

NASA CR 73211

A STUDY TO DETERMINE AN EFFICIENT DATA FORMAT
AND DATA SYSTEM FOR A LIGHTWEIGHT DEEP SPACE PROBE

W. Higgins
T. Baker
S. Bigelow
E. Cohler
J. Glass

Distribution of this report is provided in the interest of
information exchange. Responsibility for the contents
resides in the author or organization that prepared it.

February 1968

Prepared under Contract No. NAS 2-3255
APPLIED RESEARCH LABORATORY
SYLVANIA ELECTRONIC SYSTEMS
An Operating Group of Sylvania Electric Products Inc.
40 Sylvan Road, Waltham, Massachusetts 02154

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

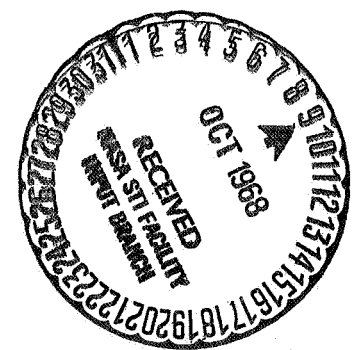
GPO PRICE \$ _____
CSFTI PRICE(S) \$ _____

Hard copy (HC) _____
Microfiche (MF) _____
v 65

N 68-35627

SESSION NUMBER	(THRU)
<u>248</u>	<u>1</u>
(PAGES)	(CODE)
<u>CR-73211</u>	<u>08</u>
(NASA CR OR TMX OR AD NUMBER)	(CATEGORY)

FACILITY FOR



A STUDY TO DETERMINE AN EFFICIENT DATA FORMAT
AND DATA SYSTEM FOR A LIGHTWEIGHT DEEP SPACE PROBE

W. Higgins
T. Baker
S. Bigelow
E. Cohler
J. Glass

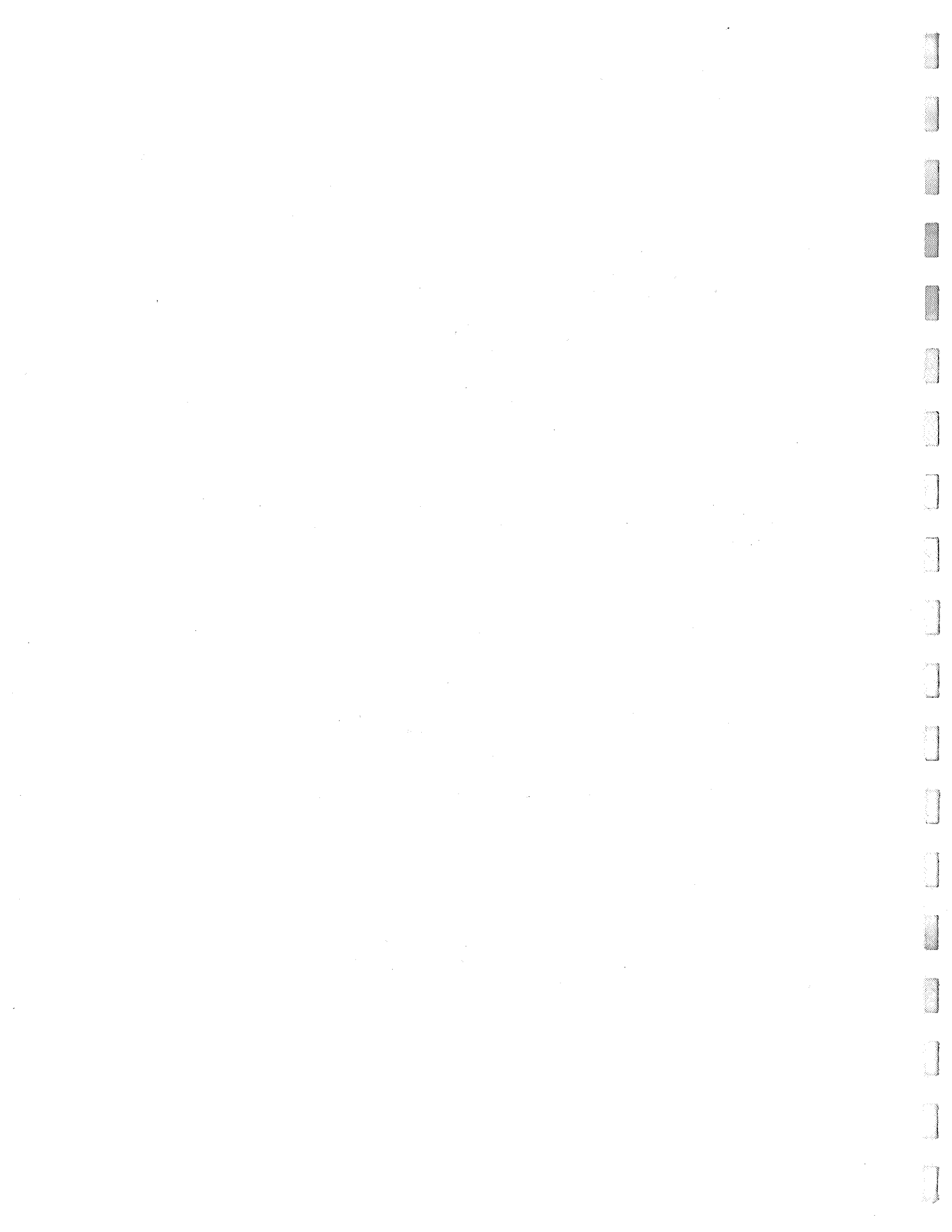
Distribution of this report is provided in the interest of
information exchange. Responsibility for the contents
resides in the author or organization that prepared it.

February 1968

Prepared under Contract No. NAS 2-3255
APPLIED RESEARCH LABORATORY
SYLVANIA ELECTRONIC SYSTEMS
An Operating Group of Sylvania Electric Products Inc.
40 Sylvan Road, Waltham, Massachusetts 02154

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION



ABSTRACT

Contract No. NAS2-3255 is a "Study to Determine an Efficient Data Format and Data System for a Lightweight Deep-Space Probe." The Applied Research Laboratory of Sylvania was awarded Contract No. NAS2-3255 in late 1965. Work began on January 1, 1966. This report covers the first phase of work performed under this contract. At the present time, the detailed logical design for the unique Multiple Pool Processor and Computer (MULTIPAC) is being performed.

Pioneer probes are aimed at mapping the radiation and particle fields of the solar corona. The central data system design must accommodate different instrument complements as mission assignments and emphasis change. Telemetry channel capacity is restricted by extreme range and practical limitations on antenna gain and transmitted power. Adaptive data sampling and telemetry formats, and sophisticated processing algorithms, which the experimenters have the option to use or not, are required to make efficient use of the available telemetry channel capacity.

The central data system requires a stored program for mission-to-mission flexibility, and ultra-reliability to meet mission effectiveness standards. Ultra-reliability is obtained through the "pooling" of logical units; the failure of one unit does not disable the computer. A diagnostic program exercises the "on-board" data system. An "on-ground" computer program isolates the fault from telemetry information and reprograms the "on-board" computer around the fault via the command link.

A system simulation model for the central data system of a scientific space probe has been designed and programmed. The goal has been to define the basic and ancillary services that the central computer can provide the experimenters. The simulation program has forced the designers to consider all aspects of the system in minute detail. The instrument complement assumed for the simulation consists of a triaxial magnetometer, cosmic ray telescope, neutron detector, radio propagation experiment, plasma probe, micrometeorite detector, and a VLF experiment, which measures the temporal correlation of the very low frequency electric and magnetic fields. Engineering data inputs are simulated and processed by the simulation program.

ABSTRACT - continued

The data sampling requirements and operational modes of the experiment are known precisely. The interface between the central computer and the experiments is influenced by instrument design and postulated solar phenomena. Simulation model ground commands specify the instrument operating modes, execution of specific processing algorithms, priority telemetry subcom, telemetry rates and formats, and parameter values that exercise the simulation model. The optional processing subroutines include quantiles, histograms, spectral analysis, digital filtering, mean and variance, and zero order prediction.

Efficient telemetry formats are provided in the simulation model. One engineering, two fixed science, and two variable science formats are available. The variable science formats adapt automatically to experiment failure or shutdown so as to maintain efficient use of the available telemetry channel capacity. The efficiency of the simulation model formats compares favorably with the formats of scientific probes that are currently operational.

An analysis and evaluation of the simulation runs provide keen insight into the computer requirements for storage, machine work size, instruction word format, instruction list, memory cycle time, arithmetic operations, execute time, data line servicing, input-output, priority interrupt, indexing, and masking. The analysis specifies the frequency with which particular instructions are executed.

The detailed requirements obtained by careful consideration of the overall system requirements and evaluation of the simulation model results are being incorporated into the logical design of the central computer. Ultra-reliability through the "pooling" of logical units is a design goal. The central computer specified in the preliminary design compares favorably in terms of weight, power, reliability, and flexibility to current operational systems.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION AND SUMMARY	1-1
	1.1 PROBLEM STATEMENT	1-1
	1.1.1 Mission Objectives and Requirements	1-1
	1.1.2 Data Input Requirements	1-2
	1.1.3 Data Output Requirements	1-4
	1.1.4 Central Data System Requirements	1-4
	1.1.5 Design Objectives and Constraints	1-5
	1.2 APPROACH	1-6
	1.2.1 Central Data System--Hardware	1-6
	1.2.2 Central Data System--Software	1-9
	1.2.3 Central Data System Simulation	1-21
	1.3 CONCLUSIONS	1-25
2	CENTRAL DATA SYSTEM INTERFACE CONSIDERATIONS	2-1
	2.1 COSMIC RAY EXPERIMENT	2-6
	2.2 MICROMETEORITE DETECTOR	2-7
	2.3 NEUTRON DETECTOR	2-7
	2.4 TRIAXIAL FLUXGATE MAGNETOMETER	2-8
	2.5 RADIO PROPAGATION EXPERIMENT	2-8
	2.6 VLF EXPERIMENT	2-8
	2.7 ARC PLASMA INSTRUMENT	2-9
	2.8 SPACECRAFT OPERATIONS AND CONTROL SUBSYSTEM	2-9
	2.9 TELEMETRY SUBSYSTEM	2-9
	2.10 INTERFACE CONSIDERATIONS SUMMARY	2-10
3	MULTIPAC COMPUTER FOR CENTRAL DATA SYSTEM	3-1
	3.1 SYSTEM DESCRIPTION	3-2
	3.2 LOGIC UNIT	3-4
	3.3 INSTRUCTION TIMING	3-6
	3.4 MEMORY FOR SPACE DATA PROBE	3-9
	3.4.1 Power Major Problem	3-9
	3.4.2 MOS Device a Competing Technique	3-9
	3.4.3 Summary	3-11

TABLE OF CONTENTS - continued

<u>Section</u>		<u>Page</u>
	3.5 INTERFACES	3-12
	3.6 INPUT-OUTPUT PROCEDURE	3-15
	3.7 COMMAND LINK	3-15
	3.8 RELIABILITY CONSIDERATIONS	3-17
	3.9 ESTIMATED SIZE OF THE CDS	3-18
	3.10 CDS DIAGNOSTIC AND CORRECTION PHILOSOPHY	3-18
	3.11 RECOMMENDATIONS FOR FUTURE WORK	3-21
	3.12 SIMPLIFIED LOGIC UNIT	3-21
	3.13 REFERENCES	3-25
4	CDS SYSTEM TEST, CHECKOUT, AND EVALUATION	4-1
	4.1 TEST AND CHECKOUT	4-1
	4.2 CAPABILITIES AND CHARACTERISTICS OF THE MULTIPAC CDS	4-2
5	EFFICIENT FORMATTING AND FORMAT ASSEMBLY	5-1
	5.1 FORMAT DESCRIPTION	5-1
	5.1.1 Fixed Engineering Format	5-2
	5.1.2 Fixed Science Format	5-2
	5.1.3 Variable Science Formats	5-5
	5.2 FORMAT EFFICIENCY ANALYSIS	5-11
	5.3 FORMAT ASSEMBLY SUBROUTINE	5-14
6	ACCEPTANCE TEST PROCEDURE FOR SYSTEM SIMULATION MODEL	6-1
7	SYSTEM SIMULATION ANALYSIS AND EVALUATION	7-1
	7.1 PROGRAM MAP	7-1
	7.2 EVALUATION PROCEDURES	7-3
	7.3 SYSTEM SIMULATION RESULTS AND CONCLUSIONS	7-8

TABLE OF CONTENTS - continued

<u>Appendix</u>		<u>Page</u>
I	SYSTEM SIMULATION MODEL	I-1
	I.1 DATA SIMULATION SUBROUTINE	I-1
	I.1.1 Temporal Sampling	I-1
	I.1.2 Data Generation	I-4
	I.2 SIMULATION MODEL SUMMARY	I-9
II	ACCEPTANCE TEST PROCEDURE FOR SYSTEM SIMULATION MODEL	II-1
	II.1 FORMAT CHECKOUT PROCEDURE	II-2
	II.1.1 Fixed Engineering Format (FENGF)	II-2
	II.1.2 Print Out One Frame of FENGF	II-2
	II.1.3 Change Command Table Values	II-3
	II.1.4 Print Out One Frame of FENGF	II-3
	II.2 FIXED HIGH BIT RATE FORMAT (FHBRF)	II-3
	II.2.1 Print Out One Frame of FHBRF	II-4
	II.2.2 Change Command Table of FHBRF	II-4
	II.3 FIXED LOW BIT RATE FORMAT (FLBRF)	II-4
	II.3.1 FLBRF (All Instruments Off)	II-5
	II.4 VARIABLE LOW BIT RATE FORMAT (VLBRF)	II-5
	II.5 VARIABLE HIGH BIT RATE FORMAT (VHBRF)	II-6
	II.6 SUMMARY OF FORMAT CHECKS	II-6
	II.7 VARIABLE FORMAT COMMAND CHECKOUTS	II-7
	II.8 PROCESSING ALGORITHM CHECK PROCEDURES	II-8
	II.8.1 Simple Processing Algorithms	II-8
	II.8.2 Digital Filtering Subroutine	II-8
	II.8.3 Spectral Analysis Subroutine	II-9
	II.8.4 Quantiles of Histograms Subroutine	II-9
	II.8.5 Mean and Variance Subroutine	II-9
	II.8.6 Checkout Processing Algorithms in System Context	II-9
	II.9 TEST ACCEPTANCE PROCEDURE	II-9
	II.9.1 Engineering Format (FENGF)	II-10
	II.9.2 Fixed Low Bit Rate Science Format (FLBRF)	II-10
	II.9.3 Fixed High Bit Rate Science Format (FHBRF)	II-16

TABLE OF CONTENTS - continued

<u>Appendix</u>		<u>Page</u>
	II.9.4 Variable High Bit Rate Science Format (VHBRF)	II-16
	II.9.5 Variable Low Bit Rate Science Format (VLBRF)	II-22
	II.9.6 Storage Mode Format (SMF)	II-22
III	PROCESSING CONSIDERATIONS	III-1
	III.1 HISTOGRAMS AND QUANTILES	III-1
	III.1.1 Use of Quantiles for Data Compression	III-2
	III.1.2 Properties of Quantiles	III-2
	III.1.3 Estimation of Population Parameters Using Sample Quantiles	III-6
	III.1.4 Hypothesis Testing Using Quantiles	III-22
	III.2 DIGITAL FILTERING	III-25
	III.2.1 Digital Filter Design Methods	III-25
	III.2.2 Design Example	III-31
	III.3 SPECTRAL ANALYSIS	III-33
	III.3.1 Resolution and Variability of Spectral Estimates	III-35
	III.3.2 Indirect Method for Obtaining Spectral Estimates of Sampled Data	III-38
	III.4 DATA PROCESSING CONSIDERATIONS FOR PIONEER EXPERIMENTS	III-39
	III.4.1 Neutron Detector	III-40
	III.4.2 VLF Experiment	III-41
	III.4.3 Cosmic Ray Experiment	III-42
	III.4.4 Plasma Probe	III-42
	III.4.5 Triaxial Fluxgate Magnetometer	III-43
	III.4.6 Stanford Radio Propagation Experiment	III-44
	III.4.7 Conclusions	III-44
	III.4.8 References	III-46

TABLE OF CONTENTS - continued

Appendix

	<u>Page</u>
IV CDS INPUT/OUTPUT TIMING REQUIREMENTS	IV-1
IV.1 EXPERIMENT AND SUBSYSTEM DEFINITIONS	IV-1
IV.1.1 Magnetometer	IV-1
IV.1.2 Cosmic Ray Telescope	IV-4
IV.1.3 Plasma Probe	IV-8
IV.1.4 Radio Propagation	IV-8
IV.1.5 Neutron Detector	IV-9
IV.1.6 VLF Experiment	IV-10
IV.1.7 Micrometeorite Detector	IV-11
IV.1.8 Science Line Characteristics Summary	IV-12
IV.1.9 Spacecraft Operations and Control Subsystems	IV-12
IV.1.10 Total Servicing Time Requirements for Experiments and Subsystems	IV-19
IV.2 MULTIPLEXER CONFIGURATION	IV-21
IV.3 COMMAND SWITCH CONFIGURATION	IV-28
IV.4 MICROSTRUCTURE OF SCHEDULING AND SAMPLING SUBROUTINE	IV-30
IV.5 INPUT/OUTPUT INTERFACE POSSIBILITIES	IV-40

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1-1	Ultra Reliable Computer	1-8
1-2	Engineering Format	1-14
1-3	FLBR Science Format with Parity	1-15
1-4	Block Formats for Fixed Bit Rate Telemetry	1-17
1-5	Variable LBR Telemetry Format with Parity	1-18
1-6	Program Routine Executed Each Spacecraft Revolution	1-22
2-1	Overall System Block Diagram	2-3
2-2	Interface Between CDS and "On-Board" Experiments and S/C Support Equipment	2-5
3-1	MULTIPAC Instructions	3-7
3-2	Timing Diagram of Three Consecutive Instructions	3-8
3-3	Interrupt Procedure	3-16
5-1	Fixed Engineering Format	5-3
5-2	FLBR Science Format with Parity	5-4
5-3	FHBR Science Format with Parity	5-6
5-4	Variable LBR Telemetry Format with Parity	5-8
5-5	Variable HBR Telemetry Format with Parity	5-10
5-6	Comparative Efficiencies of Pioneer and CDS Science Formats	5-13
5-7	Overall Flow Through Format Assembly Subroutine	5-16
5-8	Format Assembly Simulation Program	5-19
5-9	Correspondence Between Data Block and Its BID Code	5-23
7-1	Program Routine Executed Each Spacecraft Revolution	7-2

LIST OF ILLUSTRATIONS - continued

<u>Figure</u>		<u>Page</u>
I-1	Program Routine Executed Each Spacecraft Revolution	I-10
II-1	Exhibit 1	II-12
II-2	Exhibit 2	II-14
II-3	Exhibit 3	II-18
II-4	Exhibit 4	II-20
II-5	Exhibit 5	II-25
III-1a	Cumulative Histogram for Neutron Detector 16-Channel Pulse Height Analysis-1000 Samples, 64 Levels	III-17
III-1b	Histogram for Neutron Detector 14-bit Counter-1000 Samples, 64 Levels	III-18
III-1c	Histogram for One Axis of Magnetometer-2200 Samples, 64 Levels	III-19
III-1d	Histogram for Another Axis of Magnetometer-2220 Samples, 64 Levels	III-20
IV-1	Magnetometer Input/Output	IV-1
IV-2	A/PW Converter	IV-2
IV-3	Digital Representation of Sensor A Analog Voltage	IV-2
IV-4	Command Flow	IV-3
IV-5	CR Telescope Input/Output	IV-4
IV-6	Moving Counter Contents to CDS	IV-5
IV-7	Plasma Probe Input/Output	IV-8
IV-8	Radio Propagation Experiment Input/Output	IV-9
IV-9	Neutron Detector Input/Output	IV-9
IV-10	VLF Experiment Input/Output	IV-10

LIST OF ILLUSTRATIONS - continued

<u>Figure</u>		<u>Page</u>
IV-11	Micrometeorite Detector Input/Output	IV-11
IV-12	Orientation Subsystem Input/Output	IV-12
IV-13	Power Subsystem Input/Output	IV-15
IV-14	RF Subsystem Input/Output	IV-16
IV-15	CDS Engineering Subsystem Input/Output	IV-17
IV-16	S/C Support Subsystem Input/Output	IV-18
IV-17	Other Subsystem Input/Output	IV-19
IV-18	Multiplexer #1 Input/Output Interface	IV-26
IV-19	Command and Switch #1	IV-28
IV-20	Overall Interface	IV-29
IV-21	(128 → 512) and (8 → 64) Bit/Rev Sampling Formats	IV-34

LIST OF TABLES

<u>Table</u>		<u>Page</u>
3-1	ADD α INSTRUCTION	3-5
3-2	APPROXIMATE MAXIMUM LINE SERVICING FOR CONTROL COMPUTER	3-14
3-3	ESTIMATED FLATPACK COUNT FOR 16-BIT VERSION	3-19
4-1	MULTIPAC VS. PIONEER COMPARISON	4-4
4-2	SOLAR PROBE CDU RELIABILITY	4-5
4-3	FORTRAN PROGRAM FOR CALCULATING MULTIPAC CDU RELIABILITY	4-6
4-4	EXAMPLES OF SPEED AND POWER FOR VARIOUS TYPES OF MACHINE ORGANIZATION	4-8
5-1	TELEMETRY FORMAT EFFICIENCY FOR VARIOUS CONDITIONS	5-12
7-1	EXECUTION TIME PER PROGRAM SEGMENT IN MILLISECONDS	7-4
7-2	CALL ROUTINE/SECOND	7-5
7-3	INSTRUCTIONS EXECUTED FOR EFFICIENT CALL SUBROUTINES	7-5
7-4	INSTRUCTIONS EMPLOYED BY VARIABLE AND FIXED SCIENCE SAMPLING AND PACKING SUBROUTINES	7-6
I-1	SUNLINE CORRESPONDING TO S/C ANGULAR COUNT ((S/C)AC)	I-2
I-2	PRIORITY ORDER FOR SAMPLING WHEN MULTIPLE SAMPLES ARE REQUIRED FOR A (S/C) AC	I-3
I-3	DATA INPUT SPECIFICATIONS	I-4
I-4	INPUT FORMAT DEFINITION WITH ASSUMED PARAMETERS	I-5
I-5	STORAGE LOCATION CORRESPONDENCES	I-8
II-1	SYSTEM SIMULATION COMMAND TABLE	II-1
III-1		III-13
III-2	RESULTS OF ESTIMATES USING QUANTILES BASED ON HISTOGRAMS IN TABLE III-1	III-15

LIST OF TABLES - continued

<u>Table</u>		<u>Page</u>
III-3	MEAN, VARIANCE, AND STANDARD DEVIATION FOR THE HISTOGRAMS OF FIGURE III-1. RESULTS ARE GIVEN FOR ENTIRE HISTOGRAM AND FOR 4, 6, 8, and 10 QUANTILES	III-16
III-4	VALUE OF FILTER OUTPUT AT 150 th SAMPLING INSTANT FOR CONSTANT VALUE INPUT = 10.0	III-34
IV-1	SCIENCE LINE DESIGNATIONS AND CHARACTERISTICS	IV-13
IV-2	SCIENTIFIC INSTRUMENTS INPUT/OUTPUT (MAXIMUM TIME) SERVICING REQUIREMENTS (milliseconds/second)	IV-20
IV-3	INTERFACE LINES	IV-21
IV-4	MULTIPLEXER LINE ASSIGNMENTS	IV-23
IV-5	COMMAND LINES AND PERTINENT INSTRUMENT/SYSTEM	IV-27
IV-6	TYPICAL VALUES OF ΔS (ASSUMING $\delta = 1$)	IV-31
IV-7	SUNLINE CORRESPONDING TO SUN-PULSE CLOCK	IV-32
IV-8	SAMPLING PRIORITY	IV-35
IV-9	SAMPLING SCHEDULE	IV-36
IV-10	SERVICING TIME REQUIREMENT FOR INPUT/OUTPUT AT $TLMR \geq 512$ IN PERCENT OF TOTAL TIME	IV-40

LIST OF ABBREVIATIONS

A/D	Analog to Digital
BER	Bit Error Rate
BID	Block Identification
CDS	Central Data System
FAS	Format Assembly Subroutine
FE	Fixed Engineering
FENGF	Fixed Engineering Format
FHBR	Fixed High Bit Rate
FLBR	Fixed Low Bit Rate
HBR&P	High Bit Rate Sampling and Packing Subroutine
LB	Lower Bound
LD	Line Definition
LIR or LR	Logic Instruction Register
LSI	Large Scale Integration
LU	Logic Unit
MA	Memory Address (Register)
MD	Memory Data (Register)
OP	Operation
PC	Program Counter
R	Register
S/C	Spacecraft
SID	Subcom Identification
SMDCESW	Storage Mode Duty Cycle Execute Switch
TLM	Telemetry
UB	Upper Bound
VHBR	Variable High Bit Rate
VLBR	Variable Low Bit Rate

SECTION 1

INTRODUCTION AND SUMMARY

Scientific solar probes (such as Pioneer) are aimed at mapping the radiation and particle fields of the solar corona. The Central Data System (CDS) design must accommodate different instrument complements as mission assignments and emphasis change. Telemetry channel capacity is restricted by extreme range and practical limitations on antenna gain and transmitter power. Adaptive data sampling and telemetry formats, and sophisticated processing algorithms, which the experimenters have the option to use or not, are required to make efficient use of the available telemetry channel capacity.

The CDS requires a stored program computer for mission-to-mission flexibility and ultra-reliability to meet mission effectiveness standards. Ultra-reliability is obtained through the "pooling" of logical units in such a way that the failure of one unit does not disable the computer. A diagnostic program exercises the on-board data system. An on-ground computer program isolates the fault from telemetry information and reprograms the on-board computer around the fault via the command link.

1.1 PROBLEM STATEMENT

The general problem considered is that of designing a CDS for use on-board future unmanned solar probe vehicles. The purpose of such probes is to measure and map various physical phenomena occurring between the earth and the sun, particularly in the vicinity of the latter. In the following discussion the data handling problem is defined in terms of the objectives, requirements, and constraints which an on-board CDS must meet.

1.1.1 Mission Objectives and Requirements

The primary purpose of a solar probe vehicle is to carry sensing devices through the region of space near the sun, in order that in situ measurements of such physical phenomena as solar plasma, cosmic rays, neutrons, micrometeorites, and electric and magnetic fields may be made. The scientific data thus gathered must be telemetered to the earth in a form which permits their

interpretation and correlation. The functioning of the subsystems which make up the space vehicle must be monitored and information on their status telemetered to earth for use in checking spacecraft operation. Since environmental conditions vary widely during the course of a mission, provision must also be made for altering on-board operations by means of commands transmitted from the earth.

All of these mission-related objectives and requirements have implications for the on-board data handling system. These are examined in some detail in the following sections.

1.1.2 Data Input Requirements

The on-board data handling system must accept input data from three different classes of sources:

- 1) Scientific measurements from instruments.
- 2) Engineering status data from subsystems.
- 3) Commands from the earth.

The way in which this data is made available to the CDS depends on the particular source. Input information may be presented in the form of analog voltages, serial or parallel digital words, single bits, and pulse trains. In the latter case, the information is contained in the number of pulses per unit time, the amplitude of the pulses, or both. The input interface equipment must, therefore, be capable of accepting data in all of these forms. Furthermore, since all data sources must be sampled periodically, the CDS must control the interface equipment to select both an input source and the time of its sampling.

The data input requirements which the CDS of future solar probes will have to satisfy cannot be precisely defined. In the belief, however, that these requirements will not differ radically from those of the current Pioneer VI spacecraft, the data input requirements of this vehicle have been used to develop a model for use in this study. The model is discussed briefly below.

Scientific Instrument Inputs

The complement of scientific instruments assumed for our model consists of the following:

- 1) Micrometeorite Detector--Measures velocity, momentum, energy, and radiant of each impacting micrometeorite.
- 2) Neutron Detector--Measures flux and energy distribution of incident neutrons.
- 3) Cosmic Ray Detector--Measures flux and energy distribution of cosmic rays as a function of angular direction from the sun-line reference direction.
- 4) Plasma Probe--Measures flux and energy distribution of positively and negatively charged plasma particles as a function of their direction of arrival.
- 5) Magnetometer--Measures magnetic field components in three orthogonal directions.
- 6) Radio Propagation Experiment--Measures average electron density and electron-induced changes in polarization over the earth-spacecraft path.
- 7) VLF Experiment--Measures electric and magnetic fields and their correlations at VLF.

The data output characteristics of these instruments are sufficiently well defined to permit the detailed specification of the interface and servicing requirements which the CDS must satisfy in order to input the scientific data. This interface is shown in simplified form in Figure 2-2.

Engineering Sensor Inputs

Information concerning the status and operation of the various spacecraft subsystems is obtained by sensing temperatures, voltages, currents, and switch positions. These data are entered into the CDS for subsequent telemetering to earth and for use in the control of spacecraft functions.

The number of engineering data lines assumed in the model is 64, of which two-thirds carry analog information with from four- to six-bit precision and one-third carry bi-level switch position information.

A special purpose device which should be mentioned in the present context is the sun sensor. Its purpose is to establish an orientation reference for the spinning vehicle, which can be used to control the time at which scientific measurements are made.

Ground Command Inputs

Spacecraft operations must be changed from time-to-time during a mission. These changes are effected by transmitting coded commands from the earth. Such commands are entered after demodulation into the CDS. This inputting function must be performed without delay whenever a ground command is received.

1.1.3 Data Output Requirements

Two classes of data must be outputted by the CDS: Telemetry data for immediate or delayed transmission to earth and command data for the control of other on-board subsystems. Expected telemetry data rates are from 8 to 2048 bits/sec. The types of output commands anticipated include addressing and timing signals to the input and output multiplexers, commands to turn equipment on or off and commands to select different operating modes for the scientific instruments and support subsystems.

1.1.4 Central Data System Requirements

The CDS is required to perform a number of functions. In the discussion these have been grouped into three categories.

Data Processing

The CDS must be capable of performing data processing operations aimed at reducing redundancy or converting raw data into a more useful form. Examples of such operations are filtering, compensation for spacecraft spin, selection of maxima and minima from sets of samples, and the computation of spectral and statistical characteristics.

Data Formatting and Identification

Data samples which are to be telemetered to earth must be arranged in a known order or format prior to transmission. Additional information concerning the time at which samples were taken and the range and orientation of the instruments must be included for use in identifying and interpreting the telemetry data. Such formatting operations must be flexible enough to adapt to changing environmental and mission requirements.

Timing and Control

The CDS must accept, interpret, and execute ground commands. It must sample scientific and engineering data lines and output telemetry data and control signals in accordance with an appropriate timing sequence. A priority interrupt capability is also required in the handling of ground commands and other high priority events.

1.1.5 Design Objectives and Constraints

The CDS must satisfy all of the general requirements outlined above. Since the CDS is to be carried on a relatively small spacecraft, it should be realizable with sufficiently low size, weight, and power consumption to be practical. It is also desired that the operation of the CDS be flexible enough that it may be readily modified to accommodate changes in specific requirements during the course of a particular mission and from mission-to-mission. Such operational flexibility has the potential not only for improved data handling efficiency during a mission but also for satisfying with a single, general-purpose design the data handling requirements of widely differing missions.

An additional design constraint is that the CDS be highly reliable. Because of the central role it plays in all on-board operations, any malfunction of the CDS could render the spacecraft useless. For this reason the CDS design should not only be as reliable as it is possible to make it, but also should be configured in such a way that component and subsystem failures result in graceful degradation of its operation rather than complete system outage.

1.2 APPROACH

In view of the CDS requirements outlined in the above problem statement, particularly those relating to operational flexibility, it was felt that the best approach to the CDS design problem would be to use a stored-program digital computer. A preliminary investigation of current and foreseeable developments in digital computer hardware indicated that it was quite probable that all of the CDS design requirements could be met by such a device.

On the basis of this belief, the CDS design study was split into two main tasks; the one concerned with hardware and the other concerned with software. These two areas were investigated in parallel and the results combined to obtain the final CDS design.

The aim of the hardware study was to develop a logical organization for a stored-program computer which would be capable of performing the necessary operations and would possess the requisite reliability and closely controlled failure characteristics.

The approach taken in the software study was to work out in complete detail a computer simulation program for all CDS operations required for the chosen system model and mission. The purpose in doing this was twofold: First, it required the careful examination of a CDS operations which would be essential or highly desirable in an actual system, and second, it enabled one, by running and analyzing the simulation program, to establish the CDS storage capacity, speed, and instruction repertoire needed to satisfy typical mission requirements.

This method for designing a spacecraft CDS has proved to be quite effective. The results of the study are presented in the following sections.

1.2.1 Central Data System--Hardware

Use has been made of a unique conceptual design for a general purpose programmable CDS. A general purpose computer approach, which gives enormous flexibility, is combined with a pooling technique to allow survival after a component failure. The survival is achieved by reprogramming around the failed unit, since each unit is a member of a pool of program addressable

identical modules. Each basic module is small enough that it does not represent a large portion of the CDS.

CDS Logical Design

A computer using this pooling technique is shown in Figure 1-1. Each of the four memories satisfies one-quarter of the total memory requirement. The 32 registers, some of which are the memory address and memory data registers of the four memories, are identical. In this example, there are four independent but identical logic units each with its own set of input and output bus structures.

All the registers in the machine form a single register store. Eight bus structures are connected to the output of the registers. Each bus can select one register. Each logic unit has two of these busses connected to its two inputs, and two output busses to these same registers. Thus, there are eight busses on the output of the register store and eight busses feeding the register inputs.

The logic unit will perform the operation specified by the bits in the logic instruction register (LIR) associated with each logic unit. This logic instruction register also specifies the registers selected for the two input busses and two output busses connected to that logic unit. This logic unit can perform all necessary instructions and control operations required in the computer. For example, by specifying the proper bits in the logic instruction register, the logic unit can add two registers together and send the results to a third and fourth register. It can also perform control operations such as adding 1 to a program counter and sending it to a memory address register to fetch an instruction.

The instruction word will contain four fields of four bits each for a total instruction code of 16 bits. One of the 4-bit fields specifies one of 16 operations. The other three fields specify the registers to be operated on and where the results are to be sent.

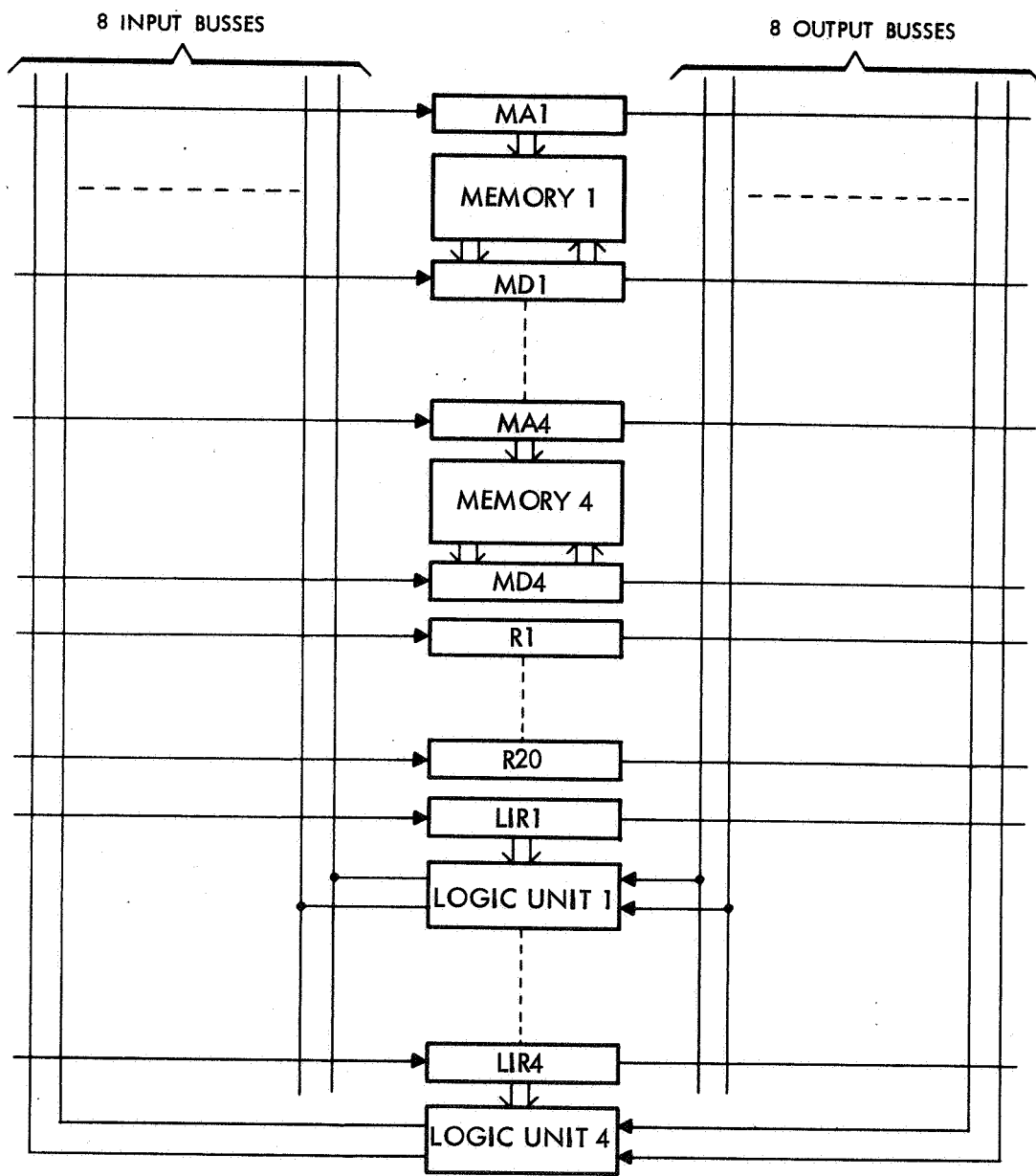


Figure 1-1. Ultra Reliable Computer

Reliability Considerations

The reliability of this design is obtained through the use of pooling techniques instead of equipment duplication. Recovery after a failure is accomplished through reprogramming. The maximum effort of such a failure will be to reduce the computational rate of the system. For real-time applications, one can either have an extra logic unit, an extra register, and an extra memory above that needed for normal functions, or else plan to eliminate the least important function to be performed when a failure occurs.

The CDS must contain provisions to test for errors, to diagnose and locate the faulty module, to roll back to a minimum configuration while repair is being performed, and to repair the CDS.

Repair is accomplished through replacement of the faulty module with a good module by reprogramming all programs that refer to the faulty module. When the programs are first written, all references to memory and to hardware needed will be done symbolically. An assembler will assign absolute addresses to the symbolic addresses. At the same time, it will produce tables of these symbolic absolute equivalences, and what instructions use what addresses. This information will be kept in a computer on the ground; and, in the event of failure, these tables will be searched to find all references to the faulty unit. At this point, uplink commands will be made to change all required addresses. If a tape recorder is on-board with sufficient capacity, then some of these tables can be kept on the spacecraft. In this case, fewer commands will be needed to make these reference changes.

The most critical information path in the CDS is the command link. If this link is broken, one cannot change modes or "repair" the CDS by reprogramming.

1.2.2 Central Data System--Software

The CDS software or program requirements are discussed briefly in the following paragraphs which are concerned in turn with the functions of processing, identification and formatting, and timing and control.

Data Processing

The essential purpose of any data processing technique is to convert raw data samples into a form in which they are more useful and/or less redundant than in their unconverted form. A number of data processing techniques have been examined in connection with this study. These include filtering of samples to reduce frequency aliasing, compensation to remove modulation caused by spacecraft spinning, extraction of maximum and minimum values from sets of data, the formation of histograms and the computation of their quantiles, and the estimation of the power spectrum of a finite-length time function.

While all of these processing techniques (and others as well) may be included in the CDS program, the decision to process the raw data from a particular scientific instrument prior to transmission must be made by the instrument designer or experimenter. Consequently, this study has simply demonstrated the feasibility of including various processing algorithms in the CDS program.

Certain data processing functions selected as constituting a representative and typical set have been incorporated into the CDS simulation program. These consist of the following:

- 1) Removal of spin modulation from the magnetometer data.
- 2) Extraction of maximum and minimum values from the radio propagation polarization data taken over one spacecraft rotation. These two values are sufficient to define the shape of the polarization ellipse.
- 3) Variable bandwidth low-pass digital filtering of several instrument outputs. The filter cutoff frequencies are functions of the current sampling rates.
- 4) Computation of six quantiles from 1000 samples of the cosmic ray counter.
- 5) Computation of the mean and variance of neutron energy level from a sample of 1000 neutron events.
- 6) Spectral analysis of the output of the magnetic field sensor in the VLF experiment. Ten weighted spectral components are computed based on a 1000-sample record.

Data Identification and Formatting

Before data gathered from the various on-board sources can be telemetered to earth or stored for later transmission, they must be arranged in an identifiable pattern and tagged with such additional information as may be required for their subsequent interpretation. The following important considerations must be taken into account in developing suitable data formats.

- 1) For any given on-board transmitting equipment and ground-based receiving equipment the bit rate which the telemetry-link will support with acceptable error rates is strongly dependent on range. Since the range of a solar probe vehicle can vary from 0 to 2 AU, the telemetry-link bit rate should be variable as a function of range so that the available channel capacity may be used efficiently. For the purpose of this study telemetry rates of 8, 16, 32, 64, 128, 256, 512, and 1024 bits/sec have been assumed.
- 2) The rate at which data are gathered by the CDS must be controlled in some way to match the output telemetry rate in use at any given time. Various techniques may be used to accomplish this. In addition to the data compression methods discussed in the section on data processing, the rates at which instruments are sampled can be adjusted in proportion to changes in available telemetry rate.
- 3) Provision should be made for the efficient handling of the situation in which one or more of the scientific instruments is turned off, either because of some malfunction or because its output is not expected to be of interest at a particular time in a mission.

One approach to the formatting problem is to set up a fixed cyclic sequence of data words in which all of the desired information appears. If this sequence contains n bits and the available telemetry rate is r bits/sec, then the CDS input and output rates can be matched provided that instrument sampling rates can be chosen to bring in n bits every n/r seconds. Such a fixed format has two important advantages. First only a relatively small number of bits (frame sync bits) need be transmitted to mark the start of the known sequence. Once these are found in the received bit stream, the source of other bits may be inferred from their location in the bit stream relative to the frame sync marker. Second the same sequence can be used at different bit rates simply by making suitable adjustments in the input sampling rates. The chief disadvantage of such fixed formatting is that, when instruments are turned off,

meaningless bits must be inserted into the telemetry stream in place of those which would normally come from the inoperative instruments in order to preserve the fixed sequential telemetry pattern.

Another approach is to generate a variable format in which data are arranged in source associated blocks which contain relatively small numbers of bits in a fixed order. The overall format may then be generated by transmitting these blocks in a variable sequence, provided only that each block carries its own identification bits and that these block ID bits can be distinguished in some way from ordinary data bits. Such a variable formatting scheme has the advantage that blocks associated with inoperative sources can be dropped from the telemetry sequence and the sampling rates for the operating instruments adjusted to maintain input-output rate matching. This technique eliminates the inefficiency involved in transmitting meaningless bits, an advantage which can more than offset the disadvantage of including block ID bits.

In either case, the telemetry format must include not only the source data and frame sync or block ID bits, but also any additional information needed to interpret the source data and monitor spacecraft operations. This includes such things as current instrument ranges, sensitivities and spatial orientations, certain engineering data indicating system status, and timing data.

In the present design study both of these formatting schemes have been included in the CDS program in order to have available efficient formats for all operating conditions. Five different formats have been developed. These are:

- 1) Fixed Engineering (FE) - This format contains only engineering data. The arrangement of engineering words within a frame is fixed.
- 2) Fixed Low Bit Rate Science (FLBR) - This format contains primarily science data, but some engineering and status data is also included. The arrangement of words within a frame is fixed. This format has been tailored to handle science data when the downlink telemetry rate is 64 bits/sec or less.
- 3) Fixed High Bit Rate Science (FHBR) - This format has essentially the same characteristics as the FLBR format, with the exception that it has been designed to work at telemetry rates of 128 bits/sec or more.

- 4) Variable Low Bit Rate Science (VLBR) - This format is designed for use when one or more of the scientific instruments is turned off by ground command. Under such conditions the sampling rates for the functioning instruments are automatically increased and the telemetry format altered to maintain nearly complete utilization of the available downlink telemetry rate. This particular format is designed for use at telemetry rates of 64 bits/sec or less.
- 5) Variable High Bit Rate Science (VHBR) - This format has features similar to those of the VLBR format, but is intended for use at bit rates of 128 bits/sec or more.

A detailed description of these formats and a discussion of their utility is given in the following paragraphs.

