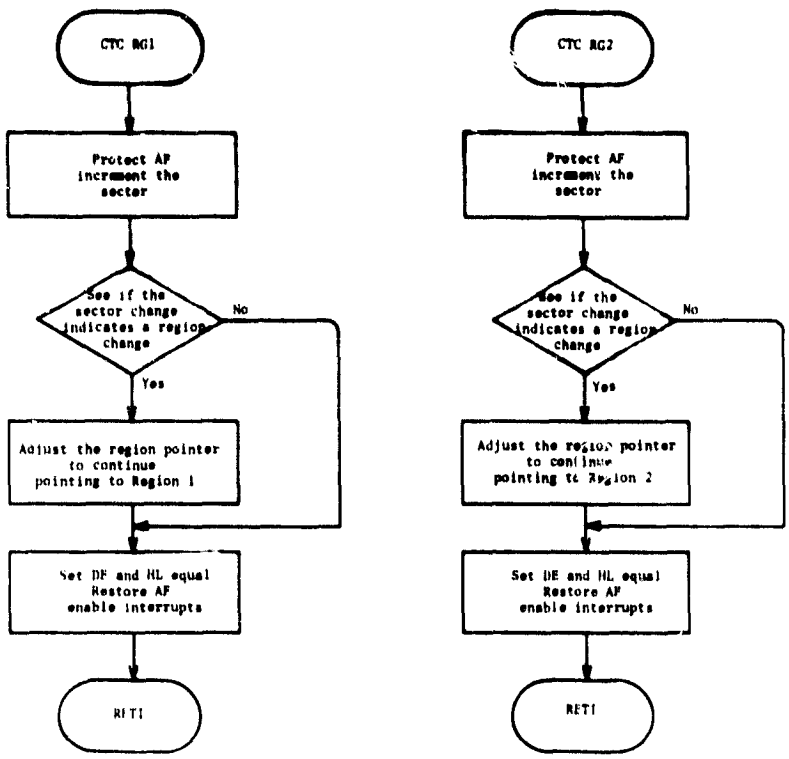


Figure III.4 (a) Flight program lines 218-231 (Appendix II.1). This routine collects data from the PHA board via the microprocessor PIO port A and accumulates the counts in [illegible]; (b) Flight program lines 372-393 (Appendix II.1). If the magnetometer fails, this routine will simulate a spin rate of 5 rps.



(a) (b)

Figure III.5 Flight program lines (a) 395-409; (b) 411-425 (Appendix II.1). CTC RG1 and CTC RG2 are responsible for incrementing the sectors from 0 to 31.

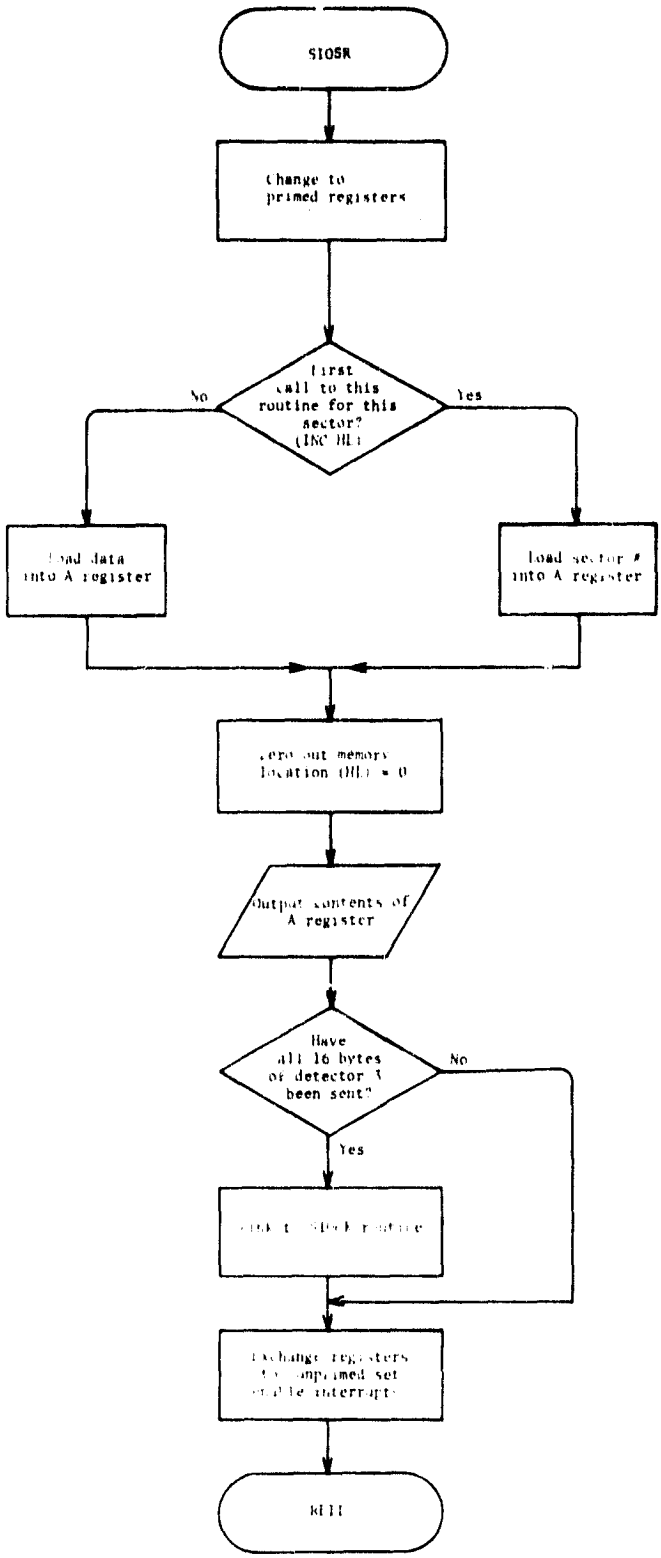


Figure III.6 Flight program lines 314-339 (Appendix II.1). This routine is responsible for transmitting the detector 3 information which includes the sector number in bin 0.