Fixed Engineering Format: One complete frame of this format consisting of 128-bit lines is shown in Figure 1-2. Each line begins with an 11-bit line sync code, followed by an 18-bit status word. Six different such words appear in each frame. These words contain current information on time, spacecraft rotations, commanded telemetry rate and format, instrument on-off commands, command link bit rate, storage mode commands, and data processing commands. The remaining 99 bits in each line are used to telemeter out the entire contents of the engineering list which provides information on the status of all on-board systems. One parity bit for each eight data bits is generated and added to this section of the format. Two complete readings of the engineering list are included in each six-line frame.

Fixed Science Formats: The fixed science formats have been designed to include a maximum amount of useful data bits and a minimum amount of identification bits. Most of the data in these formats has one parity bit added for each eight data bits. Engineering and status data are included as subcom along with the science data in each frame.

The fixed low bit rate format is shown in Figure 1-3. It consists of a sequence of 18 128-bit lines. Each line starts with an 11-bit sync code. In every other line the sync code is followed by a 13-bit engineering subcom word. The first 7 bits of each engineering word form a unique subcom identification (SID) code indicating the source of the last 6 bits which are the engineering data. The SID is necessary because, while the engineering data

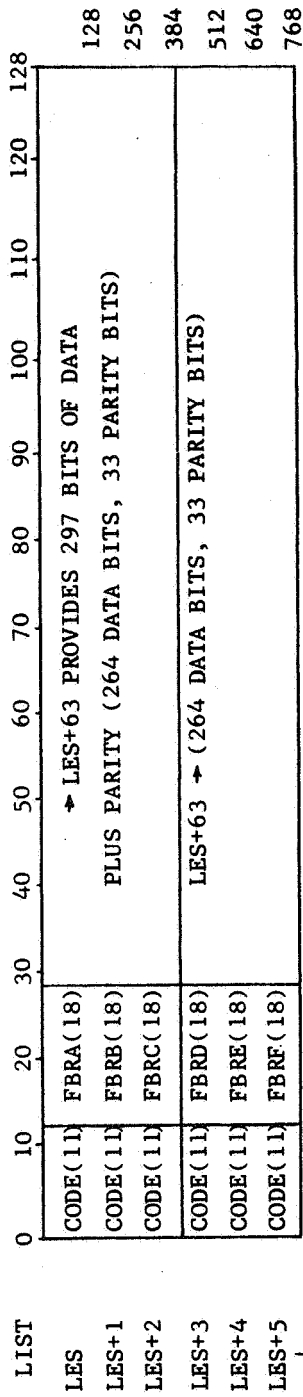
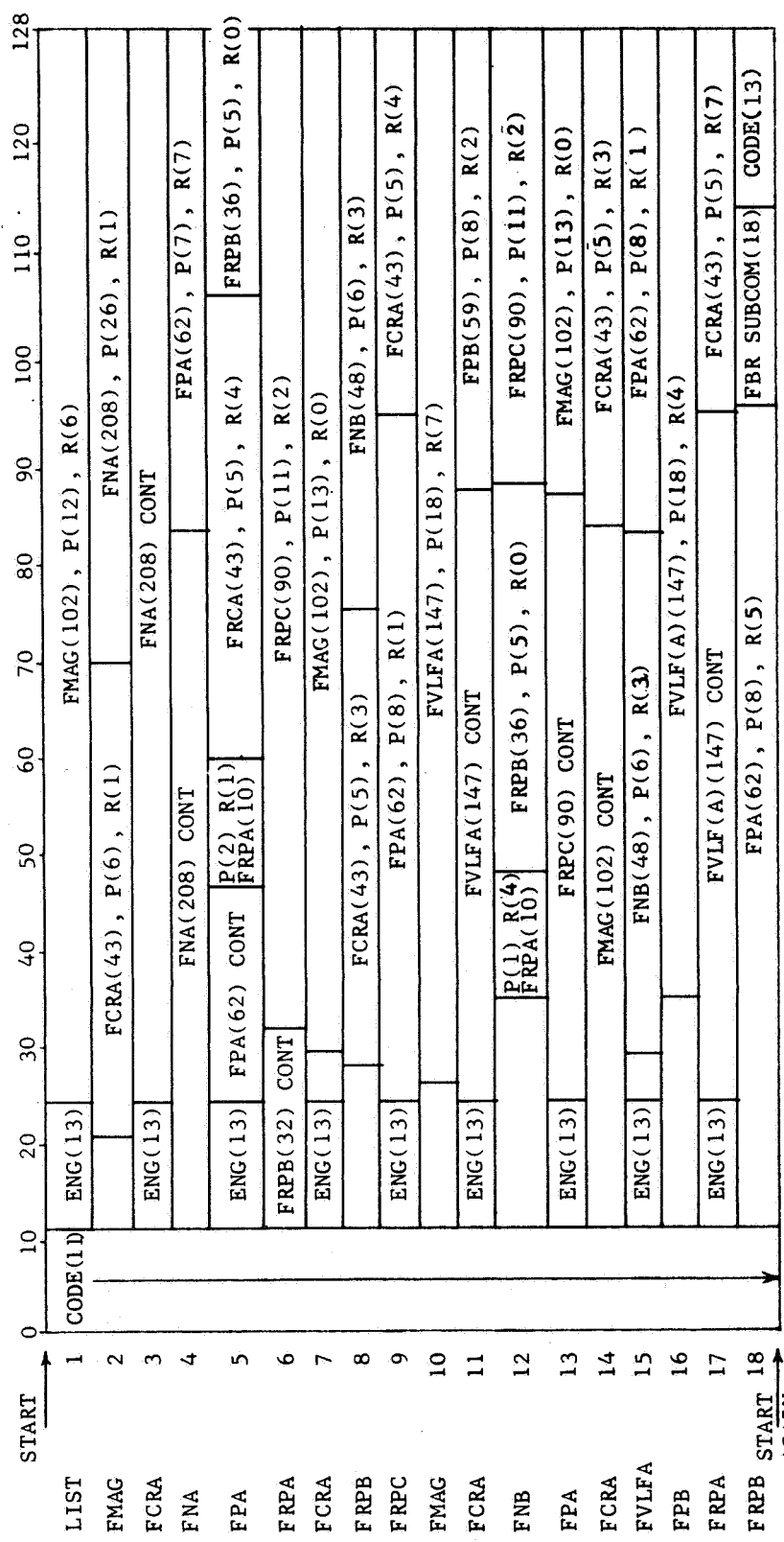


Figure 1-2. Engineering Format



FORMAT COMPOSITION (FRAME)

DATA + PARITY BITS	1953
ENG BITS	117
CODE SYNC BITS	198
SUBCOM + CODE(13)	-31
TOTAL BITS PER FRAME	2304

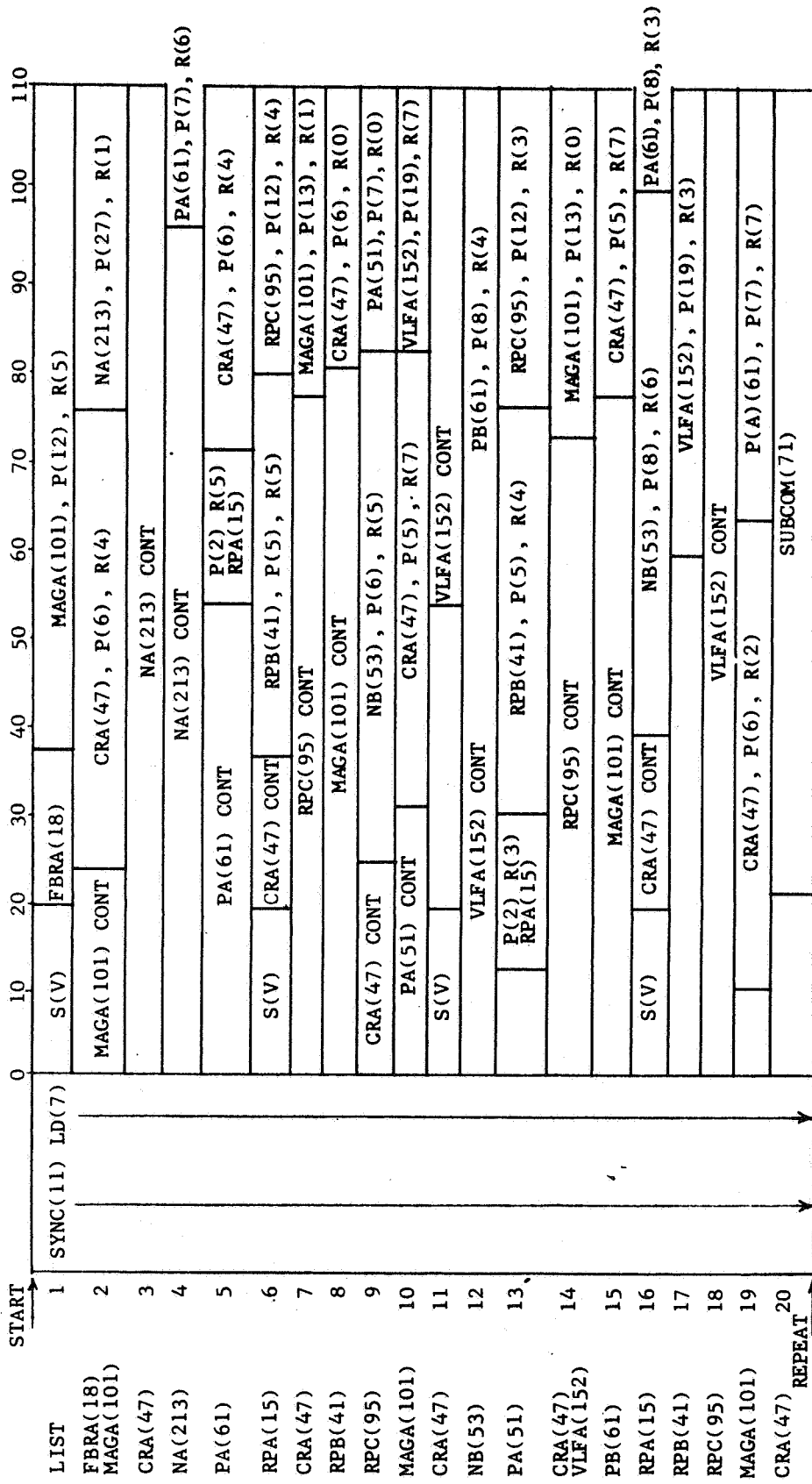
Figure 1-3. FLBR Science Format with Parity

list is scanned cyclically, only those engineering words that have changed significantly since the last time they were transmitted are picked up and put into the telemetry stream. Since many engineering measurements change very slowly or not at all, this technique results in a net compression of the engineering data. The balance of each line is used to telemeter out the various instrument-associated data blocks in a fixed sequence. For a detailed picture of the contents of these data blocks see Figure 1-4. One parity bit is added for each eight data block bits, except in the eighteenth line. Here the last five bits of the FPA data block, the FBR subcom block, and the end-of-frame code are transmitted without parity. The FBR subcom block contains a cyclic commutation of the six FBR status blocks, the complete cycle requiring six successive frames.

The fixed high bit rate format is arranged in a similar fashion but with some differences in detail. The engineering subcom words appear every third line, the entire frame is 24 lines long, the sequence of data blocks is modified, and two FBR subcom blocks instead of one appear in the last line of the frame. At a bit rate of 512 bits/sec the FHBR format can handle all instrument data taken at the maximum sampling rates, it will provide a complete cycle of FBR status data every 18 seconds, and it will scan all 64 entries in the engineering list at least once every 48 seconds.

Variable Science Formats: The variable science formats have been devised so that the CDS program can automatically modify the instrument sampling rates and the telemetry format to maintain efficient use of the downlink channel capacity in the event that one or more instruments is turned off. As with the fixed science formats, data are assembled as sampled into instrument-associated blocks. The telemetry stream is then generated by outputting these blocks in a cyclic sequence. The data blocks for use with the variable science formats differ from those used with fixed formats primarily in that a 5-bit block identification (BID) code is inserted at the beginning of each block.

The variable low bit rate (VLBR) format is shown in Figure 1-5. Each line starts with the usual 11-bit line sync code followed by 7 bits of line definition (LD). The LD is a binary number in the range 0 to 110 (decimal)



FORMAT COMPOSITION (FRAME)

DATA + PARITY BITS	2041
SUBCOM	159
SYNC(11) + LD(7)	360
TOTAL BITS	2560

Figure 1-5. Variable LBR Telemetry Format with Parity

REPEAT

SUBCOM

indicating at which of the following 110 bits in the line the first BID occurs. An LD of zero signifies that no new blocks start in that line. The presence of the LD permits format disassembly on a line-by-line basis, since once a BID is located in a line, foreknowledge of the length of that particular block and its internal arrangement can be used to separate the telemetry data into interpretable groups. The use of so large a fraction (7/128) of the available bit rate to realize this line-by-line disassembly feature may be unnecessary. Improvement in this respect is readily attainable by providing line definition information only for every n^{th} line rather than for every line of the format.

The scientific, engineering, and status data are assembled into each line following the line sync and definition bits. The greater part of this data is assembled by scanning the sequence list (see Figure 1-5), which determines the order in which scientific data blocks are to be picked up. If certain instruments are off, then the corresponding entries in the sequence list are automatically skipped over in the scanning process, and the sampling rates on those instruments that remain on are increased to maintain an overall match between data gathering and downlink telemetry rates. One parity bit is added to each eight science data bits.

In addition to the main sequence of scientific data blocks, other types of data blocks are inserted into the telemetry stream. One such is the subcom block (S). These appear after the LD in every fifth line of the VLBR format to assure that subcom data are transmitted at least this often. Each subcom block consists of 5 BID bits to identify the block, 7 subcom identification (SID) bits to identify the subcom data, and a variable number of data bits. The data may be priority engineering or status data, or if no priority data is available, engineering data picked up from the engineering list in the same way as for the engineering blocks in the fixed science formats. Subcom blocks are also inserted in the bit stream to equalize the CDS input and output data rates. The instrument sampling rates are chosen so that the data input rate is always slightly less than can be accommodated by the downlink telemetry rate. As a result there will be times when the next data block in the main block sequence has not been filled with samples and is, therefore, temporarily unavailable. This condition is detected by means of a system of flags, and

additional subcom blocks are assembled into the format as filler, until the awaited science block becomes available. The main block sequence is then resumed.

There are two additional types of irregularly and infrequently occurring data blocks that are inserted into the telemetry stream. One of these is instrument-associated subcom. These blocks contain information about the status of particular instruments, which is essential to the interpretation of that instrument's science data. These are inserted as needed immediately before the main sequence block with which they are associated. The other such block type contains processed data. The last three entries in the main sequence list for the VLBR format are of this type. These are handled exactly like other main sequence blocks. However, the instrument on-off flag for these blocks is set to indicate a fictitious off condition except when processed data is available for insertion into the bit stream.

The operation of the variable format assembly program is such that it is not possible to predict or show the exact arrangement of data within the format under all conditions. However, the information necessary for interpreting the received telemetry data is present in the bit stream at all times. The illustration given in Figure 1-5 is only intended to give an approximate picture of what the VLBR format could look like under certain conditions and to demonstrate that a sufficient telemetry bit rate is available to handle the required data flow.

While the variable high bit rate (VHBR) format uses a different main block sequence list and has subcom blocks regularly inserted in every third rather than every fifth line, it is nevertheless assembled in the same manner as is the VLBR format. Indeed the same program is used to assemble both these formats.

Timing and Control

The CDS program must time the sampling of input data lines at the desired rates. In the case of directional instruments such as the plasma probe and cosmic ray detector, not only is the sampling rate important but also the time at which samples are taken in relation to the rotation of the spacecraft.

The flow of formatted data to the telemetry transmitter must be controlled by the CDS. The CDS must also exercise control over its own internal operations. It must perform certain operations in a set sequence, others at particular instants of time. It must accept interruptions in routine operations to service ground commands and other relatively infrequent and irregular events. It must fit low priority operations with noncritical timing requirements into the time periods when it is not otherwise tied up.

The CDS serves as a central controller for all on-board systems. On the basis of both stored programs and ground commands, it will turn on and off and change the operating mode of various systems. Providing the CDS with the capability to accept program modifications via the command link makes the implementation of these timing and control functions quite straightforward.

1.2.3 Central Data System Simulation

The CDS software functions discussed above have been programmed for a CDC-3200 general purpose scientific computer. A brief description of this simulation program is given in the following paragraphs. This is followed by a discussion of the simulation study results.

Simulation Program

The simulation performs all the operations of a flight model of the CDS in nonreal time. Input science data is generated by a data generation subroutine and stored on magnetic tape. When the simulation model is exercised, input science data is read into the CDC-3200 in blocks. The simulation model contains processing algorithms, format assembly subroutines, parity subroutines, and allows command changes from the operator's console.

Figure 1-6 summarizes the program executed each spacecraft revolution. At START the Science Data List is inputted to the computer. The Spin-Demodulation subroutine removes rotational effects from the magnetometer data. The Polarization Ellipse Max-Min subroutine determines the maximum and minimum values of the polarization data from the radio propagation experiment. The Plasma Probe Maximum and Radiant subroutine specifies the maximum flux, and the associated channel and sector for the most recent 120 data samples.

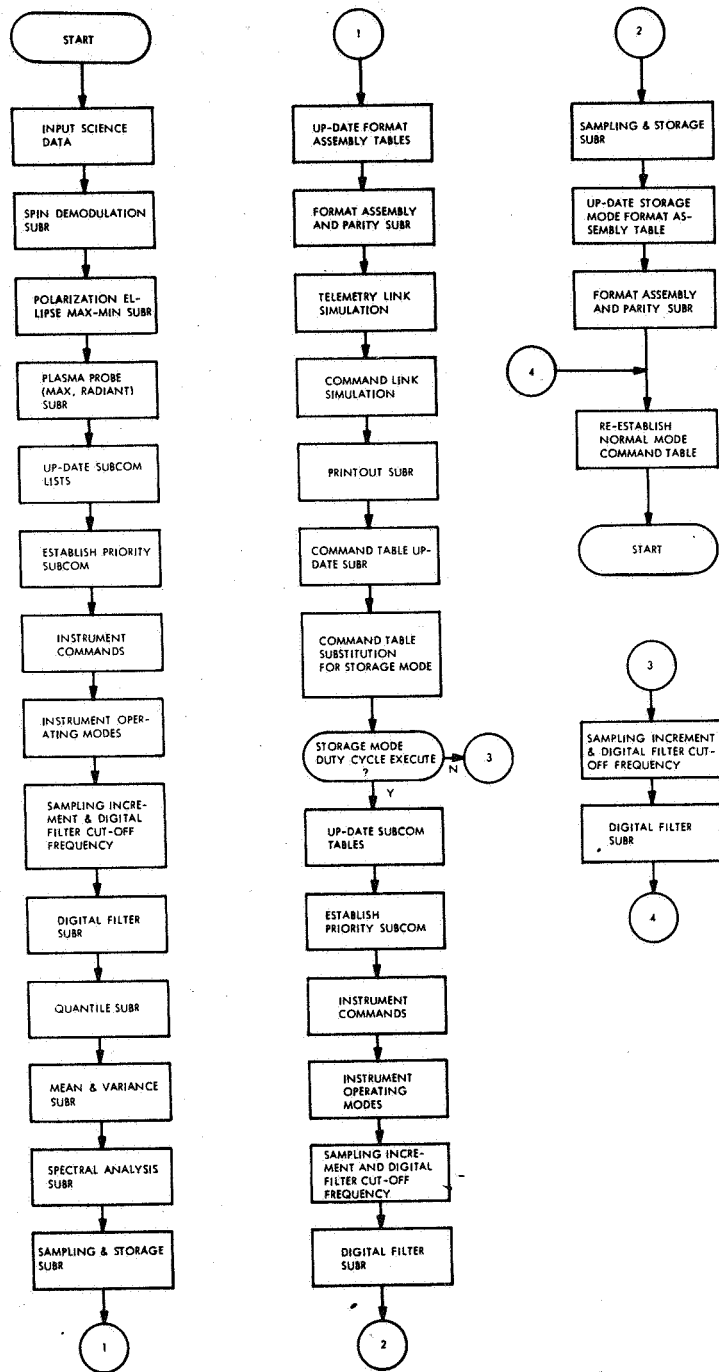


Figure 1-6. Program Routine Executed Each Spacecraft Revolution

Ancillary subcom lists are updated from the Command Table. Priority subcom is identified and moved into the Priority Assembly table. Instrument commands are executed and the format assembly tables are updated. Instrument-operating modes are examined and the format assembly tables are updated when necessary. The rate at which the sampling format is to be executed is computed. The cutoff frequency for the Digital Filter subroutine is computed. The Digital Filtering, Quantile, Mean and Variance, and Spectral Analysis algorithms are executed. The Sampling and Storage subroutine samples appropriate data lines at specified spacecraft revolutions and moves data into storage blocks. The format assembly tables are then updated. The Format Assembly subroutine assembles the telemetry format. The bit error rates and the signal-to-noise ratios are computed for the telemetry and command links. The printout subroutine outputs measures of system performance, Command Table parameters, and telemetry as required. The Command Table is updated.

The routine enters Storage Mode operation which is executed on command and results in the assembly of data in the 512 bit/sec VHBR format for on-board storage and later transmission. It is executed on a duty-cycle basis once every 256, 512, 1024, or 2048 spacecraft rotations in accordance with the commanded duty cycle. At the beginning of each Storage Mode cycle one complete frame of Storage Mode data is assembled by scanning the VHBR block sequence list once. The actual length of such a frame depends, of course, on which instruments are on.

If the Storage Mode is not to be executed, the program branches to ③ where the Digital Filter subroutine is executed for magnetometer data. The program branches to ④ where normal mode commands are restored to the Command Table. The program branches back to START. If, on the other hand, the Storage Mode is scheduled, the appropriate Storage Mode lists and tables are updated. Instrument command and operational changes are executed. Sampling times are computed and stored. The digital filter cutoff frequency is computed; magnetometer data are smoothed by the Digital Filter subroutine. Data are sampled and stored in storage mode data blocks. The Storage Mode Format Assembly subroutine assembles the format and assigns parity. Normal mode parameters are restored to the Command Table. The program branches to START.

System Simulation Results and Conclusions

The system simulation results have been evaluated in terms of the CDS memory and macro-instruction execute times which are practical in view of the physical constraints on power, volume, weight, and reliability.

The "map" of instruction and memory locations and associated analysis reveal that the CDC-3200 memory requirement for instructions and constants can be contained in roughly 4000 memory locations. The 2500 instructions employ 24-bit words. The 1500 storage locations required for constants generally contain less than 16 data bits. Thus, the CDS storage requirement is approximately 6000 words of 16 bits each.

The FORTRAN program compiled into 9600 CDC-3200 instructions. By clocking the execution times of specified program segments and then writing an efficient machine code for these segments, it was found that time expansions of 20 → 45/1, due to the way the FORTRAN program compiled into CDC-3200 language, were the rule rather than the exception. If simple operational and procedural changes are made, the entire program can be executed in approximately 10 → 20 milliseconds at 512 bits/sec and 15 → 40 milliseconds at 1024 bits/sec. A revised flow and complete CDC-3200 language program would be needed to refine the estimates. If the ratio of CDS-CDC macro-instruction execution time is 10, the CDS will be occupied 100 → 200 milliseconds/sec with execution of the instruction list. The input/output servicing time requirements may add another 100 milliseconds/sec to the execute time requirements at high bit rates. In addition, the executive routine which assigns tasks in time sequence may add another 100 milliseconds/sec to the time required. This implies that the CDS machine may be busy 350 → 600 milliseconds/sec at 1024 bits/sec.

A machine word size of 16 bits appears to be appropriate. Double precision is required for the processing algorithms. Virtually all data are specified by less than 16 bits. The principal advantage of a 16-bit word size lies in power conservation. A four-bit operation code and a 12-bit address field are specified. Each 16-bit data word has 15 bits for data and one bit for sign. Since there is a tendency to underestimate storage requirements, the total storage may end up in the 100,000- to 200,000-bit range.

The postulated machine instructions include add, subtract, move, shift left or right, test for zero and branch, mask, read and write; mask, move, and pack instructions occur frequently during format assembly. Multiply and divide instructions occur very infrequently and can probably be provided by software.

Rough calculations based on the assumption that the CDS will require approximately 1500 logic gates and 100,000 bits of memory yield the following results. The CDS would have approximately the same weight and size as the present Pioneer data system (weight = 20 pounds, volume = 1/2 ft.³). It would require 2 to 4 times as much power (10 → 20 watts). It would support a telemetry rate of 8 → 2048 bits per second. In addition to performing all of the functions of the current Pioneer data system, it would provide data processing and variable formatting capabilities.

To have been able to give explicit answers which would have made the CDS design seem comfortably obtainable would have required that the simulation be tailored to a postulated CDS design. The most significant answer we can offer for the CDS design is that the designer must squeeze every bit of performance out of the system within the restraints. The present system simulation holds to a reasonable degree for a specific species of experiments. The CDS must accommodate a wider breed of experiments than those postulated for the simulation model.

1.3 CONCLUSIONS

The results of the study reported here indicate that a stored program digital computer can be used to advantage as a CDS on-board unmanned solar probe spacecraft. High reliability performance is obtained by means of a novel logical design which uses pooled components and programmed interconnections.

It has been demonstrated that a stored program CDS can provide more data processing and formatting capability than the present Pioneer data system without exceeding reasonable size, weight, and power specifications. Furthermore, such a CDS makes possible efficient use of the telemetry channel capacity under all operating conditions.

Because this CDS has a stored program, it could readily be used without hardware modification to meet the data handling requirements of a wide variety of scientific missions. Clearly such interchangeability of hardware will become more and more important economically as efforts aimed at the exploration of space increase in frequency and scope.

SECTION 2

CENTRAL DATA SYSTEM INTERFACE CONSIDERATIONS

The CDS should be capable of counting, analog-to-digital conversion, storage, instrument mode selection, and pulse height analysis to interface flexibly with the scientific instruments. These functions are currently performed within certain scientific instruments.

The experiments are described in Research Report No. 2, NASA Contract No. NAS2-3254, 29 August 1966.

The overall system block diagram is shown in Figure 2-1. The CDS interfaces with the telemetry subsystem, the scientific instruments, and the spacecraft operations and control subsystem. The latter contains auxiliary storage, engineering sensors, sun sensor, as well as power, guidance, and RF subsystems.

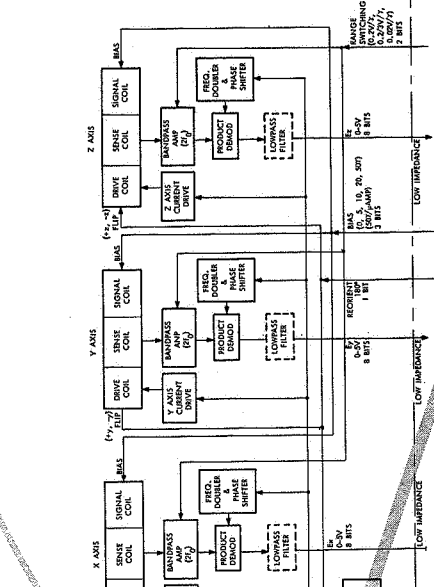
Some functions currently implemented within each scientific instrument may be performed by the CDS. The interface definition involves trade-offs between instrument and CDS design. The functional boxes within the instruments or subsystems that can be integrated into the CDS are indicated by segmented lines in Figure 2-1.

Figure 2-2 promulgates the interface between the experiments, spacecraft support equipment, and the CDS. The experiments employ 56 data lines and 10 command lines. The data lines are either analog or digital. There are 75 data lines for the engineering sensors and 80 command lines.

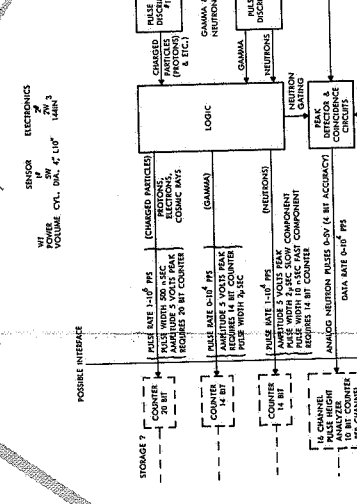
The basic CDS design philosophy is to provide maximum practical service to the experimenters, yet leave them the option of using the available service or not. All input data lines are multiplexed from the experiments to the CDS. Two or more multiplexers are recommended to increase the response capability and allow for graceful degradation in the event of a multiplexer failure. Appendix III describes suggested processing considerations for each experiment.

Stanford Radio Propagation Experiment

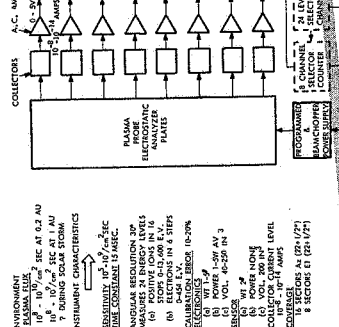
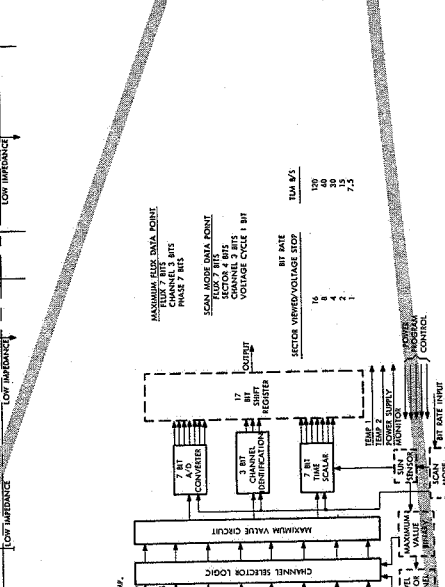
Triaxial Fluxgate Magnetometer



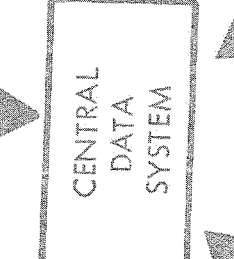
Neutron Detector (K 0.5 AU)



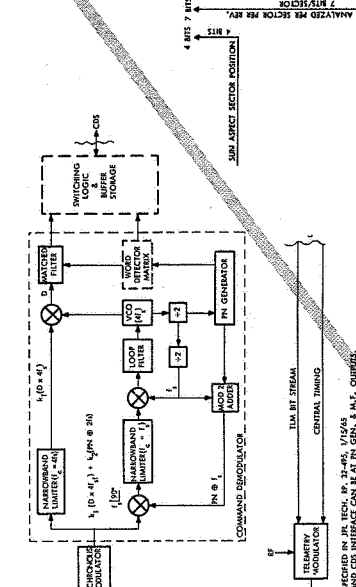
ARC Plasma Instrument



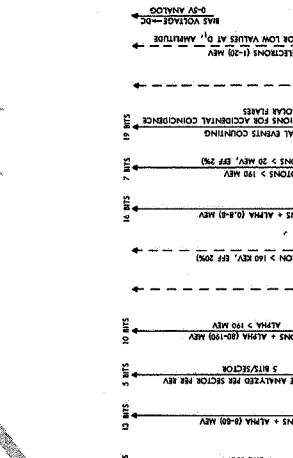
CENTRAL DATA SYSTEM



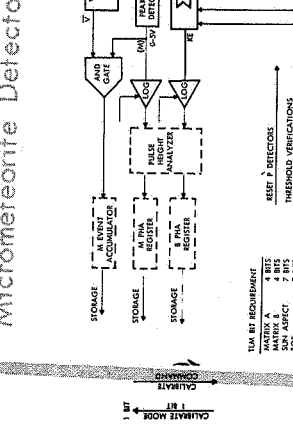
Telemetry Subsystem



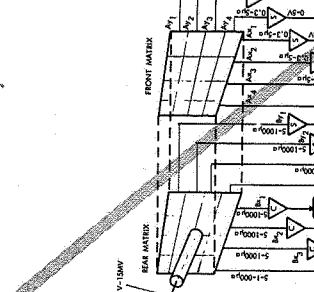
Cosmic Ray Experiment



Micrometeorite Detector



s/c Ops & Control Subsystems



VLF Experiment

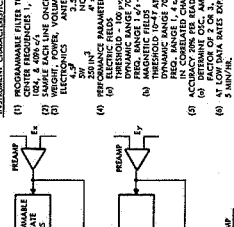


Figure 2-1. Overall System Block Diagram

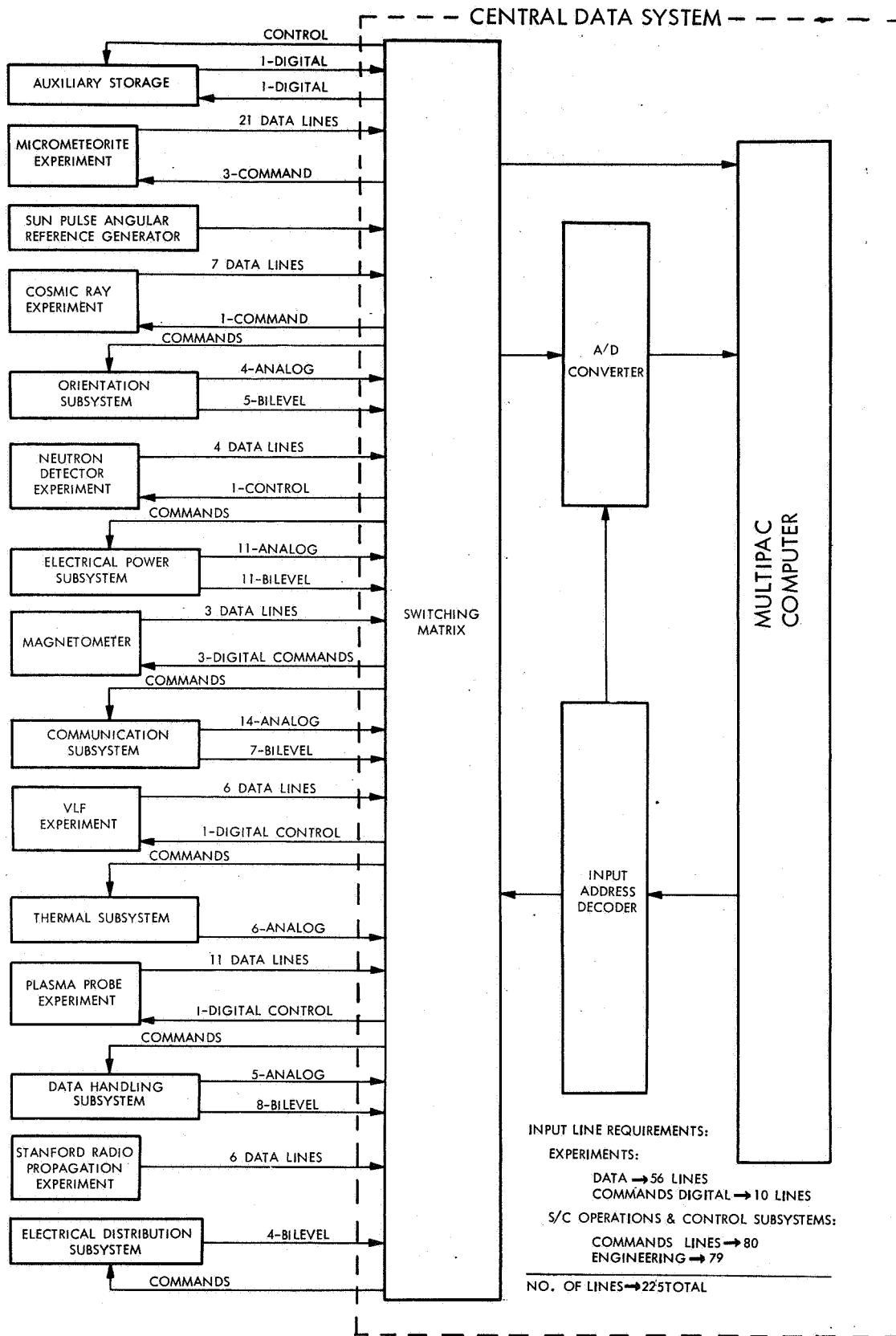


Figure 2-2. Interface Between CDS and "On-Board" Experiments and S/C Support Equipment

Analog inputs require 5 → 8 bit conversions. If the conversion is provided within the computer, sophisticated multiplexing switches are needed. If these are converted to pulse width at the experiment, standard digital circuitry can be used for the multiplexer. Each conversion must be tailored to the instrument. The exact technique employed will be specified during the logical design of the CDS where specific input/output interfaces must be defined.

2.1 COSMIC RAY EXPERIMENT

The counters and pulse height analyzers contained in the cosmic ray instrument (shown in Figure 2-1) may be incorporated into the central system. The four inputs to the logic and signal conditioning are each 0-5 volts in amplitude. The D_4 pulse duration is 30 nanoseconds. The D_1 , D_2 , and D_3 pulse durations are of the order of one microsecond. The logic and signal conditioning function is most reasonably performed within the instrument. The 17-bit D_4 counter must handle 10^5 counts per second. The scalar in series with the 17-bit counter must handle 10^7 counts per second. There are pro and con arguments for placing the D_4 counter in the CDS.

The 32 and 128 level pulse height analyzers operate on one pulse per second each. The pulse height analysis can be performed by the central computer.

The requirements for the cosmic ray experiment have changed during the course of the study, resulting in lower counting rates and fewer transmitted bits per second. The number of bits to be transmitted is indicated in Figure 2-1. If 6 bits convey information generated by the 17-bit D_4 counter, the 17-bit count must be partitioned into one of 64 possible ranges. Information on the exact range assignments is not available. A similar situation exists for the other data lines of this experiment.* The histogramming required can be provided by the CDS.

*The significant bits for these data lines have been defined by the experimenter.

2.2 MICROMETEORITE DETECTOR

The micrometeorite detector shown in Figure 2-1 measures the velocity, momentum, energy, and radiant of micrometeorites. The maximum number of expected micrometeorite impacts per day is 100. Each event is specified by 43 bits. The electronics and the "AND" and "OR" logic shown in Figure 2-1 should remain in the instrument package. The pulse height analysis can be performed by the CDS. The time of flight counter should remain in the instrument package. The reset, threshold verification, and data sample commands should be implemented by the central computer.

2.3 NEUTRON DETECTOR

The neutron detector contains four counters and a 16-channel pulse height analyzer. All pulse durations are of the order of a microsecond. The pulse shape discrimination process is best handled in the instrument package. The peak detector and coincidence circuits should remain in the instrument.

The counters require counting rates of 10^6 per second for charged particles and 10^4 per second for gamma rays and neutrons. The counters and pulse height analyzer channels are sampled once per second. Each channel of the pulse height analyzer uses a 10-bit counter. The charged particle counter capacity is 20 bits; the gamma ray and neutron counter capacities are 14 bits each. Time resolution requirements for each event are one second per day (17 bits). Engineering sensors associated with the detector (not shown in Figure 2-1) add negligibly to the data flow from the neutron detector.

The peak detector output can be sampled in several different ways. In one mode, the central computer interrogates the peak detector fast enough to count all pulses; a second mode of operation uses a priority interrupt. However, the most reasonable approach is probably to retain the counters in the instrument but provide ancillary counters when instrument counter failures occur. The selection of the preferred method will be resolved in the detailed design considerations of the computer.

2.4 TRIAXIAL FLUXGATE MAGNETOMETER

The fluxgate magnetometer has three analog data channels. The analog signals are converted to 8-bit words. Each magnetometer channel is sampled at a maximum rate of four times a spacecraft revolution generating 96 bits of magnetometer data per spacecraft revolution.

The A/D conversion can be handled by the CDS. The commands setting the bias level, orientation, and receiver sensitivity are executed by the CDS.

The low-pass filtering for each channel can be provided by the CDS. Digital filtering subroutines are available; their use can result in considerable savings in hardware.

2.5 RADIO PROPAGATION EXPERIMENT

The functional description of the Radio Propagation Experiment is shown in Figure 2-1. The 10-bit counter counts wavelengths of propagation path difference and can be incorporated into the CDS. The analog outputs, S_2 , S_3 , S_4 , and S_6 can be A/D converted by the CDS consistent with the sampling requirements indicated in Figure 2-1. A 6-bit precision is required. The analog signal, S_5 , associated with the polarization experiment requires at least 36 samples per second.

There is no significant interface problem with this experiment.

2.6 VLF EXPERIMENT

The VLF experiment contains six analog output lines that are sampled once per second, each line being converted to a 6-bit word. There are two command lines for switching filters, three bits being required for the switching function on each line. The data rate is 36 bits per second, and there is no interface problem.

2.7 ARC PLASMA INSTRUMENT

The plasma detector contains many digital elements. The selection of energy level, channel, and mode of operation could be executed by a CDS. The A/D conversion and data word synthesis can be executed by the CDS.

The ecliptic plane is divided into 15 angular sectors. In the maximum flux mode, one 17-bit mode word containing flux magnitude, channel number, and azimuth angle relative to the sun line is generated every spacecraft revolution. In the scan mode, when the telemetry downlink data rate is 512 bits per second, 15 words of 15 bits each are generated in one spacecraft revolution. Maximum flux is measured in the next revolution.

The CDS can provide all functions performed currently by the wired-in instrument digital logic. Mode changes, channel selection, and energy selection can be implemented by the CDS.

2.8 SPACECRAFT OPERATIONS AND CONTROL SUBSYSTEM

This subsystem contains the sun sensor that generates the angular reference for all spacecraft functions. The sun sensor generated clock reference is asynchronous with the down-telemetry and the central computer clock. A sun sensor clock reference can be generated from sun sensor pulses by the CDS.

The CDS is required to input data into auxiliary storage. The data from various engineering sensors are A/D converted for incorporation into the telemetry format.

Functions to be performed by the power, guidance, and RF power tubes, antennas, etc., will be executed by program subroutines. A switching matrix is required in each subsystem.

2.9 TELEMETRY SUBSYSTEM

The telemetry subsystem defines the interface between the up-command decoder and the central computer on the uplink and the interface between the

central computer and the telemetry modulator on the downlink. A specific up-command decoder has been assumed in order to exhibit the interface. The word detector matrix provides bit sync and word stop markers, and the information necessary for placing command words into the up-command buffer. The word detector matrix and the up-command storage can be incorporated into the CDS. The up-command link data rate may be 1 → 100 bits per second. The CDS can service this interface easily.

2.10 INTERFACE CONSIDERATIONS SUMMARY

A number of instrument components can be incorporated into the central computer. The savings in terms of weight and volume to the experimenters is not large. Although the interface is simplified for the instrument designer when a CDS is used, the real advantage to the experimenter is the flexibility offered by a programmable computer.

The Sunpulse Angular Reference Generator provides a sampling reference for the CDS. It will be serviced frequently enough to allow a computer clock to be synchronized to it. A servicing rate once per spacecraft revolution is recommended.

The cosmic ray experiment can be serviced frequently or infrequently depending on the interface. If all counters are contained in the instrument, the CDS services the 7 data lines once per spacecraft revolution. The maximum counting (on the D_4 data line) rate is 10^5 bits per second. This implies that the CDS must service the D_4 line in excess of 10^5 times a second when counts are to be accumulated. It makes more sense to provide a scalar, and service this line less frequently or incorporate the 17-bit counter in the instrument. Four other event counters are included in the cosmic ray experiment. They count 10^2 → 10^4 events per second, and could be serviced by the CDS.

The servicing time required for the Orientation, Electrical Power, Thermal, and Electrical Distribution Subsystem is negligible compared to the cosmic ray requirements. The Communication and Data Handling Subsystem will require service approximately each millisecond maximum.

The VLF experiment requires a servicing rate of once per spacecraft rotation, 6-bit A/D conversion being required. There are 6 data lines and 2 command lines. The Radio Propagation experiment requires servicing no more than 133 times each S/C rotation. The servicing requirements for the magnetometer and plasma probe are modest. The magnetometer requires the servicing of 4 data lines 4 times per spacecraft revolution. A/D conversion accuracy to 8 bits is required. The plasma probe requires 8 data lines to be serviced each spacecraft revolution. A/D conversion accuracy to 7 bits is required.

The servicing requirements are dominated by the cosmic ray and neutron experiments. They dictate a servicing period of 50-100 microseconds. The dwell time on each line will probably average 5-10 microseconds. A timing diagram for all possible interface configurations is necessary to completely detail the interface servicing requirements. This is a task left for the logical design study. Appendix IV lists CDS input/output timing requirements.

SECTION 3

MULTIPAC COMPUTER FOR CENTRAL DATA SYSTEM

A single central data system (CDS) design to be used on many solar probe missions must have a large amount of flexibility. The CDS should be capable of counting, analog-to-digital conversion, storage, mode selection, switching, and pulse height analysis in order to interface flexibly with the scientific instruments. These functions are currently being performed within the scientific instruments.

In addition, the ability to perform data reduction and other data management functions should be available. The basic design philosophy is to give maximum practical service to the experimenters, yet give them the option of using the available service or not.

The CDS must be capable of (1) counting pulses whose rates vary from less than a hertz to 10 MHz; (2) handling analog inputs of 4 to 8 bit accuracy at rates from 1 Hz to 10 kHz; (3) recording time of occurrence of events; and (4) accepting digital information from the experiments. Compromises may be made in the above requirements. The analog inputs may be converted to pulse widths before processing. In addition, the CDS must interface with the telemetry in such a manner that component failures cannot prevent further experimentation.

Vulnerability of system performance due to component failures must not be the price paid for this flexibility and centralization. Some degradation of system performance can usually be allowed due to unlikely failures, but complete outage of the system is intolerable.

For these reasons, a unique conceptual design for a general-purpose programmable CDS has been developed. A general-purpose computer approach, giving enormous flexibility, is combined with a pooling technique to allow survival after a component failure. The survival is achieved by reprogramming around the failed unit, since each unit is a member of a pool of program addressable identical modules. Each basic module is small enough that it does not represent a large portion of the CDS.

The design must provide freedom from catastrophic failures. No component failure in the CDS should cause the complete system to fail. The worst that a failure or failures should cause is a reduction in capability. For example, if for some reason a major failure occurred in the CDS such that it cannot perform all of its functions, the CDS should still be able to transmit the experimental data to earth via the telemetry link. It may be necessary to forego some of the data collection but the failure should not be catastrophic. There must under all circumstances be the ability to transmit experimental data to earth and to accept commands error-free from the earth.

3.1 SYSTEM DESCRIPTION

The CDS acts like a stored program computer. A stored word in it does not perform a full instruction as does a stored word in a normal computer, but instead performs a single operation. Three of these operations are required to perform a typical computer instruction, such as ADD the contents of a memory location to a register. Since the data can be obtained from one memory at the same time as an operation is obtained from another, our CDS takes three memory-cycle times to perform an instruction that takes two memory-cycle times on the CDC-3200 computer. A 16-bit word length was chosen, which will allow 4 bits for an operational code and three fields of 4 bits each to specify registers. Sixteen 4-bit AND gates plus drivers are needed to decode the operation. The 16 different operations allowed appear at this time to be sufficient.

A computer using this pooling technique is shown in Figure 1-1. Each of the four memories is one-quarter of the total memory requirements. The 32 registers, some of which are the memory address and memory data registers of the four memories, are identical. In this example, there are four independent but identical logic units, each with its own set of input and output bus structures.

All the registers in the machine form a single register-store. Eight bus structures are connected to the outputs of the registers. Each bus can select one register. Each logic unit has two of these busses connected to its two inputs, and two output busses to these same registers. Thus, we have eight busses on the output of the register store and eight busses feeding the register inputs.

The logic unit will perform the operation specified by the bits in the logic instruction register associated with each logic unit. This logic instruction register also specifies the registers selected for the two input busses and two output busses connected to that logic unit. This logic unit can perform all necessary instructions and control operations required in the computer. For example, by specifying the proper bits in the logic register, the logic unit can add two registers together and send the results to a third and fourth register. It can also perform control operations such as adding 1 to a program counter and sending it to a memory address register to fetch an instruction.

A small amount of control circuitry remains that is common to all units in the machine. This is a simple control unit not dependent on the state of the computer. It simply sends timing pulses to the memory and logic units to perform their functions at a repetitive rate. This clock will cause the memory to go through a read-write cycle continually. After every memory cycle, the memory data register will contain the contents of the locations specified by the memory address register. Once every memory cycle, all four logic units will perform the functions specified by the logic instruction register.

The operation of this computer is best explained by an example. Consider the routine to perform the normal computer instruction of adding the contents of a memory location to a register. In normal computing systems, a single instruction is used to specify this function and the address of the data in memory. In this computer, three instructions will be required to perform this.

The following initial conditions must have been set up previously. Logic instruction register 1 (LIR1) attached to logic unit 1 (LU1) will contain an instruction that will add 1 to memory address register 1 (MA1). This operation will be performed every memory cycle. The effect of this instruction will be to read consecutive locations from memory 1 to memory 1 data register (MD1). LIR2 of logic unit 2 (LU2) will contain the operation of transferring the contents of memory data register 1 (MD1) to the logic instruction register 3 (LIR3) of logic unit 3 (LU3). This operation will also be performed every memory cycle. The effect of this operation is to transfer the instructions that will appear in memory register 1 to logic unit 3. Thus, the logic unit 1 and logic

unit 2 will perform the control functions of this computer. Any two logic units could be chosen and this could be changed by the program itself.

One more initial condition is necessary. A register (R1) contains the locations in memory 2 where the data addresses for the instructions can be found. Consecutive locations will contain the addresses needed for the instructions in memory 1 in the order in which they are required. A transfer to another routine in memory will probably result in the use of another register or require changing the contents of register 1.

The ADD instruction will require three consecutive instructions in memory 1 and the operational sequence shown in Table 3-1. Logic unit 1 and logic unit 2 will continuously perform the same operation. The first operation that logic unit 3 (LU3) will perform is to add 1 to register 1, and then send the results to memory address register 2 (MA2). By simply placing a new address in MA2, the contents of that location in memory will appear on the memory data register after the next memory cycle. The second operation of logic unit 3 will be to move the address that now appears in memory data register 2 (MD2), because of the first instruction, to memory address register 2. This is the data address that normally appears in the address field of a standard computer instruction. On the next memory cycle the data will appear at the memory data register 2. The third operation of logic unit 3 will then perform the addition of the contents of α to the specified register.

3.2 LOGIC UNIT

The initial design concept for the logic unit uses the simplified logic unit described in detail in paragraph 3.12. This unit had a very simple design easily adapted to today's LSI (Large Scale Integration) integrated circuits. This design would require 24 to 32 bits for each micro-instruction. Since about three micro-instructions were needed for each CDC-3200 type macro-instruction, we felt that the increased memory requirements more than offset the simplification of the logic unit in view of power requirements.

A logic unit of 12 to 16 bits with about a 16 micro-instruction order code is now recommended. Four bits on the word would specify the order, and the remaining bits the registers used.

TABLE 3-1

ADD α INSTRUCTIONInstruction 1 = (R1) + 1 \rightarrow R1, MA2Instruction 2 = (MD2) \rightarrow MA2Instruction 3 = ADD (MD2) to (Rx) \rightarrow Ry

<u>LU1</u>	<u>LU2</u>	<u>LU3</u>	<u>Comments</u>
(PC) + 1 \rightarrow PC, MA1	(MD1) \rightarrow LIR3	-----	First instruction to LU3.
(PC) + 1 \rightarrow PC, MA1	(MD1) \rightarrow LIR3	(R1) + 1 \rightarrow R1, MA2	Request α from memory 2 and send second instruction to LU3.
(PC) + 1 \rightarrow PC, MA1	(MD1) \rightarrow LIR3	(MD2) \rightarrow MA2	Request α from memory 2 and send third instruction to LU3.
(PC) + 1 \rightarrow PC, MA1	(MD1) \rightarrow LIR3	(MD2) + (Rx) \rightarrow Ry	Perform function.

Glossary:

MA1, MA2 are the memory address registers of memories 1 and 2.

MD1, MD2 are the memory data registers of memories 1 and 2.

R1 is one of the addressable registers which contains the location of α .

α is the address of the data to be operated on.

PC is one of the addressable registers used as a program counter.

LU1, LU2, LU3 are the three of the logic units.

LIR3 is the logic instruction register which controls logic unit 3.

() means contents of register inside parenthesis.

The following descriptions generally apply whether the logic unit is serial or parallel (by bit). However, a serial logic unit that is fast enough relative to the memory module will tend to be favored and combined with serial transmission not only to reduce the interconnections (a growing threat to reliability, relative to the integrated circuits) but also to fit the state of current packaging technology in terms of transistors per chip and number of leads per chip.

The instruction word will contain four fields of four bits each for a total instruction code of 16 bits.

4	4	4	4
OP	R1	R2	R3

One of the 4-bit fields (OP) specified one of 16 operations. The other three fields (R1, R2, R3) specify the registers to be operated on and where the results are to be sent. A tentative list of instructions is shown in Figure 3-1.

3.3 INSTRUCTION TIMING

A timing diagram of three typical instructions is shown in Figure 3-2. This diagram shows the delay in executing an instruction from the time it appears in the memory data register. This does not normally produce difficulty except when transferring control to another location in memory such as the transfer instruction shown in this figure. The seventh instruction performs no operation but must be there since it appears at the logic unit before the effect of the transfer takes place.

Instructions

- | | |
|--|---|
| 1. $\left\{ \begin{array}{l} \text{DEC} \\ \text{INC} \end{array} \right\} \text{Rx Ry}$ | $\left\{ \begin{array}{l} \text{Decrement} \\ \text{Increment} \end{array} \right\} \text{Register Rx} \rightarrow \text{Rx, Ry}$ |
| 2. COP Rx Ry | Copy Rx to Ry |
| 3. CCP Rx Ry LUz | Conditionally copy Rx to Ry if LUz overflowed. |
| 4. TNZ Rx | Test Rx. If non-zero, set overflow FF. |
| 5. NOP α | No operation. α ignored. |
| 6. ADD Rx Ry Rz | Add Rx to Ry \rightarrow Rz |
| 7. SUB Rx Ry | Subtract Rx from Ry \rightarrow Ry |
| 8. SHR Rx, Ry | Shift Right 1 Rx into Ry |
| 9. MLY Rx Ry Rz | Multiply Rx x Ry \rightarrow Rz |
| 10. MSK Rx Ry | Logical AND of Rx : Ry \rightarrow Ry |
| 11. EOR Rx Ry | Exclusive OR Rx + Ry \rightarrow Ry |
| 12. Parity Rx | Set overflow on parity |
| 13. FMO Rx Ry | Find most significant "one" in Rx and place bit number in Ry. |
| 14. TLB Rx | Test least significant bit of Rx and set overflow \neq to bit. |
| 15. C-ADD Rx Ry LUz | Conditionally ADD Rx to Ry if LUz is overflowed. (Should this be C-ADD-SHR?) |

Figure 3-1. MULTIPAC Instructions

TIMING DIAGRAM OF THREE CONSECUTIVE INSTRUCTIONS

INSTRUCTION MICRO-SEQUENCE

- 1. COP MD1, MA2
- 2. NOP
- 3. ADD MD2, RX, BX
- 4. COP BX, MD2
- 5. COP MD1, PC
- 6. NOP
- 7. NOP

STORE BX, a

TRANSFER UNCONDITIONAL TO a

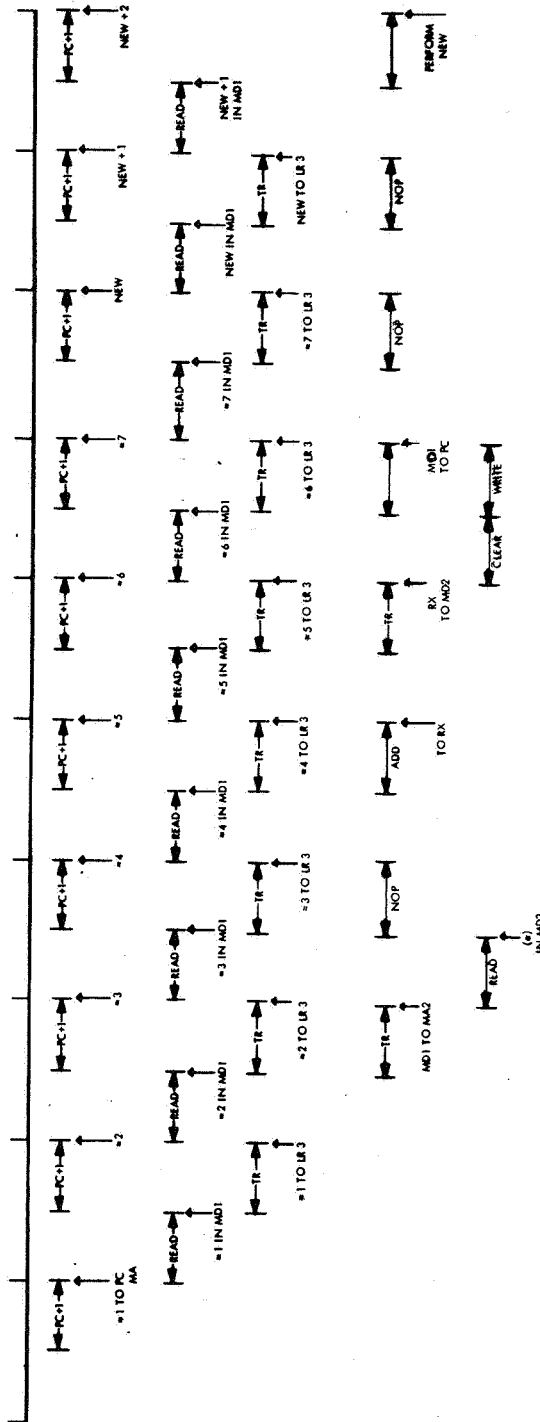


Figure 3-2. Timing Diagram of Three Consecutive Instructions

3.4 MEMORY FOR SPACE DATA PROBE

3.4.1 Power Major Problem

A major consideration in choosing a memory technique for a space probe central processor is the power consumption of the memory. Based on an actual formatting program, the data processing, as we would propose to do it, for the CDS will require between 4K and 16K of 16-bit memory, or several hundred thousand bits of memory. Since the allowable dissipation in this memory is of the order of 5 watts, this means that a dissipation in the memory of approximately 10-50 microwatts per bit is desired. At the present time, core memories can operate at levels of 50 to 150 microwatts per bit. For example, the Di/An Model MSA-1A-INT¹ stores 82,000 bits with an average dissipation of 5 watts at a 100 microsecond per word rate. In another case, Sylvania manufactures a ground-based memory, in which no major attempt has been made to minimize power, and 400,000 bits consume 60 watts at a 2-microsecond word rate, or 150 microwatts per bit. With little design effort this could be reduced to about 40 watts. Another factor of ten reduction in power would need new design ideas.

Of course, the core memory may be turned off during nonoperating periods in order to conserve power. However, it is expected that peak data processing loads will occur at the times when other equipments are placing peak power requirements on the solar cells. Thus, the small standby power of a core memory will not be effectively used. Therefore, the standard of performance by which we must gauge other techniques will be the dissipation in the core memory when it is being used at a high rate of computation.

3.4.2 MOS Device A Competing Technique

The only technique that seems to compete with the core memory is the MOS device. MOS devices have been built in memory configurations consuming 33 microwatts per bit² at a cycle time (read or write) of 50 nanoseconds. Furthermore, it appears that, as time goes by, improvements in the characteristic devices will lead to stability in devices with even lower power requirements. The problems of the MOS devices and some of the solutions will be briefly reviewed.

The initial promise of the MOS device was that because of the requirement for fewer diffusions and simpler kinds of diffusion, the devices could be made more cheaply and at greater density.^{3,4,5} While there still seems to be some argument about these two advantages they are pretty well admitted in the industry. A factor of from two-to-five in the cost of diffusion and a similar factor in the density of devices seems to be agreed upon by manufacturers of bipolar and MOS devices.

The early MOS devices were found to be unstable primarily because MOS's are controlled by charge on the surface of an insulator and various mechanisms lead to ionic charges moving into the vicinity of the surface.^{6,7,8,9,10} Continued research in the fabrication of these devices has led to improvements in this stability of about a factor of 20.^{6,11} Originally, a threshold voltage of about 4 volts was found to vary as much as 20 volts depending on previous history and aging. With new devices this variation seems to be easily brought to within one volt, and thus allows usable gain in the device.

While the MOS device itself is very fast, its output is a relatively high resistance while its input is a capacitance. Thus capacitive loading combined with high impedance output have led to existing devices being rather slow in operation. In memory operation, however, the effects of the outside world and its heavy capacitive loadings can be minimized by employing bipolar integrated circuits as input and output devices for the MOS memory array.¹² With these techniques, memory speeds well under 0.1 microsecond have been achieved.

In cost, the MOS elements have not yet reached the point where we can honestly predict a production cost for a few years hence. The present-era yields are relatively low, but improving rapidly, and the research and development is relatively high and decreasing rapidly. The 64-bit device that was employed to build the memory mentioned above¹² will cost about 50 cents per bit next year.² In about two years, industry representatives estimate costs of a few cents per bit, thus making possible MOS memory costs comparable to the core memory. Eventual limits on cost are predicted at about one-eighth of a cent per bit based on present fabrication costs and a 10 percent yield of 64 x 64 devices at high density. These costs would definitely give MOS's the edge for all but the mass-memory field.

In the area of memory density, an 8 x 8 memory array now goes on a chip 80 x 105 mils. J. Schmidt of Fairchild predicts that eventual techniques will allow a 4096 bit plane (64 x 64 elements) to go on a chip 210 x 210 mils. Even with large allowances for packaging, this density is much greater than core memory.

One remaining question which arises is that of long-term reliability. Some data in this regard are favorable. NASA has reported 360 MOS integrated circuits operating for 9500 hours (about one year) without failure.¹³ The sensitivity of the devices to radiation effects is also somewhat questionable. Recent results¹⁴ show that 10^{14} neutrons/cm² cause significant permanent changes in leakage currents and transient doses of 10^7 r/sec cause rather large transient currents. These dosages are quite large and their relation to expected doses on the Solar Probe is not known.

One possible drawback we see with the MOS memory is that of volatility. Since the MOS memory elements are active flip-flops any loss of power or momentary high noise can cause loss of memory. This is not particularly bad if the programs are also stored off-line in a tape machine so that they can be re-entered. It certainly is not detrimental in the scratch pad part of the memory.

3.4.3 Summary

The MOS integrated circuit promises high speed, lower power, and a less costly memory for space applications than is obtainable with core or thin film memories. Its density can be much greater than core when LSI circuits are available. However, the core technology is here now and the MOS LSI is still a few years away. Nonetheless, in the era when we have to make final decisions in this matter, the MOS may well offer the best solution to the memory problem for the CDS. Furthermore, the MOS offers the advantage of speed over the core memory, particularly when we talk about low power. Speed can show up as power and equipment economy in the rest of the CDS. Some questions on reliability and space qualification (particularly under radiation conditions) are yet to be answered.

3.5 INTERFACES

Table 3-2 has been prepared to aid in the definition of the inputs used in routine for the simulation model. The CDS clock and the sun pulse angular reference generator will generally be asynchronous. In order to determine when to sample the experiments, the program will continually address the sun pulse angular reference generator. Experimental data will be inputted to the CDS as a function of the reading obtained from the sun pulse angular generator. Since the cosmic ray, plasma probe, and VLF experiments must be sampled at various angles relative to the sun line for successive spacecraft revolutions, ancillary counters will be employed to do the required bookkeeping. The pulse height analysis of the neutron detector requires almost continuous surveillance of one line of this instrument. The S_5 line of the radio propagation experiment must be sampled 36 times a second. The cosmic ray experiment must be sampled at a rate consistent with the maximum count rates on each line. About 10 input lines dominate the servicing time for the 225 line input. Most engineering measurements require virtually no input time a few days after launch.

Our basic philosophy is to design for optimum processing capability to give maximum practical service to the experimenters, yet leave them the option of using the available service or not. The design recommended multiplexes all inputs from the experiments to the computer. There would be two or more multiplexers to increase the response capability and to allow for graceful degradation in the event of a failure of one multiplexer. This computer would count pulses from the experiments up to a rate of about 10^4 pulses per second directly and 10^7 pulses per second on a sampled basis. It would convert analog inputs to 8-bit accuracy up to 10^4 per second per input. In addition, the computer will accept information from registers in the experiments.

In order to count pulses up to 10^7 pulses per second from an experiment, two modes of counting are needed. For counts below 10^4 per second, the experiment could be sampled at a rate equal to 10^4 per second. When previous data has indicated that these counts have approached or gone above 10^4 per second, the program would go into a second mode. In this mode, the multiplexer would service this one input for a fixed length of time, such as 100 microseconds,

and repeat this at some fixed rate, such as every 10 milliseconds. The 100 microseconds and 10 milliseconds are programmable and hence can be dictated by the experimenter. If the experimenter does not feel that this sampling method will give him true statistics, he will use his own counter; and the counter will be read like any other register in an experiment.

Analog inputs, in some cases, require 7- and 8-bit conversions. If this is done completely within the computer, reasonably sophisticated multiplexing switches will be needed. If, on the other hand, these are converted to pulse width at the experiment, then standard digital circuitry can be used for the multiplexer. In addition to the inputs discussed above, it seems desirable to include a real-time counter in the computer which, when it receives a signal from an experiment, will store the time of occurrence in memory. This time can represent either clock time or time from the last sun pulse.

The micrometeorite experiment requires servicing very infrequently (1-100 times a day). The present probes contain all the digital logic needed in the experiment. Another method is to put very little digital logic in the experiment and, when an event occurs, to interrupt the computer and have the computer completely dedicated to this experiment during the period of this event. The latter implies that no other experiment will get service during a short period ($\sim 100 \mu\text{s}$) after a micrometeorite has entered the detector.

Table 3-2 indicates the approximate number of lines that must be serviced by the computer per second. The servicing requirements are modest except for the pulse height analysis of the neutron detector output and the need to measure time of flight of micrometeoroids. A servicing rate capability of from one data line per millisecond to one data line per 100 microseconds may be required, depending on how things are done. If one line is to be serviced at a rate of 10^4 per second and if each sample is converted to a 4-bit word and stored in one of 16 counters, the machine memory cycle time required would probably be of the order of a few microseconds. The memory cycle time is also obviously affected by the processing algorithms employed.

TABLE 3-2

APPROXIMATE MAXIMUM LINE SERVICING REQUIREMENTS FOR CONTROL COMPUTER

Instrument or Subsystem	Lines	Servicing Rate	Bits/Line	Service Rate Total/Second
VLF Experiment	E _{x1}	1 sample/sec.	6	
	E _{x2}	1 sample/sec.	6	
	E _{y1}	1 sample/sec.	6	
	E _{y2}	1 sample/sec.	6	
	M ₁	1 sample/sec.	6	9
	M ₂	1 sample/sec.	6	
	Filter Command 1	1 sample/sec.	3	
	Filter Command 2	1 sample/sec.	3	
	Solar Aspect Angle	1 sample/sec.	7	
Cosmic Ray Exp.	D ₄	1 sample/sec.	19	
	($\bar{D}_1 D_2 D_3 \bar{D}_4$)	1 sample/sec.	7	
	(D ₁ $\bar{D}_2 \bar{D}_4$)	1 sample/sec.	16	
	(D ₁ D ₂ D ₃ \bar{D}_4)	1 sample/sec.	10	8
	PHA (D ₁ D ₂ D ₃ \bar{D}_4)	1 sample/sec.	5	
	(D ₁ D ₂ $\bar{D}_3 \bar{D}_4$)	1 sample/sec.	13	
	PHA (D ₁ D ₂ $\bar{D}_3 \bar{D}_4$)	1 sample/sec.	7	
	Solar Aspect Angle	1 sample/sec.	7	
Command Decoder	---	1 sample/sec.	1	1
TLM Modulator	---	512 samples/sec.		512
Plasma Detector	Flux Amplitude	16 samples/sec.	7	
	Channel Ident.	16 samples/sec.	3	
	Sun Aspect Angle	16 samples/sec.	4	64
	Voltage Cycle	16 samples/sec.	1	
Stanford Radio Propagation Exp.	S ₁	1 sample/10 sec.	10	
	S ₂	1 sample/5 sec.	6	
	S ₃	1 sample/30 sec.	7	73
	S ₄	1 sample/sec.	6	
	S ₅	36 samples/sec.	6	
	S ₆	1 sample/5 sec.	6	
	Sun Aspect Angle	36 samples/sec.	7	
Triaxial Fluxgate Experiment	E _x	4 samples/sec.	8	
	E _y	4 samples/sec.	8	
	E _z	4 samples/sec.	8	
	Sun Aspect Angle	4 samples/sec.	2	28
	Bias	4 samples/sec.	3	
	Receiver Sensitivity	4 samples/sec.	2	
	Orientation	4 samples/sec.	1	
Neutron Detector	Proton	1/sec.	20	
	Gamma Ray	1/sec.	14	10 ⁴ or 19*
	Neutron	1/sec.	14	
	PHA (Neutrons)	(10 ⁴ /sec. or 16/sec.*)	10	
Spacecraft Operations & Control Subsystem				Very low generally
Micrometeorite Detector	Time of Flight	Priority Interrupt	7	Very low generally. Question exists on ability to interrupt fast enough to retain 0.5 μsec. TOF measurement accuracy

* NOTE: Depends upon whether PHA (pulse height analysis) is done in central computer or within neutron detector.

3.6 INPUT-OUTPUT PROCEDURE

Transferring data across the computer interface is accomplished by assigning register addresses to input-output channels. The problem of interrupt seems to be best solved by a combination of hardware and software. The hardware is the addition of test instruction in the logic unit instruction repertoire.

If the computer has a 16-bit word, we would assign a register address to groups of 16 interrupts. One logic unit (e.g., - LU5) will examine these interrupts as though it were a 16-bit word, and will test for one or more ones in the "register." If one or more of the interrupt lines is high, this instruction will set the overflow flip-flop of the logic unit (LU5) to 1. A second logic unit (e.g., - LU6) will be assigned to perform a Conditional Copy micro-instruction. This micro-instruction will examine the overflow flip-flop of LU5 and, if set, will transfer a new micro-instruction stored in a spare register to logic instruction register 1, the logic instruction register controlling the logic unit, which is updating the program counter. This new instruction will use a different register assignment for the program counter, but will still perform the same type of a function of updating a program counter. The new program counter will contain the address of the interrupt procedure. From this point on, a stored program will determine which device is requesting service and then service the device.

A typical program for this interrupt procedure is shown in Figure 3-3. It should be noted that we enter this subroutine at the second instruction. When the interrupt routine is finished, it will return to the first instruction of the interrupt procedure. This instruction returns control to the original program counter and since the interrupt program counter has been incremented, it will be in proper position for the next interrupt. Figure 3-3 only shows those instructions that determine the device requesting service and will transfer control to the proper input or output subroutine.

3.7 COMMAND LINK

The most critical information path in the CDS is the command link. If this link is broken, then we cannot change modes or "repair" the CDS by reprogramming.

Interrupt Determining Subroutine

- Exit point → 1. COP R_{MP} , IR2
- Entry point → 2. CCP, MD1, PC_{int} , LU4
3. NOP α_{int} lines A routine
- These are interrupt lines B routine (similar set for A).
4. FMO Int lines B, R1
Bit location of interrupt to R1.
5. ADD MD1, R1
6. NOP α_{table} of jump locations.
7. COP, R1, MA2
8. COP, MD2, PC_{int}
- (Gets proper jump location to MA2
Jump to proper I/O routine.

Exit from Interrupt Routine

1. COP MD1, PC_{int}
2. NOP α_{Exit} point
- (α_{Exit} point is instruction 1 of Interrupt Determining Routine Above.)

Notes:

1. R_{MP} contains instruction:
(PC) + 1 → PC, MA1,
the normal main control program.
2. Entry is at instruction no. 2.
3. PC_{int} is name of PC for this routine.
4. Two sets of interrupt lines assumed (A and B)

Figure 3-3. Interrupt Procedure

The most promising solution to the uplink problem seems to be to feed a full 16-bit word directly into two different bus structures. The receiver will collect bits until 20 bits are received. Four of these bits will specify the register to be transferred to, and 16 bits will be the word to be transferred. For reliability, the word will go on to two busses and the word will be transferred to two different locations. The software in the computer will address only one of these locations but may use the other as a check on the validity of the word. The two busses each go to a different set of 16 registers to prevent a failure from taking out a whole bus system.

3.8 RELIABILITY CONSIDERATIONS

The reliability of this design is obtained through the use of pooling techniques instead of equipment duplication. Recovery after a failure is accomplished through reprogramming. The maximum effect of such a failure will be to reduce the computational rate of the system. For real time applications, we can either have an extra logic unit, an extra register, and an extra memory above that needed for normal functions, or else plan to eliminate the least important function to be performed when a failure occurs. A compiler that will do this reprogramming automatically must be programmed.

At first glance, the critical structure seems to be the bussing system, which looks like a telephone switching crosspoint matrix. However, each of these crosspoints consists of an AND followed by an OR gate. A single failure in this structure does not allow both the bus and all the registers to fail. A failure should only disable either the register or the bus but not both. However, to guard against the possibility of catastrophic failure, it is recommended that not all registers be gated onto all busses.

Some failures may be "repaired" without replacing the failed unit. The logic unit may be able to continue with slightly degraded performance after a failure. It is possible in some cases that operations which cannot be performed because of the failure can be replaced by two or three other operations. For example, an ADD might be replaced by a subtract followed by a 2's complement. This combined with the fact that any logic unit can perform the control-

ling functions can make it possible to program around any failure in some cases without loss of computing power.

3.9 ESTIMATED SIZE OF THE CDS

The memory should be divided into at least two parts, and four seems a reasonable selection. In addition to the four memory units, there will be eight logic units, 16 busses, 32 registers, and a central clock. The 32 registers include the memory data registers, the memory address registers, and the logic instruction registers. We are left with 16 general-purpose registers, which are probably too many.

Table 3-3 is an estimate of the flatpack count for a 16-bit version. The flatpacks used for estimating may not exist in their exact form today, but they are well within the component density and pin limitations of today's 14-pin flatpack. For example, Sylvania, Transitron, and Texas Instruments have under production today a 16-bit flip-flop storage unit on a single wafer in a standard 14-pin flatpack.

3.10 CDS DIAGNOSTIC AND CORRECTION PHILOSOPHY

The CDS depends on reprogramming to obtain its high probability of continued performance. The system must contain provisions to test for errors, to diagnose and locate the faulty module, to roll back to a minimum configuration while repair is being performed, and to repair the CDS. Two different types of tests would be used. In programming the various functions performed between the CDS, it is good programming practice to check the results of computation for consistency. For example, if an experiment has a dynamic range from 1 to 100 units, then any computation that results in our sending a reading, or average of readings, to earth outside of this range is inconsistent with physical fact and implies an existence of a fault. In addition to these consistency checks, there should be on-board a program that, upon command, periodically can check whether or not all units are functioning. This check may simply be performing the same task with different hardware, and seeing whether the results match. The degree of sophistication of this checkout is a function of each particular mission.

TABLE 3-3

ESTIMATED FLATPACK COUNT FOR 16-BIT VERSION

<u>Quantity</u>	<u>Function</u>	<u>Count</u>
8	Logic Units	256
32	Registers (16 bits)	256
	8--LIR	
	4--MD	
	4--MA	
	16--General purpose	
16	Busses--2 flatpacks per bus	32
Clock	(Estimated)	<u>32</u>
	Total	576

Diagnosing an error to a completely failed module can be accomplished by means of substitution. (This technique is only possible because of the "pooled" concept.) A set of hardware can be used to try a known problem; and, if a fault exists, substitution of half the hardware will determine in which half the error is. At this point, we then substitute half of this half again; and, by continuation of this halving process, we can zero-in on the faulty module. At this point, we can make a thorough test of the module to see what functions it is capable of performing. The difficult problem in fault location occurs when a unit has an intermittent failure. For these, the only sure solution is persistent checks.

As soon as a failure is detected, it is incumbent on the CDS to revert to a simple mode such as transmitting unprocessed data to earth before fault location and repair is attempted. Since at this point it is not known what module is at fault, it will be necessary that the program to do this be simple and stored in two memories and, in addition, that each of the identical programs uses completely different hardware. In this way, first one and then the other can be tried to find a good rollback program.

Repair is accomplished through substitution of the faulty module with a good module by reprogramming all programs that refer to the faulty module. When the programs are first written, all references to memory and to hardware needed will be done symbolically. An assembler will assign absolute addresses to the symbolic addresses. At the same time, it will produce tables of these symbolic absolute equivalences, and what instructions use what addresses. This information will be kept in a computer on ground; and, in the event of failure, these tables will be searched to find all references to the faulty unit. At this point, uplink commands will be made to change all required addresses. If a tape recorder is on-board with sufficient capacity, then some of these tables can be kept on the spacecraft. In this case, fewer commands will be needed to make these reference changes.

In any case, these commands will enter the computer through the two bus structures as explained in paragraph 3.7. The command is 20 bits, and four of these bits specify to what register to transfer the remaining 16 bits. The rollback program referred to above will be sampling this register from time

to time and through software will make the required changes. In the extreme event, when the written rollback program is not working properly, the 20-bit command will have to specify a logic unit and boot-strap in the program.

3.11 RECOMMENDATIONS FOR FUTURE WORK

More work needs to be done on the design presented in programming and further failure analysis. Even though it appears presently that no single failure can cause complete system failure, the detailed analysis to support this lies ahead.

Some examination of the choice of modules must be made to make sure that a complete module failure will not prevent us from recovery. Even though a module may contain many circuits, it is always possible for the complete module to fail by such means as extreme local heating melting the integrated circuit, or a group of integrated circuits losing power.

We recommend a study of the development of LSI (Large Scale Integration) integrated circuits for this CDS organization. This pooled type organization lends itself naturally to the use of LSI. The packaging of a logic unit in a few "chips" is within today's state of the art. This organization, which uses very few types of circuits, makes LSI development practical, economic, and reliable. The reliability will be increased through reduction of connections and concentrated engineering effort on fewer "chip" types. The flexibility of the universal type, which can be used for any purpose, improves the logistics of spares for replacing failures and for meeting varied needs, both anticipated and unforeseen, of differing missions and from time-to-time in one mission. The use of LSI does introduce one reliability question that must be answered in any study. "Can recovery be made if an entire chip fails?" The answer to this question can affect processor design and chip definition.

3.12 SIMPLIFIED LOGIC UNIT

A simplified logic unit was considered early in the study and has been since discarded in view of the extra memory required to store the micro-instructions. A description of this unit follows. This logic unit consists of

a pair of identical circuits for each bit in the data word,* which has the power to synthesize all useful logic functions of two registers. Each of the two identical circuits consists of eight 4 input AND gates ORed together. The unique feature of the logic unit is the method of control that eliminates the large amount of circuitry normally required to decode the gate specified logic circuitry since the operation is specified explicitly by 16 bits of the logic register.

The logic unit can implement the expressions indicated by Eqs. (3-1) and (3-2).

$$S_n = f_0 \bar{J}_n \bar{K}_n \bar{C}_{n-1} + f_1 J_n \bar{K}_n \bar{C}_{n-1} + f_2 \bar{J}_n K_n \bar{C}_{n-1} + f_3 J_n K_n \bar{C}_{n-1} + f_4 \bar{J}_n \bar{K}_n C_{n-1} + f_5 J_n \bar{K}_n C_{n-1} + f_6 \bar{J}_n K_n C_{n-1} + f_7 J_n K_n C_{n-1} \quad (3-1)$$

$$C_n = f_8 \bar{J}_n \bar{K}_n C_{n-1} + f_9 J_n \bar{K}_n C_{n-1} + f_{10} \bar{J}_n K_n C_{n-1} + f_{11} J_n K_n C_{n-1} + f_{12} \bar{J}_n \bar{K}_n C_{n-1} + f_{13} J_n \bar{K}_n C_{n-1} + f_{14} \bar{J}_n K_n C_{n-1} + f_{15} J_n K_n C_{n-1} \quad (3-2)$$

J and K represent the two inputs to the logic unit. J_n and K_n are the n^{th} bits of these two inputs and C_{n-1} is the C_n of Eq. (3-2) derived from the less significant bits. The sixteen f coefficients of the two equations are either 0 or 1 as determined by 16 bits of the logic register. The recursive term, C_n , makes the output term, S_n , dependent on the outcome of the previous bit computation for those operations requiring a propagating carry such as the carry function in the algebraic ADD instruction. It will be observed by examining Eqs. (3-1) and (3-2) that all 8 min-terms of the three variables J_n , K_n , and C_{n-1} are multiplied by a constant that is either 0 or 1. When the corresponding f value is a 0, that min-term does not appear in the equation;

*A serial computer would only require a single pair. However, this introduces more clocking circuitry.

and when the f value is a one, the min-term does appear. By a Boolean summation of remaining min-terms, the output function is uniquely established. For example, consider the logical Exclusive OR of two registers. The f values for the S_n term are (0, 1, 1, 0, 0, 1, 1, 0) while the f values for the C_n term are (0, 0, 0, 0, 0, 0, 0, 0). Actually $f_8, f_9,$ or f_{15} can assume either a "1" or "0" (don't care condition) and not affect the final answer. Inserting these numbers for f_0 through f_{15} into Eq. (3-1) yields the output expression in min-term form.

$$S_n = J_n \bar{K}_n \bar{C}_{n-1} + \bar{J}_n K_n \bar{C}_{n-1} + J_n \bar{K}_n C_{n-1} + \bar{J}_n K_n C_{n-1}$$

that reduces to the carry (C_{n-1}) independent equation: $S_n = J_n \oplus K_n$.

Another example would be the algebraic addition of two variables, the f_{0-7} values assume (0, 1, 1, 0, 1, 0, 0, 1) and the f_{8-15} values assume (0, 0, 0, 1, 0, 1, 1, 1). For this combination of f's, the generated min-terms lead to the following equations:

$$S_n = J_n \bar{K}_n \bar{C}_{n-1} + \bar{J}_n \bar{K}_n C_{n-1} + J_n \bar{K}_n C_{n-1} + J_n K_n C_{n-1}$$

$$C_n = \bar{J}_n K_n C_{n-1} + \bar{J}_n \bar{K}_n C_{n-1} + J_n \bar{K}_n C_{n-1} + J_n K_n C_{n-1}$$

These reduce to the familiar sum and carry equations:

$$\text{Sum} = S_n = J_n \oplus K_n \oplus C_{n-1}$$

$$\text{Carry} = C_n = J_n K_n + C_{n-1} (J_n + K_n)$$

An instruction for such a machine will require 32 bits: 16 bits to specify the function of the logic unit and four fields of four bits each to specify one of 16 registers into the two input busses and the two output busses of the logic unit. If a 32-bit data word is desired, this would be very con-

venient. If, on the other hand, a shorter data word such as 16 bits seems a better choice, a simple modification will allow the use of two 16-bit words for each of the instructions while using one 16-bit word for data.

In one design using the shorter word length, the logic register of the logic units will have two addresses, one for the lower half and one for the upper half. The fourth logic unit will have to be initially loaded with the operation: complement the least significant bit of logic register 2. The four least significant bits of each logic register will specify one of the two output busses of the logic unit. Thus, the operation of logic unit 4 alternately allows logic unit 2 to select the upper and lower half of the logic register 3. In this manner, two consecutive locations stored in memory 1 will be transferred properly to logic register 3. One more modification is necessary: the machine word should really be 17 bits. The 17th bit will be used to tell whether it is the first half or the second half of the 32-bit instruction. The logic register will contain 33 bits. Each time 17 bits are loaded into one-half of the logic register, the 17th bit of each half will load the same 33rd bit of the logic register. The first time the 33rd bit will be made a zero and the second time a one. When this 33rd bit is made a one, the logic unit will perform its function on the next memory cycle.

3.13 REFERENCES

1. Di/An Controls, Inc., Bulletin 62-45, Boston, Massachusetts.
2. D. Farina, Philco, private communication. See also Reference 12.
3. Warner, R. M., Jr., "A Comparison of MOS and Bipolar Integrated Circuits," Texas Instruments, Inc., Bulletin SC0637, 7 October 1966.
4. Warner, R. M., Jr., "A Comparison of MOS and Bipolar Integrated Circuits," Presented at the NEREM Meeting, November 1966, NEREM Record, pp. 68-69.
5. Petritz, R L., "Technological Foundations and Future Directions of Large-Scale Integrated Electronics," Proceedings 1966 FJCC, vol. 29, Spartan Books, Washington, D. C., pp. 65-87; 1966.
6. Mitchell, M. M., and Ditrick, N. H., "Stability Effects in MOS Enhancement Transistors," Solid State Design, pp. 19-22; November 1965.

This reference shows the effect the circuit terms in one preliminary solution to the problem.
7. IEEE Trans. on Electron Devices, vol. ED-13, no. 2; February 1966.

Most of the articles in this issue refer to the problem of stability in field effect devices.
8. Sah, C. T., and Pao, H. C., "The Effects of Fixed Bulk Charge on the Characteristics of Metal-Oxide-Semiconductor Transistors," IEEE Trans. on Electron Devices, vol. ED-13, no. 4, pp. 393-409; April 1966.
9. Sah, C. T., Wu, S. Y., and Hielscher, F. H., "The Effects of Fixed Bulk Charge on the Thermal Noise in Metal-Oxide-Semiconductor Transistors," IEEE Trans. on Electron Devices, vol. ED-13, no. 4, pp. 410-414; April 1966.
10. IBM Journal of Research and Development, vol. 8, no. 4; September 1964.

Most of the articles in this issue refer to the problem of stability in field effect devices.
11. Wegener, H. A. R., "Silicon Nitride and Silicon Oxide in Insulated-Gate Field-Effect Transistors," Presented at the NEREM Meeting, November 1966, NEREM Record, pp. 70-71.

12. Pleshko, P., and Terman, L. M., "An Investigation of the Potential of MOS Transistor Memories," IEEE Trans. on Electronic Computers, vol. EC-15, no. 4, pp. 423-427; August 1966.
13. Electronics Newsletter: Electronics, p. 26, 7 February 1966.
14. Long, D. M., Fleischer, H. L., and Harrison, S., "Radiation Effects on Insulated Gate Field-Effect (MOS) Integrated Circuits," First Quarterly Report ASTIA 630344 April 1966, Second Quarterly Report ASTIA 635198 July 1966.

SECTION 4

CDS SYSTEM TEST, CHECKOUT, AND EVALUATION

4.1 TEST AND CHECKOUT

The CDS depends on reprogramming to obtain its high probability of continued performance. The system must contain provisions to test for errors; to diagnose and locate the faulty module; to roll back to a minimum configuration while repair is being performed; finally, to repair the CDS. Two different types of tests are used. In programming the various functions performed within the CDS, it is good programming practice to check the results of computation for consistency. For example, if an experiment has a dynamic range from 1 to 100 units, then any computation which results in our sending a reading or average of readings to earth outside of this range is inconsistent with physical fact and implies an existence of a fault. In addition to these consistency checks, there should be on-board a program which upon command periodically can check whether or not all units are functioning. This check may simply be performing the same task with different hardware, and seeing if the results match. The degree of sophistication of this checkout is a function of each particular mission.

Diagnosing an error to a completely failed module can be accomplished by means of substitution. A set of hardware can be used to try a known problem; if a fault exists, substitution of half the hardware will determine in which half the error is. At this point, substitute half of this half again; and, by continuation of this halving process, one can zero-in on the faulty module. At this point, a thorough test of the module can be made to see what functions it is capable of performing. The difficult problem in fault location occurs when a unit has an intermittent failure. The only sure solution is persistent checks.

As soon as a failure is detected, it is incumbent on the CDS to revert to a simple mode such as transmitting unprocessed data to earth before fault location and repair is attempted. At this point, it is not known what module is at fault; it is necessary that the fault-detection program be simple, and

stored in two memories and that each of the identical programs use completely different hardware. In this way, first one and then the other can be tried to find a good rollback program.

Repair is accomplished through substitution of the faulty module for a good module, and by reprogramming all programs which refer to the faulty module. When the programs are first written, all references to memory and to hardware needed will be done symbolically. An assembler will assign absolute addresses to the symbolic addresses. At the same time, it will produce tables of these symbolic absolute equivalences and what instructions use what addresses. This information will be kept in a computer on ground; in the event of failure, these tables will be searched to find all references to the faulty unit. At this point, uplink commands will be made to change all required addresses. If a tape recorder is on-board with sufficient capacity, then some of these tables can be kept on the spacecraft. In this case, fewer commands will be needed to make these reference changes.

These commands will enter the computer through two different bus structures. The receiver will collect bits until 20 bits are received. Four of these bits will specify the register to be transferred to, and 16 bits will be the word to be transferred. For reliability the word will go to two busses and the word will be transferred to two different locations. The software in the computer will address only one of these locations, but may use the other as a check on the validity of the word. The two busses each go to a different set of 16 registers to prevent a failure from stopping a whole bus system.

The rollback program referred to above will be sampling this register from time-to-time, and through software will make the required changes. In the extreme event when the written-rollback program is not working properly, the 20-bit command will have to specify a logic unit and boot-strap in the program.

4.2 CAPABILITIES AND CHARACTERISTICS OF THE MULTIPAC CDS

The following section gives a preliminary estimate of the proposed MULTIPAC CDS capabilities and comparison to Pioneer VI. This work will continue in the contract extension and will be more fully reported in the final report.

Table 4-1 gives an estimate of the gross characteristics of the proposed MULTIPAC data processor as compared to the figures for Pioneer VI. The use of integrated circuits prevents a significant increase in weight and volume with the added complexity, but the MULTIPAC will probably require about three times the power of the simpler machine. The added complexity does not lead to a lowered reliability figure though, partly because of the use of integrated circuits, which reduces the parts count, and partly because of the MULTIPAC redundancy concept.

Table 4-2 is the computed reliability of the MULTIPAC; Table 4-3 is the program which generated it. A mathematical statement of its operation is as follows:

$$RM = e^{-NPC \cdot FAILR \cdot 730 \cdot NOMO}$$

where RM = Module reliability.

NPC = Number of integrated circuits per module.

FAILR = Failure rate per integrated circuit-hour.

730 = Number of hours in a month.

NOMO = Mission time in months.

$$RP = \sum_{NS=NMR}^N \frac{N!}{NS! (N-NS)!} (RM)^{NS} (1-RM)^{N-NS}$$

where RP = Reliability of a module pool.

N = Number of modules in the pool.

NS = Number of surviving modules.

NMR = Number of modules required for operation.