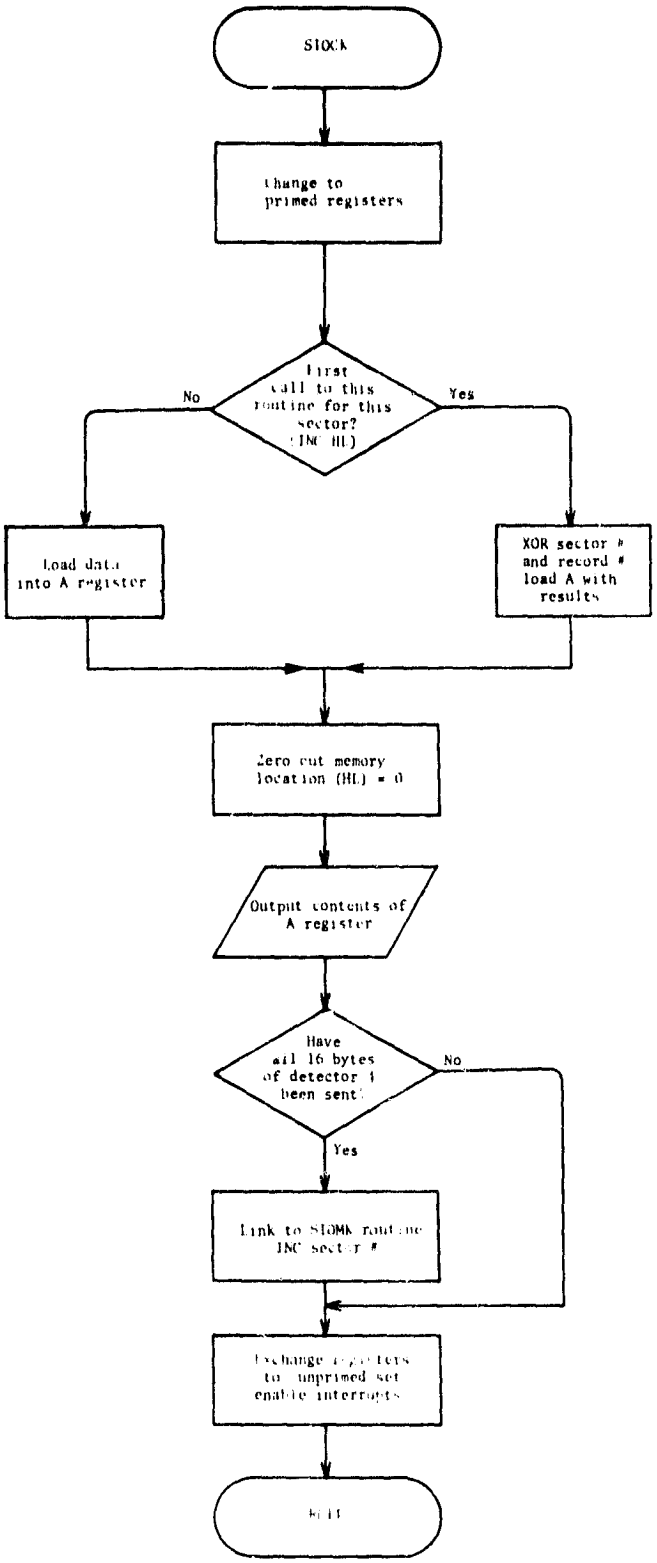


Figure III.7 Flight program lines 341-369 (Appendix II.1). This routine is responsible for transmitting the detector 4 information which includes sector XOR block number information in bin 0.

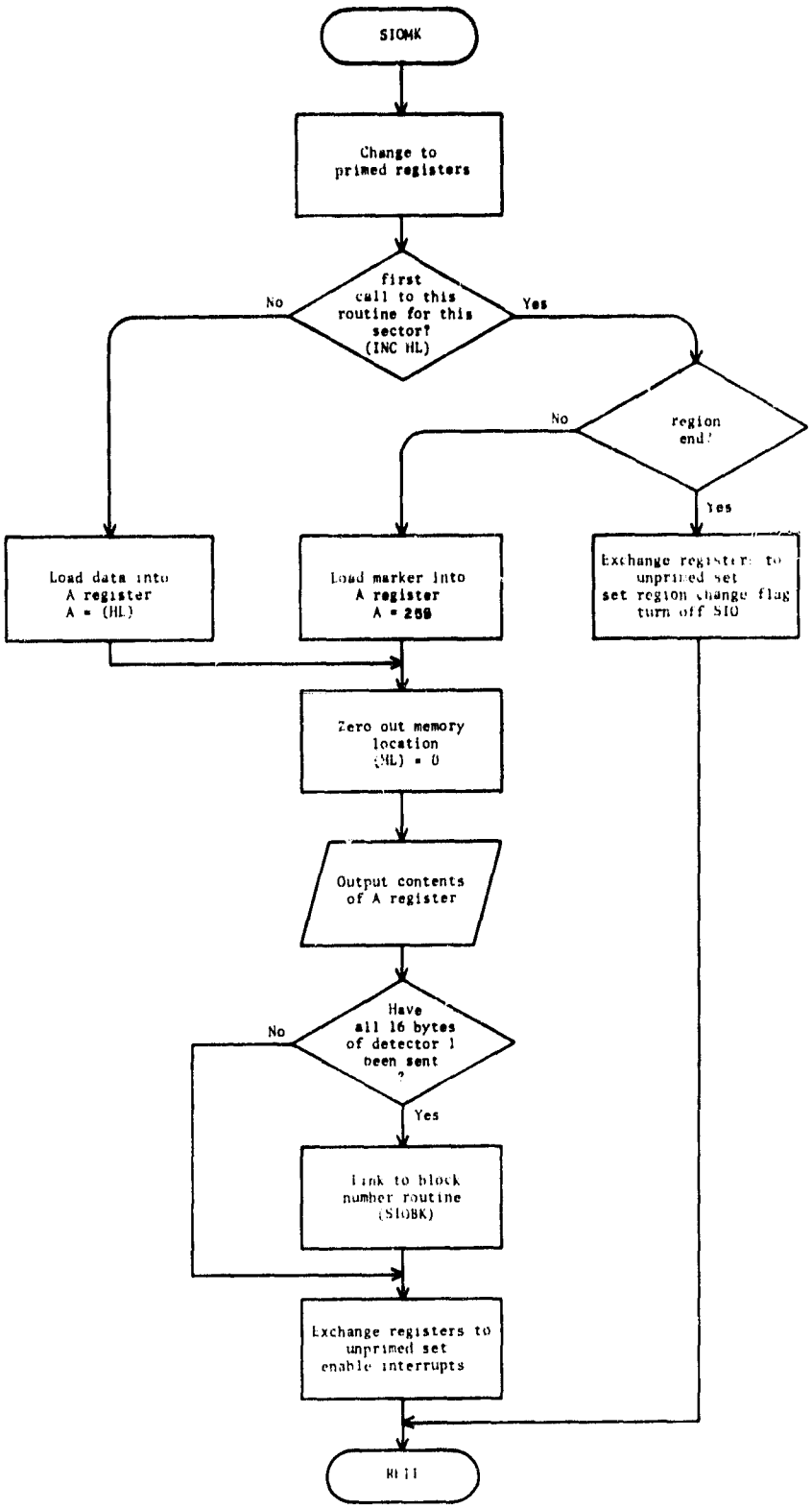


Figure III.8 Flight program lines 235-286 (Appendix II.1). This routine transmits the detector 1 information which includes a marker pulse of 255 in energy bin 0. At the end of each block of 2K bytes, this routine turns off the SIO until the next NMI occurs.