RM = Module reliability (from above).

$$SYSR = RP_1 \cdot RP_2 \cdot RP_3 \cdot RP_4 \cdot RP_5 \cdot RP_6 \cdot RP_7$$

where RP_n = Reliability of module pool n.

SYSR = Series system reliability.

TABLE 4-1

MULTIPAC VS. PIONEER COMPARISON

Pioneer:

Module	Weight (lbs.)	Volume (in ³)	Power (max.)	One-Year Reliability
Digital Telemetry Unit	8.46	260	4.1	[0.5087]
Digital Storage Unit	1.69	70	0.139	
Command Distribution Unit	5.08	251	1.38	unk.
Command Decoder	5.64	189	0.133	unk.
Totals	20.87 lbs.	770 in ³	5.752 w.	<0.5087

Multipac (preliminary calculations):

Section	IC Parts Count	Weight (lbs.)*	Volume (in ³)**	Power (max.)	One-Year Reliability***
Processor	1752	[6.9]	[173]	4.0	[0.67]
I/O and Multiplexer	670			1.5	
Memories (4)	----	≈16	≈400	≈10	unk.
Totals	2422 IC's	22.9 lbs	573 in ³	15.5 w.	<0.67

Multipac (projected estimate):

3500 IC's 25 lbs 600 in³ 18 w. 0.5

* Based upon 500 flat packs per lb mounted and 70% utilization of mounting spaces.

** Based upon 20 flat packs per in³ mounting and 70% utilization of mounting spaces.

***For normal mode operation and based upon a failure rate of 100 x 10⁻⁹ failures per integrated circuit-hour.

TABLE 4-2

SOLAR PROBE CDU RELIABILITY

FAILURE RATE = $100 \cdot 0E-09$ FAILURES PER INTEGRATED CIRCUIT-HOUR

MODULE TYPE	NO. MODULES	NO. MOD REQ	PARTS COUNT
LOGIC UNIT	8	5	171
MEMORY CONTROL	4	3	48
REGISTER	11	5	12
MULTIPLEXER	3	2	176
TELEMETRY INPUT	2	1	24
TELEMETRY OUTPUT	2	1	72
TIMING AND CONTROL	1	1	10

MISSION TIME 12 MONTHS:

REL OF MOD	REL OF POOL	REL OF SYSTEM
.86088358	.98357988	
.95882375	.99037698	
.98954306	.99999990	
.85712115	.94459047	
.97919546	.99956717	
.93887587	.99626384	
.99127826	.99127826	
		.90831328

MISSION TIME 24 MONTHS:

REL OF MOD	REL OF POOL	REL OF SYSTEM
.74112054	.87353112	
.91934299	.96503746	
.97919546	.99999999	
.73465667	.82614285	
.95882375	.99830452	
.88148790	.98595488	
.98263258	.98263258	
		.67357962

MISSION TIME 36 MONTHS:

REL OF MOD	REL OF POOL	REL OF SYSTEM
.63801850	.68036908	
.88148790	.92845363	
.96895608	.99999998	
.62968978	.69017204	
.93887587	.99626384	
.82760772	.97028090	
.97406231	.97406231	
		.41050716

FORTRAN PROGRAM FOR CALCULATING MULTIPAC CDS RELIABILITY

```

100-----FORTAN PROGRAM TO CALCULATE CDS RELIABILITY-----
110-
120-
130----NFACT:  A FACTORIAL FUNCTION-----
140
150 REAL FUNCTION NFACT(J)
160 NFACT=1
170 DO 10 I=1,J
180 10 NFACT = NFACT*I
190 END
195-----PMNS:  BINOMIAL DISTRIBUTION FUNCTION---
200 FUNCTION PMNS (N,MNS,PS)
210 PMNS=0
220 NS=N
230 40 NF=N-NS
240 POS=(NFACT(N)/(NFACT(NS)*NFACT(NF)))*PS**NS*(1-PS)**(N-NS)
250 PMNS = PMNS + POS
260 NS=NS-1
270 IF(NS-MNS) 30,40,40
280 30 END
290-
300-
310-----MAIN PROGRAM-----
320
330 DIMENSION NPC(7), NM(7), NMR(7),RM(7),RP(7)
340 PRINT 100
350 100 FORMAT (21X,27HSOLAR PROBE CDU RELIABILITY,//
360 + "FAILURE RATE:")
370-
380-----INPUT FAILURE RATE + PART COUNTS-----
390 INPUT,FAILR
400 PRINT"NO. MODULES; NO.MODULES REQUIRED, PARTS COUNT:"
410 INPUT,(NM(I),NMR(I),NPC(I),I=1,7)
420 PRINT 200. FAILR*10**9
430 200 FORMAT(///,"FAILURE RATE = ",F5.1,"E-09 FAILURES PER
440 + INTEGRATED CIRCUIT-HOUR")
450 PRINT 300
460 300 FORMAT(//,11X,"NO. MODULES",7X,"NO. MOD REQ",7X,
470 +"PARTS COUNT",/)
480 PRINT 400, (NM(I),NMR(I),NPC(I),I=1,7)
490 400 FORMAT(15X,13,15X,13,15X,13)
500 DO 850,NOMO=12,36,12
510 PRINT 500,NOMO
520 500 FORMAT(///,12HMISSION TIME,X,12,X,7HMONTHS://,16X
530 +10HREL OF MOD,7X,11HREL OF POOL,5X,13HREL OF SYSTEM,/)
540-
550-----CALCULATIONS-----
560 DO 700,I=1,7
570 RM(I)=EXP(-NPC(I)*FAILR*730*NOMO)
580 RP(I)=PMNS(NM(I),NMR(I),RM(I))
590 700 PRINT 800, RM(I),RP(I)
600 800 FORMAT(15X,F9.8,8X,F9.8)
610 SYSR=RP(1)*RP(2)*RP(3)*RP(4)*RP(5)*RP(6)*RP(7)
620 850 PRINT 900,SYSR
630 900 FORMAT(49X,F9.8)
640 END

```

Since the MULTIPAC operates as a group of interchangeable modules connected together by software assignment, successful operation of the MULTIPAC in a particular mode only requires that a sufficient number of each module type be operational. The figures in the "Number of Modules Required" column are those for normal processing capability as presently envisioned. Failure of this mode does not mean mission failure, however, since simpler modes can be implemented using the remaining operational modules and the mission can be continued with only slight degradation in processor capability. Thus, this reliability figure should not be compared to those predicting the likelihood of absolute failure. Another factor which should be noted also is that the failure rate for integrated circuits is arbitrarily chosen, intentionally pessimistic relative to other components for which data is known, until actual circuits are chosen and statistical data on them is available.

The capabilities of the MULTIPAC processor depend on its ability to perform as a stored program data processor. As such it is able to format data and perform calculations with great power and flexibility so long as a sufficient number of program steps are allowed it. This is limited by the real time requirement and the execution time of the program steps. The simplicity and generality of the MULTIPAC hardware, while enhancing reliability, also leads to a very simple machine-level instruction set. Approximately three of these micro-instructions are needed to execute a typical large computer instruction. The low power budget of the data system in the spacecraft demands the use of very low-powered, slow, logic modules. The result is a design conflict between processor capability and power consumption. Therefore, the question of serial vs. parallel machine organization was considered. Some results of examining preliminary designs are tabulated as Table 4-4. It was concluded from these figures that comparable systems in speed and power can be built by using either low speed logic in parallel or high speed logic in serial. The parallel organization has a consistent edge in the speed-power ratio but a disadvantage in parts count and reliability, and the overall power of either serial or parallel organization is excessive for the design goal. One method of meeting the latter is by using mixed logic: high-powered fast logic in critical propagation paths, and slow low-powered logic in areas where it does

TABLE 4-4

EXAMPLES OF SPEED AND POWER FOR VARIOUS TYPES OF MACHINE ORGANIZATION

<u>Logic Type</u>	<u>Serial</u>		<u>Parallel</u>	
	<u>12 bit</u>	<u>16 bit</u>	<u>12 bit</u>	<u>16 bit</u>
Fairchild LPDTuL 4mw/FF: 180 ns 2mw/gate: 65 ns	(based on 1752 pcs)	(based on 2584 pcs)	(based on 8768 pcs)	(based on 11776 pcs)
1) Power consumption	3.4w	4.4w	7.9w	10.6w
2) Clock frequency	1.1mc	1.1mc	0.47mc	0.38mc
3) Microinstruction time	11.6us	15.2us	2.1us	2.7us
4) Macroinstruction time*	46.5us	45.6us	8.2us	8.1us
5) Speed/power ratio**	6.3Kc/w	2.9Kc/w	15.4Kc/w	11.6Kc/w
Philco MwIII RTuL 15mw/FF: 30ns 3mw/gate: 14ns				
1) Power consumption	10.2w	13.6w	14.0w	18.9w
2) Clock frequency	5.4mc	5.4mc	2.0mc	1.7mc
3) Microinstruction time	2.4us	3.1us	0.5us	0.6us
4) Macroinstruction time*	9.6us	9.4us	2.0us	1.8us
5) Speed/power ratio**	10.2Kc/w	7.8Kc/w	35.7Kc/w	29.2Kc/w
Philco MwIII RTuL as above but with Philco MEL(TTL) FF's 2.2mw/FF: 200ns				
	<u>best choice</u>			
1) Power consumption	4.0w	4.8w	10.0w	13.5w
2) Clock frequency	5.0mc	5.0mc	1.4mc	1.2mc
3) Microinstruction time	2.6us	3.4us	0.7us	0.8us
4) Macroinstruction time*	10.4us	10.2us	2.8us	2.5us
5) Speed/power ratio**	24 Kc/w	20.4Kc/w	35.1Kc/w	30.0Kc/w

* Indexed add macroinstruction.

** Rate of performing indexed add macroinstruction per watt of power.

not determine machine operating speed. This is particularly advantageous in serial organization where most of the machine can use slow logic, and only the serial transfer paths determine machine speed. Furthermore, the fewer parts of the serial organization lead to less size, weight, and higher reliability. The final design should be serially, utilizing mixed logic families. It is this configuration which was used to determine the estimates given in Table 4-1.

It can be concluded that the MULTIPAC processor specified here would have the flexibility and processing capability of a slow general-purpose computer having an instruction execution time of 10 to 20 microseconds. This capability can be further increased during machine design and programming by the choice of an optimized instruction set for the most used data formatting routines and by attention to efficient machine language programming.

SECTION 5

EFFICIENT FORMATTING AND FORMAT ASSEMBLY

One of the main tasks of this program has been to establish the telemetry requirements for engineering and scientific data gathered by a solar probe vehicle. Another has been to develop data formats and formatting techniques that not only satisfy the telemetry requirements but that also are efficient and adaptable to changing conditions and requirements both during the course of a mission and also from mission-to-mission. The formatting scheme we have developed meets these requirements by utilizing the programming and data processing capabilities of an on-board stored program digital computer. In Section 5.1 a description of the various formats developed under this program is presented. The results of an analysis that compares the efficiencies of the various Sylvania formats to one another and to Pioneer formats are discussed in Section 5.2. A discussion of the computer program that performs the function of assembling the Sylvania formats from samples of engineering and science data is given in Section 5.3.

5.1 FORMAT DESCRIPTION

Five different formats have been developed and implemented in the simulation model. These are:

- 1) Fixed Engineering (FE)--This format contains only engineering data. The arrangement of engineering words within a frame is fixed.
- 2) Fixed Low Bit Rate Science (FLBR)--This format contains primarily science data, but some engineering and status data is also included. The arrangement of words within a frame is fixed. This format has been tailored to handle science data when the downlink telemetry rate is 64 bits/sec or less.
- 3) Fixed High Bit Rate Science (FHBR)--This format has essentially the same characteristics as the FLBR format, with the exception that it has been designed to work at telemetry rates of 128 bits/sec or more.

- 4) Variable Low Bit Rate Science (VLBR)--This format is designed for use when one or more of the scientific instruments is turned off by ground command. Under such conditions the sampling rates for the functioning instruments are automatically increased and the telemetry format altered to maintain nearly complete utilization of the available down link telemetry rate. This particular format is designed for use at telemetry rates of 64 bits/sec or less.
- 5) Variable High Bit Rate Science (VHBR)--This format has features similar to those of the VLBR format, but is intended for use at bit rates of 128 bits/sec or more.

A detailed description of these formats and a discussion of their utility are given in the following sections.

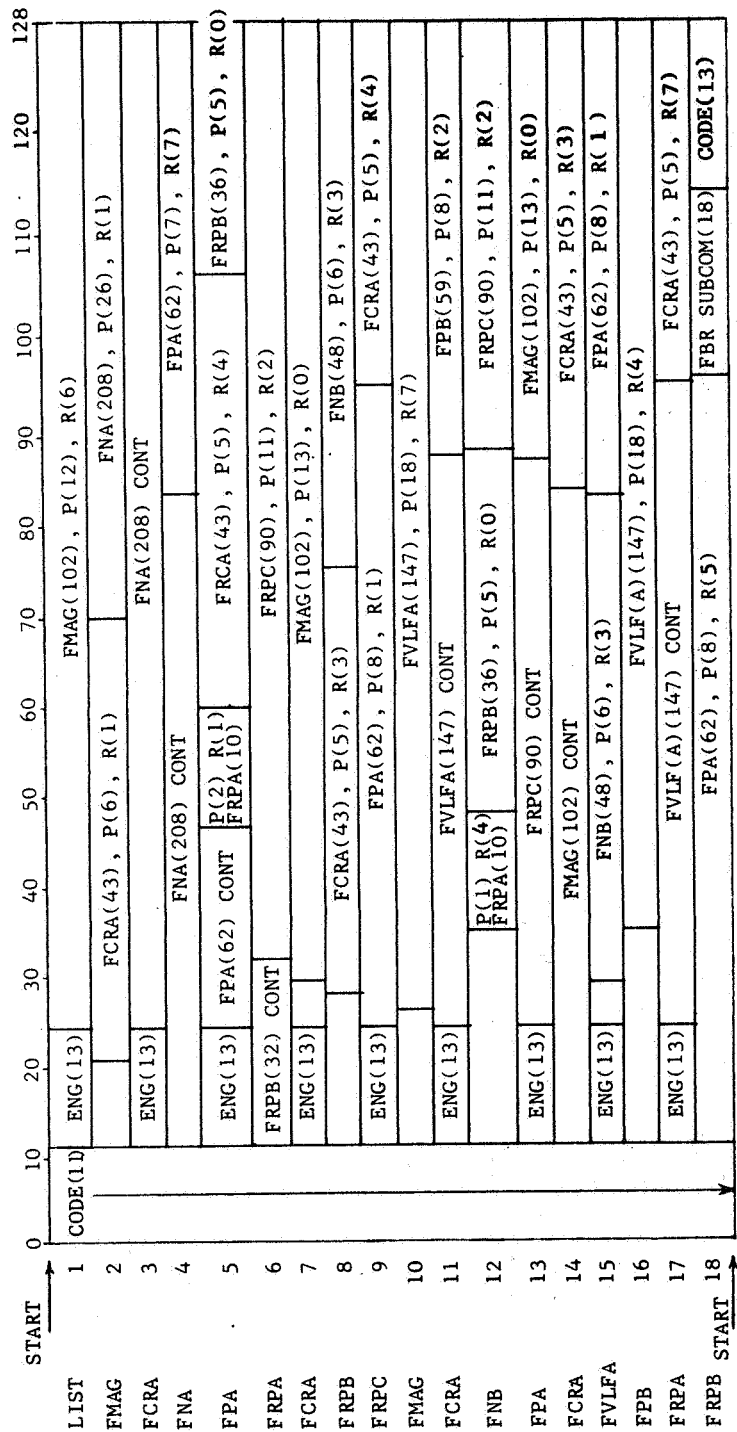
5.1.1 Fixed Engineering Format

One complete frame of this format consisting of six 128-bit lines is shown in Figure 5-1. Each line begins with an 11-bit line sync code, followed by an 18-bit status word. Six different such words appear in each frame. These words contain current information on time, spacecraft rotations, commanded telemetry rate and format, instrument on-off commands, command link bit/rate, storage mode commands, and data processing commands. The remaining 99 bits in each line are used to telemeter out the entire contents of the engineering list. They provide information on the status of all on-board systems. One parity bit for each eight data bits is generated and added to this section of the format. Two complete readings of the engineering list are included in each six-line frame.

5.1.2 Fixed Science Format

The fixed science formats have been designed to include a maximum amount of useful data bits and a minimum amount of identification bits. Most of the data in these formats has one parity bit added for each eight data bits. Engineering and status data are included as subcom along with the science data in each frame.

The fixed low bit rate (FLBR) format is shown in Figure 5-2. It consists of a sequence of 18 128-bit lines. Each line starts with an 11-bit sync code.



FORMAT COMPOSITION (FRAME)

DATA + PARITY BITS	1953
ENG BITS	117
CODE SYNC BITS	198
SUBCOM + CODE(13)	31
TOTAL BITS PER FRAME	2304

Figure 5-2. FLBR Science Format with Parity

In every other line the sync code is followed by a 13-bit engineering subcom word. The first 7 bits of each engineering word form a unique subcom identification (SID) code indicating the source of the last 6 bits which are the engineering data. The SID is necessary because, while the engineering data list is scanned cyclically, only those engineering words that have changed significantly since the last time they were transmitted are picked up and put into the telemetry stream. Since many engineering measurements change very slowly or not at all, this technique results in a net compression of the engineering data. The balance of each line is used to telemeter out the various instrument-associated data blocks in a fixed sequence. One parity bit is added for each eight data block bits, except in the eighteenth line. Here the last five bits of the FPA data block, the FBR subcom block, and the end-of-frame code are transmitted without parity. The FBR subcom block contains a cyclic commutation of the six FBR status blocks, the complete cycle requiring six successive frames.

The fixed high bit rate (FHBR) format is arranged in a similar fashion but with some differences in detail. It is shown in Figure 5-3. The engineering subcom words appear every third line, the entire frame is 24 lines long, the sequence of data blocks is modified, and two FBR subcom blocks instead of one appear in the last line of the frame. At a bit-rate of 512 bits/sec the FHBR format can handle all instrument data taken at the maximum sampling rates, it will provide a complete cycle of FBR status data every 18 seconds, and it will scan all 64 entries in the engineering list at least once every 48 seconds.

5.1.3 Variable Science Formats

The variable science formats have been devised so that the CDS program can automatically modify the instrument sampling rates and the telemetry format to maintain efficient use of the downlink channel capacity in the event that one or more instruments is turned off. As with the fixed science formats, data are assembled as sampled into instrument-associated blocks. The telemetry stream is then generated by outputting these blocks in a cyclic sequence.

LIST	0	10	20	30	40	50	60	70	80	90	100	110	120	128
FMAGA(102)	CODE(11)	ENG(13)	FMAGA(102), P(12), R(6)											
FCRA(43)			FCRA(43), P(6), R(1)		FPA(62), P(7), R(7)									
FPA(62)			FRPA(10) P(2), R(1)		FNA(208), P(26), R(1)									
FRPA(10)			FNA(208) CONT											
FNA(208)			FNA(208)											
FMAGA(102)			FMAGA(102) CONT		FMAGA(102), P(12), R(7)									
FCRA(43)			FCRA(43)		FPA(62), P(8), R(0)									
FPA(62)			FPA(62) CONT		FNA(208), P(26), R(0)									
FNA(208)			FNA(208) CONT											
FMAGA(102)			FNA(208) CONT		FMAGA(102), P(12), R(6)									
FCRA(43)			FMAGA(102)		FCRA(43), P(6), R(1)									
FPA(62)			FPA(62), P(7), R(7)		FMAGA(102), P(13), R(5)									
FMAGA(102)			FMAG(102) CONT											
FCRA(43)			FPA(62), P(7), R(7)		FCRA(43), P(6), R(0)									
FCRA(43)			FPA(62), P(7), R(7)		FRPB(36), P(5), R(2)									
FPA(62)			FNA(208), P(26), R(2)											
FRPB(36)			FNA(208) CONT											
FNA(208)			FNA(208) CONT											
FVLFA(147)			FVLFA(147), P(18), R(5)											
FMAGA(102)			FVLFA(147) CONT		FMAG(102), P(13), R(3)									
FCRA(43)			FMAGA(102) CONT		FCRA(43), P(5), R(6)									
FPA(62)			FPA(62), P(8), R(4)		FRPC(90), P(11), R(6)									
FRPC(90)			FRPC(90) CONT		FNA(208), P(26), R(6)									
FNA(208)			FNA(208) CONT											
FMAGA(102)			FNA(208) CONT		FMAGA(102), P(13), R(4)									
FCRA(43)			FMAG(102) CONT		FCRA(43), P(5), R(7)									
FPA(62)			FPA(62), P(8), R(5)		FBR SUBCOM(36)									
FBR SUBCOM			CODE(17)											

FORMAT COMPOSITION (HALF FRAME)

SYNC(11) 264
 ENG(13) 104
 DATA + PARITY 2651
 SUBCOM + CODE(17) 53
 TOTAL BITS 3072

Figure 5-3. FHBR Sceince Format with Parity

REPEAT

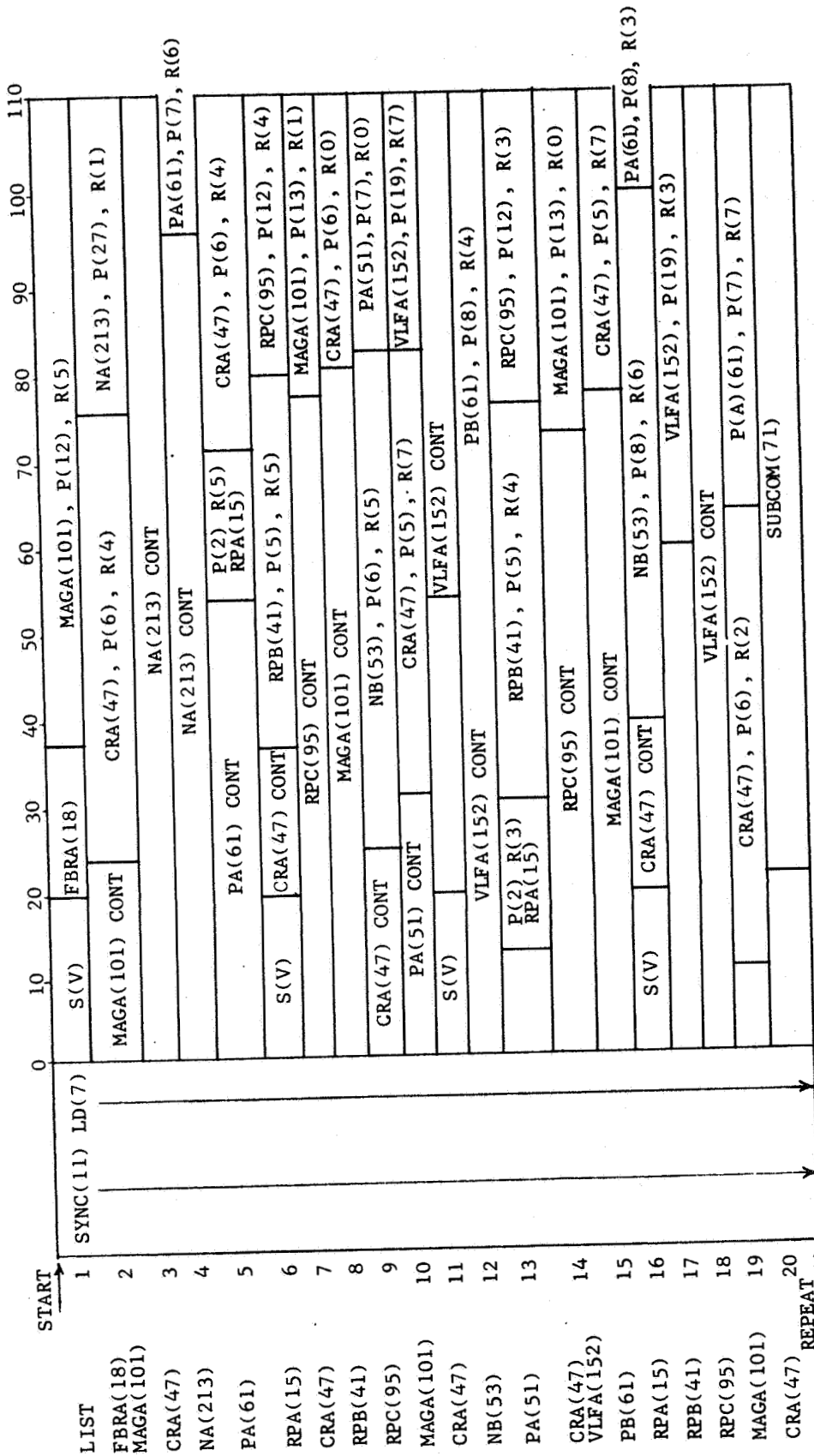
They differ from those used with fixed formats primarily in that a 5-bit block identification (BID) code is inserted at the beginning of each block.*

The variable low bit rate (VLBR) format is shown in Figure 5-4. Each line starts with the usual 11-bit line sync code followed by 7 bits of line definition (LD). The LD is a binary number in the range 0 to 110 (decimal) indicating at which of the following 110 bits in the line the first BID occurs. An LD of zero signifies that no new blocks start in that line. The presence of the LD permits format disassembly on a line-by-line basis, since once a BID is located in a line, foreknowledge of the length of that particular block and its internal arrangement can be used to separate the telemetry data into interpretable groups. The use of so large a fraction (7/128) of the available bit rate to realize this line-by-line disassembly feature may be unnecessary. Improvement in this respect is readily attainable by providing line definition information only for every n^{th} line rather than for every line of the format.

The scientific, engineering, and status data are assembled into each line following the line sync and definition bits. The greater part of this data is assembled by scanning the sequence list (see Figure 5-4), which determines the order in which scientific data blocks are to be picked up. If certain instruments are off, then the corresponding entries in the sequence list are automatically skipped over in the scanning process, and the sampling rates on those instruments that remain on are increased to maintain an overall match between data gathering and downlink telemetry rates. One parity bit is added to each eight science data bits.

In addition to the main sequence of scientific data blocks, other types of data blocks are inserted into the telemetry stream. One such is the subcom block(S). These appear after the LD in every fifth line of the VLBR format to assure that subcom data are transmitted at least this often. Each subcom block consists of 5 BID bits to identify the block, 7 subcom identification (SID) bits to identify the subcom data, and a variable number of data bits. The data may be priority engineering or status data, or if no priority data is available, engineering data picked up from the engineering list in the same way as for the engineering blocks in the fixed science formats. Subcom blocks are also inserted in the bit stream to equalize the CDS input and output data

* See Figure 5-9.



FORMAT COMPOSITION (FRAME)

DATA + PARITY BITS 2041
 SUBCOM 159
 SYNC(1D) + LD(7) 360
 TOTAL BITS 2560

Figure 5-4. Variable LBR Telemetry Format with Parity

REPEAT

rates. The instrument sampling rates are chosen so that the data input rate is always slightly less than can be accommodated by the downlink telemetry rate. As a result there will be times when the next data block in the main block sequence has not been filled with samples and is, therefore, temporarily unavailable. This condition is detected by means of a system of flags, and additional subcom blocks are assembled into the format as filler, until the awaited science block becomes available. The main block sequence is then resumed.

There are two additional types of irregularly and infrequently occurring data blocks that are inserted into the telemetry stream. One of these is instrument-associated subcom. These blocks contain information about the status of particular instruments, which is essential to the interpretation of that instrument's science data. These are inserted as needed immediately before the main sequence block with which they are associated. The other such block type contains processed data. The last three entries in the main sequence list for the VLBR format are of this type. These are handled exactly like other main sequence blocks. However, the instrument on-off flag for these blocks is set to indicate a fictitious off condition except when processed data is available for insertion into the bit stream.

The operation of the variable format assembly program is such that it is not possible to predict or show the exact arrangement of data within the format under all conditions. However, the information necessary for interpreting the received telemetry data is present in the bit stream at all times. The illustration given in Figure 5-4 is only intended to give an approximate picture of what the VLBR format could look like under certain conditions and to demonstrate that a sufficient telemetry bit rate is available to handle the required data flow.

The variable high bit rate (VHBR) format is shown in Figure 5-5. While it uses a different main block sequence list and has subcom blocks regularly inserted in every third rather than every fifth line, it is nevertheless assembled in the same manner as is the VLBR format. Indeed the same program is used to assemble both these formats.

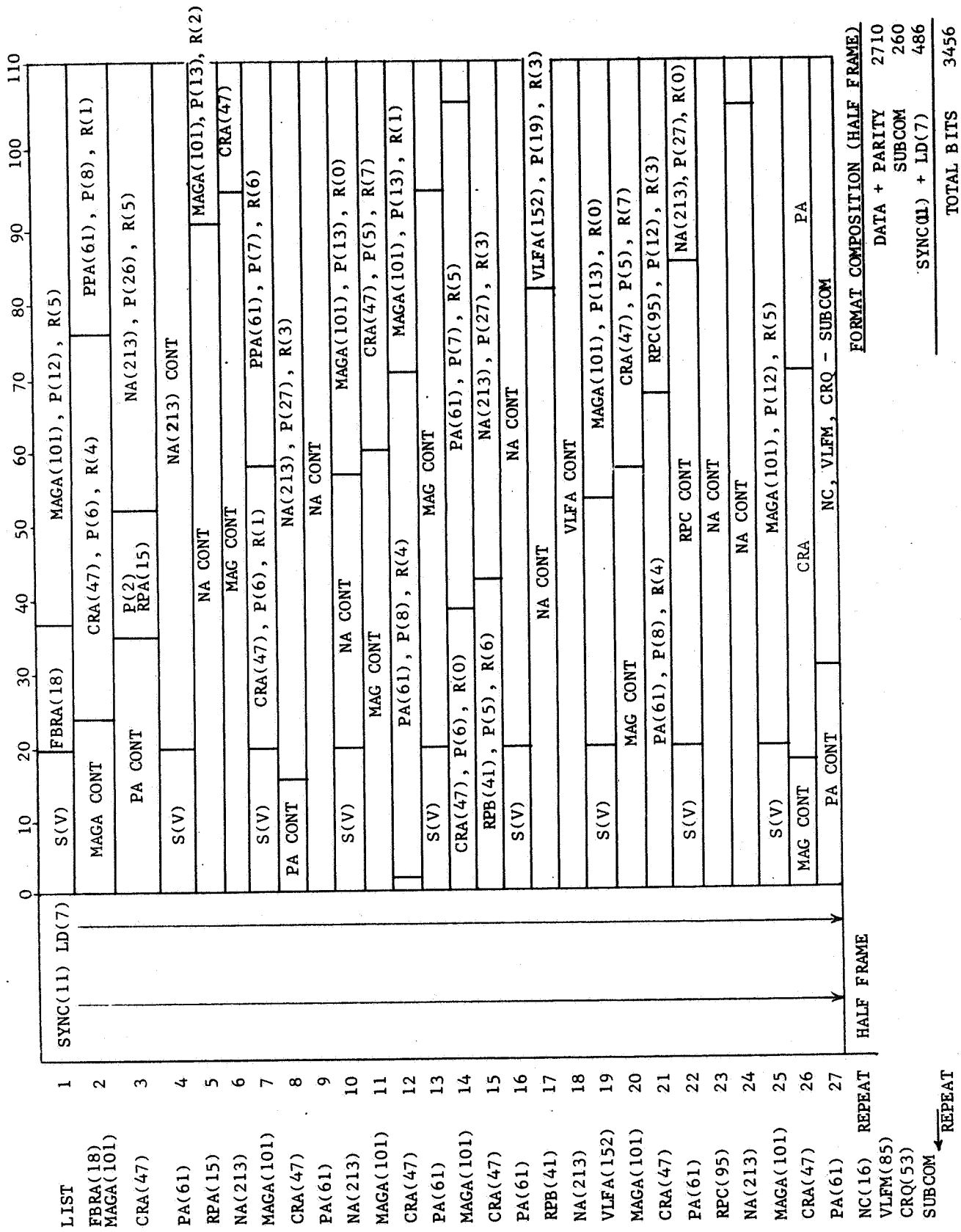


Figure 5-5. Variable HBR Telemetry Format with Parity

5.2 FORMAT EFFICIENCY ANALYSIS

This section presents the results of an analysis performed to ascertain the efficiencies of the various formats described in paragraph 5.1 relative to one another and to the Pioneer Formats B (Science) and C (Engineering) described in Specification PC-121.01, 22 January 1965, issued by the ARC Pioneer Project Office.

The format efficiency has been defined in the following way. All of the formats considered consist of frames in which data gathered in a cyclic sequence from the various sources are assembled, together with sufficient additional information to identify the data source and its status. Each frame is subdivided into a number of lines and each such line begins with a readily identifiable sync-code, which facilitates decommutation of the received telemetry stream. Since such sync bits are an essential part of any telemetry format, they have been excluded from consideration in determining efficiency. They do constitute a significant fraction of the total telemetry stream, however. In the case of the Pioneer formats there are 6 sync bits in each 96-bit line, i.e., 6.25 percent of the total, and in the case of the various formats Sylvania has developed there are 11 sync bits in each 128-bit line, or 8.60 percent of the total. The efficiency of a particular format has been taken to be the ratio of the number of bits of science and/or engineering data in a frame to the total number of bits in a frame excluding line sync bits. Those bits include such things as mode and block ID bits, frame sync bits, parity bits, and data identification bits (time, channel, sector, voltage level, amplitude and frequency range, etc.).

The results of this analysis are presented in Table 5-1 for various instrument on-off conditions. A remark on the influence of parity bits on these figures is in order at this point. In generating efficiency figures for the Pioneer formats, parity bits have not been counted in the total of non-sync bits/frame, whereas they have been included in the corresponding totals for the Sylvania formats. Since all the latter formats include from 7 to 10 percent parity bits, the comparison between Pioneer and Sylvania formats on the basis of Table 5-1 is weighted somewhat in favor of Pioneer. Despite this fact the Sylvania formats are in every case more efficient with parity than are the corresponding Pioneer formats without parity.

TABLE 5-1

TELEMETRY FORMAT EFFICIENCY FOR VARIOUS CONDITIONS

Format	Instruments On-Off				
	All On	1 Off		2 Off	
		Best	Worst	Best	Worst
Pioneer C, Engr.	0.811				
Pioneer B, Sci.	0.648	0.620	0.448	0.517	
Sylvania, Engr.	0.892				
Sylvania, FLBR	0.816	0.715	0.672	0.585	
Sylvania, FHBR	0.828	0.789	0.622	0.729	
Sylvania, VLBR	0.723	0.742	0.724	0.734	0.709
Sylvania, VHBR	0.744	0.751	0.734	0.755	0.694

In the case of the Pioneer and Sylvania fixed engineering formats, the fact that the latter is completely packed and the former is not, more than offsets the decrease in efficiency resulting from the inclusion of parity bits. A graphic comparison of the various science data formats is presented in Figure 5-6 for different instrument on-off conditions. In the case where all instruments are on and functioning the Sylvania fixed low and high bit rate (FLBR and FHBR) formats have efficiencies between 80 and 85 percent while the Pioneer B format is only about 65 percent efficient. Under these conditions the variable formats (VLBR and VHBR) are less efficient than the fixed Sylvania formats, but more efficient than Pioneer B.

In the case where one or more instruments are turned off or malfunctioning, the consequent degradation in all of the fixed formats is in proportion to the number of data bits contributed by nonfunctioning instruments relative to the total number of data bits in the particular format. Certain best and worst cases have been analyzed for both one instrument and two instruments off.

With one instrument off, Figure 5-6 shows that the efficiencies of both variable formats (VLBR and VHBR) remain essentially unchanged independent of

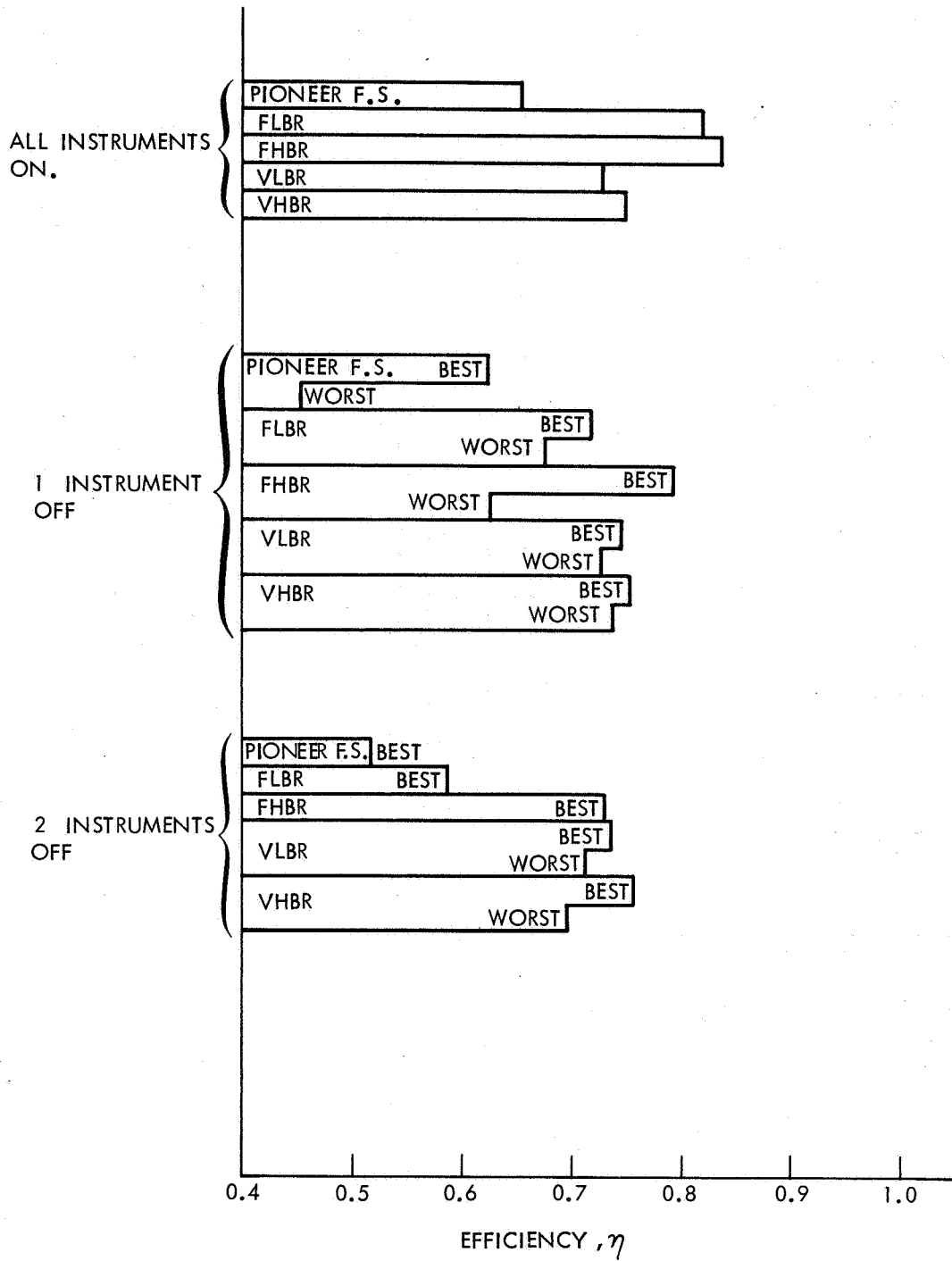


Figure 5-6. Comparative Efficiencies of Pioneer and CDS Science Formats

which instrument is turned off. The fixed formats, both Pioneer and CDS, are degraded and the best and worst efficiencies of any particular format are quite widely spread. Only the best case, FHBR format maintains an efficiency in excess of the worst case VLBR and VHBR formats.

With two instruments off, the two variable formats maintain their efficiencies well, while all of the fixed format efficiencies are further reduced. The best case Pioneer B efficiency is down to about 50 percent. Only the best case FHBR format continues to be comparable in efficiency to the two variable formats.

The conclusions to be drawn from this analysis are as follows:

- 1) The CDS formats are significantly more efficient even when parity is included than are the corresponding Pioneer formats.
- 2) The CDS variable science formats are less efficient than are the fixed science formats when all experiments are working.
- 3) The fixed science format efficiencies drop off rapidly as data is lost due to inoperative instruments. The variable science format efficiencies are, however, relatively constant under these conditions.
- 4) With two or more instruments not functioning, the variable formats are more efficient than are the fixed formats. Even with only one instrument off, the worst case variable formats are significantly more efficient than are the worst case fixed formats.

5.3 FORMAT ASSEMBLY SUBROUTINE

The Format Assembly Subroutine (FAS), which is part of the data system simulation program, is used to generate the five different formats described in paragraph 5.1. It is entered once each spacecraft rotation and in accordance with the commanded format and telemetry rate it outputs a segment of telemetry data consisting of a number of bits equal to the current telemetry rate in bits/sec. It then returns to the calling program. If the duty cycle storage mode has been commanded and a storage mode frame is being assembled, the FAS will be entered a second time before the spacecraft rotation counter is advanced. At such a second entry the FAS generates 512 bits of telemetry data in the VHBR format as part of a storage mode frame. It then returns again to the calling program.

The overall program flow control within the FAS is shown in Figure 5-7. Before the simulation program is started, initial conditions are set which force the flow indicated by double arrows in this figure. Thus at the first entry to the FAS, flow through the storage mode flag test is to the left. The telemetry rate (TLMR) and format in use by the FAS are then compared against those in the command table. If a change has occurred, the new commands will be accepted after the current format line is completed. This delayed execution of new commands assures no disturbance in the periodicity of line sync codes in the telemetry stream. The FAS then assembles a number of bits equal to the TLMR in the selected format. At this point the main mode telemetry assembly for the current rotation has been completed.

The remaining steps in the flow chart of Figure 5-7 are all related to the assembly of storage mode telemetry data. The storage mode of operation is executed on command and results in the assembly of data in the 512 bit/sec VHBR format for on-board storage and later transmission. It is executed on a duty-cycle basis once every 256, 512, 1024, or 2048 spacecraft rotations in accordance with the commanded duty cycle. At the beginning of each storage mode cycle one complete frame of storage mode data is assembled by scanning the VHBR block sequence list once. The actual length of such a frame depends, of course, on which instruments are on.

Referring once more to Figure 5-7, the storage mode change inhibit test serves to prevent the acceptance of a command for discontinuance of the storage mode of operation during a time when a storage mode data frame is being assembled. The variable program linkage labeled ENTRY is used to control flow around that part of the program that accepts and interprets storage mode commands. This bypass path is followed when a storage mode change is inhibited or when no change is detected in the storage mode commands. If the storage mode is turned on, ENTRY connects to point S; if it is turned off, ENTRY connects to point X. In the latter case, the FAS is exited, and will be reentered during the next spacecraft rotation.

If, however, the storage mode is turned on, the next step in the FAS is to test for the start of a new storage mode frame. This decision is based on whether or not the number of rotations comprising the storage mode duty cycle

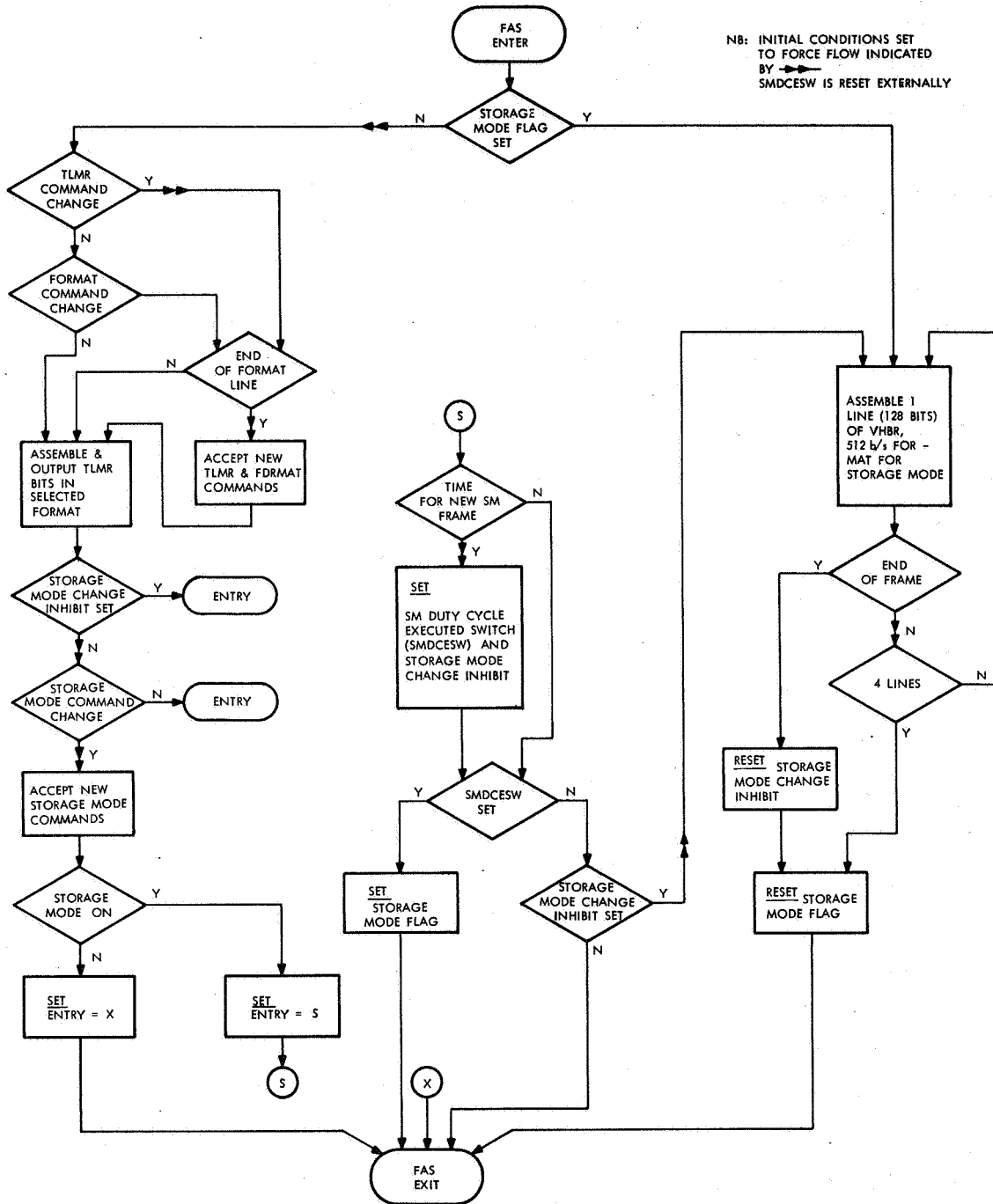


Figure 5-7. Overall Flow Through Format Assembly Subroutine

has been completed since the last storage mode frame was assembled. If not, the FAS will be exited until the next rotation. If it is time to start a new frame, then the storage mode duty cycle execute switch (SMDCESW), the storage mode inhibit flag, and the storage mode flag are all set. Control is then returned to the calling program. At every such return the SMDCESW is interrogated. If it is not set, the simulation advances to the next spacecraft rotation. If, however, it is set, then the data sampling routine is used to gather and arrange in block storage data for the storage mode frame. After this is done, the FAS is reentered. Flow is now to the right in Figure 5-7 because the storage mode flag is set. The storage mode data frame is assembled, one 128-bit line at a time. This assembly terminates either at the end of the frame or after a total of 512 bits has been assembled. In the former case both the storage mode change inhibit and the storage mode flags are reset. In the latter case only the storage mode flag is reset, since the present frame is still incomplete.