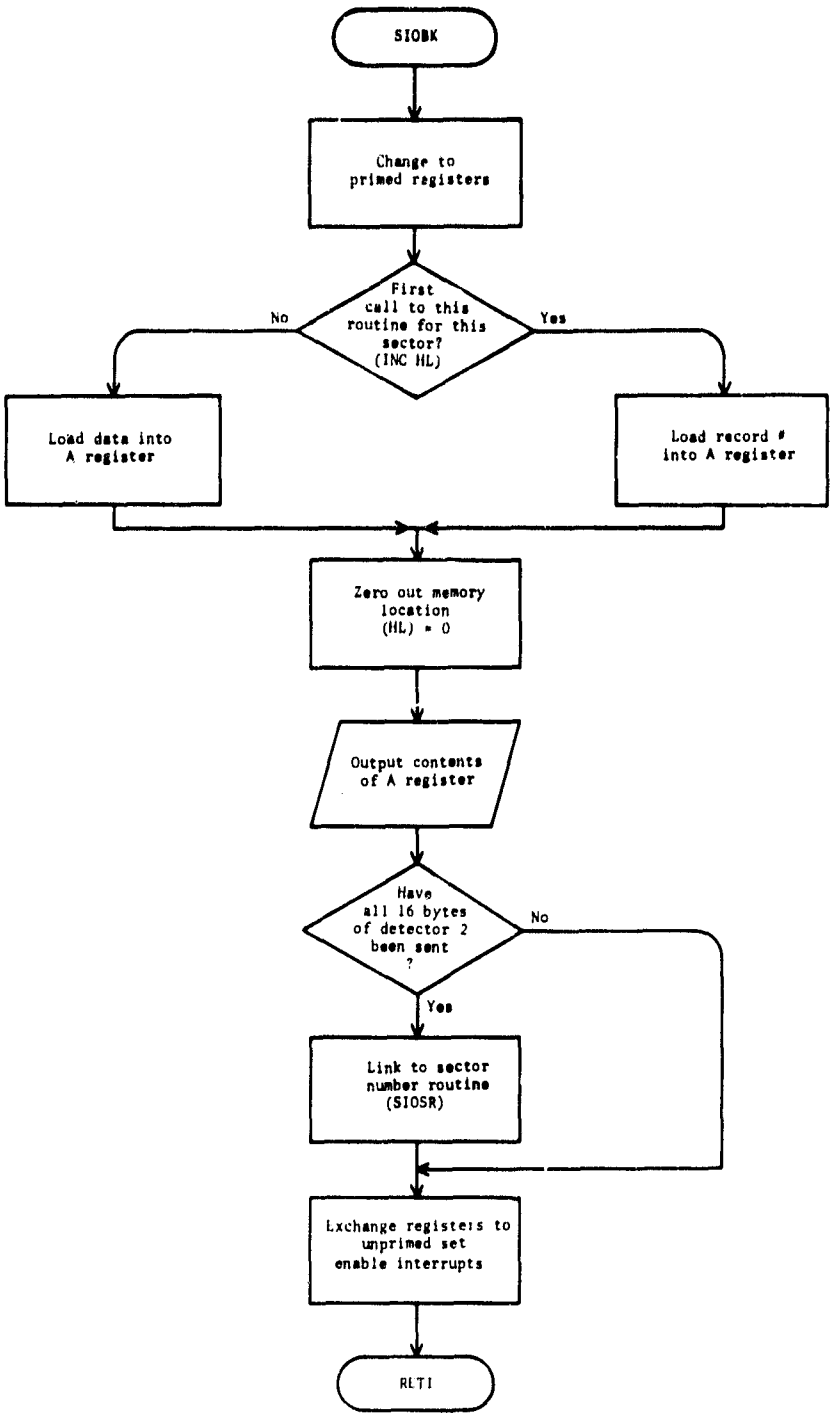


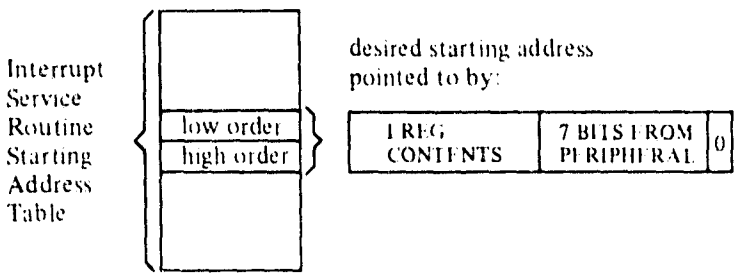
Figure III.9 Flight program lines 287-312 (Appendix II.1). This routine transmits the detector 2 information which includes the block number in energy bin 0. Each block of 2K bytes has an unique block number (0-255) associated with it.

APPENDIX IV. Z80 vectored interrupt description
[*MOSTEK Z80 Technical Manual*, 1977].

MODE 2 Interrupt

This mode is the most powerful interrupt response mode. With a single 8 bit byte from the user an indirect call can be made to any memory location.

With this mode the programmer maintains a table of 16 bit starting addresses for every interrupt service routine. This table may be located anywhere in memory. When an interrupt is accepted, a 16 bit pointer must be formed to obtain the desired interrupt service routine starting address from the table. The upper 8 bits of this pointer is formed from the contents of the I register. The I register must have been previously loaded with the desired value by the programmer, i.e. LD I, A. Note that a CPU reset clears the I register so that it is initialized to zero. The lower eight bits of the pointer must be supplied by the interrupting device. Actually, only 7 bits are required from the interrupting device as the least significant bit must be a zero. This is required since the pointer is used to get two adjacent bytes to form a complete 16 bit service routine starting address and the addresses must always start in even locations.



The first byte in the table is the least significant (low order) portion of the address. The programmer must obviously fill this table in with the desired addresses before any interrupts are to be accepted.

Note that this table can be changed at any time by the programmer (if it is stored in Read/Write Memory) to allow different peripherals to be serviced by different service routines.

Once the interrupting device supplies the lower portion of the pointer, the CPU automatically pushed the program counter onto the stack, obtaining the starting address from the table and does a jump to this address. This

mode of response requires 19 clock periods to complete (7 to fetch the lower 8 bits from the interrupting device, 6 to save the program counter, and 6 to obtain the jump address.)

Note that the Z80 peripheral devices all include a daisy chain priority interrupt structure that automatically supplies the programmed vector to the CPU during interrupt acknowledge. Refer to the Z80-PIO, Z80-SIO and Z80-CTC manuals for details.

APPENDIX V. Schematic diagrams

V.1 *CPU Board*

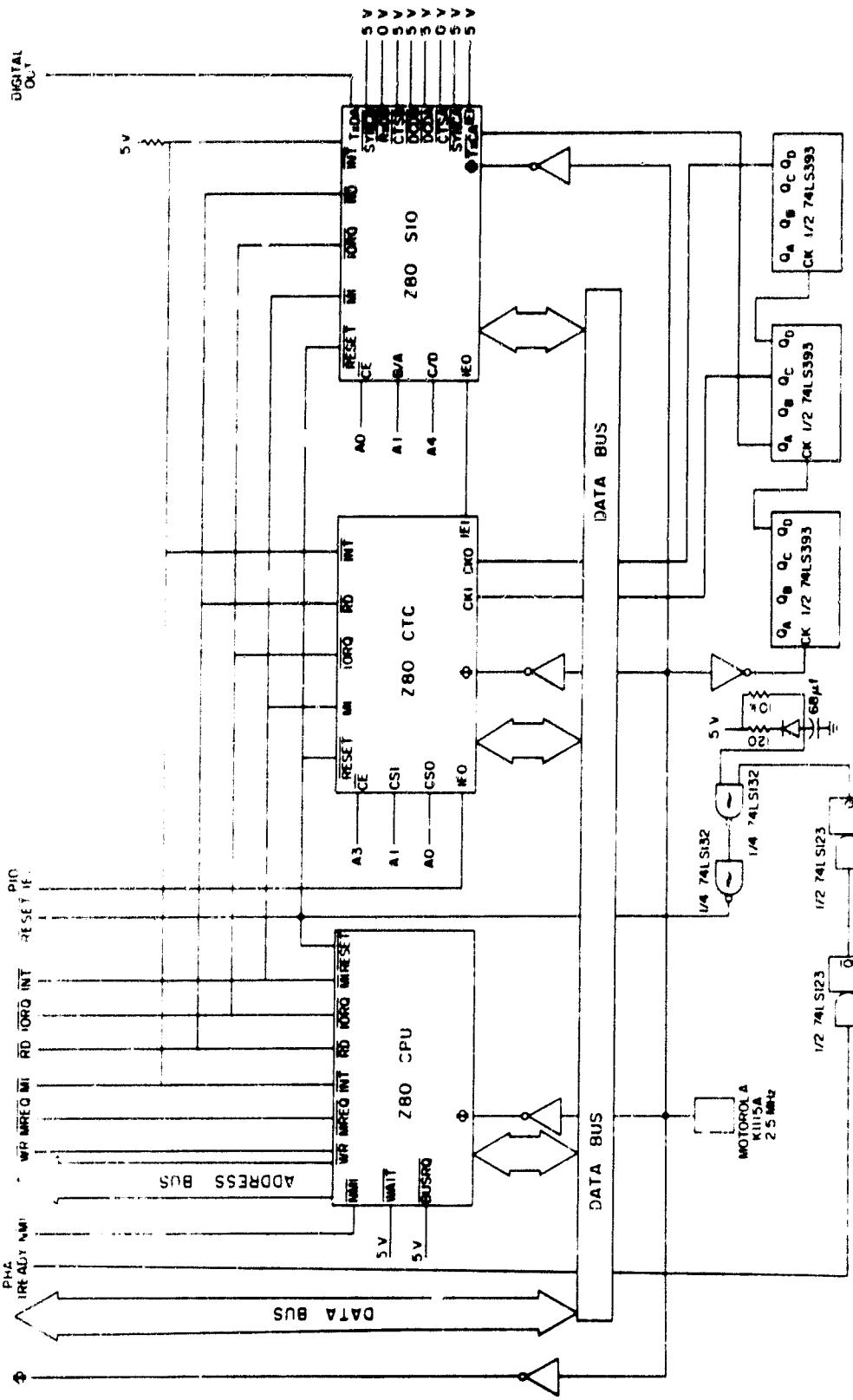


Figure V.1 The Z80 CPU is the heart of the microprocessor system and is responsible for the execution of the instructions in memory. The CTC handles all the timing involved in dividing the rocket's spin into azimuth bins. The SIO transmits the results of the experiment to the rocket telemetry system. The 74LS393's generate the proper clock frequencies for the CTC and SIO. The resistor, diode and capacitor combination will reset the system upon power-up. The 74LS123's make up the automatic reset circuit described in Figure 2.23.