The process of gathering and arranging in block storage all of the data that make up a complete storage mode frame may require from one to twelve spacecraft rotations depending on which instruments are on and which are off. When the process has been completed, the SMDCESW is reset by the sampling program. The resetting of the SMDCESW is detected by the test shown in the lower center of Figure 5-7. Since this event will normally occur prior to the complete assembly of a storage mode frame, the storage mode change inhibit flag is used to force program flow directly to the storage mode assembly block until the current frame is complete. Thereafter, the storage mode assembly block is bypassed until it is time to start the next storage mode frame.

The flow chart used in coding the format assembly simulation program is presented in Figure 5-8. Here the steps required to implement the various bit-packing and parity-forming operations, to process the block sequence lists, to effect variable control of program linkages, and to decode and interpret commands are shown in complete detail.

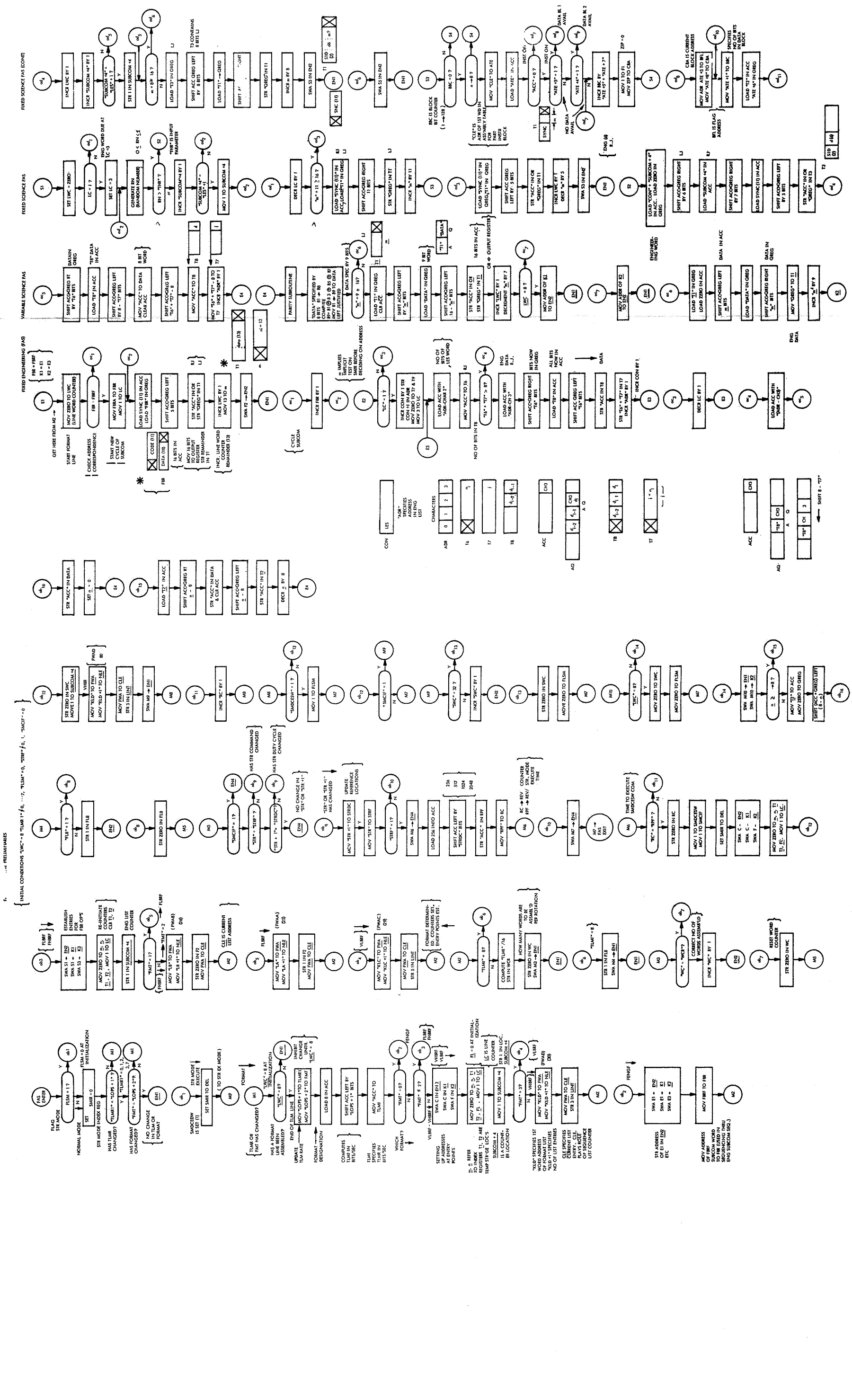


Figure 5-8. Format Assembly Simulation Program

DATA BLOCK DESIGNATION

BID CODE

FBRA	0
SUBCOM B	1
SUBCOM C	2
SUBCOM D	3
SUBCOM E	4
SUBCOM F	5
SUBCOM W	6
MAGA	7
CRA	8
CRQ	9
PA	10
PB	11
RPA	12
RPB	13
RPC	14
NA	15
NB	16
NC	17
VLFA	18
VLFM	19
MAGSA	20
PSAB	21
PS2A	22
PS1A	23
PSB	24
ENG SUBCOM	25
CRSA	26

Figure 5-9. Correspondence Between Data Block and Its BID Code

SECTION 6

ACCEPTANCE TEST PROCEDURE FOR SYSTEM SIMULATION MODEL

The system simulation model and the telemetry formats have been defined in Appendix I and Section 5. The Acceptance Test Plan and Procedure was specified in Section 3 of the August 1966 Progress Report.

The output of the system simulation exercise is a telemetry bit stream. The telemetry format and bit rate are variable (five formats and eight bit rates). The Acceptance Test Plan and Procedure specifies that all bits in the bit stream are to be identified either as data from an instrument sampled at a particular time, parity, subcom, sync, or line definition. The correctness of each processing algorithm is to be confirmed by comparison with previously prepared hand computations, or the processing algorithms are to be isolated from the main program and exercised with special test functions for which the processed output is known.

The data input consists of 10,000 records of simulated data. The first 300 records of data were printed out and employed to confirm that the correct data were appearing at specific locations in the telemetry format. The System Simulation Command Table (Appendix II, Table II-1) is inputted into the program by typewriter. It specifies operational parameters for the simulation. The command and telemetry bit rates and the format are specified for each simulation run. Specific science and engineering lines may be called. Range, system, and antenna gain are input parameters. All experiments may be switched "on" or "off." Specific instrument operating states may be commanded. Five processing subroutines may be executed via the System Simulation Command Table. The storage mode and associated duty cycle can be commanded. Printout of signal-to-noise ratio and bit error rate (BER) can be commanded. Each simulation run is executed over a specific number of spacecraft revolutions. Provisions are available for reinitializing the counters and storage flags each simulation run, to facilitate the checkout procedures. Typewriter code 37 is the signal for the program to execute.

A formal checkout procedure was prepared for the Acceptance Test. The checkout procedure was employed in debugging and exercising the simulation program. The checkout procedure is detailed in Appendix II.

SECTION 7

SYSTEM SIMULATION ANALYSIS AND EVALUATION

The system simulation defines the basic and ancillary functions that a stored program computer can provide the experimenters. The external characteristics of the CDS are dominated by power, weight, volume, component "state-of-the-art", and reliability considerations. The system simulation analysis tailors the simulation results to a computer which is dictated by component state-of-the-art and power budgets.

FORTRAN was agreed upon as the universal language to be used for the simulation program. However, FORTRAN is grossly inefficient for some program segments; machine language coding has been used for these segments.

7.1 PROGRAM MAP

The simulation program requires 15,900 CDC-3200 memory locations. The main program comprises 9600 instructions. 4800 storage locations are used for constants, tables, and intermediate storage; 1000 of these storage locations are reserved for the spectral analysis subroutine. 1450 locations are required for call subroutines; 700 of these are for the printout subroutines. 1000 storage locations are used for ancillary subroutines such as sine, cosine, error function, random number generator, etc. Six hundred of the 9600 instructions are required for the processing subroutines. Program segments (901) → (902), (902) → (905), (905) → (940), (940) → (957), (957) → (964), (964) → (935), and (935) → (4015)* require 115, 312, 717, 648, 1177, 2452, and 2503 instructions, respectively.

The simulation program is coded with approximately 2500 FORTRAN instructions. A close scrutiny of the flow charts and program listings leads to the conclusion that the entire instruction list could be coded in approximately 2500 machine language instructions. The 4800 storage locations can be reduced to 3800 by eliminating the spectral analysis subroutine. It is estimated that the storage locations required for constants, tables, etc., can be reduced to

* Specified in Figure 7-1.

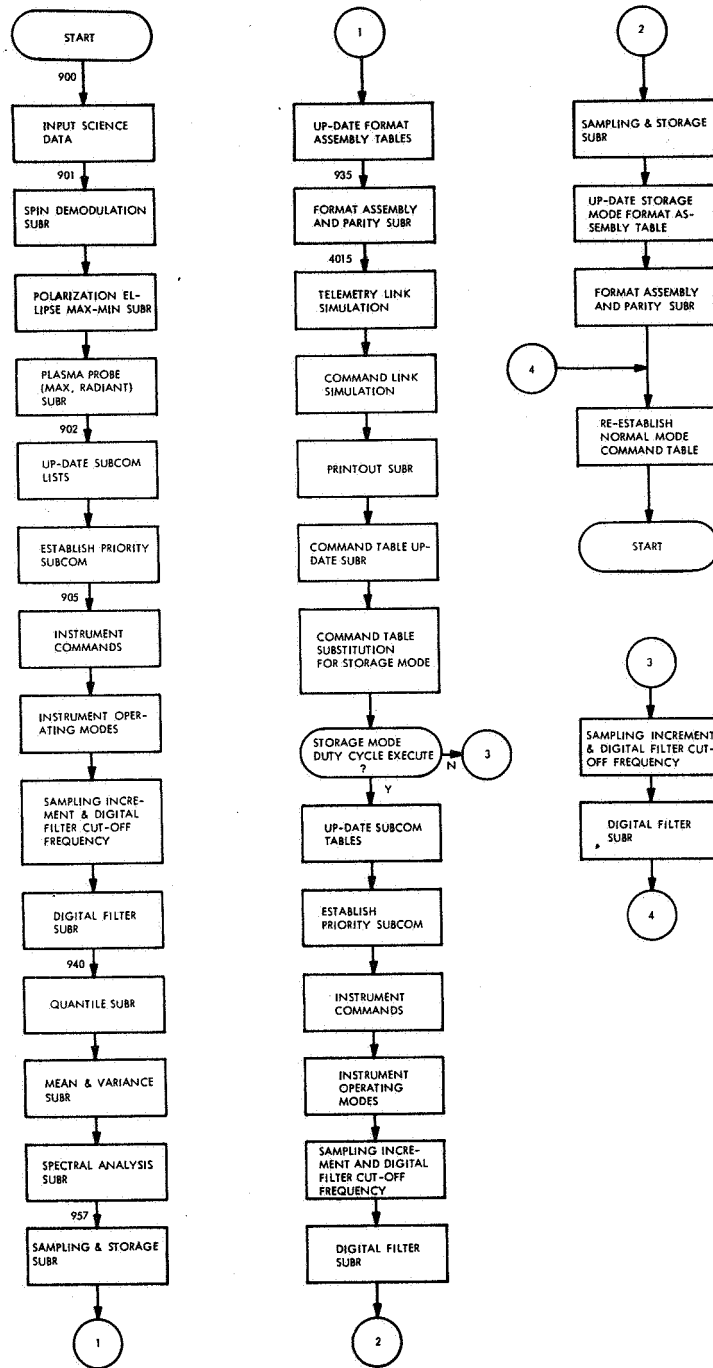


Figure 7-1. Program Routine Executed Each Spacecraft Revolution

1500 by improved flow charting and consolidation of the format assembly and other ancillary tables. Thus, the entire program can be packed in 4000 CDC-3200 storage locations.

7.2 EVALUATION PROCEDURES

The program execution times from instruction list entries (901) → (902) → (905) → (940) → (957) → (964) → (935) → (4015) have been "clocked." The program segment (901) → (902) starts after data input and includes the RP S5 line maximum and minimum subroutines, the PP maximum subroutine, and the spin demodulation subroutine. Program segment (902) → (905) updates subcom tables and various spacecraft revolution counters. The program segment (905) → (940) includes updating priority subcom, indexing instrument-associated subcom in accordance with command table requests, and establishing quadrant, sector, and voltage flags; it also includes the sampling period computation subroutine and the digital filter subroutine. The program segment (940) → (957) includes the quantile, mean and variance, and spectral analysis processing subroutines. The program segment (957) → (964) includes the high bit rate sampling and packing subroutine (HBRSP). The program segment (964) → (935) includes the low bit rate and storage mode packing and sampling subroutines and updates the format assembly tables. The program segment (935) → (4015) constructs the format.

Table 7-1 shows the "clocked" execution times for specific program segments in the simulation program for telemetry rates of 512 and 1024 bits/sec. There is no significant difference in execution time for the variable and fixed science formats. The FAS (Format Assembly Subroutine) dominates the time required at high bit rates.

The HBRSP subroutine, program segment (957) → (964), requires between 60 to 90 milliseconds per second for execution at bit rates exceeding 512 bits/sec. This subroutine comprises 1177 instructions. However, there are 45 calls to FILL and 19 calls to MOV at high bit rates during the execution of this subroutine. The FILL and MOV subroutines comprise 187 and 48 instructions, respectively. The number of instructions executed for this subroutine exceeds 12,000 per second at high bit rates. The average execution time per CDC-3200

TABLE 7-1

EXECUTION TIME PER PROGRAM SEGMENT IN MILLISECONDS

PROGRAM SEGMENT	VHBRF; TLMR = 512 B/S				VHBRF; TLMR = 1024 B/S			
	ALL PROC "ON"		ALL PROC "OFF"		ALL PROC "ON"		ALL PROC "OFF"	
	UB	LB	UB	LB	UB	LB	UB	LB
(901) → (902)	90	60	90	60	90	60	90	60
(902) → (905)	30*	0	30	0	30	0	30	0
(905) → (940)	90	60	90	60	90	60	90	60
(940) → (957)	30	0	30	0	30	0	30	0
(957) → (964)	90	60	90	60	90	60	90	60
(964) → (935)	30	0	30	0	30	0	30	0
(935) → (4015)	360	240	360	240	720	480	720	480
TOTAL TIME	720	420	720	420	1080	660	1080	660

*Minimum time resolution is 30 milliseconds.

instruction is 2.5 microseconds. The total execution time is estimated to be greater than 30 milliseconds. The measured value is 60 to 90 milliseconds. An efficient code has been written for program segment (957) → (964). The efficient instruction list contains 600 instructions and requires 1500 microseconds per second for execution. The time ratio improvement obtained by efficient coding ranges from (20-45)/1 for program segment (957) → (964).

Five of the 19 call subroutines employed are executed frequently, and require approximately 7 percent of the computer running time at 512 bits/sec. These are called KSHIFT, K2SHIFT, MOV, FILL, and PARITY. Counts have been made of the average number of times the call routines are executed per second at a telemetry rate of 512 bits/sec. Table 7-2 shows the number of times each call routine is employed per second for a science format and the time required per call.

TABLE 7-2
CALL ROUTINE/SECOND

<u>Call Routines</u>	<u>No. of Calls/Second</u>	<u>Time/Call (Microseconds)</u>
KSHIFT	30	45
K2SHIFT	212	75
MOV	19	100
FILL	45	400
PARITY	50	600

Table 7-3 shows the percentage of time instructions executed for efficient call subroutines. Fifty-five percent of the time is spent loading and storing; 20 percent of the time is spent shifting; and the remaining time is dedicated to ancillary functions such as indexing, testing, and branching.

TABLE 7-3
INSTRUCTIONS EXECUTED FOR EFFICIENT CALL SUBROUTINES

<u>Instructions</u>	<u>% of Time Executed</u>
LDA (Load A)	30
STA (Store A)	19
SHA (Shift A)	13
SHAQ(Shift AQ)	7
SWA (Store Word Address)	6
INA (Increase A)	4
AZJ (Test and Branch)	2
LDQ (Load Q)	2
STQ (Store Q)	2
STI (Store Index)	2
UJP (Unconditional Jump)	2
ETC	2

When the call subroutines are reprogrammed efficiently in machine language, the execution time is reduced by a factor of 7 for the KSHIFT and K2SHIFT subroutines, by a factor of 5 for the MOV subroutine, 28 for the FILL subroutine, and 14 for the PARITY call subroutine.

The instructions executed for an efficient machine language program have been analyzed for the HBRSP subroutine. The percentages of time individual instructions are executed for the HBRSP subroutine are shown in Table 7-4. in CDC-3200 instruction code.

TABLE 7-4
INSTRUCTIONS EMPLOYED BY VARIABLE AND FIXED
SCIENCE SAMPLING AND PACKING SUBROUTINES

<u>Instructions</u>	<u>% of Time Executed</u>
LDA (Load Accumulator)	28
AZJ (Test Accumulator and Branch)	21
STA (Store Accumulator)	11
STQ (Store Contents of Q Register)	10
SHAQ(Shift AQ Registers)	10
LDQ (Load Q Register)	10
SHA (Shift Accumulator)	7
INI (Increase Index)	2
ADA (Add to Accumulator)	1

Table 7-4 is representative of the instructions employed by the variable and fixed science sampling and packing subroutines. Approximately 40 percent of the time is dedicated to loading the accumulator or Q register. Approximately 25 percent of the time is spent in indexing, testing, and branching; 15 percent of the time is spent shifting, and 20 percent storing. The arithmetic computations are negligible.

Selected portions of the FAS subroutine have been examined. The call subroutines used by the FAS subroutine are dominated by K2SHIFT and KSHIFT. The FAS subroutine is unsuited for FORTRAN. Detailed scrutiny of statistically selected segments of this program have produced a lower bound to the time-

expansion factor relative to an efficiently encoded machine language version of the subroutine which is $20 \rightarrow 45/1$. To arrive at a precise number would require coding the flow from (935) \rightarrow (4015) in CDC-3200 language and "clocking" the execution time. It has already been mentioned that the time expansion factor for the program segment (957) \rightarrow (964) is $20 \rightarrow 45/1$. This means that 70 percent of the program has a time expansion factor of at least 20 to 1.

The various processing subroutines require approximately 15 percent of the program time at TLMR = 512 bits/sec. The subcom identification, tagging, and format assembly and command table updating require an additional 15 percent at TLMR = 512 bits/sec. All or some of the processing subroutines can be eliminated if a time bind develops. A minimum-system model configuration is concerned only with sampling, packing, format assembly, and outputting.

If the spin demodulation, plasma probe maximum radiant, and radio propagation experiment S_5 maximum and minimum subroutines are eliminated, program segment (901) \rightarrow (902) requires no program time. If the subcom tables are only updated every minute, for example, the program segment (902) \rightarrow (905) will average less than one millisecond/second of running time each spacecraft revolution. The program segment (905) \rightarrow (940) updates instrument-associated subcom, executes commands, computes the sampling instant, and filters magnetometer data. A rollback to a minimum configuration would result in the elimination of the digital filter subroutine. This would reduce the execution time for this program segment to less than 10 milliseconds. Since this program segment is dominated by test and move instructions, more efficient coding would reduce the original execution time to less than two milliseconds for the program segment (905) \rightarrow (940). If the quantile, mean variance, and spectral analysis subroutines are eliminated, program segment (940) \rightarrow (957) will require no execution time. The estimated program execution time for the rollback configuration is approximately $10 \rightarrow 20$ milliseconds for a TLMR of 512 bits/sec. Note that only the FAS subroutine execution time will increase for telemetry rates of 1024 bits/sec and higher. This is because the experiments are being sampled at their maximum rates at 512 bits/sec and the program recycles only once a second regardless of the telemetry rate. The time to execute the FAS subroutine doubles as the TLMR goes from 512 to 1024.

If the average ratio of macro-instructions for the CDS/CDC-3200 is 10/1, the CDS will require 100 → 200 milliseconds per second to service a rollback configuration at 512 bits/sec. The average macro-instruction execution time for the CDC-3200 for the simulation program is 2.5 microseconds. At a telemetry rate of 1024, the execution time is approximately 150 → 400 milliseconds per second. A more accurate estimate requires recoding the simulation model in machine language.

7.3 SYSTEM SIMULATION RESULTS AND CONCLUSIONS

The system simulation results have been evaluated in terms of the CDS memory and macro-instruction execute times which are practical in view of the physical constraints on power, volume, weight, and reliability.

The "map" of instruction and memory locations and associated analysis reveal that the CDS storage requirement is approximately 6000 words of 16 bits each.

The FORTRAN program compiled into 9600 CDC-3200 instructions. By clocking the execution times of specified program segments and then writing an efficient machine code for these segments, it was found that time expansions of 20 → 45/1, due to the way the FORTRAN program compiled into CDC-3200 language, were the rule rather than the exception. If simple operational and procedural changes are made, the entire program can be executed in approximately 10 → 20 milliseconds at 512 bits/sec and 15 → 40 milliseconds at 1024 bits/sec. A revised flow and complete CDC-3200 language program would be needed to refine the estimates. If the ratio of CDS/CDC macro-instruction execution time is 10, the CDS will be occupied 100 → 200 milliseconds/second with execution of the instruction list. The input/output servicing time requirements may add another 100 milliseconds/second to the execute time requirements at high bit rates. In addition, the executive routine which assigns tasks in time sequence may add another 100 milliseconds/second to the time required. This implies that the CDS machine may be busy 350 → 600 milliseconds/second at 1024 bits/sec.

A machine word size of 16 bits appears to be appropriate. Double precision is required for the processing algorithms. Virtually all data is specified by less than 16 bits. The principal advantage of a 16-bit word size lies in power conservation. A four-bit operation code and a 12-bit address field are specified. Each 16-bit data word has 15 bits for data and one bit for sign. Since there is a tendency to underestimate storage requirements, the total storage may end up in the 100,000- to 200,000-bit range.

The postulated machine instructions include add, subtract, move, shift left or right, test for zero and branch, mask, read, and write; mask, move, and pack instructions occur frequently during format assembly. Multiply and divide instructions occur very infrequently and can probably be provided by software.

Rough calculations based on the assumption that the CDS will require approximately 1500 logic gates and 100,000 bits of memory yield the following results. The CDS would have approximately the same weight and size as the present Pioneer data system (weight = 20 pounds, volume = $1/2 \text{ ft.}^3$). It would require 2 to 4 times as much power (10 → 20 watts). It would support a telemetry rate of 8 → 2048 bits/sec. In addition to performing all of the functions of the current Pioneer data system, it would provide data processing and variable formatting capabilities.

To have been able to give explicit answers which would have made the CDS design seem comfortably obtainable would have required that the simulation be tailored to a postulated CDS design. The most significant answer we can offer for the CDS design is that the designer must squeeze every bit of performance out of the system within the restraints. The present system simulation holds to a reasonable degree for a specific set of experiments. The CDS must accommodate a wider breed of experiments than those postulated for the simulation model.

APPENDIX I

SYSTEM SIMULATION MODEL

The Data Simulation subroutine generates data for exercising the system. The Simulation Model Summary outlines the program cycle executed each spacecraft revolution. The Detailed Flow Chart description of the Simulation Model verbalizes flow chart logic.

I.1 DATA SIMULATION SUBROUTINE

The Data Simulation subroutine generates data representative of what has been measured in the solar environment for exercising the simulation model. Estimated temporal and dynamic variations of the solar environment have been incorporated into the Data Simulation Subroutine. The estimates are based on insufficient experimental evidence and theories that require experimental evaluation.

The output of the Data Simulation subroutine is a list. Data samples are ordered in the list independently of the temporal sampling sequence.

I.1.1 Temporal Sampling

A spoke-wheel angular reference generator (spacecraft (S/C) angular reference clock) divides a S/C rotation of 360° into 128 equal sectors. As the S/C rotates, the sun is successively scanned by the magnetometer, plasma probe, cosmic ray telescope, VLF experiment, neutron detector, and radio propagation experiment.

The sunline for each scientific instrument corresponds to the (S/C)AC indicated in Table I-1. The sunline for the magnetometer corresponds to the instant that the S/C Angular Count, (S/C)AC, equals zero.

TABLE I-1

SUNLINE CORRESPONDING TO S/C ANGULAR COUNT ((S/C)AC)

	Magne- tometer Sunline	Stanford Rad. Exp. Sunline	Neutron Detector Sunline	VLF Exp. Sunline	Cosmic Ray Sunline	Plasma Probe Sunline
(S/C)AC Modulo 128	0	107	85	64	43	21

The sunline for the radio propagation experiment is located $(107/128) \cdot 360^\circ$ from the magnetometer sunline reference.

The (S/C)AC, in conjunction with spacecraft rotation counters, specify the instants at which data lines are sampled. The magnetometer, radio propagation, and neutron detector experiments are sampled at the (S/C)AC promulgated in Table I-1 each S/C revolution; the VLF, cosmic ray, and plasma probe experiments are sampled cyclically. The VLF experiment is sampled at

$(S/C)AC = \left[64 + \eta_4 \frac{128}{4} \right] \text{ modulo } 128$. η_4 is a quadrant counter; it is incremented every spacecraft rotation, tested for $\eta_4 = 4$, and reset to zero as necessary. The cosmic ray experiment is sampled at $(S/C)AC = \left[43 + \eta_{16} \frac{128}{16} \right] \text{ modulo } 128$. The η_{16} counter is incremented each spacecraft revolution, tested for $\eta_{16} = 16$, and reset as necessary. The Plasma Probe experiment is sampled at $(S/C)AC = \left[21 + \eta_{15} \frac{128}{15} \right] \text{ nearest integer modulo } 128$. The η_{15} counter is incremented every spacecraft revolution, tested for $\eta_{15} = 15$, and reset as necessary.

If the angular sectors are not equal, the (S/C)AC at which the sampling takes place is specified by a more complicated expression than exhibited. For the purpose of generating data to exercise the simulation model it has not seemed necessary to create a one-to-one correspondence between the sampling instants at which data is generated and the time at which it would be sampled for a real instrument.

Since it is occasionally necessary to sample more than one instrument at a given (S/C)AC and S/C rotation count, a priority order is assigned in Table I-2.

TABLE I-2

PRIORITY ORDER FOR SAMPLING
WHEN MULTIPLE SAMPLES ARE REQUIRED FOR A (S/C)AC

Priorities

- (1) Stanford Radio Propagation Exp. Line S₅
- (2) Magnetometer
- (3) Stanford Radio Propagation Exp. Lines S₁, S₂, S₃, S₄, S₅
- (4) Neutron Detector
- (5) VLF Experiment
- (6) Cosmic Ray Experiment
- (7) Plasma Probe Experiment

The magnetometer lines defined in Table I-3 are sampled uniformly four times per S/C revolution. The plasma probe lines are sampled once per S/C revolution; 15 different azimuth sectors are sampled sequentially (one sector per S/C revolution). The cosmic ray lines are sampled once per S/C revolution. On successive revolutions, the azimuth at which a sample is taken is stepped; it has been assumed that the measurement cycle will be completed in 16 S/C revolutions. The VLF data lines are sampled once per S/C revolution; the measurement cycle is four S/C revolutions. The neutron detector lines are sampled once per S/C revolution. The Stanford Radio Propagation Experiment lines, S₁, S₂, S₃, S₄, and S₆, are sampled once per S/C revolution. The S₅ line is sampled 128 times per S/C revolution.

TABLE I-3
DATA INPUT SPECIFICATIONS

Instruments	Data Lines	Bits per Line	Sampling Rate per Line
Magnetometer	B_x, B_y, B_z	8	4 times per S/C revolution
Plasma Probe	$E_1 \dots E_8$	7	once per S/C revolution
Cosmic Ray Exp.	2 (PHA) 5 counters	(5,7) (9,9,13,7,17)	once per S/C revolution once per S/C revolution
VLF Exp.	$E_{x_1}, E_{x_2}, E_{z_1}, E_{z_2},$ M_1, M_2	6	once per S/C revolution
Neutron Detector	16 (PHA) 3 counters	10 (14,14,20)	once per S/C revolution once per S/C revolution
Stanford Radio Prop. Exp.	S_1 (counter) (S_2, S_3, S_4, S_6) S_5	10 6 6	once per S/C revolution once per S/C revolution 128 times per S/C revolution

I.1.2 Data Generation

The data word has the general mathematical form promulgated in Table I-4. The waveform parameters are stipulated by computer cards during initialization. The $k_1^{(i)}$, $k_2^{(i)}$, and $k_3^{(i)}$ are constants. The parameters $\omega_0^{(i)}$, A_i , and ξ_i are waveform parameters; $\omega_0^{(i)}$ is the instantaneous radian frequency. The waveform parameter, Δ_i , is a random variable that perturbs the instantaneous radian frequency, $\omega_0^{(i)}$; ξ_i is a random phase. The "i" subscript or superscript denotes the "ith" data line or instrument. (RV) is a uniformly distributed or Gauss random variable.

The Data Simulation subroutine generates data at the time specified by (S/C)AC modulo 128; the temporal sequence is recorded in the science data list. The correspondence between the temporal sampling sequence and the science data list is promulgated in Table I-5.

TABLE I-4 (cont.)

Experiment	Data Line	Samples (S/C) ~ Word	Waveform Parameters					Azimuth Sector			
			$k_1^{(i)}$	Δ_i	$\xi_i^{(i)}$	$k_2^{(i)}$	$k_3^{(i)}$		$\omega_0^{(i)}$		
Neutron Detector (contd.)	Neutron Counter	1	14	-	-	-	-	-	-	-	
	Gamma Ray Counter	1	14	0	-	-	1	0	0	-	
	Charged Particle Counter	1	20	0	-	-	1	0	0	-	
Stanford Radio Prop. Exp.	S ₁	1	10	4	gauss m=0 $\sigma = .0628$	0	0	0	16	.00628	<p>ξ_i is to be changed each S/C revolution. Set (A + B = 64, A = 48, B = 16).</p> <p>NOTE: * Each data word formed from A sin (6.28 t) + B sin (6.28 t + ξ_i) where ξ_i is selected from un(d.d) (0-0.1); A and B are parameters.</p>
	S ₂	1	6	16	gauss m=0 $\sigma = .0628$	0	0	48	0	6.28	
	S ₃	1	6	6	6	0	0	0	0	0	
	S ₄	1	6	6	6	0	0	0	0	0	
	S ₅	128	6	6	*	0	Un(d.d) 0-0.1	0	0	6.28	

TABLE I-5

STORAGE LOCATION CORRESPONDENCES

Experiment	Ancillary Counters	(S/C)AC Reference	Science List Storage Locations
Radio Propagation (S ₅)	--	(0,1,2, ... 127)	LIS (58-185)
Magnetometer	--	LAS (0), 0; LAS (1), 32; LAS (2), 64; LAS (3), 96	LIS (0-2; 3-5; 6-8; 9-11);
Radio Propagation (S ₁ ... S ₄ , S ₆)	--	LAS (5); 107	LIS (53-57; S ₁ , S ₂ , S ₃ , S ₄ , S ₆)
Neutron Detector	--	LAS (4); 85	LIS (33-52; PHA 1-16, neutron, gamma, charged particle counts, respectively)
VLF	n ₄ (0, 1, 2, 3)	LAS (6); (64 + n ₄ 32) 128	LIS (27-32; E _{x1} , E _{z1} , E _{x2} , E _{z2} , M ₁ , M ₂)
Cosmic Ray	n ₁₆ (0 ... 15)	LAS (7); $\left[43 + n_{16} \frac{128}{16}\right]_{128}$	LIS (20-26; $\hat{D}_1, \hat{D}_3,$ D ₁ D ₂ $\bar{D}_3\bar{D}_4, D_1D_2D_3\bar{D}_4,$ D ₁ $\bar{D}_2\bar{D}_4, \bar{D}_1D_2D_3\bar{D}_4, D_4$)
Plasma Probe	n ₁₅ (0 ... 14)	LAS (8); $\left[21 + n_{15} \frac{128}{15}\right]_{128}$ to nearest integer	LIS (12-19; E ₁ ... E ₈)

Data words are written on magnetic tape each S/C revolution. LIS (0-11; $B_{x_1}, B_{y_1}, B_{z_1}; B_{x_2}, B_{y_2}, B_{z_2}; B_{x_3}, B_{y_3}, B_{z_3}; B_{x_4}, B_{y_4}, B_{z_4}$) define the storage locations for the magnetometer data. LIS (12-19; $E_1, E_2, E_3, E_4, E_5, E_6, E_7, E_8$) define the storage locations for the plasma probe data. LIS (20-26; $\hat{D}_1, \hat{D}_3, D_1 D_2 \bar{D}_3 \bar{D}_4, D_1 D_2 D_3 \bar{D}_4, D_1 \bar{D}_2 \bar{D}_4, \bar{D}_1 D_2 D_3 \bar{D}_4, D_4$) define the storage locations for the cosmic ray experiment. LIS (27-32; $E_{x_1}, E_{z_1}, E_{x_2}, E_{z_2}, M_1, M_2$) define the storage locations for the VLF experiment. LIS (33-52; PHA 1...16, neutron count, gamma count, charged particle count) define the storage locations for the neutron detector experiment. LIS (53-57; S_1, S_2, S_3, S_4, S_6) define five lines of the radio propagation experiment. LIS (58-185; S_5) represent the 128 readings of the S_5 line of the radio propagation experiment. Locations LIS (186-285) are reserved. The S/C Angular Count, (S/C)AC, is the primary counter modulo (128) used in the program. The (S/C)AC is initially set to zero. Ancillary counters, modulo 4, 15, and 16, designated $n_4, n_{15},$ and n_{16} , respectively, are set to zero initially. A queue sequence counter, Q, used in sequence bookkeeping procedures, is reset to zero whenever the (S/C)AC changes to zero modulo 128. The correspondences between (S/C)AC and LIS are stored in ancillary storage locations, LAS \rightarrow LAS + 8.

Data from lines $D_1 D_2 \bar{D}_3 \bar{D}_4$ (9), $D_1 D_2 D_3 \bar{D}_4$ (9), $D_1 \bar{D}_2 \bar{D}_4$ (13), $\bar{D}_1 D_2 D_3 \bar{D}_4$, and D_4 (17) promulgated in Table I-4 undergo histogramming, and are stored with a reduction in precision as $D_1 D_2 \bar{D}_3 \bar{D}_4$ (5), $D_1 D_2 D_3 \bar{D}_4$ (3), $D_1 \bar{D}_2 \bar{D}_4$ (4), $\bar{D}_1 D_2 D_3 \bar{D}_4$ (3), and D_4 (6) in the science data list.

The simulated input science data is transferred in blocks of 285 words into memory locations in the CDC 3200 once per S/C revolution.

I.2 SIMULATION MODEL SUMMARY

Figure I-1 summarizes the program executed each spacecraft revolution. At (START) the Science Data List is inputted to the computer. The Spin-Demodulation subroutine removes rotational effects from the magnetometer data. The Polarization Ellipse Max-Min subroutine determines the maximum and minimum

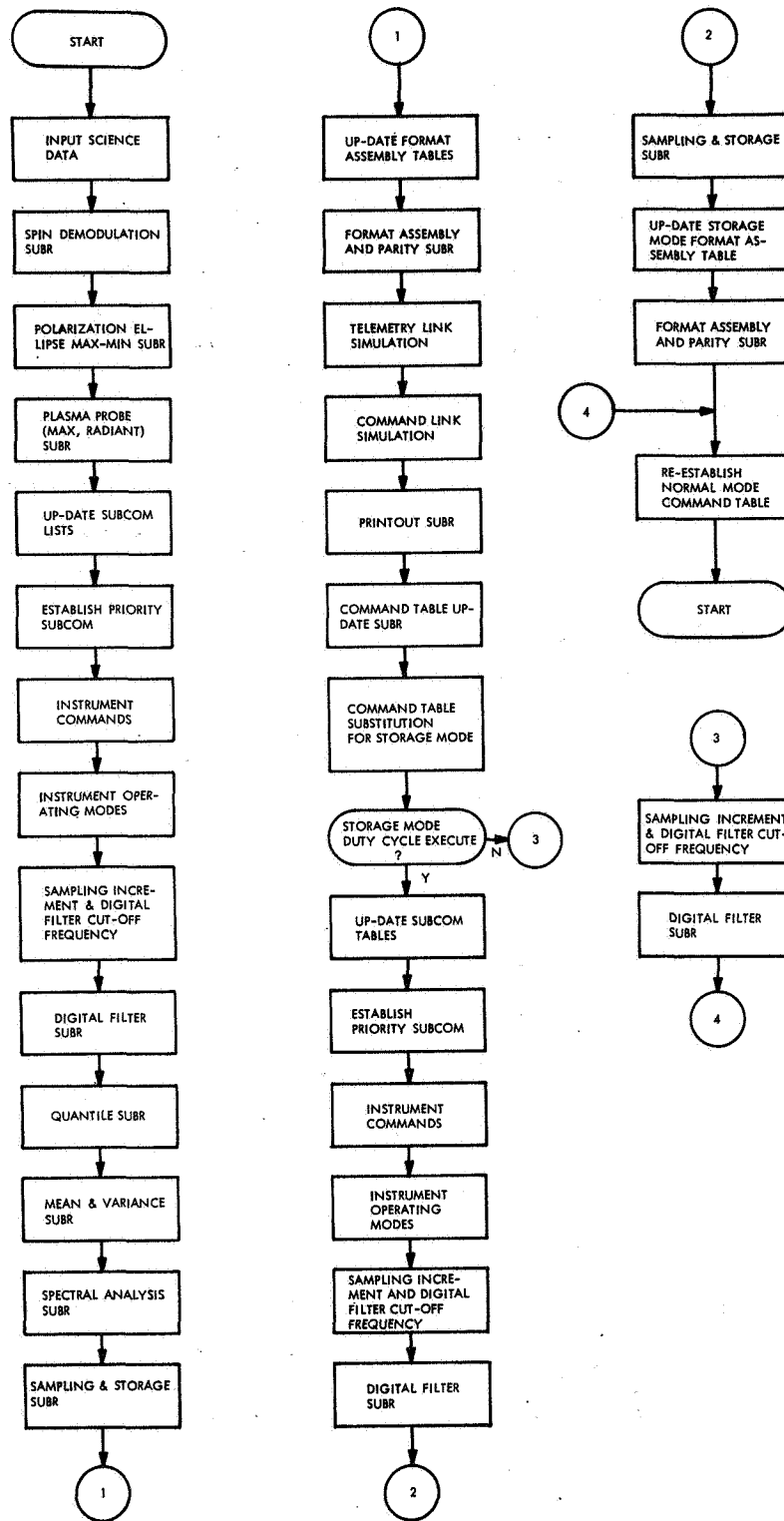


Figure I-1. Program Routine Executed Each Spacecraft Revolution

values of the S_5 line of the radio propagation experiment. The Plasma Probe Maximum and Radiant subroutine specifies the maximum flux, and the associated channel and sector for the most recent 120 data samples. Ancillary subcom lists are up-dated from the Command Table. Priority subcom is identified and moved into the Priority Assembly table. Instrument commands are executed and the format assembly tables are updated. Instrument-operating modes are examined and the format assembly tables are updated when necessary. The rate at which the sampling format is to be executed is computed. The cutoff frequency for the Digital Filter subroutine is computed. The Digital Filter subroutine is executed. The Quantile subroutine computes six quantiles from 1000 samples of the D_4 line of the Cosmic Ray experiment. The Mean and Variance subroutine computes the mean energy level and variance of a 64-bucket, 1000-sample histogram for neutron events. The Spectral Analysis subroutine computes 10 weighted spectral components of the M1 line of the VLF experiment for 1000 data samples. The Sampling and Storage subroutine samples appropriate data lines at specified spacecraft revolutions and moves data into storage blocks. The format assembly tables are then updated. The Format Assembly subroutine assembles the telemetry format. The bit error rates and the signal-to-noise ratios are computed for the telemetry and command links. The print-out subroutine outputs measures of system performance, Command Table parameters, and telemetry as required. The Command Table is updated.

The routine enters Storage Mode operation. If the Storage Execute Mode is not executed, the program branches to (3) where the Digital Filter subroutine is executed for magnetometer data. The program branches to (4) where normal mode commands are restored to the Command Table. The program branches back to (START). If, on the other hand, the Storage Execute Mode is scheduled, the appropriate Storage Mode lists and tables are updated. Instrument command and operational changes are executed. Sampling times are computed and stored. The digital filter cutoff frequency is computed; magnetometer data are smoothed by the Digital Filter subroutine. Data are sampled and stored in storage mode data blocks. The Storage Mode Format Assembly subroutine assembles the format and assigns parity. Normal mode parameters are restored to the Command Table. The program branches to (START).

APPENDIX II

ACCEPTANCE TEST PROCEDURE FOR SYSTEM SIMULATION MODEL

TABLE II-1

SYSTEM SIMULATION COMMAND TABLE

TYPEWRITER CODE	MACHINE ADDRESS	DESCRIPTION	OPERATIONAL CODE
1	LOPS	Command Link Bit Rate (1, 2, 4, 8 B/S)	(0 → 3)
2	LOPS+1	Telemetry Bit Rate (8, 16, 32, 64, 128, 256, 512, 1024 B/S)	(0 → 7)
3	LOPS+2	Telemetry Format (ENG, FLBR, FHBR, VLBR, VHBR)	(0 → 4)
4	LOPS+3	Science Subcom Identification (7 Bits)	
5	LOPS+4	Engineering Subcom Identification (7 Bits)	
6	LOPS+5	Range (MKM)	
7	LOPS+6	SKDL (Scaled by 10^{-4} in Computer) (23 Bits)	
8	LOPS+7	SKUL (Scaled by 10^{-2} in Computer) (23 Bits)	
9	LOPS+8	Antenna Gain (23 Bits)	
10	LSW	MAG (On/Off) (1)	(0 → ON, 1 → OFF)
11	LSW+1	PP (On/Off) (1)	(0 → ON, 1 → OFF)
12	LSW+2	CR (On/Off) (1)	(0 → ON, 1 → OFF)
13	LSW+3	VLF (On/Off) (1)	(0 → ON, 1 → OFF)
14	LSW+4	ND (On/Off) (1)	(0 → ON, 1 → OFF)
15	LSW+5	RP (On/Off) (1)	(0 → ON, 1 → OFF)
16	LSW+6	PP Energy Level Command (1)	(1 → EXECUTE)
17	LSW+7	VLF Filter Command (1)	(1 → EXECUTE)
18	LSW+8	Science Subcom Command (1)	(1 → EXECUTE)
19	LSW+9	Engineering Subcom Command (1)	(1 → EXECUTE)
20	LSW+10	CR Calibrate Command (1)	(1 → EXECUTE)
21	LSW+11	Magnetometer Operation Commands (1)	(1 → EXECUTE)
22	LSC	MAG Orientation (1)	
23	LSC+1	MAG Sensitivity (2)	
24	LSC+2	MAG Bias (3)	
25	LSC+3	PP - Voltage Level (5)	
26	LSC+4	VLF - Filter Center Frequency (3)	
27	PROC	ND - Mean & Variance Subroutine (1)	(1 → EXECUTE)
28	PROC+1	CR - Quantile Subroutine (1)	(1 → EXECUTE)
29	PROC+2	VLF - Spectral Analysis Subroutine (1)	(1 → EXECUTE)
30	STR	Storage Mode (1)	(1 → EXECUTE)
31	STR+1	Storage Mode Duty Cycle (1)	(1 → EXECUTE)
32	PRINTOUT	S/N, BER for TLM & Command Links (1)	(1 → EXECUTE)
33	KKROT	Exercise Program for Specified No. of Rotations (23)	
34	KILLSPIN	Spin-Demodulation Subroutine (1)	(1 → INHIBIT)
35	KILLDIG	Digital Filter Subroutine (1)	(1 → INHIBIT)
36	INIT	Initialize Before Run Starts (1)	(1 → EXECUTE)
37	GO	Starts Simulation Run	

II.1 FORMAT CHECKOUT PROCEDURE

- 1) Print out 300 records of scientific data tape. Reroll tape.
- 2) Print out Engineering List.
- 3) Print out Command Table.