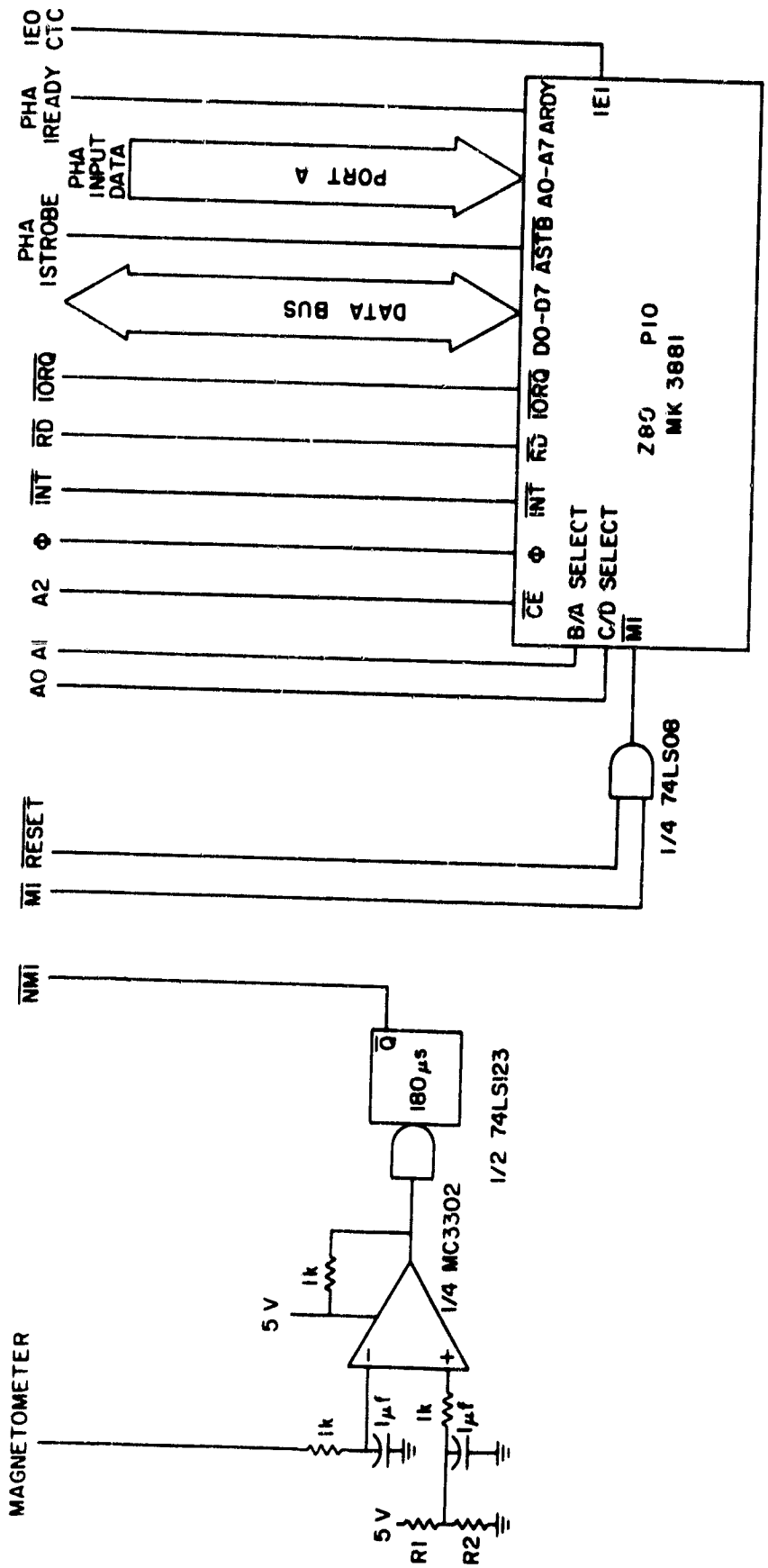


Figure V.2 The magnetometer pulse generator (left side) generates an NMI pulse at every "zero crossing" of the magnetometer as shown in Figure 3.2. The PIO port coordinates communication between the PHA board and the microprocessor, and inputs detector and pulse height information approximately once every 50 μs.

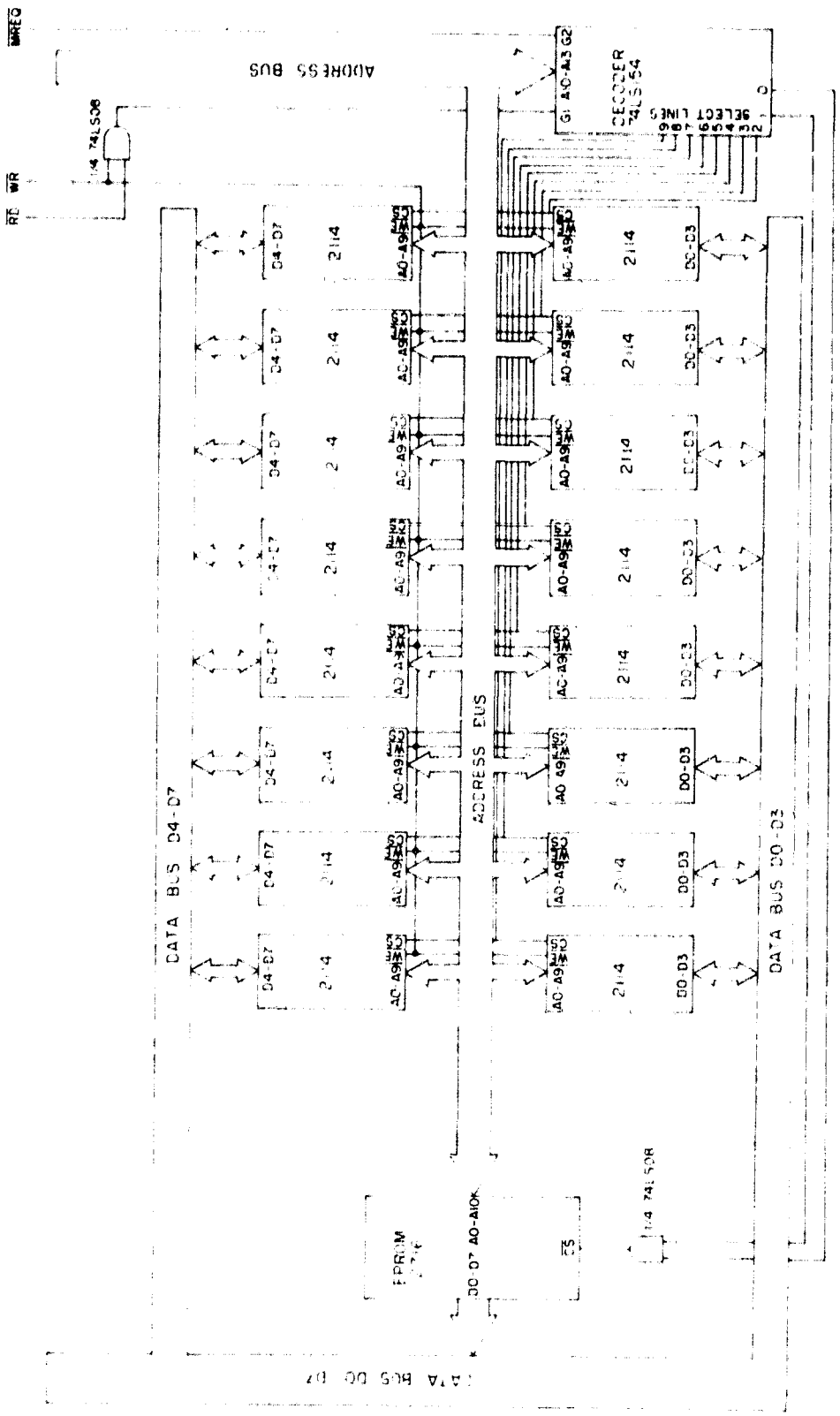


Figure V.3 The operating system software (see Appendix II.1) resides in the EPROM while the data is manipulated in the 2114 RAM chips. The decoder selects which area of memory is to be active at any given time. The logical organization of the microprocessor memory area is shown in Figure 2.19.