II.1.1 Fixed Engineering Format (FENGF)

- 1) Move 1 to LOPS.
- 2) Move 512 to LOPS+1.
- 3) Designate FENGF in LOPS+2.
- 4) Move zero to LOPS+3 and LOPS+4.
- 5) Move $2^{23}-1$ to LOPS+5.
- 6) Move 1 to LOPS+6 and LOPS+7.
- 7) Move $2^{23}-1$ to LOPS+8.
- 8) Move zero to LSW → LSW+11.
- 9) Move zero to PROC → PROC+2.
- 10) Move zero to STR → STR+1.
- 11) Move 1 to PRINTOUT.

II.1.2 Print Out One Frame of FENGF

- 1) Check CODE (11).
- 2) Check data and parity versus Engineering List for specified parity algorithm.
- 3) Check BER's.
- 4) Repeat procedure for all telemetry rates.

II.1.3 Change Command Table Values

- 1) Move 1 to LSW → LSW+11.
- 2) Move 1 to LSC → LSC+4.
- 3) Move 1 to LOPS+3 → LOPS+4.
- 4) Move 1 to PROC → PROC+2.
- 5) Move 1 to STR → STR+1.
- 6) Move 1 to LES+63.
- 7) Move 512 to LOPS+1.

II.1.4 Print Out One Frame of FENGF

- 1) Check FBR List.
- 2) Check parity on last data word.
- 3) Compare against printout of II.1.1.
- 4) FENGF has been checked out.

II.2 FIXED HIGH BIT RATE FORMAT (FHBRF)

- 1) Reroll data tape to beginning.
- 2) Move 8 to LOPS.
- 3) Move 512 to LOPS+1.
- 4) Move FHBRF designation to LOPS+2.
- 5) Move zero to LOPS+3 → LOPS+4.
- 6) Move zero to LSW → LSW+11.
- 7) Move zero to LSC → LSC+4.
- 8) Move zero to PROC → PROC+2.
- 9) Move zero to STR → STR+1.

II.2.1 Print Out Frame of FHBRF

- 1) Six records of input are required.
- 2) Check CODE(11).
- 3) Check ENG(13).
- 4) Check CODE(17).
- 5) Check data and parity in detail.
- 6) Check FBR subcom.
- 7) Check BER's.

II.2.2 Change Command Table for FHBRF

- 1) Move 1 to LSW → LSW+5.
- 2) Print out a frame of FHBRF.
- 3) Data bits should be all zeros.
- 4) "LES+8" → "LES+15" should appear in FHBRF frame.
- 5) Check FBR subcom.
- 6) Check BER's.
- 7) Reroll input data tape.
- 8) Move zero to LSW → LSW+5.
- 9) Print out frames of FHBRF at TLMR's of 128, 256, 1024 bits/second.
- 10) Checkout of FHBRF complete.

II.3 FIXED LOW BIT RATE FORMAT (FLBRF)

- 1) Reroll input data tape.
- 2) Move 64 to LOPS+1.
- 3) Move FLBRF designation to LOPS+2.
- 4) Print out a frame of FLBRF (requires 36 input records).

- 5) Check CODE(11).
- 6) Check ENG(13).
- 7) Check CODE(13).
- 8) Check FBR subcom.
- 9) Check data and parity.

II.3.1 FLBRF (All Instruments Off)

- 1) Move 1 to LSW → LSW+5.
- 2) Print out frame of FLBRF.
- 3) Check ENG(13).
- 4) Check FBR subcom.
- 5) Check CODE(13).
- 6) Check parity.

II.4 VARIABLE LOW BIT RATE FORMAT (VLBRF)

- 1) Reroll input data tape.
- 2) Move 4 to LOPS.
- 3) Move 64 to LOPS+1.
- 4) Move VLBRF designation to LOPS+2.
- 5) Print out one frame of VLBRF (requires 40 input records).
- 6) Check SYNC(11).
- 7) Check LD(7) for known format.
- 8) Check S(V).
- 9) Check data and parity against input for known parity algorithm.
- 10) Check block formats and BID(5) for each block.
- 11) Reroll input data tape.

- 12) Move 1 to LSW.
- 13) Print out one frame of VLBRF.
- 14) Check against input and previous frame.
- 15) Reroll input data tape.
- 16) Check against possible instrument "on-off" combinations.
- 17) Reroll input tape.
- 18) Try other telemetry rates.

II.5 VARIABLE HIGH BIT RATE FORMAT (VHBRF)

- 1) Reroll data tape.
- 2) Move 512 to LOPS+1.
- 3) Move VHBRF designation to LOPS+2.
- 4) Move 1 to LOPS.
- 5) Move zero to LSW → LSW+11.
- 6) Print out frame of VHBRF (requires 14 input records).
- 7) Check CODE(11).
- 8) Check LD(7).
- 9) Check S(V).
- 10) Check data and parity against input for known parity algorithm.
- 11) Reroll input data tape and repeat above procedure at TLMR's of 128, 256, 1024 bits/second.
- 12) Reroll input tape.
- 13) Repeat above procedures for combinations of instruments "on" and "off".

II.6 SUMMARY OF FORMAT CHECKS

- 1) FENGF checked out.
- 2) VHBRF and VLBRF checked out.

- 3) FHBRF and FLBRF checked out.
- 4) Engineering subcom and Engineering List checked out.
- 5) FBR subcom checked out.
- 6) Parity algorithm checked out.
- 7) Operation at different formats and telemetry rates checked out.

II.7 VARIABLE FORMAT COMMAND CHECKOUTS

- 1) Reroll input data tape.
- 2) Move 1 to LOPS.
- 3) Move 512 to LOPS+1.
- 4) Move VHBRF designation to LOPS+2.
- 5) Move 7 to LOPS+3 and LOPS+4 [SID(7) = 7, SID(E)(7) = 7].
- 6) Move zero to LSW → LSW+5.
- 7) Move 1 to LSW+8 → LSW+9.
- 8) Move zero to LSC → LSC+4.
- 9) Move zero to PROC → PROC+2.
- 10) Move zero to LSW+6, LSW+7, LSW+10, and LSW+11.
- 11) Move zero to STR → STR+1.
- 12) PRINTOUT (zero or 1) at operator option.
- 13) Print out frame of VHBRF.
- 14) Check S(V) and filter subcom.
- 15) Reroll input tape.
- 16) Move zero to LSW+8 → LSW+9.
- 17) Move 1 to LSC → LSC+2.
- 18) Print out frame of VHBRF.
- 19) Check "MAGSA".

- 20) Check "LSW+11".
- 21) Reroll input data tape.
- 22) Move 1 to LSC+3 → LSC+4.
- 23) Move 1 to LSW+6 → LSW+7.
- 24) Print out a frame of VHBRF.
- 25) Check data preamble for Plasma Probe and Cosmic Ray data blocks.
- 26) Reroll input data tape.
- 27) Move 1 to STR and STR+1.
- 28) Print out one frame of Storage Mode Format.
- 29) Check against VHBRF for TLMR = 512 bits/second.

II.8 PROCESSING ALGORITHM CHECKOUT PROCEDURES

II.8.1 Simple Processing Algorithms

- 1) Maximum and Minimum Values of S_5 Line of RP Experiment
 - a) Print out input data list.
 - b) Compare with "LS5MAX" and "LS5MIN".
- 2) Max Flux and Radiant for Plasma Probe
 - a) Print out input data list over 15 S/C rotations.
 - b) Compare with "LFMAX".
- 3) Spin-Demodulation Subroutine
 - a) Compare data list before subroutine with data list after subroutine.
 - b) Compare with pre-computed values.

II.8.2 Digital Filtering Subroutine

- 1) Isolate subroutine.
- 2) Input test function (constant).
- 3) Print out the output.

II.8.3 Spectral Analysis Subroutine

- 1) Isolate subroutine.
- 2) Input test function (1/4 cps sine wave).
- 3) Print out $M_1 \rightarrow M_{10}$.

II.8.4 Quantiles of Histograms Subroutine

- 1) Isolate subroutine.
- 2) Input test function (uniformly distributed RV $\rightarrow 0 \rightarrow 2^6$).
- 3) Print out $Q_1 \rightarrow Q_6$.

II.8.5 Mean and Variance Subroutine

- 1) Isolate subroutine.
- 2) Input test function (Gauss RV with $\bar{X} = 0$, $\bar{X}^2 = 1$).
- 3) Print out mean and variance.

II.8.6 Checkout Processing Algorithms in System Context

- 1) Reroll data tape.
- 2) Move VHBRF to LOPS+2.
- 3) Move 512 to LOPS+1.
- 4) Move 1 to PROC, PROC+1, PROC+2.
- 5) Print out format frame starting at S/C rotation count equal to 1000.
- 6) Check against expected results.

II.9 TEST ACCEPTANCE PROCEDURE

Experience with operating the simulation program indicated that it was not necessary to follow the formal checkout procedure in detail. The important checks were to ascertain that data were inputted into each format correctly and

that format structure could be broken down into data words, parity bits, sync words, subcom words, line definition, and instrument-associated subcom.

II.9.1 Engineering Format (FENGF)

The engineering format was checked out first. It has a simple structure in the simulation model since the data is invariant (all 1's). The sync code is composed of eleven 1's in sequence; this makes it easy to distinguish. The checkout of the engineering format consisted of checking the subcom to confirm that it agreed with the System Simulation Command Table and that relevant S/C rotation counters incremented each S/C rotation. Various Command Table parameters were tried. The program checked out. Exhibit 1 (Figure II-1) promulgates the Command Table parameters and one frame of FENGF. It adds the 1st, 3rd, 5th, and 7th data bits of eligible data bits modulo 2 and the result is the 9th (parity) bit.

II.9.2 Fixed Low Bit Rate Science Format (FLBRF)

The FLBRF is shown in Figure 1-3. The corresponding sampling format is shown in Figure IV-21. A mask (Mask - FLBRF) has been constructed to decode the format. The entry points and sampling times are determined from the mask and FBR SUBCOM words which appear in the format. Time resolution to approximately (1/128) seconds is obtainable.

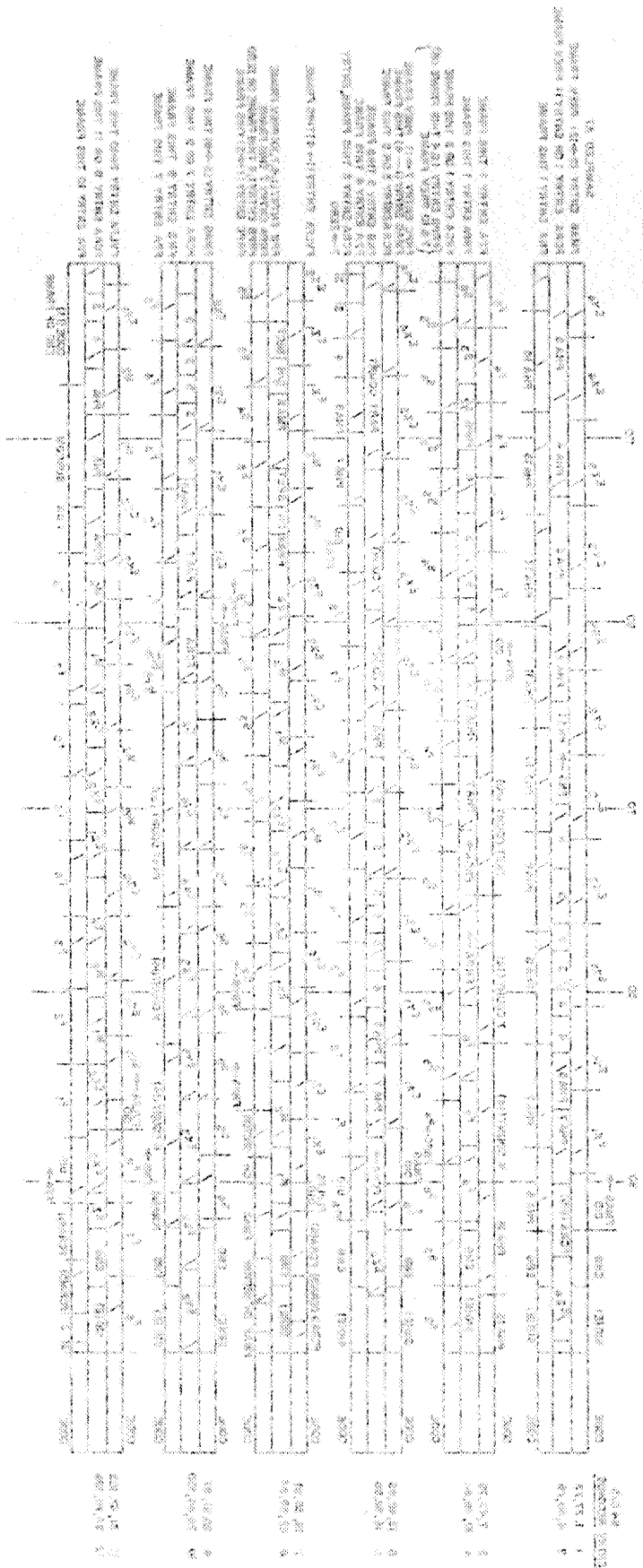
Exhibit 2 (Figure II-2) promulgates one of the simulation runs employed for checking out the FLBRF. The first output specifies the signal-to-noise and bit error rates for the parameters given in the command Table. The FLBRF mask shows the structure of the format in detail. At the right-hand edge of the mask, the entries at which samples are entered is indicated. At the left-hand edge of the mask are shown a limited number of input records from which samples are taken at 64 B/S. The initial format frame exhibits transient behavior since data is not immediately available from the magnetometer storage block due to the sampling algorithm employed. The first format frame employs input records 1, 4, 7, 10, 13, 16, 19, 22, 25, 28, 31, and 34. After the first format frame, the steady state condition is reached; thereafter a one-to-one correspondence between sampling time and the time at which

LOPS = 8	COMMAND LINK (BITS/SEC)
LOPS+1 = 1024	TELEMETRY LINK (BITS/SEC)
LOPS+2 = FENGF	TELEMETRY FORMAT COMMAND
LOPS+3 = 0	SCIENCE SUBCOM IDENTIFICATION
LOPS+4 = 0	ENGINEERING SURCOM IDENTIFICATION
LOPS+5 = 160	RANGE (MKM)
LOPS+6 = 0	SKDL (SYSTEM GAIN DOWN-LINK)
LOPS+7 = .01	SKUL (SYSTEM GAIN UP-LINK)
LOPS+8 = 8388607	ANTENNA GAIN
Lsw = 0	MAGNETOMETER ON-OFF
Lsw+1 = 0	PLASMA PROBE ON-OFF
Lsw+2 = 0	COSMIC RAY ON-OFF
Lsw+3 = 0	VLF ON-OFF
Lsw+4 = 0	NEUTRON DETECTOR ON-OFF
Lsw+5 = 0	RADIO PROPAGATION ON-OFF
Lsw+6 = 1	PLASMA PROBE - ENERGY LEVEL EXECUTE COMMAND
Lsw+7 = 1	VLF - FILTER EXECUTE COMMAND
Lsw+8 = 0	SCIENCE SUBCOM - EXECUTE COMMAND
Lsw+9 = 0	ENGINEERING SUBCOM - EXECUTE COMMAND
Lsw+10 = 1	COSMIC RAY - CALIBRATE COMMAND
Lsw+11 = 0	MAGNETOMETER - VERIFY COMMAND
LSC = 1	MAGNETOMETER ORIENTATION
LSC+1 = 2	MAGNETOMETER SENSITIVITY
LSC+2 = 7	MAGNETOMETER BIAS
LSC+3 = 15	PLASMA PROBE VOLTAGE LEVEL
LSC+4 = 6	VLF - FILTER CENTER FREQUENCY
PROC = 0	NEUTRON DETECTOR - EXECUTE MEAN AND VARIANCE SUBROUTINE
PROC+1 = 0	COSMIC RAY - EXECUTE QUANTILES SUBROUTINE FOR D4 COUNTER
PROC+2 = 0	VLF - EXECUTE SPECTRAL ANALYSIS OF M1 LINE
STR = 0	STORAGE EXECUTE COMMAND
STR+1 = 0	STORAGE MODE IDENTIFICATION
SPIN DEMODULATION INHIBIT = 0	
DIGITAL FILTER INHIBIT = 0	
REINITIALIZE = 0	

Figure II-1. Exhibit 1 (Sheet 1 of 2)

LPS = 1	COMMAND LINK (BITS/SEC)
LPS+1 = 16	TELEMETRY LINK (BITS/SEC)
LPS+2 = FLRRF	TELEMETRY FORMAT COMMAND
LPS+3 = 0	SCIENCE SURCOM IDENTIFICATION
LPS+4 = 0	ENGINEERING SURCOM IDENTIFICATION
LPS+5 = 160	RANGE (MKM)
LPS+6 = 0	SKDL (SYSTEM GAIN DOWN-LINK)
LPS+7 = .01	SKUL (SYSTEM GAIN UP-LINK)
LPS+8 = 8388607	ANTENNA GAIN
LS+ = 0	MAGNETOMETER ON-OFF
LS+1 = 0	PLASMA PROBE ON-OFF
LS+2 = 0	COSMIC RAY ON-OFF
LS+3 = 0	VLF ON-OFF
LS+4 = 0	NEUTRON DETECTOR ON-OFF
LS+5 = 0	RADIO PROPAGATION ON-OFF
LS+6 = 1	PLASMA PROBE - ENERGY LEVEL EXECUTE COMMAND
LS+7 = 1	VLF - FILTER EXECUTE COMMAND
LS+8 = 0	SCIENCE SURCOM - EXECUTE COMMAND
LS+9 = 0	ENGINEERING SURCOM - EXECUTE COMMAND
LS+10 = 1	COSMIC RAY - CALIBRATE COMMAND
LS+11 = 1	MAGNETOMETER - VERIFY COMMAND
LSC = 1	MAGNETOMETER ORIENTATION
LSC+1 = 2	MAGNETOMETER SENSITIVITY
LSC+2 = 7	MAGNETOMETER BIAS
LSC+3 = 15	PLASMA PROBE VOLTAGE LEVEL
LSC+4 = 6	VLF - FILTER CENTER FREQUENCY
PROC = 0	NEUTRON DETECTOR - EXECUTE MEAN AND VARIANCE SUBROUTINE
PROC+1 = 0	COSMIC RAY - EXECUTE QUANTILES SUBROUTINE FOR D4 COUNTER
PROC+2 = 0	VLF - EXECUTE SPECTRAL ANALYSIS OF M1 LINE
SIR = 0	STORAGE EXECUTE COMMAND
SIR+1 = 0	STORAGE MODE IDENTIFICATION
SPIN DEMODULATION INHIBIT = 0	
DIGITAL FILTER INHIBIT = 0	
REINITIALIZE = 1	
BIT ERROR RATES	
$(S/N)_{DL} = 0 (P_{BE})_{DL} = 5.00000000E-01 (S/N)_{UL} = 3.27679961E 00 (P_{BE})_{UL} = 3.51329118E-02$	

Figure II-2. Exhibit 2 (Sheet 1 of 2)



data is entered into the format exists. The FLBRF was checked out for various Command Table parameters over a number of telemetry frames.

II.9.3 Fixed High Bit Rate Science Format (FHBRF)

The FHBRF is shown in 5-3. The corresponding sampling format is shown in Figure IV-21. The FHBRF mask of Exhibit 3 (Figure II-3) specifies the details of the format and the sampling table entries at which the experiments have been sampled. Time and spacecraft rotations are entered into the format frequently as subcom to specify the spacecraft rotation and time at which each experiment has been sampled. Time resolution is one second. Time resolution to approximately (1/128) seconds can be deduced since each experiment is sampled at one of 128 angular sectors relative to the sunline.

Exhibit 3 shows the first format frame of one of the simulation runs for the given set of Command Table parameters. The signal-to-noise and bit error rates on the command and telemetry link were checked out. The initial format has transient behavior. The checkout procedure consisted of checking data against the input record list. The DID words were checked to ascertain that counters were incrementing and flags were set as needed. The format checked out.

II.9.4 Variable High Bit Rate Science Format (VHBRF)

A typical VHBRF (Figure 5-5) uses the sampling format of Figure IV-21. The format assembly subroutine (FAS) scans the list shown at the left of Figure 5-5 and assembles data as available. The SYNC, LD, and S(V) are the fixed structure of the VHBRF. The list starting at FBRA(18) may originate at any point within the fixed structure. Each data block or subcom word is recognized directly by its five-bit identification tag (BID). Spacecraft revolution counts and time are entered into the format periodically so that the sampling time for each experiment can be resolved to within one second.

Further resolution to within (1/128) second is possible since the angular sector within which the experiments are sampled is known. FBRA(18) is incremented every S/C revolution. The experiments are sampled at:

$$\left\{ \text{FBRA} + K \right\}_{\text{Modulo } 6\Delta S} = (EN-1)\Delta S \quad \text{least integer value.}$$

The entry number EN takes on the values 1 → 12 modulo 6. The record number or spacecraft revolution at which a sample is taken is $\text{FBRA} + K+1 \left\{ \text{Modulo } 2^{13} \right\}$. FBRA appears as a format entry. The first data entry following FBRA occurs at the spacecraft revolution specified by:

$$\text{FBRA} + K \left\{ \text{Modulo } 6\Delta S \right\} = 0$$

where K is the smallest integer value satisfying the equality. The spacecraft revolution (hence, record number, and time) at which successive samples are taken may be computed as EN ranges from (1 → 12 modulo 6). K takes on integer values both positive and negative; the range of K is restricted to $\pm [6\Delta S]$ least integer value.

A transient occurs when a format changes, e.g., from the fixed high rate format to variable high bit rate format. There are two obvious ways to handle such a transient. One is to reinitialize and start again with FBRA reset to zero. The second way is to interpret the interface frame based upon the previous and new format. The interface decoding requires a computer program. Our simple logic is to reinitialize for the simulation model; an operational model should include the logic to allow a smooth transition as formats are changed.

Exhibit 4 (Figure II-4) contains a frame of VHBRF. The frame has been partially decomposed into sync, line definition, subcom, and data words to exhibit the structure. No simple mask will decommutate this format since the list FBRA, MAG, CRA, PA, etc., "snakes" thru the fixed format structure. Each format must be decommutated uniquely. This was done for the acceptance

LPS = 8	COMMAND LINK (BITS/SEC)
LPS+1 = 512	TELEMETRY LINK (BITS/SEC)
LPS+2 = FHHRF	TELEMETRY FORMAT COMMAND
LPS+3 = 12	SCIENCE SURCOM IDENTIFICATION
LPS+4 = 35	ENGINEERING SURCOM IDENTIFICATION
LPS+5 = 10	RANGE (MKM)
LPS+6 = .5000	SKUL (SYSTEM GAIN DOWN-LINK)
LPS+7 = 20.00	SKUL (SYSTEM GAIN UP-LINK)
LPS+8 = 8388607	ANTENNA GAIN
LS = 0	MAGNETOMETER ON-OFF
LS+1 = 0	PLASMA PROBE ON-OFF
LS+2 = 0	COSMIC RAY ON-OFF
LS+3 = 0	VLF ON-OFF
LS+4 = 0	NEUTRON DETECTOR ON-OFF
LS+5 = 0	RADIO PROPAGATION ON-OFF
LS+6 = 0	PLASMA PROBE - ENERGY LEVEL EXECUTE COMMAND
LS+7 = 0	VLF - FILTER EXECUTE COMMAND
LS+8 = 1	SCIENCE SURCOM - EXECUTE COMMAND
LS+9 = 1	ENGINEERING SURCOM - EXECUTE COMMAND
LS+10 = 0	COSMIC RAY - CALIBRATE COMMAND
LS+11 = 1	MAGNETOMETER - VERIFY COMMAND
LSC = 0	MAGNETOMETER ORIENTATION
LSC+1 = 2	MAGNETOMETER SENSITIVITY
LSC+2 = 1	MAGNETOMETER BIAS
LSC+3 = 0	PLASMA PROBE VOLTAGE LEVEL
LSC+4 = 0	VLF - FILTER CENTER FREQUENCY
PROC = 1	NEUTRON DETECTOR - EXECUTE MEAN AND VARIANCE SUBROUTINE
PROC+1 = 1	COSMIC RAY - EXECUTE QUANTILES SUBROUTINE FOR D4 COUNTER
PROC+2 = 1	VLF - EXECUTE SPECTRAL ANALYSIS OF M1 LINE
STR = 0	STORAGE EXECUTE COMMAND
STR+1 = 0	STORAGE MODE IDENTIFICATION
SPIN DEMODULATION INHIBIT = 1	
DIGITAL FILTER INHIBIT = 1	
REINITIALIZE = 1	
BIT ERROR RATES	
$(S/N_{DL}) = 8.19199902E - 0 (PBE)_{DL} = 7.08543679E - 20 (S/N)_{DL} = 2.09715175E - 05 PBE_{UL} = 0$	

Figure II-3. Exhibit 3 (Sheet 1 of 2)

FHBRF - MASK

10	20	30	40	50	60	70	80	90	100	110	120
CODE	SID (E)	ENG	FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
1			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
2			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
3			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
4			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
5			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
6			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
7			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
8			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
9			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
10			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
11			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
12			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
13			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
14			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
15			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
16			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
17			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
18			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
19			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
20			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
21			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
22			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
23			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)
24			FMAG (102)	ENG	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)	PHAS (10)

SAMPLED AT

FMAG ENTRY 1 (INITIALLY)
FCRA ENTRY 1 (INITIALLY)
FPA ENTRY 1 (INITIALLY)
FNA ENTRY 1

FMAG ENTRY 2 (EXCEPT ENTRY 1 INITIALLY)
FCRA ENTRY 2 (EXCEPT ENTRY 1 INITIALLY)
FPA ENTRY 2 (EXCEPT ENTRY 1 INITIALLY)
FNA ENTRY 2

FMAG ENTRY 3 (INITIALLY)

FCRA ENTRY 3 (EXCEPT ENTRY 2 INITIALLY)
FPA ENTRY 3 (EXCEPT ENTRY 2 INITIALLY)
FCRA ENTRY 3 (ZERO INITIALLY)
FPA ENTRY 3 (ZERO INITIALLY)

FPA ENTRY 4 (EXCEPT ENTRY 3 INITIALLY)
FPA ENTRY 4

FVLFA ENTRY 1, 2, 3, 4

FMAG ENTRY 4 (EXCEPT ENTRY 3 INITIALLY)
FCRA ENTRY 4 (EXCEPT ENTRY 3 INITIALLY)
FPA ENTRY 4 (EXCEPT ENTRY 3 INITIALLY)

FRPC ENTRY 1 (→ 3)

FMAG ENTRY 5 (EXCEPT ENTRY 4 INITIALLY)
FCRA ENTRY 5 (EXCEPT ENTRY 4 INITIALLY)
FPA ENTRY 5 (EXCEPT ENTRY 4 INITIALLY)

120

100

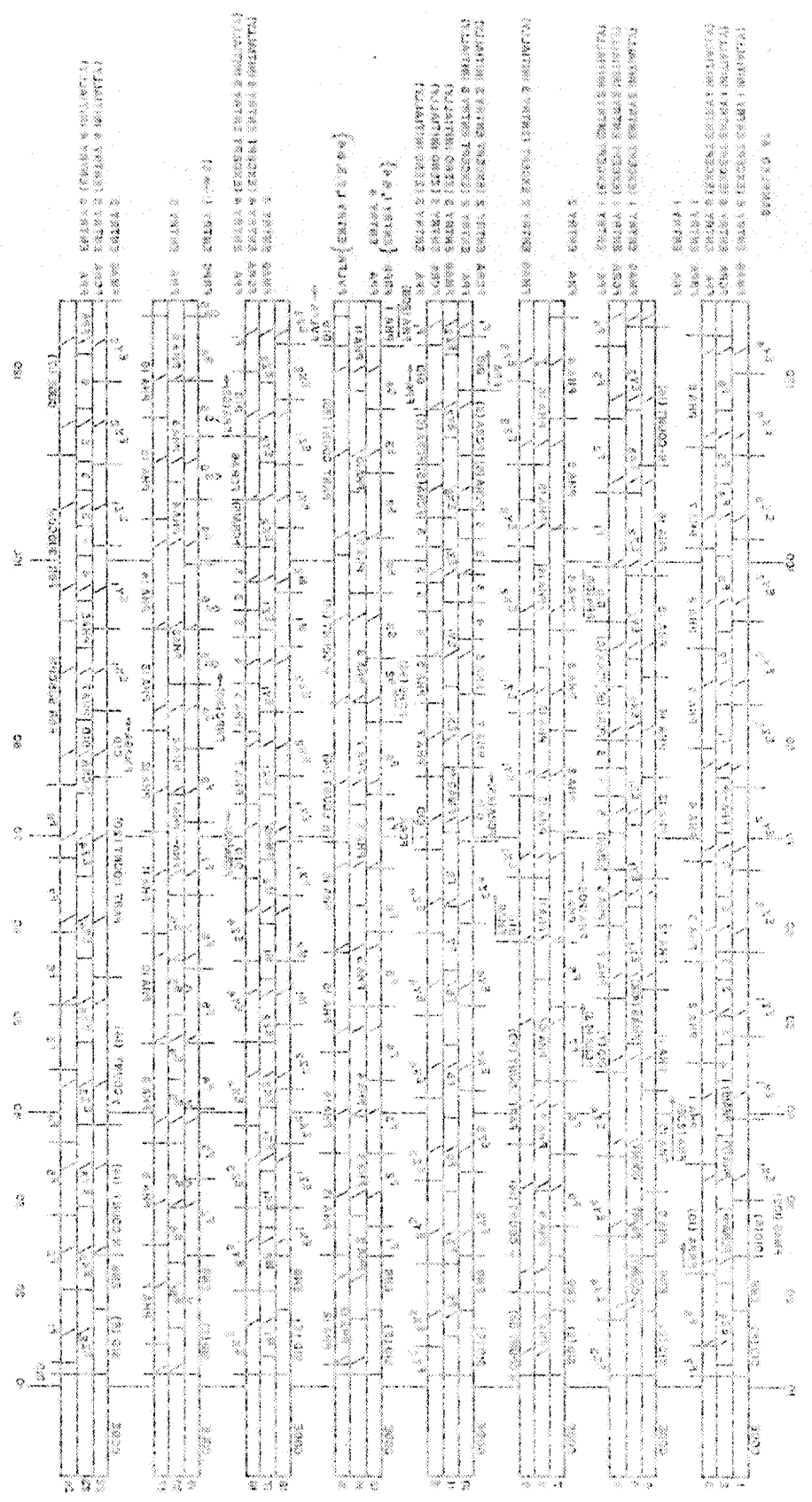
80

60

40

20

10



512000-1000

LOPS = 2	COMMAND LINK (BITS/SEC)
LOPS+1 = 512	TELEMETRY LINK (BITS/SEC)
LOPS+2 = VHHRF	TELEMETRY FORMAT COMMAND
LOPS+3 = 0	SCIENCE SUBCOM IDENTIFICATION
LOPS+4 = 0	ENGINEERING SUBCOM IDENTIFICATION
LOPS+5 = 8388607X10**6	RANGE (KM)
LOPS+6 = 1.0000	SKDL (SYSTEM GAIN DOWN-LINK)
LOPS+7 = 1.00	SKUL (SYSTEM GAIN UP-LINK)
LOPS+8 = 8388607	ANTENNA GAIN
LSW = 0	MAGNETOMETER ON-OFF
LSW+1 = 0	PLASMA PROBE ON-OFF
LSW+2 = 0	COSMIC RAY ON-OFF
LSW+3 = 0	VLF ON-OFF
LSW+4 = 0	NEUTRON DETECTOR ON-OFF
LSW+5 = 0	RADIO PROPAGATION ON-OFF
LSW+6 = 0	PLASMA PROBE - ENERGY LEVEL EXECUTE COMMAND
LSW+7 = 0	VLF - FILTER EXECUTE COMMAND
LSW+8 = 0	SCIENCE SUBCOM - EXECUTE COMMAND
LSW+9 = 0	ENGINEERING SUBCOM - EXECUTE COMMAND
LSW+10 = 0	COSMIC RAY - CALIBRATE COMMAND
LSW+11 = 1	MAGNETOMETER - VERIFY COMMAND
LSC = 0	MAGNETOMETER ORIENTATION
LSC+1 = 0	MAGNETOMETER SENSITIVITY
LSC+2 = 0	MAGNETOMETER BIAS
LSC+3 = 0	PLASMA PROBE VOLTAGE LEVEL
LSC+4 = 0	VLF - FILTER CENTER FREQUENCY
PROC = 1	NEUTRON DETECTOR - EXECUTE MEAN AND VARIANCE SUBROUTINE
PROC+1 = 1	COSMIC RAY - EXECUTE QUANTILES SUBROUTINE FOR D4 COUNTER
PROC+2 = 1	VLF - EXECUTE SPECTRAL ANALYSIS OF MODE INFO
STR = 1	STORAGE EXECUTE COMMAND
STR+1 = 0	STORAGE MODE IDENTIFICATION
SPIN DEMODULATION INHIBIT = 1	
DIGITAL FILTER INHIBIT = 1	
REINITIALIZE = 0	

Figure II-4. Exhibit 4 (Sheet 1 of 2)

VARIABLE HIGH BIT RATE FORMAT

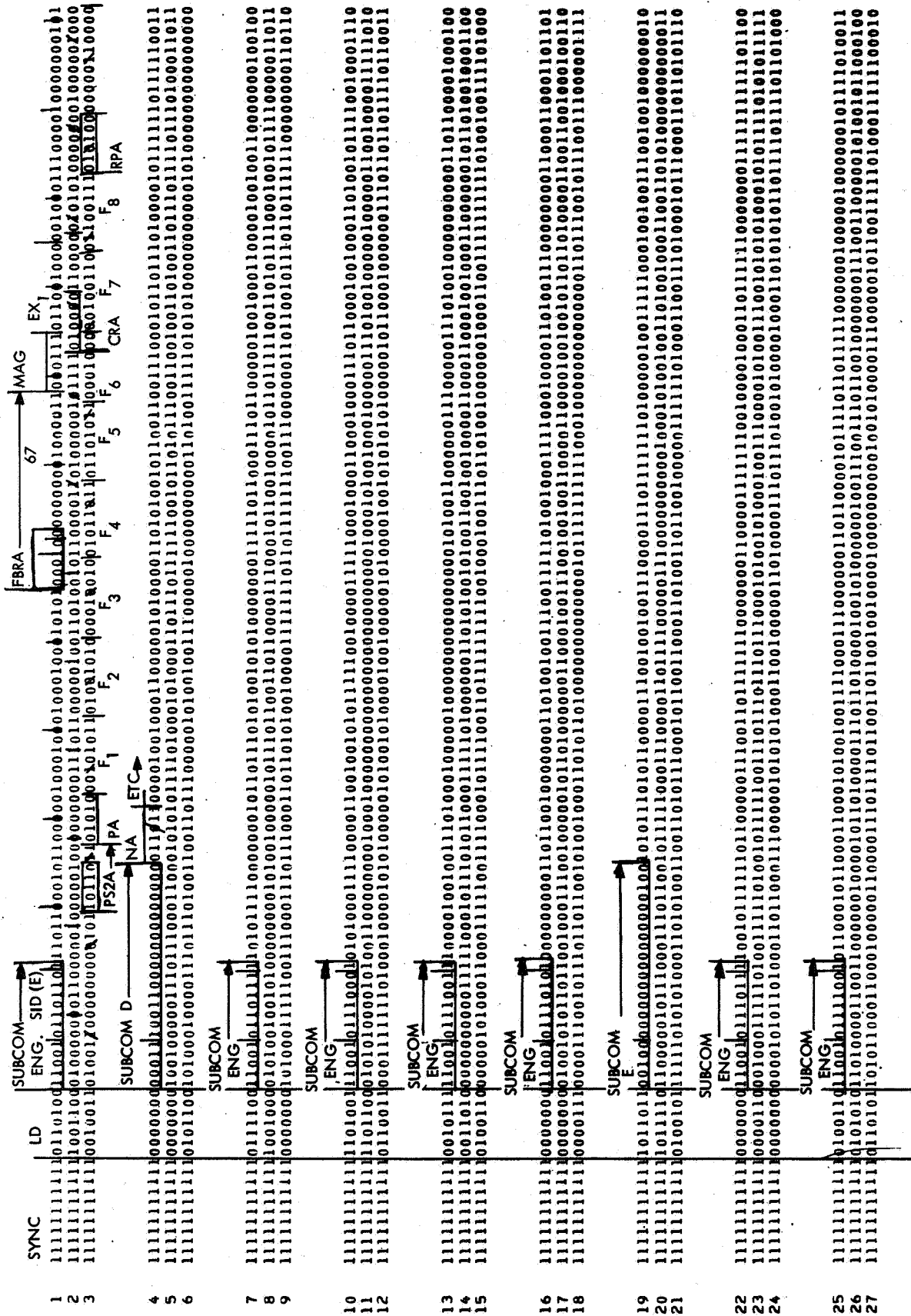


Figure II-4. Exhibit 4 (Sheet 2 of 2)

test. FBRA equals 67 in the first format line of exhibit 4; this means that the magnetometer data which follows was taken from input record 68. The magnetometer data was correlated with input record 68 and the correspondence established. Note that the sector cycle flag is set in the third line of the format. This designates that the plasma probe data sample which follows was taken from sector 1. The test acceptance procedure ascertained that all bits in the format could be accounted for logically. The format checked out satisfactorily.

II.9.5 Variable Low Bit Rate Science Format (VLBRF)

A typical VLBRF is shown in Figure 1-5. The associated sampling format is shown in Figure IV-24. Tagging the data blocks by spacecraft revolution and time is accomplished in the way described above for the VHBRF. The relevant equation is:

$$\left\{ \begin{array}{l} \text{FBRA} + \text{K} \\ \text{Modulo} \\ 6\Delta\text{S} \end{array} \right\} = (\text{EN}-1)\Delta\text{S} \quad \text{least integer value}$$

where EN = 1 → 12 modulo 6. However, the sampling formats for the VLBRF and VHBRF are different. Consequently, elements of each data block may have been sampled at different spacecraft revolutions. The magnetometer block telemetered at VLBRF is composed of samples taken at four different entry points. Furthermore, the block cannot be telemetered until it has been completely assembled. Consequently, the last three data words included in the first magnetometer data block following FBRA are taken at the spacecraft revolution corresponding {EN = 4, K minimum}.

II.9.6 Storage Mode Format (SMF)

The SMF is identical to the VHBRF except the sampling format recycles every 6SMAS. Consequently, the defining equation is:

$$\left\{ \begin{array}{l} \text{FBRA} + \text{K} \\ \text{Modulo} \\ 6\text{SMAS} \end{array} \right\} = (\text{EN}-1)\text{SMAS} \quad \text{least integer value.}$$

In the above equation, EN ranges from 1 - 6.

Exhibit 5 (Figure II-5) promulgates the VLBRF and the Storage Mode Format. The storage mode may be executed with any of the selectable formats. The Command Table of Exhibit 5 promulgates the command parameters. In the simulation run of Exhibit 5, the exercise was a duration of 1200 spacecraft rotations. The Storage Mode Format is essentially the VHBRF operating at 512 bits/second. It is operating in sequence with the VLBRF which provides the telemetry bit stream. In an operational system, the Storage Mode Format would be stored on a magnetic tape and read out upon ground command. A readout has been provided in Exhibit 5 for the demonstration. Note that the Storage Mode Format has an end of frame filler which fills out the last line of the format with zeros. This is the one point at which the VHBRF and the SMF differ.

The signal-to-noise and probability of bit error are shown in Exhibit 5. These have been checked out for the relevant parameters shown in the Command Table.

The VLBRF has a fixed structure consisting of sync, line definition, and subcom. The elements of the list shown at the left of Figure 1-5 may occur anywhere within the fixed structure. Note that FBRA, which heads the list, starts in line 5 of the VLBRF for the first frame promulgated in Exhibit 5. In the Test Acceptance Procedure, the data were checked against input data and the correspondence demonstrated. The processing subroutines PROC 1 → 3 were selected at the start. The associated data blocks (NC, VLFM, and CRQ) are available after 1000 S/C rotations. The output blocks, NC, VLFM, and CRQ are outputted in lines 5, 6, and 7 of the second VLBRF frame of Exhibit A. The processing subroutines were checked by hand computation and by operating on signals of known statistical characteristics. The program just missed picking up the NC, VLFM, and CRQ blocks in line 5 of the first frame of the VLBRF since the end of list test occurred before the NC, VLFM, and CRQ data blocks were ready for outputting.

Some elements of the Storage Mode Format have been identified in Exhibit 5. There is no simple decomposition algorithm for separating out the data from the VHBRF. A computer program is necessary to do the same job as has been done manually for Exhibit 5 using rather simple aids.

LOPS = 1
 LOPS+1 = 16
 LOPS+2 = VLRRF
 LOPS+3 = 3
 LOPS+4 = 63
 LOPS+5 = 320
 LOPS+6 = .4000
 LOPS+7 = 7.00
 LOPS+8 = 8388607

LSW = 0
 LSW+1 = 0
 LSW+2 = 0
 LSW+3 = 0
 LSW+4 = 0
 LSW+5 = 0
 LSW+6 = 0
 LSW+7 = 0
 LSW+8 = 1
 LSW+9 = 1
 LSW+10 = 1
 LSW+11 = 1

LSC = 1
 LSC+1 = 2
 LSC+2 = 5
 LSC+3 = 0
 LSC+4 = 0

PROC = 1
 PROC+1 = 1
 PROC+2 = 1

STR = 1
 STR+1 = 0

SPIN DEMODULATION INHIBIT = 1
 DIGITAL FILTER INHIBIT = 1

REINITIALIZF = 0

COMMAND LINK (BITS/SEC)
 TELEMETRY LINK (BITS/SEC)
 TELEMETRY FORMAT COMMAND
 SCIENCE SURCOM IDENTIFICATION
 ENGINEERING SURCOM IDENTIFICATION
 RANGE (MKM)
 SKDL (SYSTEM GAIN DOWN-LINK)
 SKUL (SYSTEM GAIN UP-LINK)
 ANTENNA GAIN

MAGNETOMETER ON-OFF
 PLASMA PROBE ON-OFF
 COSMIC RAY ON-OFF
 VLF ON-OFF
 NEUTRON DETECTOR ON-OFF
 RADIO PROPAGATION ON-OFF
 PLASMA PROBE - ENERGY LEVEL EXECUTE COMMAND
 VLF - FILTER EXECUTE COMMAND
 SCIENCE SURCOM - EXECUTE COMMAND
 ENGINEERING SURCOM - EXECUTE COMMAND
 COSMIC RAY - CALIBRATE COMMAND
 MAGNETOMETER - VERIFY COMMAND

MAGNETOMETER ORIENTATION
 MAGNETOMETER SENSITIVITY
 MAGNETOMETER BIAS
 PLASMA PROBE VOLTAGE LEVEL
 VLF - FILTER CENTER FREQUENCY

NEUTRON DETECTOR - EXECUTE MEAN AND VARTANCE SUBROUTINE
 COSMIC RAY - EXECUTE QUANTILES SUBROUTINE FOR D4 COUNTER
 VLF - EXECUTE SPECTRAL ANALYSIS OF M1 LTNF

STORAGE EXECUTE COMMAND
 STORAGE MODE IDENTIFICATION

BIT ERROR RATES

$(S/N)_{DL} = 2.04799976E-00$ $(P_{BE})_{DL} = 7.62031542E-02$ $(S/N)_{UL} = 1.63839980E-02$ $(P_{BE})_{UL} = 8.19764235E-38$

Figure II-5. Exhibit 5 (Sheet 1 of 4)

VARIABLE LOW BIT RATE FORMAT

SYNC	LD	SUBCOM W		DATA	CRSA		PHA (7)	DID	CRA	CRA BID	CRA BID	DID	NB BID	VLFA
		BID	SID		BID	BID								
1	1111111111	1001011100	1100000000	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
2	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
3	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
4	1111111111	1001011100	1100000000	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
5	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
6	1111111111	1001011100	1100000000	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
7	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
8	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000
9	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1111111111	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000	1000000000

IPB →

IPA →

MAG

FBRA = 1001

SUBCOM FILLER

STORAGE MODE FORMAT

SYNC	LD	SUBCOM W		DATA	DATA	MAG	EX ₁	ETC
		BID	SID					
1	1111111111	1001010100	1100000000	1000000000	1010101000	1000000000	1000000000	1000000000
2	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
3	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
4	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
5	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
6	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
7	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
8	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
9	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
10	1111111111	1001010100	1100000000	1000000000	1010101000	1000000000	1000000000	1000000000
11	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
12	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
13	1111111111	1001010100	1100000000	1000000000	1010101000	1000000000	1000000000	1000000000
14	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
15	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
16	1111111111	1001010100	1100000000	1000000000	1010101000	1000000000	1000000000	1000000000
17	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000
18	1111111111	1000000000	1111010100	1000000000	1010101000	1000000000	1000000000	1000000000

Figure II-5. Exhibit 5 (Sheet 3 of 4)

APPENDIX III

PROCESSING CONSIDERATIONS

A principal effort during this study has been to devise data compression algorithms. "Meaningful data compression" allows the experimenter to match instrument characteristics to the available channel capacity via processing algorithms.

The particular experiment, user requirements, and ultimate ground processing modes dictate the type of processing algorithm that should be used. Adaptive thresholding methods, if implemented in their full complexity, generate a severe buffer queue control problem. At low telemetry rates, assuming the data activity to be "high" for all instruments, large thresholds or priority servicing algorithms must be employed to control the buffer. Such thresholding schemes mask small data variations and require extra bits to identify the appropriate data sources. Our goal has been to understand those continuous data processing algorithms best suited for the complement of instruments described in Section 2.

III.1 HISTOGRAMS AND QUANTILES

The central data system (CDS) is ideally suited to the formation of histograms and the subsequent computation of statistics from these histograms. The cosmic ray and neutron experiments described in Section 2 employ counters. Rather than transmit counts each revolution, a histogram of the counts can be accumulated over a large number of S/C revolutions. The mean, variance, mode, etc. for each histogram can be computed and transmitted. Alternatively, the quantiles of the histograms can be computed and transmitted. In either case large compression ratios can be achieved.

In addition to their data compression potential, histograms (and quantiles) require very little processing time for their implementation. Normally this would not be a primary consideration because the speed of the CDS is sufficient to handle more sophisticated processing algorithms. In the event

of component failure in the CDS, however, a degradation in computing capacity occurs. Those processing algorithms could be retained that make the fewest demands on the CDS and still allow other essential operations such as multiplexing and formatting to be performed.

III.1.1 Use of Quantiles for Data Compression

The object here is to summarize existing results pertinent to the use of order statistics, or quantiles, in estimating population parameters and distributions, and in testing hypotheses. Results are stated without proofs, and details are largely omitted except for a few instances in which they enhance understanding of the method. Section III.1.2 discusses some properties of quantiles and their application in finding tolerance limits on the population and in establishing confidence bands for the distribution function. Section III.1.3 demonstrates the use of quantiles in estimating population parameters and Section III.1.4 illustrates their use in hypothesis testing.

III.1.2 Properties of Quantiles

A. Definition of Order Statistics and Quantiles

Consider a sample of size n taken from a continuous cumulative distribution function (cdf), $F(x)$, with corresponding density function $F'(x) = f(x)$. Represent the sample as: X_1, X_2, \dots, X_n . Now arrange the sample values in increasing order of magnitude, from least to greatest, denoting the ordered samples as $X_{(1)}, X_{(2)}, \dots, X_{(n)}$. These ordered values of the random variable (r.v.) X are called order statistics, $X_{(k)}$ being the k^{th} order statistic, $k = 1, 2, \dots, N$, or the order statistic of rank k . Often we denote the k^{th} order statistic from a sample of size n by $\xi_k^{(n)}$. Thus $\xi_k^{(n)}$ is the function of X_1, X_2, \dots, X_n , the n -dimensional random vector, that takes on the value $X_{(k)}$ in each possible sequence $X_{(1)}, X_{(2)}, \dots, X_{(n)}$.

Suppose that in a sample of size n we choose $k = [np] + 1$, $0 < p < 1$, where $[np]$ denotes the greatest integer not exceeding np . We define ξ_p , the p^{th} quantile, or the quantile of order p of the population, as the value of x satisfying the relation

$$\int_{-\infty}^{\xi_p} f(x) dx = p.$$

Equivalently, the p^{th} quantile can be defined as the root of the equation

$$F(\xi) = p.$$

We refer to $\xi_k^{(n)}$, with $k = [np] + 1$, as the sample quantile (also denoted by z_p) corresponding to the population quantile ξ_p . We observe that

$$\lim_{n \rightarrow \infty} k/n = p$$

$$n \rightarrow \infty.$$

It is observed that for n odd, the choice $k = [n/2] + 1$ gives the sample median; while for n even, say $n = 2\ell$, the $(\ell+1)^{\text{st}}$ element is taken as the sample median.

B. Limit Distributions of a Sample Quantile

If in a population being sampled, the r.v. X is continuous, with density $f(x)$, and if this density is continuous and positive at the point $x = \xi_p$, where ξ_p is the p^{th} -order population quantile, then the corresponding sample quantile is asymptotically normally distributed, with mean ξ_p and variance $p(1-p)/nf^2(\xi_p)$.^{1*} In fact, the joint distributions of any number of sample quantiles are asymptotically normal, with means ξ_{p_i} , variances

$$\sigma_i^2 = p_i(1-p_i)/nf_i^2(\xi_{p_i}),$$

and correlations

$$\rho_{ij} = \left[\frac{p_i(1-p_j)}{p_j(1-p_i)} \right]^{1/2}, \quad i < j \text{ and } p_i < p_j.$$

*References are listed at the end of this section.

C. Confidence Limits of Quantiles

It is possible to obtain confidence limits for quantiles that do not require a knowledge of the population cdf. We denote by U the coverage composed of the sum of any r ($r \leq n$) of the elementary coverages, $C_i = \int_{I_i} dF(x)$, where $I_i = [X_{(i-1)}, X_{(i)}]$, $i = 1, 2, \dots, n+1$, $X_{(0)} = -\infty$, $X_{(n+1)} = +\infty$, and where $F(x)$ is continuous and $X_{(i)}$ is the i^{th} order statistic. Then it can be shown² that U has the probability element

$$\frac{\Gamma(n+1)}{\Gamma(r)\Gamma(n-r+1)} u^{r-1}(1-u)^{n-r} du, \quad 0 \leq u \leq 1; \quad r \leq n.$$

Thus we have

$$\begin{aligned} P_r \left[X_{(i)} < \xi_p < X_{(j)} \right] &= P_r \left[F(X_{(i)}) < p < F(X_{(j)}) \right] \\ &= P_r \left\{ F(X_{(i)}) < p < F(X_{(i)}) + [F(X_{(j)}) - F(X_{(i)})] \right\} \\ &= P_r \left[U < p < U + V \right], \end{aligned}$$

where $U + V$ is the disjoint sum of the sum of elementary coverages. There results:

$$P_r \left[X_{(i)} < \xi_p < X_{(j)} \right] = I_p(i, n-i+1) - I_p(j, n-j+1), \quad (\text{III-1})$$

where $I_x(a, b)$ is the incomplete beta function given by

$$I_x(a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^x t^{a-1}(1-t)^{b-1} dt.$$

Thus one would choose a confidence coefficient, say α , and then for each value of n select the i and j such that the right-hand side of (III-1) evaluated to not less than α .

Similarly, one can find tolerance limits on the population. That is, one can determine a sample size n , and two order statistics, $X_{(k)}$, $X_{(n-k+1)}$, so that for any cdf $F(x)$ the probability is α that the fraction of the population bounded by $X_{(k)}$ and $X_{(n-k+1)}$ is at least β . We observe that the probability element of $U = F(X_{(n-k+1)}) - F(X_{(k)})$ is²

$$\frac{\Gamma(n+1)}{\Gamma(2k)\Gamma(n-2k+1)} u^{n-2k}(1-u)^{2k-1} du .$$

Thus for a given k we solve for the smallest value of n for which

$$1 - I_{\beta}(n-2k+1, 2k) \geq \alpha .$$

D. Confidence Bands for the cdf $F(x)$

Given a sample of size n , we can obtain the sample cdf $F_n(x)$, which is defined as follows:

$$F_n(x) = \begin{cases} 0 & x < X_{(1)} \\ i/n & X_{(i)} \leq x < X_{(i+1)} \\ 1 & x \geq X_{(n)} \end{cases} .$$

It is desirable to find confidence bands for a continuous cdf $F(x)$ from the sample cdf. It can be shown² that

$$\lim_{n \rightarrow \infty} P_r \left\{ \sup_{(x)} |F(x) - F_n(x)| n^{1/2} \leq \lambda \right\} = \Phi(\lambda) ,$$

where

$$\Phi(\lambda) = \sum_{k=-\infty}^{\infty} (-1)^k e^{-2k^2 \lambda^2} .$$

Thus for a given confidence coefficient α , approximate confidence bands for $F(x)$ are given by the functions $F_n(x) \pm \lambda_\alpha/n^{1/2}$, where λ_α is found from $\Phi(\lambda_\alpha) = \alpha$. Hence for large enough n the probability is approximately α that the curve $F(x)$ is contained in the band $F_n(x) \pm \lambda_\alpha/n^{1/2}$.

III.1.3 Estimation of Population Parameters Using Sample Quantiles

We turn now to the question of how to use sample quantiles to obtain estimates of the mean and variance of the underlying population, and how to do so optimally; that is, so as to minimize the variance of the estimates. It is well to stress at the outset that determination of the optimal quantiles requires a knowledge of the density function of the underlying population whose parameters are being estimated. Thus the results will not be optimal when applied to populations whose densities depart from that assumed in finding the quantiles. The derivations in the literature have assumed populations with normal distributions. In this section we will largely summarize the results obtained by Eisenberger and Posner³ and Mosteller⁴.

A. Estimate of the Mean Using a Single Quantile

We recall from Section III.1.2 B that the variance of a sample quantile, z_p , is asymptotically normal with mean ξ_p and variance

$$\sigma^2(z_p) = \frac{p(1-p)}{nf^2(\xi_p)} ,$$

where $f(x)$ is the population density function and ξ_p is determined from

$$\int_{-\infty}^{\xi_p} f(x) dx = p .$$

The minimum of $\sigma^2(z_p)$ will occur for $p = 0.5$ when $f(x)$ is symmetric. If the population is normal, with $N(0;1)$, we have

$$\sigma^2(z_p) = \frac{(0.5)^2}{nf^2(0)} = \frac{1.57}{n} .$$

Thus the estimation of the mean, μ , is taken as the median; that is, the sample quantile of order 0.5. The efficiency of this estimate is 0.637. We use as the definition of efficiency the ratio of the variance of the estimate using the entire sample, where the underlying population is assumed to be normal with $N(0;1)$, to the variance of the estimate obtained using quantiles:

$$\text{eff}(\hat{\mu}) = \sigma^2/n\sigma^2(\hat{\mu})$$

$$\text{eff}(\hat{\sigma}) = \sigma^2/2n\sigma^2(\hat{\sigma}) .$$

B. Estimate of the Mean and Variance Using Two Quantiles

Let z_1 and z_2 denote the sample quantiles of orders p_1 and p_2 such that $p_1 < p_2 = 1 - p_1$. Assuming z_1 and z_2 to be jointly normal, the maximum-likelihood estimates are:

$$\hat{\mu} = \frac{1}{2}(z_1 + z_2)$$

$$\hat{\sigma} = \frac{z_2 - z_1}{\xi_2 - \xi_1} .$$

Requiring that the quantiles be symmetric, the latter expression can be written

$$\hat{\sigma} = \frac{z_2 - z_1}{2\xi_2} .$$

The variance of $\hat{\mu}$ is obtained as:

$$\begin{aligned}
 \sigma^2(\hat{\mu}) &= E\{[\hat{\mu} - E(\hat{\mu})]^2\} \\
 &= E\{\hat{\mu}^2\} - [E\{\hat{\mu}\}]^2 \\
 &= E\left\{z_1^2/4 + z_1z_2/2 + z_2^2/4\right\} - \frac{\bar{z}_1^2}{4} - \frac{\bar{z}_2^2}{4} - \frac{\bar{z}_1\bar{z}_2}{2} \\
 &= \frac{1}{4} E\left\{z_1^2 - \frac{\bar{z}_1^2}{4}\right\} + \frac{1}{4} E\left\{z_2^2 - \frac{\bar{z}_2^2}{4}\right\} + \frac{1}{2} E\left\{z_1z_2 - \frac{\bar{z}_1\bar{z}_2}{2}\right\} \\
 &= \sigma_1^2/4 + \sigma_2^2/4 + \sigma_1\sigma_2\rho_{12}/2 \quad ,
 \end{aligned}$$

But we have

$$\sigma_1^2 = \frac{p_1(1-p_1)}{nf_1^2} \quad ; \quad \sigma_2^2 = \frac{p_2(1-p_2)}{nf_2^2} \quad ,$$

where $p_2 = 1 - p_1$; $f_1 = f(\xi_1) = f_2 = f(\xi_2)$ (from symmetry). Substitution into the variance equation gives

$$\sigma^2(\hat{\mu}) = \frac{p_1}{2nf_1^2} \quad .$$

Minimization of this expression results in the value $p_1 = 0.2709$. From symmetry then, $p_2 = 0.7291$. We get for the efficiency

$$\text{eff}(\hat{\mu}) = \sigma^2 / n \sigma^2(\hat{\mu}) = \frac{2f_1^2}{p_1} = 0.81$$

Through similar manipulations we can derive the expression for the variance of the estimator $\hat{\sigma}$:

$$\begin{aligned} \sigma^2(\hat{\sigma}) &= \frac{1}{4\xi_1^2} \left[\sigma_1^2 + \sigma_2^2 - 2\sigma_1\sigma_2\rho_{12} \right] \\ &= \frac{1}{2n\xi_1^2} \frac{p_1(1 - 2p_1)}{f_1^2} \end{aligned}$$

This function has a minimum when $p_1 = 0.0694$, giving

$$\sigma^2(\hat{\sigma}) = 0.767/n$$

The efficiency is

$$\text{eff}(\hat{\sigma}) = \frac{1/2n}{0.767/n} = 0.652$$

C. Estimates Using More Than Two Quantiles

In general, one can obtain estimators using any number of quantiles. The derivations though are lengthy and will be omitted, but we will indicate their development. First, however, we wish to point out that in the preceding derivation for two quantiles the variances of the estimators were minimized separately, giving two different sets of quantiles, each optimum for the particular estimator. In practical situations, however, a single set of quantiles would have to be used. Such a set can be obtained by minimizing, say,

the sum of the variances, $\sigma^2(\hat{\mu}) + \sigma^2(\hat{\sigma})$, or some other linear combination. Although the efficiencies of the estimators obtained when minimizing the sum of the variances are less than those resulting from separate minimization, the results represent an effective compromise.

Suppose we wish to obtain estimates $\hat{\mu}$ and $\hat{\sigma}$ using four quantiles.

We take:

$$x_1 = \frac{1}{2} (z_1 + z_4)$$

$$x_2 = \frac{1}{2} (z_2 + z_3) ,$$

and $p_1 + p_4 = p_2 + p_3 = 1$. Again the use of symmetric quantiles can be shown to result in maximum efficiency. From maximum-likelihood estimation theory, the estimator for μ can be derived as:

$$\hat{\mu} = \frac{(z_1 + z_4) (p_2 f_1^2 - p_1 f_1 f_2) + (z_2 + z_3) (p_1 f_2^2 - p_1 f_1 f_2)}{2 (p_1 f_2^2 + p_2 f_1^2 - 2p_1 f_1 f_2)}$$

with

$$\sigma^2(\hat{\mu}) = \frac{p_1 p_2 - p_1^2}{2n (p_1 f_2^2 + p_2 f_1^2 - 2p_1 f_1 f_2)}$$

Minimization of $\sigma^2(\hat{\mu})$ results in the estimator

$$\hat{\mu} = 0.198 (z_1 + z_4) + 0.302 (z_2 + z_3) ,$$

where $p_1 = 0.1068$, $p_2 = 0.3512$, $p_3 = 0.6488$, $p_4 = 0.8932$, and $\text{eff}(\hat{\mu}) = 0.920$.

In a similar manner minimization of $\sigma^2(\hat{\sigma})$ results in

$$\hat{\sigma} = 0.116 (z_4 - z_1) + 0.236 (z_3 - z_2) ,$$

where we have used

$$x_1 = \frac{z_4 - z_1}{2\xi_4} , \quad x_2 = \frac{z_3 - z_2}{2\xi_3} .$$

The resulting quantiles are: $p_1 = 0.023$, $p_2 = 0.1271$, $p_3 = 0.8729$, $p_4 = 0.977$;
and the efficiency is: $\text{eff}(\hat{\sigma}) = 0.824$.

If we choose to minimize the sum $\sigma^2(\hat{\mu}) + \sigma^2(\hat{\sigma})$ for four quantiles
we get

$$\hat{\mu} = 0.141 (z_1 + z_4) + 0.359 (z_2 + z_3)$$

$$\hat{\sigma} = 0.258 (z_4 - z_1) + 0.205 (z_3 - z_2) ,$$

where

$$p_1 = 0.668, p_2 = 0.2912, p_3 = 0.7088, p_4 = 0.9322, \text{ with efficiencies}$$

$$\text{eff}(\hat{\mu}) = 0.908$$

$$\text{eff}(\hat{\sigma}) = 0.735 .$$

Results using from six to twenty quantiles are tabulated in Reference [3] for a variety of linear combinations of $\sigma^2(\hat{\mu}) + b\sigma^2(\hat{\sigma})$. The derivations are carried out by reducing the number of quantiles to a previously treated case through a change of variables. The efficiencies of the estimators are increased for a larger number of quantiles, but the data-compression ratio is correspondingly decreased. Also, the equipment complexity increases with a

larger number of quantiles if a special-purpose device is used for implementation. Although the tabulated results in Reference [3] were obtained assuming an underlying normal population, $N(0;1)$, it can be shown that the results apply equally to the distribution $N(\mu;\sigma)$.

As an illustration of the use of quantiles for estimating the mean and the variance we include the following examples. Table III-1 contains the histograms for a data channel obtained with 64 and 32 quantization levels respectively. Table III-2 shows a comparison between estimates obtained directly from the histograms and those obtained by use of quantiles. The appropriate quantile formulas, listed below, are obtained by minimizing the sum of the variances $\sigma^2(\hat{\mu}) + \sigma^2(\hat{\sigma})$. Table III-2 shows quite clearly that for the severely non-normal distributions of Table III-1 the estimates of the variance are very poor indeed. Figure III-1 shows a number of histograms generated using the data generation program created for the simulation study (see Section 3). Corresponding values for mean and standard deviation are given in Table III-3, along with the estimates obtained using quantiles. In this case the agreement is much better than in Table III-2.

TABLE III-1

<u>Sample No.</u>	<u>A 64 Levels</u>	<u>B 32 Levels</u>
1	2	275
2	280	200
3	0	1764
4	320	2
5	1627	17
6	10	0
7	0	0
8	2	0
9	14	2
10	3	0
11	0	0
12	0	0
13	0	0
14	0	1
15	0	1
16	0	0
17	2	0
18	0	0
19	0	0
20	0	0
21	0	5612
22	0	0
23	0	0
24	0	0
25	0	0
26	0	0
27	0	0
28	1	0
29	0	0
30	1	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0
41	865	0
42	4747	0
43	0	0
44	0	0
45	0	0

TABLE III-1 (continued)

<u>Sample No.</u>	<u>A</u> <u>64 Levels</u>	<u>B</u> <u>32 Levels</u>
46	0	
47	0	
48	0	
49	0	
50	0	
51	0	
52	0	
53	0	
54	0	
55	0	
56	0	
57	0	
58	0	
59	0	
60	0	
61	0	
62	0	
63	0	
64	0	

TABLE III-2

RESULTS OF ESTIMATES USING QUANTILES BASED ON HISTOGRAMS IN TABLE III-1

		Data Set A	Data Set B
Histogram Mean		31.13	15.74
Quantile Mean	4 quantiles	36.42	18.60
	6 "	34.03	17.76
	8 "	32.34	16.58
Histogram Variance		285.40	68.76
Quantile Variance	4 quantiles	105.41	24.02
	6 "	139.38	38.75
	8 "	176.26	43.51

TABLE III-3

MEAN, VARIANCE, AND STANDARD DEVIATION FOR THE HISTOGRAMS OF FIGURE III-1. RESULTS ARE GIVEN FOR ENTIRE HISTOGRAM AND FOR 4, 6, 8, AND 10 QUANTILES.

<u>MEAN</u>	<u>VAR</u>	<u>S.D.</u>		
31.429	20.113	4.485	Entire Hist.	} a)
31.500	19.176	4.379	4 Quantiles	
31.500	19.055	4.365	6 Quantiles	
31.424	20.255	4.500	8 Quantiles	
31.500	20.142	4.488	10 Quantiles	
31.146	349.646	18.699	Entire Hist.	} b)
31.359	399.320	19.983	4 Quantiles	
31.035	360.073	18.976	6 Quantiles	
31.166	345.893	18.598	8 Quantiles	
31.125	351.173	18.740	10 Quantiles	
32.177	509.805	22.579	Entire Hist.	} c)
32.577	578.114	24.044	4 Quantiles	
32.386	497.326	22.301	6 Quantiles	
32.226	488.591	22.104	8 Quantiles	
32.091	483.771	21.995	10 Quantiles	
15.307	128.092	11.318	Entire Hist.	} d)
15.141	146.362	12.098	4 Quantiles	
15.205	120.965	10.998	6 Quantiles	
15.101	123.541	11.115	8 Quantiles	
15.263	118.111	10.868	10 Quantiles	

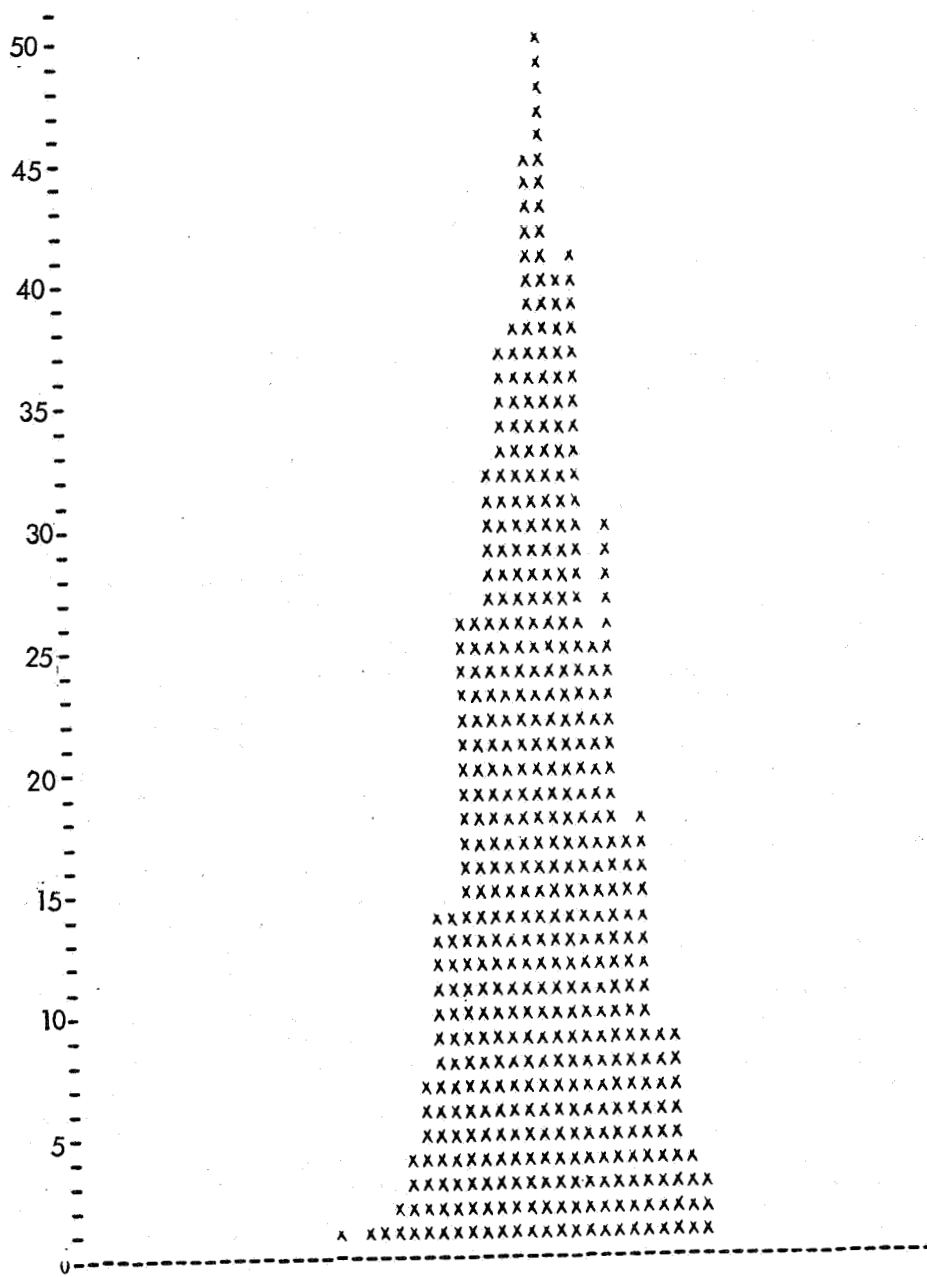


Figure III-1a. Cumulative Histogram for Neutron Detector 16-Channel Pulse Height Analysis - 1000 Samples, 64 Levels. (Actual Scale is 2 x that shown.)

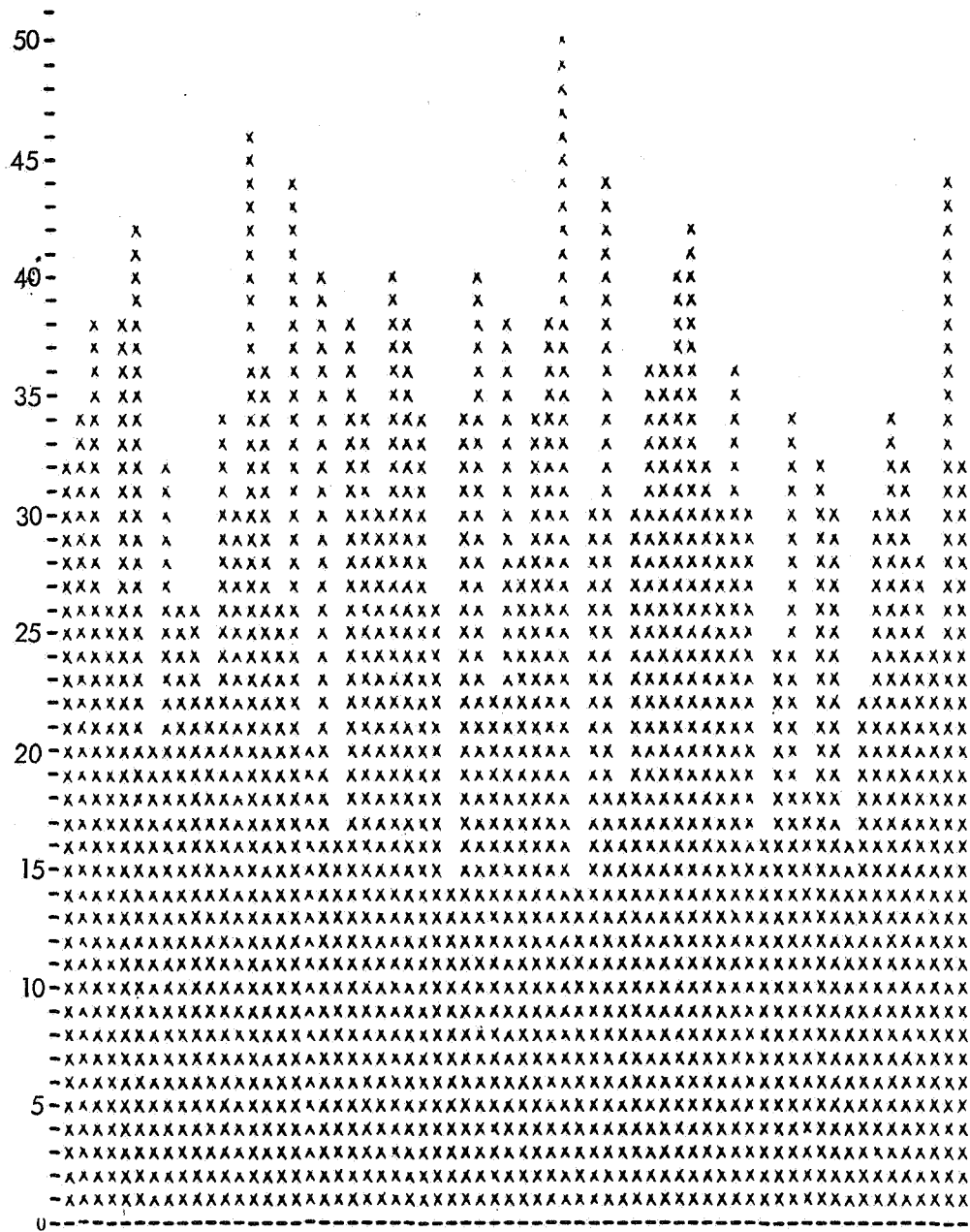


Figure III-1b. Histogram for Neutron Detector 14-bit Counter-1000 Samples, 64 Levels.
 (Actual Scale is 0.46 x that shown.)

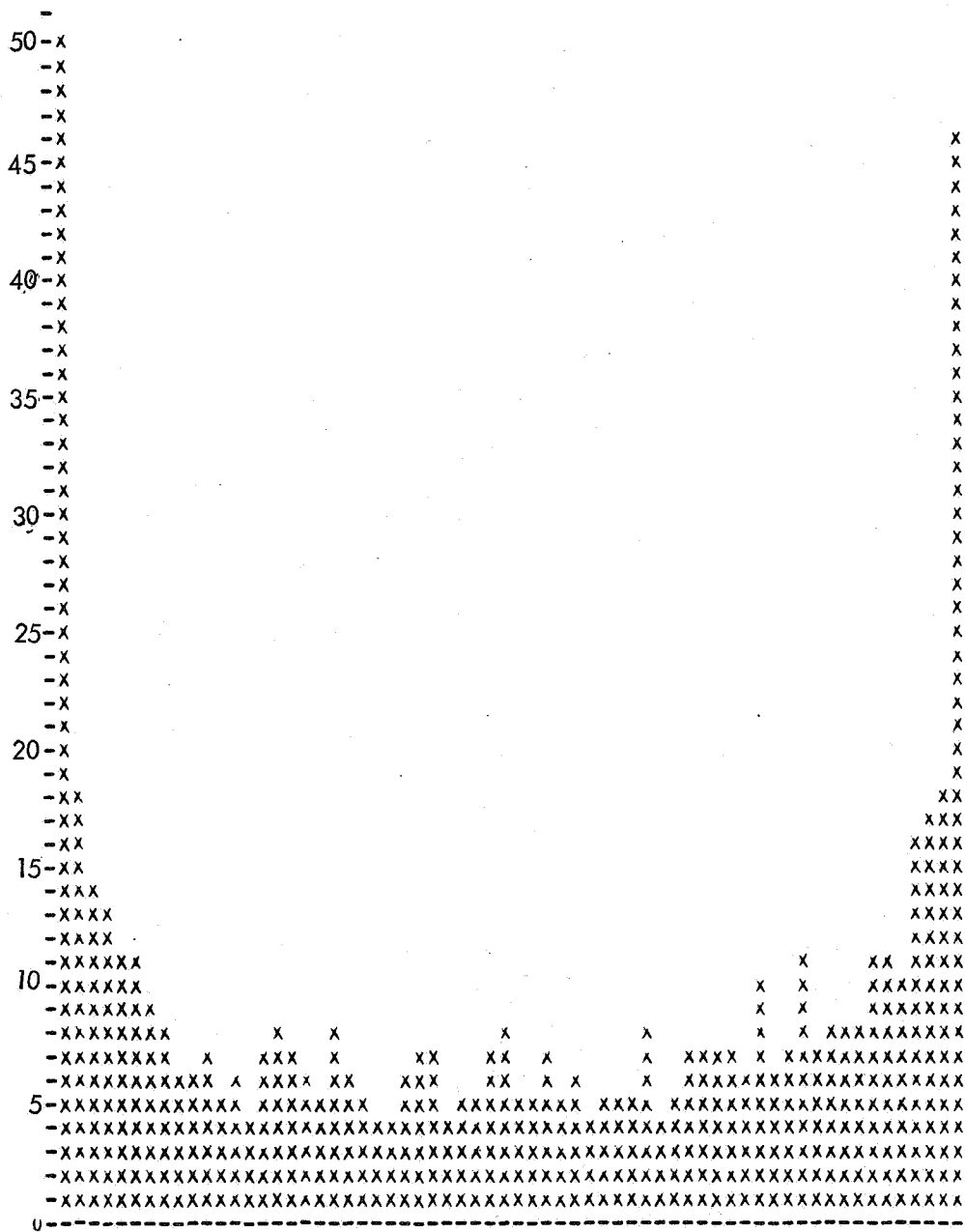


Figure III-1c. Histogram for One Axis of Magnetometer - 2220 Samples, 64 Levels.
 (Actual Scale is 3.7 x that Shown.)

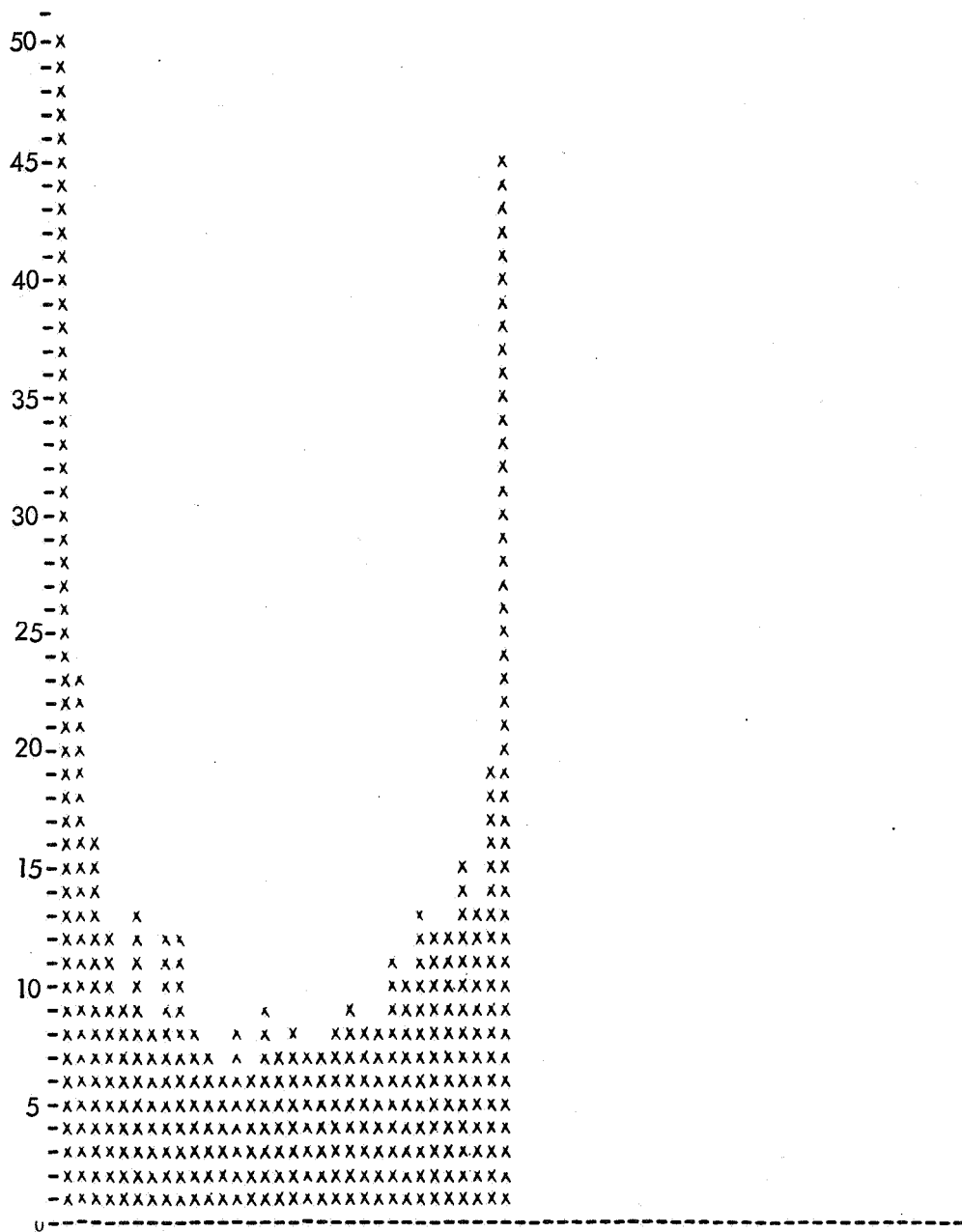


Figure III-Id. Histogram for Another Axis of Magnetometer - 2220 Samples, 64 Levels.
 (Actual Scale is 5.22 x that shown.)

for 4 quantiles:

$$\hat{\mu} = 0.141(z_1 + z_4) + 0.359(z_2 + z_3)$$

$$\hat{\sigma} = 0.258(z_4 - z_1) + 0.205(z_3 - z_2)$$

where

z_1, z_2, z_3, z_4 are of orders $p_1 = 0.0668, p_2 = 0.2912,$

$p_3 = 0.7088, p_4 = 0.9332; \text{eff } \hat{\mu} = 0.9080; \text{eff } \hat{\sigma} = 0.7350$

for 6 quantiles

$$\hat{\mu} = 0.0497(z_1 + z_6) + 0.1550(z_2 + z_5) + 0.2953(z_3 + z_4)$$

$$\hat{\sigma} = 0.1088(z_6 - z_1) + 0.1952(z_5 - z_2) + 0.1228(z_4 - z_3)$$

where

$z_1, z_2, z_3, z_4, z_5, z_6$ are of orders $p_1 = 0.0231, p_2 = 0.1180,$

$p_3 = 0.3369, p_4 = 0.6631, p_5 = 0.8820, p_6 = 0.9769$

$\text{eff } \hat{\mu} = 0.9459; \text{eff } \hat{\sigma} = 0.8541$

for 8 quantiles

$$\hat{\mu} = 0.0249(z_1 + z_8) + 0.0764(z_2 + z_7) + 0.1568(z_3 + z_6) + 0.2419(z_4 + z_5)$$

$$\hat{\sigma} = 0.0600(z_8 - z_1) + 0.1249(z_7 - z_2) + 0.1528(z_6 - z_3) + 0.0789(z_5 - z_4)$$

where

$z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8$ are of orders $p_1 = 0.0119,$

$p_2 = 0.0604, p_3 = 0.1721, p_4 = 0.3711, p_5 = 0.6289,$

$p_6 = 0.8279, p_7 = 0.9396, p_8 = 0.9881$

$\text{eff } \hat{\mu} = 0.9659; \quad \text{eff } \hat{\sigma} = 0.9050$

III.1.4 Hypothesis Testing Using Quantiles

In Section III.1.3 it was shown that using as few as four quantiles we can estimate the mean and variance of a normal population with efficiencies no lower than approximately 0.9 and 0.73, respectively. It is also possible to develop quantile tests that decide whether or not it is reasonable to regard a set of n observed values of a random variable as originating from a population with a specified distribution function. Similarly, one can construct quantile tests to decide whether the parameters of a normal distribution are μ_1 and σ_1 , or μ_2 and σ_2 ; or, if one of the parameters, say σ , is known, whether the mean is μ_1 or μ_2 . It is possible to choose the quantiles optimally so as to maximize the power of the test; that is, the probability of rejecting the null hypothesis when it is false (and thereby accepting the alternative hypothesis). The details of such tests are given in References [3], [5], and [6], and only a few of the results will be stated here.

Suppose that H_0 is the null hypothesis that the parent population has a normal probability density function $f(x)$, and H_1 is the alternative hypothesis that the density function is not $f(x)$. It is desired to establish a criterion that will enable one to accept or reject H_0 as a function of the sample quantiles z_i , so that the probability of rejecting H_0 when it is true is less than some given ϵ .

The r.v.'s z_i are not independent, but they can be transformed into independent r.v.'s x_i by the Gram-Schmidt orthogonalization procedure. The x_i will then be independent, normally distributed r.v.'s under the null hypothesis, each with zero mean and unit variance. The likelihood function is then the joint density function of the x_i , or

$$L(x_1, \dots, x_k) = \frac{1}{(2\pi)^{k/2}} \exp \left[-\frac{1}{2} \sum_{i=1}^k x_i^2 \right],$$

The critical region is then an interval $0 < L < A$, where $A < (2\pi)^{-k/2}$, A being determined such that given H_0 the probability of $L(x_1, \dots, x_k)$ lying in the interval is ϵ . Note, though, that $L(x_1, \dots, x_k)$ is a monotone function of

$$y = \sum_{i=1}^k x_i^2,$$

and $y = 0$ for $L = (2\pi)^{-k/2}$, and $y \rightarrow \infty$ for $L \rightarrow 0$. Thus the critical region $0 < L < A$ is equivalent to a critical region $y > B$, where B is determined so that the probability of $y > B$ equals ϵ . We see that y has the chi-squared distribution with k degrees of freedom. Denoting the density function of y by $K(y)$, we have that B is the solution to the equation

$$\int_0^B K(y) dy = 1 - \epsilon.$$

In order to carry out the above orthogonalization it is necessary to have estimates of the parameters of the parent population, as well as having the orders of the quantiles used. These may be obtained as discussed in Section III.1.3 by minimizing, say, the sum of the variances of μ and σ for the case of a normal population with unknown mean and variance.

Often it may be useful to have procedures that enable one to test various hypotheses concerning the parameters of a population. For example, one may know the variance of a normal population and wish to test the null hypothesis

$$H_0: f(x) = N(\mu_1; \sigma)$$

against the simple alternative

$$H_1: f(x) = N(\mu_2; \sigma) , \quad \text{for } \mu_2 > \mu_1 .$$

Or, perhaps the variance is unknown and one again wishes to test the population means. Additional tests may be required when one has two sets of independent samples available and either or both of each of the parent normal population parameters are unknown. Such tests, and others, are thoroughly described in [5] and [6], using different numbers of quantiles. The objective in designing the tests is to choose the orders of the quantiles so as to maximize the power of the test (i.e., the probability of rejecting H_0 when H_1 is true) for a fixed probability ϵ of Type I error (i.e., the probability of rejecting H_0 when H_0 is true). The greater number of quantiles used, the greater will be the power of the test. The power will also depend critically on the nature of H_1 , as well, of course, as on ϵ .

One can also design tests to give high power against specific distributions, in particular against a bimodal distribution. Thus it may happen that arriving particles come from two possible sources. The resulting density function will be

$$f(x) = pf_1(x) + (1-p)f_2(x) ,$$

where p is the probability that a particle originates from the first source, and f_1 and f_2 are the respective density functions. Under the null hypothesis, H_0 , we assume that the observations come from a normally distributed population with unknown mean and variance. Thus we must first obtain estimates for μ and σ by minimizing, say, $\sigma^2(\mu) + \sigma^2(\sigma)$. Then one can form various linear combinations of the normalized sample quantiles so as to test for both the symmetry of a unimodal distribution and the local minimum between the maxima of a bimodal distribution.

III.2 DIGITAL FILTERING

Increasing attention is being paid to digital filtering in data processing. As computing speeds continue to increase, the possibility exists of using digital filtering at high sampling rates. Some functions currently implemented by analog means in communication systems may be supplanted by digital techniques. The availability of the CDS leads to the consideration of employing digital filtering for data processing and replacing analog filters in the instrument electronics. Digital filtering can be employed to reduce signal bandwidths and provide estimates of spectral energy at different frequencies. Lowpass, bandpass, and highpass filters can be implemented digitally in the CDS.

III.2.1 Digital Filter Design Methods

The design of a digital filter can be accomplished by a variety of methods. Generally, the starting point is a continuous filter, certain of whose characteristics it is desired to preserve in the digitalization. In other cases, however, one may wish to specify the digital design directly, as for example, in deriving the optimum weighting sequence for sampled data in the presence of noise. Our attention in this section is devoted to the former type of design procedures. An excellent reference for the latter is Blackman.⁷

Assume a transfer function $H(s)$, where s denotes complex frequency. One way to obtain a sampled-data approximation is to apply the standard z -transform to $H(s)$. This can be done by expanding $H(s)$ into partial fractions and applying the z -transform term by term, (Freeman,⁸ Kaiser,⁹ Rader and Gold,¹⁰ Ragazzini and Franklin¹¹). Thus for a simple pole we have

$$\frac{A_i}{s + s_i} \Rightarrow \frac{A_i}{1 - e^{-s_i T} z^{-1}},$$

where T is the sampling period.