V.4 *Serial-to-Analog Converter*

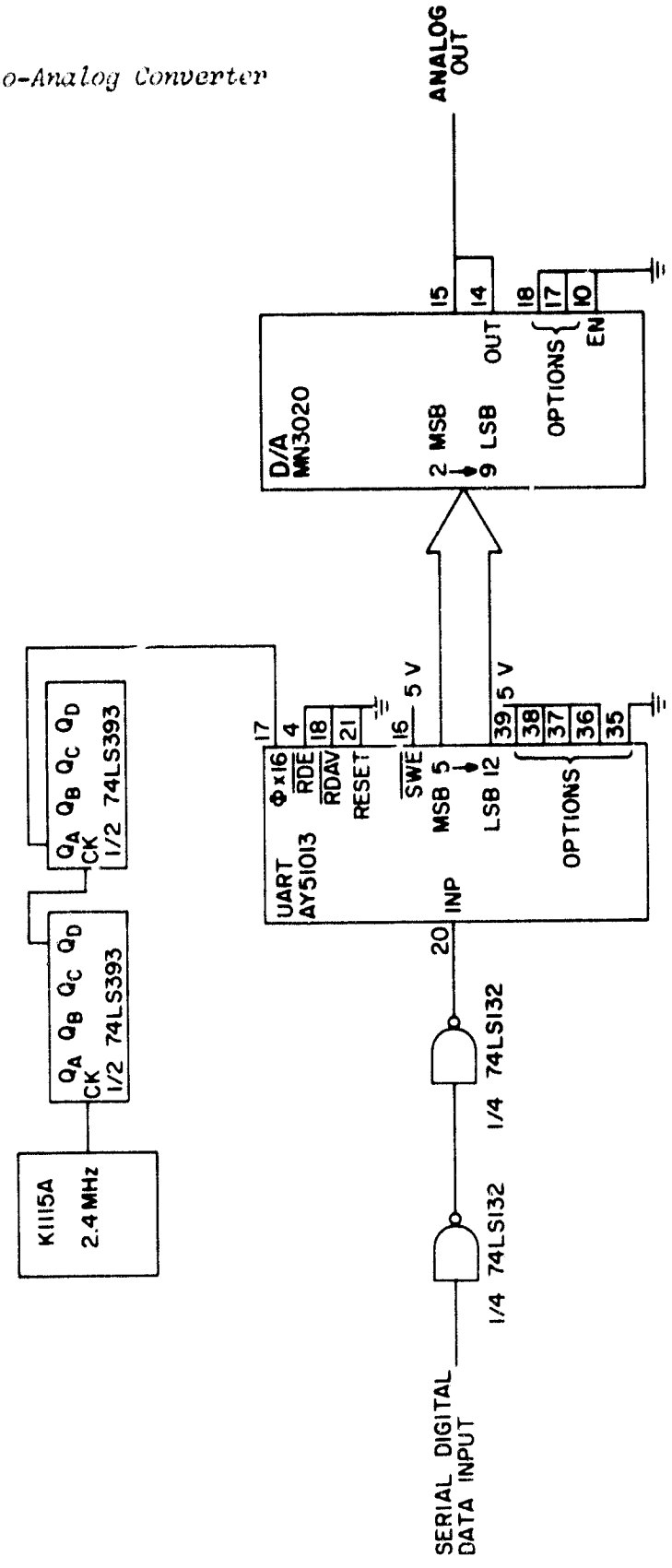


Figure V.4 The UART is used to decode the serial digital data stream coming from the microprocessor and supplies the 8-bit words in parallel to the D/A converter. The analog output (Figure 6.9 is an example) can be used to monitor the particle experiment.

REFERENCES

Barbier, D. [1958], The auroral activity at low latitudes, *Ann. Geophys.*, *14*, 334-355.

Chappell, C. R., K. K. Harris, and G. W. Sharp [1971], OGO 5 measurements of the plasmasphere during observations of stable auroral red arcs, *J. Geophys. Res.*, *81*, 608-616.

Cole, K. D. [1965], Stable auroral red arcs, sinks for energy of D_{ST} main phase, *J. Geophys. Res.*, *70*, 1689-1706.

Cole, K. D. [1975], Coulomb collisions of ring current particles--Indirect source of heat for the ionosphere, *Rep. X-621-75-108*, Goddard Space Flight Center, Greenbelt, MD.

Cornwall, J. M., F. V. Coroniti, and R. M. Thorne [1971], Unified theory of SAR arc formation at the plasmapause, *J. Geophys. Res.*, *76*, 4428-4445.

Dalgarno, A. and J. C. G. Walker [1964], Red line of atomic oxygen in the day airglow, *J. Atmos. Sci.*, *21*, 463-474.

Davis, L. L., L. G. Smith and H. D. Voss [1979], A rocket-borne data-manipulation experiment using a microprocessor, *Aeron. Rep. No. 84*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Evans, J. V. [1970], *F* region heating observed during the main phase of magnetic storms, *J. Geophys. Res.*, *75*, 4815-4823.

Hasegawa, A. and L. Chen [1975], Kinetic process of plasma heating due to Alfven wave excitation, *Phys. Rev. Lett.*, 370.

Hasegawa, A. and K. Mima [1978], Anomalous transport produced by kinetic Alfven wave turbulence, *J. Geophys. Res.*, *83*, 1117-1123.

Hernandez, G. [1972], Spectroscopic studies of the arc of March 8-9, 1970, *Planet. Space Sci.*, *20*, 1309-1321.

Hoch, R. J. and K. C. Clark [1970], Recent occurrences of stable auroral red arcs, *J. Geophys. Res.*, *75*, 2511-2515.

Hultqvist, B., W. Riedler and H. Borg [1976], Ring current protons in the upper atmosphere within the plasmapause, *Planet. Space Sci.*, *24*, 783-797.

King, G. A. M., and F. E. Roach [1961], Relationship between red auroral arcs and ionospheric recombination, *J. Res. Nat. Bur. Stand., Sect. D 65*, 129-135.

Lundblad, J. A., and F. Soraas [1978], Proton observations supporting the ion-cyclotron wave heating theory of SAR arc formation, *Planet. Space Sci.*, *26*, 245-254.

Megill, L. R., and N. P. Carleton [1964], Excitation by local electric fields in the aurora and airglow, *J. Geophys. Res.*, *69*, 101-122.

Mikhailovskii, A. B. [1967], Oscillations of an inhomogeneous plasma, in *Review of Plasma Physics*, *3*, 159, edited by M. A. Leontovich, Consultants Bureau, New York.

MOSTEK Z80 Technical Manual [copyright 1977], MK 3880 Central Processing Unit, MOSTEK Corp, Texas.

Nagy, A. F., W. B. Hanson, R. J. Hoch, and T. L. Aggson [1972], Satellite and ground-based observations of a red arc, *J. Geophys. Res.*, *77*, 3613-3617.

Okuda, M., T. Old, and J. S. Kim [1971], Midlatitude auroral arcs of 6300 Å (*OI*) and the concurrent ionospheric current system, *Radio Sci.*, *6*, 887-891.

Reed, E. I., and J. E. Blamont [1968], OGO 4 observations of the September 1970 M arc, *EOS Trans. AGU*, *49*, 731.

Rees, M. H. [1961], Excitation of high altitude red auroral arcs, *Planet. Space Sci.*, *8*, 59-67.

Rees, M. H., and S. J. Akasofu [1963], On the association between subvisual red arcs and the D_{st} (*H*) decrease, *Planet. Space Sci.*, *11*, 105-107.

Rees, M. H., and R. G. Roble [1975], Observations and theory of the formation of stable auroral red arcs, *Rev. Geophys. Space Phys.*, *13*, 201-242.

Roach, F. E., and J. R. Roach [1963], Stable 6300 Å auroral arcs in mid-latitudes, *Planet. Space Sci.*, *11*, 523-545.

Roble, R. G., R. B. Norton, J. A. Findlay, and E. Marovich [1971], Calculated and observed features of stable auroral red arcs during three geomagnetic storms, *J. Geophys. Res.*, *76*, 7648-7662.

Shepherd, G. G., L. L. Cogger, and J. R. Burrows [1976], Mid-latitude auroras and SAR arcs observed from the ISIS 2 spacecraft during the August 1972 geomagnetic storm, *J. Geophys. Res.*, *81*, 4597-4602.

Smith, L. G. in Edwards, B., Editor [1979], Research in Aeronomy April 1 - September 30, 1979, *Prog. Rep. No. 79-2*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Smith, P. H., and R. A. Hoffman [1973], Ring current particle distributions during the magnetic storms of December 16-18, 1971, *J. Geophys. Res.*, *78*, 4731-4738.

Voss, H. D., and L. G. Smith [1977], Energetic particles and ionization in the nighttime middle and low latitude ionosphere, *Aeron. Rep. No. 78*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.

Voss, H. D., and L. G. Smith [1979], Nighttime ionization by energetic particles at Wallops Island in the altitude region 120 to 200 km, *Geophys. Res. Lett.*, *6*, 93-96.

Voss, H. D., L. G. Smith, and F. M. Braswell [1979], Rocket measurements of energetic particles in the midlatitude precipitation zone, *Space Res.*, *20*, 149-152.

Williams, D. J., T. A. Fritz, and A. Konradi [1973], Observation of proton spectra ($1.0 \leq Ep \leq 300$ keV) and fluxes at the plasmapause, *J. Geophys. Res.*, *78*, 4751-4755.

Williams, D. J., G. Hernandez, and L. R. Lyons [1976], Simultaneous observations of the proton ring current and stable auroral red arcs, *J. Geophys. Res.*, *81*, 608-616.

Zimmerman, R. K., Jr., and L. G. Smith [1980], Rocket measurements of electron temperature in the E region, *Aeron. Rep. No. 92*, Aeron. Lab., Dep. Elec. Eng., Univ. Ill., Urbana-Champaign.