For multiple poles the transformation gives

$$\frac{A_i}{(s+s_i)^m} \Rightarrow \left[\frac{(-1)^{m-1}}{(m-1)!} \frac{\partial^{m-1}}{\partial a^{m-1}} \left(\frac{A_i}{1 - e^{-aT} z^{-1}} \right) \right] \Bigg|_{a=s_i}$$

It is also possible to find the z-transform as follows (Ragazzini and Franklin¹¹):

$$H(z) = \sum_{\substack{\text{poles of} \\ H(s)}} \text{res} \left[H(p_k) \frac{1}{1 - e^{p_k T} z^{-1}} \right], \quad (\text{III-2})$$

where the poles of $H(s)$ are designated as p_k . (This expression arises as a result of considering the integral involved in writing the Laplace transform of a sampled time function.) The direct z-transform has the property that the impulse response of the digital filter will be equal to the sampled impulse response of the continuous filter.

The bilinear z-transformation (Kaiser⁹, Golden and Kaiser¹², Radar and Gold¹⁰) is another way of obtaining a digital representation of $H(s)$. It is defined by the mapping transformation

$$s = \frac{2}{T} \tanh(s_1 T/2). \quad (\text{III-3})$$

Considering only the principal values of ω_1 , $-\pi/T < \omega_1 < \pi/T$, Eq. (III-3) maps the entire s-plane into the strip in the s_1 -plane bounded by $\omega_1 = -\pi/T$ and $\omega_1 = \pi/T$. Thus, noting that $z^{-1} = e^{-s_1 T}$, Eq. (III-3) becomes

$$s = \frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})},$$

and we obtain

$$H^*(z) = H(s) \Big|_{s = \frac{2}{T} \frac{(1-z^{-1})}{(1+z^{-1})}}$$

as the bilinear z-form of $H(s)$.

From Eq. (III-3), substituting $j\omega$ for s we get

$$\omega = \frac{2}{T} \tan(\omega_1 T/2). \quad (\text{III-4})$$

Equation (III-4) shows that the bilinear z -form will introduce a nonlinear warping in the frequency scale. Compensation for this warping can be made by pre-warping the critical frequencies of the continuous filter in such a way that the bilinear z -form will shift them back to the desired values (Golden and Kaiser¹²). Unlike the direct z -transform, the bilinear z -form will give a sampled data filter with amplitude characteristics identical to those of the continuous filter in the baseband (if the warping effect is compensated).

A digital filter can also be designed on the basis of an assumed squared magnitude function (Rader and Gold¹⁰). Thus, for example, in the design of continuous filters we would have

$$|F(j\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2n}}$$

and

$$|F(j\omega)|^2 = \frac{1}{1 + e^{2V_n^2 \left(\frac{\omega}{\omega_c}\right)^2}}$$

as the squared magnitude functions for Butterworth and Chebyshev filters, respectively. Now for z on the unit circle, the digital system function $H(z)$ corresponding to a linear difference equation will be a rational function of $e^{j\omega T}$. Thus, the squared magnitude $|H(e^{j\omega T})|^2$ will always be a ratio of two trigonometric functions of ωT . Hence the problem of designing the digital filter is to choose a squared magnitude function suitable for the desired filtering application and then to obtain the corresponding digital filter specified by that function. Rader and Gold¹⁰ show that the necessary conditions on the squared magnitude function for realizability are: any pole

inside the unit circle must have a mate outside the circle of inverse magnitude and the same angle; all poles must appear in complex conjugate pairs. Then the required digital filter is obtained by: discarding all poles and zeros at the origin; putting the remaining poles and zeros into a one-to-one correspondence with respect to the unit circle as described above; discarding one of each corresponding pair of poles and zeros with the retained poles all within the unit circle; and forming the transfer function from the retained poles and zeros. The resulting transfer function will then be a rational function of z with real coefficients, and will be stable.

It turns out that the above realizability conditions for a stable transfer function will be met if, in the squared magnitude function, ωT only appears as a squared trigonometric function of $\omega T/2$, and if these expressions are combined only by addition, subtraction, multiplication, and division. A similar development has been given by Holtz and Leondes.¹³ Their starting point is to find a class of functions whose amplitude frequency response approximates that of an ideal lowpass or highpass filter. Then the corresponding Q-transform* is determined; it has an identical amplitude frequency response over the entire frequency range and is stable. As in Rader and Gold,¹⁰ it is found that the class of real, even, rational polynomials of the trigonometric functions gives a stable, realizable digital filter. Further, these filters will approach the ideal lowpass or highpass characteristic as the order of the polynomial is increased. Bandpass and bandreject filters can be obtained by suitable combination of the lowpass and highpass designs.

Another method of digital filter design is to cascade sections of elementary designs so as to obtain a frequency shaping. Thus, for example, the difference equation

$$y(nT) = x(nT) - x(nT-T)$$

*The Q-transform corresponds to the bilateral z -transform, $z = e^{j\omega T}$, where ω is restricted to real values.

has a squared magnitude function (power transfer function)

$$H(\omega) = 2 - 2 \cos \omega T \quad .$$

$H(\omega)$ will have the value zero at $\omega = 0, 2\pi/T, 4\pi/T, \dots$, and will have maximum transmission at $\pi/T, 3\pi/T, 5\pi/T, \dots$. Such a filter is called a comb filter. Similarly, one can find the power transfer function for other difference equations of both recursive and nonrecursive type. Whittlesey¹⁴ has shown the power transfer characteristics for a number of composite filters to achieve both lowpass and bandpass operation.

Rader and Gold¹⁰ have also considered composite filters, calling their method the "frequency sampling technique." The idea is to cascade a comb filter with a resonator such that the poles of the resonator cancel, say, the k^{th} zero of the comb filter, and its conjugate. Then the composite filter will have zero gain at all frequencies for which the comb filter is zero, except the frequency corresponding to the k^{th} zero. Thus by multiplying the composite filter by an appropriate constant one can "synthesize" a desired gain at the frequency corresponding to the k^{th} zero of the comb filter. Now if the "teeth" of the comb filter are spaced relatively close, a resonator can be designed for each comb frequency in a desired band, and the multiplying constants chosen so as to synthesize any desired magnitude response. This process is quite analogous to the way in which a band-limited function is formed from a weighted sum of $(\sin \tau/\tau)$ -functions. The authors further show that the phase response of the synthesized filter will be linear and that the filter will have an impulse response of duration m sample periods, where m is the number of zeros of the comb filter.

Still another procedure for designing a digital filter is the method of moments, suggested by Blackman.⁷ Assume that one has a series expansion of a continuous filter.

$$H(s) = c_0 + c_1 s + c_2 s^2 + \dots \quad . \quad \text{(III-5)}$$

It is desired to obtain a sampled data approximation to $H(s)$ of the form

$$H^*(z) = \frac{\sum_{i=0}^M a_i z^{-i}}{1 + \sum_{j=1}^N b_j z^{-j}} \quad (III-6)$$

Since $z = e^{sT}$, we write

$$e^{msT} = 1 + msT + \frac{(msT)^2}{2!} + \frac{(msT)^3}{3!} + \dots \quad (III-7)$$

Then, substituting the appropriate form of Eq. (III-7) into Eq. (III-6), and equating the result to Eq. (III-5), we can obtain simultaneous equations to solve for the a_i , b_j in Eq. (III-6). The principal disadvantage of the method (aside from having to know $H(s)$ in expanded form and solve a set of simultaneous equations), is that the poles of discrete filter may be significantly different from those of $H(s)$, and hence instability may result.

In describing all of the above design methods no mention was made of their actual implementation, either in a general purpose or a special purpose digital computer. Clearly some schemes will have longer computing times and larger memory requirements than others. These considerations aside, though, there are questions that must be examined as to how best to implement a specified digital transfer function. These questions involve such factors as: required computer word length to maintain coefficient accuracy necessary for stability; effect of roundoff noise introduced at the input and at intermediate computations; effect of various arrangements (i.e., direct, cascade or parallel) of the digital filter as regards the coefficient accuracy and round-off noise problems.

Kaiser¹⁵ has given an analysis of the required coefficient accuracy necessary to ensure stability. He shows that the bound on coefficient accu-

acy depends on both the order of the filter and the sampling rate (for sampling rates large compared to the moduli of the filter poles). For a given sampling rate, doubling the order of the filter will require approximately twice as many digits for representation of the coefficients. Doubling the sampling rate for an n^{th} order filter will require an additional $n \log_{10} 2$ decimal digits in the representation of the coefficients. These bounds appear to be pessimistic. Further, they are valid only for the direct realization of a filter.

Gold and Rader¹⁶ have considered a simple noise model for a second-order filter. They compute the mean square output noise for two different realizations of the filter, a direct and a cascade (canonical), and find that the latter is less sensitive to noise. Their paper also includes experimental results to show the effect of lower precision arithmetic on the noise measure and comparison of these results with the values predicted from the analysis. The results appear to provide verification of the model for the second-order filter considered.

III.2.2 Design Example

Using the bilinear z-transformation discussed in Section III.2.1, two possible realizations of a third-order, lowpass Butterworth filter are illustrated. For cutoff frequency ω_0 , the Butterworth transfer function is given by:

$$H(s) = \frac{1}{\left(\frac{s}{\omega_0}\right)^3 + 2\left(\frac{s}{\omega_0}\right)^2 + 2\left(\frac{s}{\omega_0}\right) + 1} .$$

The corresponding digital transfer function becomes

$$H'(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}{1 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3}} . \quad (\text{III-8})$$

Letting $a = \tan \frac{\omega_0 T}{2}$, where T is the sampling period, the coefficients in Eq. (III-8) are given by

$$\begin{aligned}
 a_0 &= \frac{a^3}{a^3 + 2a^2 + 2a + 1} & b_0 &= \frac{3a^3 + 2a^2 - 2a - 3}{a^3 + 2a^2 + 2a + 1} \\
 a_1 &= \frac{3a^3}{a^3 + 2a^2 + 2a + 1} & b_2 &= \frac{3a^3 - 2a^2 - 2a + 3}{a^3 + 2a^2 + 2a + 1} \\
 a_2 &= a_1 & b_3 &= \frac{a^3 - 2a^2 + 2a - 1}{a^3 + 2a^2 + 2a + 1} \\
 a_3 &= a_0
 \end{aligned}$$

for

$$[-\pi < \omega_0 T < \pi].$$

The transfer function, Eq. (III-8), can be implemented directly, by a single difference equation, or canonically, by a pair of simultaneous difference equations. In the first case the result is:

$$\begin{aligned}
 y(nT) &= a_0 x(nT) + a_1 x(nT-T) + a_2 x(nT-2T) + a_3 x(nT-3T) \\
 &\quad - b_1 y(nT-T) - b_2 y(nT-2T) - b_3 y(nT-3T).
 \end{aligned}$$

To obtain the canonical form we write:

$$H'(z) = W(z) \left[\sum_i a_i z^{-i} \right].$$

Thus we obtain:

$$w(nT) = x(nT) - b_1w(nT-T) - b_2w(nT-2T) - b_3w(nT-3T)$$

and

$$y(nT) = a_0w(nT) + a_1w(nT-T) + a_2w(nT-2T) + a_3w(nT-3T) .$$

The canonical form is generally to be preferred in terms of ensuring greater protection against noise due to truncation and roundoff effects.¹⁶ In a spaceborne computer, where word size is likely to be restricted, one must be careful to investigate the best implementation for the limited precision. The differences in the two implementations are clearly shown in Table III-4. The entries in Table III-4 represent the filter response at the 150th sampling instant to a constant input of value 10.0 for the various precisions listed and for each of the two realizations. The canonical form is seen to offer far greater protection against computation noise, particularly below a 24-bit word length.

III.3 SPECTRAL ANALYSIS

Spectral analysis is a mathematical tool for estimating the power spectrum of a time function from a finite-length record. The spectral density of a finite-length record differs from that of the entire function; we can never hope to approach the true value except by taking longer records, which has obvious practical drawbacks dictated by storage and computational limitations. The physical process generating the time function must in general be considered nonstationary. Longer record lengths may not reveal the nonstationarity of the process as precisely as desired by the experimenter. There is a basic conflict when making spectral estimates between the spectral resolution that can be obtained and the stability (or variability) of the estimate. The finer one tries to resolve individual spectral lines in a signal the greater the spread of the confidence range about the estimate. Conversely, reducing the variability of the estimate reduces the resolution of spectral lines.

TABLE III-4

VALUE OF FILTER OUTPUT AT 150th SAMPLING
INSTANT FOR CONSTANT VALUE INPUT = 10.0.

<u>PRECISION</u>	<u>DIRECT</u>	<u>CANONICAL</u>
36 bits	10.000000	10.000000
30 bits	9.999998	10.000000
24 bits	9.999698	9.999903
18 bits	9.983826	9.992493
13 bits	9.248047	9.833984

In Section III.3.1, the mathematical basis of the resolution-stability tradeoff is shown, and in Section III.3.2, the steps for performing spectral analysis via the indirect method (Fourier transformation of the autocovariance function), are given.

III.3.1 Resolution and Variability of Spectral Estimates

A thorough treatment of the resolution-stability conflict is given in Blackman and Tukey;¹⁷ a few of the salient features are summarized below.

We let $\hat{P}_i(f_1)$ represent a smoothed estimate of the true spectral density $P(f)$ near frequency f_1 , the subscript "i" referring to the type of smoothing. This smoothing in the frequency domain is the equivalent of using a "lag window" in the time domain when forming the autocorrelation function. (Smoothing is essential because only a finite-length time record is available and particular frequencies cannot be identified exactly; i.e., the presence of neighboring frequencies with random phases has effects similar to those of noise in preventing such identification.) We can interpret the estimate $\hat{P}_i(f_1)$ according to the formula:

$$\text{ave} \{ \hat{P}_i(f) \} = \int_0^{\infty} H_i(f; f_1) P(f) df .$$

That is, the average value of the smoothed spectral estimate is the result of passing the true spectrum through a network with power transfer function $H_i(f; f_1)$. For compactness, we define $P_{i1}(f) = H_i(f; f_1) P(f)$.

We form the dimensionless variability of $\hat{P}_i(f_1)$ as:

$$\frac{\text{var} \{ \hat{P}_i(f_1) \}}{\left[\text{ave} \{ \hat{P}_i(f_1) \} \right]^2} = \frac{1}{T_n' W_e} , \quad (\text{III-9})$$

where T_n' is the effective record length (actually an approximation to it).

$$W_e = \frac{\left[\int_0^{\infty} P_{i1}(f) df \right]^2}{\int_0^{\infty} [P_{i1}(f)]^2 df} \quad (\text{III-10})$$

is the equivalent width of P_{i1} , and

$$\begin{aligned} \text{var} \{P_i(f_1)\} &= \text{cov} \{P_i(f_1), P_i(f_1)\} \\ &\cong \frac{1}{T'_n} \int_0^{\infty} P_{i1}(f) P_{i1}(f) df \end{aligned} \quad (\text{III-11})$$

It can be shown that Eqs. (III-9) through (III-11) will be valid only if $P(f)$ changes slowly in relation to $1/T'_n$. From Eq.(III-9), we note that the larger the record and the wider the equivalent width, the more stable the estimate. However, it is also seen that increasing the width makes the estimate refer to an average power density over a wider frequency range, thereby decreasing the resolution of the estimate. If $P(f)$ does not change slowly but consists of a sharp peak, the dimensionless variability can be shown to be approximately unity, a result that will be independent of T'_n so long as $1/T'_n$ is not small enough to be comparable with the width of the peak. Thus, resolution of spectral lines is obtained at the expense of stability of the estimate.

Perhaps a more convenient description of stability can be obtained by consideration of the equivalent number of degrees of freedom of the estimate; that is, the number of degrees of freedom in a chi-square distribution some multiple of which the estimate resembles. Thus, the coefficient of variation of $\chi_k^2 = y_1^2 + y_2^2 + \dots + y_k^2$, the ratio of rms deviation to average value, is $(2/k)^{1/2}$, where the y_i are independent Gaussian variables with zero mean and unit variance. We see that χ_k^2 becomes relatively less variable as k increases, and this is also true for any multiple of χ_k^2 .

We can obtain the value of k from:

$$k = \frac{2 (\text{average})^2}{\text{variance}} = \frac{2}{(\text{coeff. of variation})^2} .$$

Thus, when $P(f)$ is smooth, the number of degrees of freedom is, from Eq. (III-9):

$$k = 2T'_n W_e = \frac{W_e}{\Delta f} , \quad (\text{III-12})$$

where the second form expresses the number of degrees of freedom as the number of elementary frequency bands of width Δf contained in the equivalent width W_e . Equation (III-12) shows that the longer the record length and the wider the equivalent width, the larger the number of degrees of freedom and, hence, the more stable the estimate. Use can be made of standard chi-square tables to obtain confidence intervals for estimates.

On the other hand, if $P(f)$ has a single sharp peak, we find, using the result mentioned previously, that $k \approx 2$, provided that $1/T'_n$ is not small enough to be comparable with the width of the peak. Thus, we again observe the sacrifice in stability of the estimate in achieving resolution. Note that a single spectral line corresponds to specifying amplitude and phase and, hence, will have two degrees of freedom.

This discussion is intended to point out some of the pitfalls inherent in obtaining spectral estimates. The principal point to be emphasized is that reliable estimates are possible only if the experimenter has a rough idea of the actual shape of the spectrum being estimated. Such knowledge will then enable an intelligent choice of record length and smoothing filter. Another factor to be considered when dealing with digital implementation of spectral estimation is the sampling frequency. This frequency must be selected high enough to minimize possible aliasing errors, a selection that demands a knowledge of the spectrum shape. Generally, the spacecraft instruments have a well-defined bandwidth so that selection of the necessary sampling rate (Nyquist rate) is straightforward. It might be mentioned that sampling above

the Nyquist rate offers no increase in the number of degrees of freedom associated with the estimate. However, at rates below the Nyquist rate, we gain one degree of freedom for each additional observation of a record of fixed length.

III.3.2 Indirect Method for Obtaining Spectral Estimates for Sampled Data

The indirect method computes the spectral estimate from the Fourier transform of the autocovariance function. Spectral estimation via the direct method involves placing narrowband filters about the desired spectral lines and squaring and averaging the filter outputs over the observation time. The indirect method has been generally favored by investigators; it offers some computational advantages.

As discussed in Section III.3.1, we can obtain only a smoothed estimate of the true spectral density near any particular frequency. The smoothed estimate can be interpreted as the result of convolving the true spectrum with a spectral window. The spectral window ideally is like an impulse function in the frequency domain. Practically, it must be somewhat spread since finite lag times must be used in forming the autocovariance function (from which the power spectrum is derived via the Fourier transform). Herein lies the basic conflict between resolution and stability of the spectral estimate: the smaller the maximum lag used to compute the autocovariance, the stabler the estimate (i.e., the smaller the variability). On the other hand, a small lag means a wider spectral window, and hence poorer resolution of individual frequencies.

An outline of the steps necessary to produce spectral estimates using the Fourier transform of the autocovariance is given below in terms of finite-length discrete-time data:⁷

- 1) data values: x_0, x_1, \dots, x_n ; uniform spacing, Δt
- 2) compute mean value of data: $\bar{x} = \frac{1}{N+1} \sum_{i=0}^N x_i$
- 3) remove mean from data: $x_i = x_i - \bar{x}$

- 4) compute mean lagged products (autocovariance) at spacing Δt

$$C_r = \frac{1}{N - r + 1} \sum_{q=0}^{N-r} x_q x_{q+r}, \quad r = 0, 1, \dots, M$$

where $M \leq N$

- 5) compute raw spectral density estimates at frequency spacing $1/2M\Delta t$

$$V_s = 2\Delta t \left[C_0 + 2 \sum_{r=1}^{M-1} C_r \cos \frac{rs\pi}{M} + C_M \cos s\pi \right], \quad s = 0, 1, \dots, M$$

- 6) compute smoothed spectral estimates

$$P_0 = S_2 V_0 + S_2 V_1$$

$$P_s = S_1 V_{s-1} + S_2 V_s + S_3 V_{s+1}, \quad s = 1, 2, \dots, M-1$$

$$P_M = S_2 V_{M-1} + S_2 V_M$$

where S_1 , S_2 and S_3 are weights determined by the type of spectral window used.

To hold the variance of the estimates to a level well below the expected value, we should keep the ratio N/M large, typically, greater than 10. At the same time, we note that the resolution of the estimate is proportional to the reciprocal of M . Therefore, we are forced to accept a certain minimal resolution when demanding that the variability be below a fixed dB level.

III.4 DATA PROCESSING CONSIDERATIONS FOR PIONEER EXPERIMENTS

The availability of a general-purpose digital computer as the on-board central data processor leads to the consideration of algorithms for processing the data from the various instruments. Various processing possibilities

have been considered for representative instruments. The advantage of a computer in achieving data compression has been a primary concern.

III.4.1 Neutron Detector

The interface between the neutron detector and the CDS is shown in Figure 2-1. The bit requirement for the measurements alone, exclusive of timing information, amounts to 208 bits per second for the neutron detector. One way of reducing this requirement is to accumulate data over only a few hours during a day or to sample less frequently, thereby lowering the average bit rate. This may be undesirable since it sacrifices what might be valuable information. Use of logarithmic counters would immediately reduce the bit rate.

Processing Alternatives

1. The simplest processing uses zero-order thresholding of the count accumulations in the successive one-second measurement intervals. Thus, the gamma ray count for interval $k + 1$ is not transmitted unless it varies by more than, say, T units from the previous transmitted value. The disadvantage of such a scheme is that, during periods of rapidly fluctuating counts and for low telemetry rates, the thresholding method may offer almost no data compression. Further, it can require frequent formatting change and can put severe demands on the control of the buffer queue.

2. The accumulation of histograms for each measurement over some interval, large compared to the one-second counting time, offers several possibilities for data compression. From the histogram, the CDS can readily compute the mean, variance, and other statistical quantities of interest as well as giving the maximum and minimum counts and their relative frequencies and the count having highest frequency (the mode). Furthermore, these statistics can themselves be thresholded so that they are transmitted only when they change by more than some fixed percentage from the previously transmitted values.

Compression of the histogram data can also be achieved by use of sample quantiles. It has been shown that as few as four sample quantiles are sufficient to give efficiencies exceeding 90 percent and 73 percent, respectively,

for estimates of the mean and variance from a normal population. The disadvantage of using quantiles is that, for severely nonnormal populations, the estimate of the variance can be substantially in error. Use of more than four quantiles will give improved results, but obviously at the expense of reducing the compression ratio.

III.4.2 VLF Experiment

The instrument schematic is given in Figure 2-1. The six lines marked E_{x1} , ..., M_2 denote the interface with the CDS. The programmable filter has a bandwidth that is 10 percent of the center frequency. This means that, for the center frequencies of 16 Hz and higher, the sample rate of one sample per second is likely to result in aliasing. For each center frequency, the data output is represented by 39 bits, assuming 6 bits for each of the six lines and 3 bits to identify the center frequency (1 out of 7) used. If the center frequency is always stepped through the same cyclic order, then only one bit is needed to identify the start of the cycle.

Processing Alternatives

1. Simple thresholding as in Section III.4.1, (1).
2. At low telemetry rates, each measurement can be stored over a number of sample periods and the mean, variance, and other quantities periodically computed and transmitted. These quantities can be thresholded as in Section III.4.1, (2).
3. Each measurement can be sampled at rates greater than one sample per second and digital filtering used to determine the spectral content of the signals. This spectral information can be used either to control the rate at which samples are transmitted (providing the telemetry rate allows for such leeway) or to provide indication to the ground of the data activity. In the former case, the CDS would have the concomitant task of altering the data flow from the other experiments, under some priority schedule, to permit the increased data rate.
4. Orthogonal polynomials or Fourier series expansions for each measurement over a number of intervals. Such expansions result in best least

squares approximations to the data, and a sufficient number of terms can be computed so as to give an error term less than some desired amount. Thus, the possibility exists that M coefficients can be transmitted in place of N data samples, where $M < N$; and, hence, compression can be achieved. This method will be suitable for those measurements having lower precision requirements and where the ground station processing is likely to use least squares techniques.

III.4.3 Cosmic Ray Experiment

The instrument arrangement is shown in Figure 2-1. The bit requirement is 67 bits per sector. The sector sampled is advanced each spacecraft revolution. As with the neutron detector, logarithmic counters offer a way of reducing the bit rate.

Processing Alternatives

Same as for neutron detector, Section III.4.1.

III.4.4 Plasma Probe

Figure 2-1 showed the schematic arrangement of the instrument and the interface specifications. The instrument is programmed to cycle through a specific measurement pattern; and, as long as this pattern is in force, it is a simple matter to reduce the bit rate with no additional processing. Consider the scan mode data word: 3 bits are used to describe which of 8 channels is selected; 4 bits to give the sector (1 out of 15); 7 bits for the flux measurement; and 1 bit to specify the energy level (analyzer plate voltage) cycle. It is clear that the channel and sector bits are superfluous if the channels are selected in fixed cyclic order and the sectors sampled sequentially. Even at low telemetry rates when some of the sectors are omitted, the ordering is still fixed. Occasionally, it is desirable to transmit the channel and sector information as a check on proper instrument performance. In the maximum mode, channel identification is superfluous because of the fixed order of channel selection. The calibration signals need be sent only if they differ by more than some allowable tolerance from the desired values, a check easily implemented by the CDS. The CDS can also be used to perform the maximum mode

functions that are currently shown in Figure 2-1 as being done within the instrument. As with all other instruments, the easiest way to reduce the average bit rate is simply to restrict the operating period of the instrument.

Processing Alternatives

1. Simple thresholding as in Section III.4.1, (1). In the scan mode, successive samples of the same measurement (i.e., same energy level, channel, and sector) occur once every 384 spacecraft revolutions. Thresholding of measurements for adjacent sectors for the same channel and energy level and for adjacent channels for the same sector and energy level should also be considered as a means of obtaining compression. Similarly, thresholding can be used in the maximum mode.

2. Histogram compilation and analysis as in Section III.4.1, (2), for both the scan mode and maximum mode. A modification of the instrument sampling format might be considered whereby the measurement period could be reduced from the current value of 384 revolutions. This would produce more samples per unit time and would make the histogram analysis more meaningful by reducing the time needed to acquire an appropriately large number of samples. In order to do this, it would be necessary to multiplex all eight channels, in effect, sample them all in each of the 15 sectors.

III.4.5 Triaxial Fluxgate Magnetometer

The basic instrument schematic and interface is given in Figure 2-1. A realistic approach to data processing requires that the low-pass filter for each axis have a cutoff frequency sufficiently high to pass all desired information. For all three gates combined, a single data sample specifies either implicitly or explicitly 32 bits: 8 bits for each flux measurement; 2 bits for the sun aspect sensor position; 2 bits for the dynamic range position; 3 bits for the dc offset position; and 1 bit for the fluxgate physical orientation. This assumes that sampling is done four times per revolution. Here, as with the plasma probe, some savings can be made if the same sampling arrangement is always used. However, it may be desirable to sample more frequently in order to determine the spectral characteristics of the sensor outputs. This

spectral information can be used to determine an appropriate sampling rate or to govern the rate at which measured samples are transmitted. All processing, including spectral analysis and compensation for the spin effect of the spacecraft on the magnetic field measurements, can be performed digitally.

Processing Alternatives

1. Simple thresholding as in Section III.4.1, (1).
2. Histogram compilation and analysis as in Section III.4.1, (2).
3. Spectral analysis of sensor outputs. If the analysis indicates that signal frequency content is higher than can be handled by the available telemetry rate, then some points of the spectral distribution can be transmitted.

III.4.6 Stanford Radio Propagation Experiment

Figure 2-1 shows the schematic arrangement of the instrument and notes the appropriate interface points. The most substantial bit requirement occurs on line S_5 . We can assume that the CDS determines the maximum and minimum values (6-bit precision) of S_5 once per revolution from 36 or 128 samples. Line S_1 accumulates counts; and, therefore, histogram methods are applicable.

Processing Alternatives

1. Simple thresholding as in Section III.4.1, (1)--applicable to all lines.
2. Histogram analysis as in Section III.4.1, (2)--applicable to all lines, particularly S_1 .
3. Orthogonal polynomials as in Section III.4.2, (4)--applicable to all lines except S_1 .

III.4.7 Conclusions

Several points are clear as regards effective use of the CDS in handling the variety of signals generated by the instruments. In general, it seems better to employ averaging methods (e.g., computation of mean, variance,

and spectral distribution) rather than omit data samples when the data bandwidth exceeds the available telemetry rate. In this way, the CDS is being used to effect compression; and aliasing errors due to insufficient sampling rates are minimized.

It is also clear that a variety of algorithms could be stored by the CDS so that, by monitoring the data from each instrument, the appropriate algorithm can be selected. This "tailoring" of the processing to each channel is a distinct advantage possessed by a digital computer.

III.4.8 References

1. M. Fisz, Probability Theory and Mathematical Statistics, Third edition, Wiley and Sons, New York (1963).
2. S. S. Wilks, "Order Statistics," Bulletin, American Mathematical Society, vol. 54, pp. 6-50 (1948).
3. I. Eisenberger and E. C. Posner, "Systematic Statistics Used for Data Compression in Space Telemetry," Technical Report No. 32-510, Jet Propulsion Laboratory, Pasadena, California (October 1, 1963).
4. F. Mosteller, "On Some Useful 'Inefficient' Statistics," Annals of Mathematical Statistics, vol. 17, pp. 377-408 (1946).
5. I. Eisenberger, "Tests of Hypotheses and Estimation of the Correlation Coefficient Using Quantiles I," Technical Report 32-718, Jet Propulsion Laboratory, Pasadena, California (June 1, 1965).
6. I. Eisenberger, "Tests of Hypotheses and Estimation of the Correlation Coefficient Using Quantiles II," Technical Report 32-755, Jet Propulsion Laboratory, Pasadena, California (September 15, 1965).
7. R. B. Blackman, Data Smoothing and Prediction, Addison-Wesley, Reading, Massachusetts (1965).
8. H. Freeman, Discrete Time Systems, Wiley and Sons, New York, N.Y. (1965).
9. J. F. Kaiser, "Design Methods for Sampled-Data Filters," Proc. First Allerton Conference on Circuit and System Theory, Monticello, Illinois, pp. 221-236 (November 1963).
10. C. M. Rader and B. Gold, "Digital Filter Design Techniques," Technical Note 1965-63, Massachusetts Institute of Technology Lincoln Laboratory, Lexington, Massachusetts (December 23, 1965) (Astia Document No. 627146).
11. J. R. Ragazzini and G. F. Franklin, Sampled-Data Control Systems, McGraw-Hill Book Co., New York, N.Y. (1958).
12. R. M. Golden and J. F. Kaiser, "Design of Wideband Sampled-Data Filters," BSTJ, vol. 43, pp. 1533-1546 (July 1964).
13. H. Holtz and C. T. Leondes, "The Synthesis of Recursive Digital Filters," J. Assoc. Comp. Mach., vol. 13, no. 2, pp. 262-280 (April 1966).
14. J. R. B. Whittlesey, "A Rapid Method for Digital Filtering," Communications ACM, vol. 7, no. 9, pp. 552-556 (September 1964).

15. J. F. Kaiser, "Some Practical Considerations in the Realization of Linear Digital Filters," Proc. Third Allerton Conference on Circuit and System Theory, Monticello, Illinois, pp. 621-633 (October 1965).
16. B. Gold and C. M. Rader, "Effects of Quantization Noise in Digital Filters," AFIPS Conf. Proc., vol. 28, pp. 213-219, Spartan Books, Wash., D. C. (1966 Spring Joint Computer Conf., Boston, Massachusetts (April 26-28)).
17. R. B. Blackman and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, New York (1958).

APPENDIX IV
CDS INPUT/OUTPUT TIMING REQUIREMENTS

The input/output timing requirements are affected by the method employed to move data across the interface. The precision, sampling rate, and signal conditioning required for each data line were specified in Volume I.

IV.1 EXPERIMENT AND SUBSYSTEM DEFINITIONS

Each experiment and S/C subsystem is discussed below:

IV.1.1 Magnetometer

The input/output lines for the magnetometer are shown in Figure IV-1.

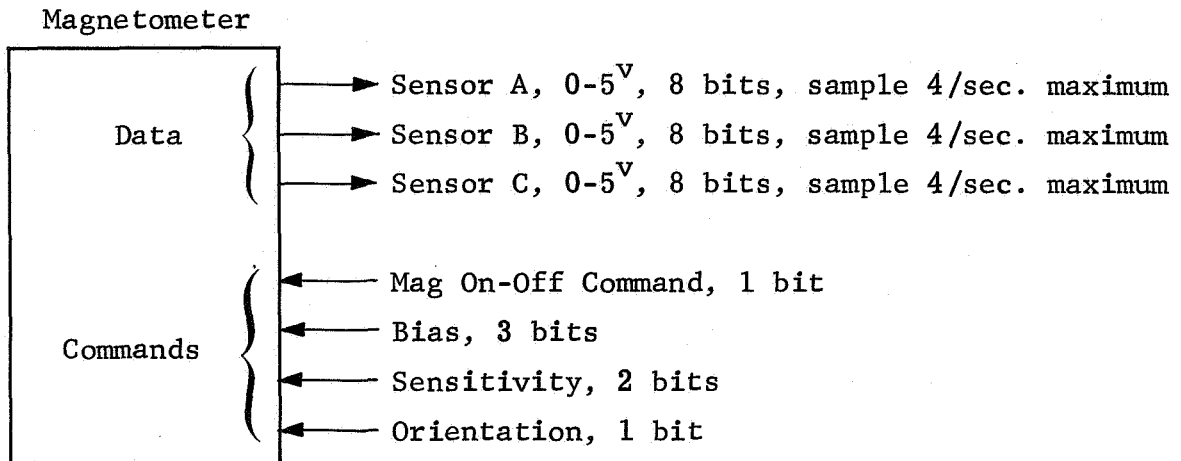


Figure IV-1. Magnetometer Input/ Output

Data lines A, B, and C are assumed to be equipped with amplitude-to-pulse-width (A/PW) converters. A typical A/PW converter is shown in Figure IV-2.

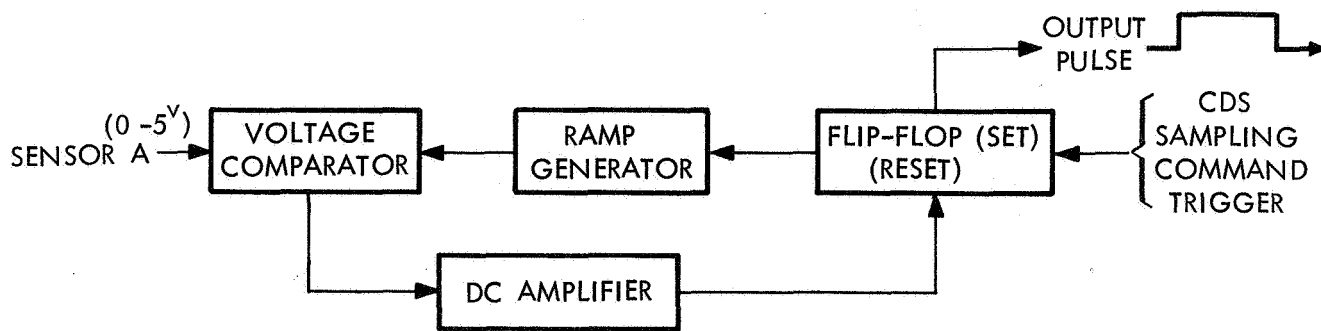


Figure IV-2. A/PW Converter

A CDS trigger sets the flip-flop shown in Figure IV-2 and ramp voltage is generated automatically. When the ramp voltage equals the voltage to be measured, the flip-flop resets automatically. The output is the pulse shown in Figure IV-2. The pulse is initiated by the set command and terminated by the reset command. The A/C pulse is coupled to the CDS. The trailing edge of the pulse triggers the gate shown in Figure IV-3 to the "off" position.

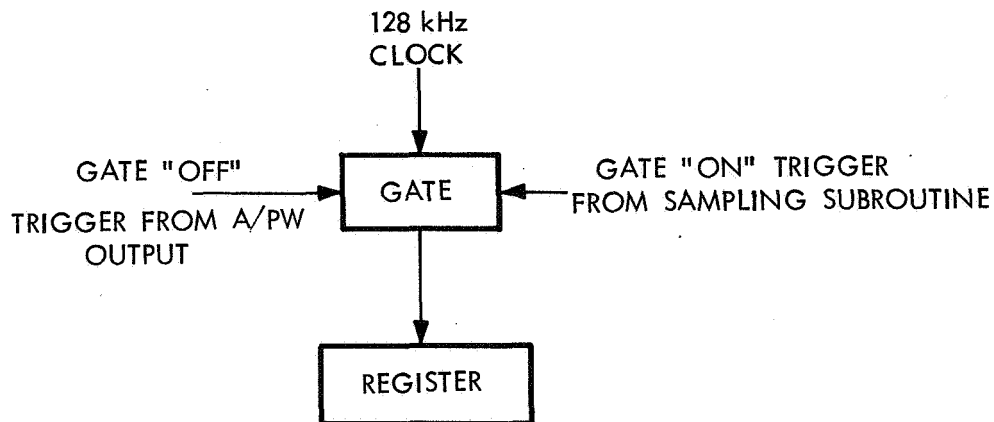


Figure IV-3. Digital Representation of Sensor A Analog Voltage

When the sampling command sets the flip-flop in the A/PW converter, it also opens the gate shown in Figure IV-3; 128 kHz clock pulses are counted. When the trailing edge of the A/PW output pulse triggers the gate "closed", the register contains a digital number representing the analog input from Sensor A. If the analog number is specified to 8 bit accuracy (9 bit resolution), the gate will be opened for 4 milliseconds maximum. The ramp generator is assumed to be calibrated to sweep 0 - 5^V in 4 milliseconds for 9-bit resolution when a 128 kHz clock is employed.

The A/PW converter minimizes potential drift problems. A/PW converters have been space qualified on "Mariner." The Mariner A/PW converters had 9-bit plus sign conversion resolution and 8-bit accuracy. Also, they can be calibrated in flight.

An A/PW converter can be implemented for bipolar analog inputs. Two converters of complementary design are required. Their inputs are connected in parallel. A logical combination of their outputs implement the bipolar converter.

Lines A, B, and C are equipped with 9-bit A/PW converters to assure 8-bit accuracy. The conversion time is 4 milliseconds per line. Since lines (A-C) are sampled four times per spacecraft revolution at high TLM bit rates, it takes approximately 48 milliseconds to service the magnetometer if the conversions are time sequenced. If lines (A-C) are converted simultaneously, the total conversion time is 16 milliseconds. The time required to execute the commands indicated in Figure IV-1 is negligible.

Commands are initiated via the command link or the "on-board" program. Commands are executed in the instruments from information supplied via the central data system. The command execute code is routed via the command switch shown in Figure IV-4. The command execute code establishes the internal magnetometer switching for the bias, sensitivity, orientation, and instrument "on" or "off".

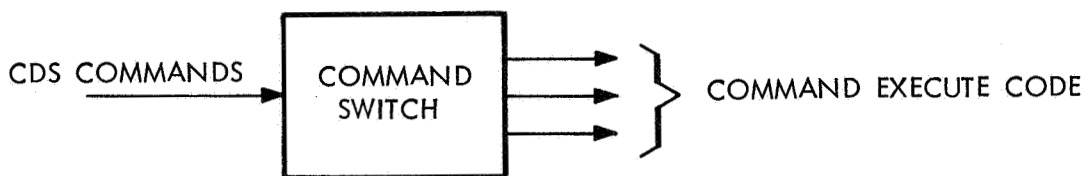


Figure IV-4. Command Flow

The requirement for sampling lines A, B, and C of the magnetometer simultaneously establishes the need for three separate multiplexers, gates and registers. The registers must combine counter and shift register characteristics. Total maximum sampling and conversion time required for the magnetometer is approximately 16 milliseconds/second, assuming simultaneous sampling of lines A, B, and C and the use of three multiplexers, each multiplexer feeding a gate and register.

IV.1.2 Cosmic Ray Telescope

Figure IV-5 shows the input/output lines for the CR telescope.

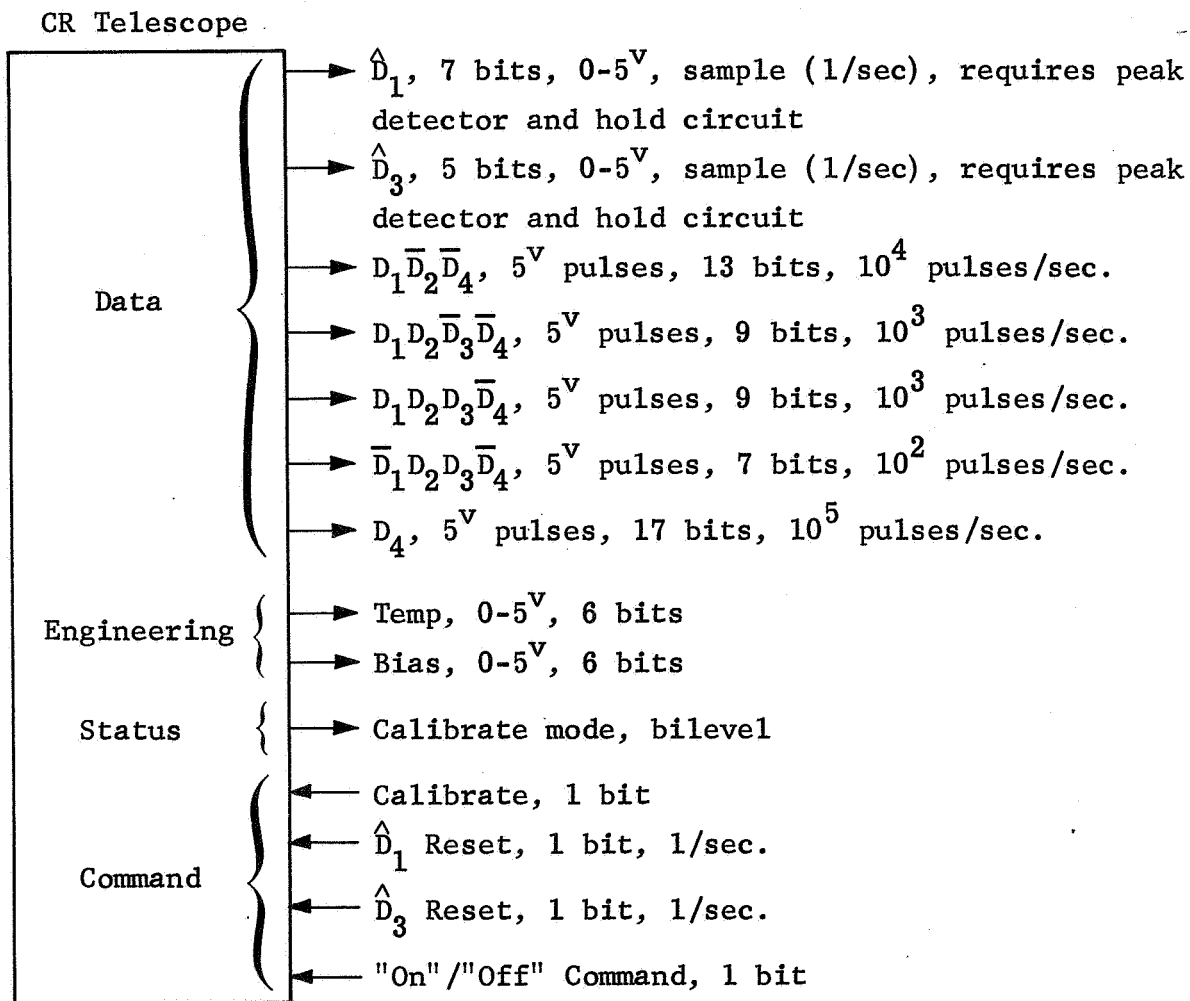


Figure IV-5. CR Telescope Input/Output

It is assumed that the peak amplitudes \hat{D}_1 and \hat{D}_3 are available for sampling. The method by which pulses are isolated is left to the instrument designer. A/PW converters are assumed to be employed for generating digital representations of \hat{D}_1 and \hat{D}_3 . At the completion of the conversion process, the peak detector outputs are zeroed by \hat{D}_1 and \hat{D}_3 reset commands. Assume that an 8-bit converter is employed with a 128-kHz clock for \hat{D}_1 . The conversion time is 2 milliseconds. A 6-bit A/PW converter is employed for \hat{D}_3 . The conversion time is 1/2 millisecond. Each line is sampled once each spacecraft revolution maximum.

The experimenters may want to retain some or all of the counters in the instrument. Consider line $\bar{D}_1 D_2 D_3 \bar{D}_4$ defined in Figure IV-5. A 7-bit counter is required. The maximum counting rate is 100 per second. One method which may be employed for moving the contents of the counter to the CDS is to drive the counter to overflow. Figure IV-6 shows a simplified functional diagram of such a scheme.

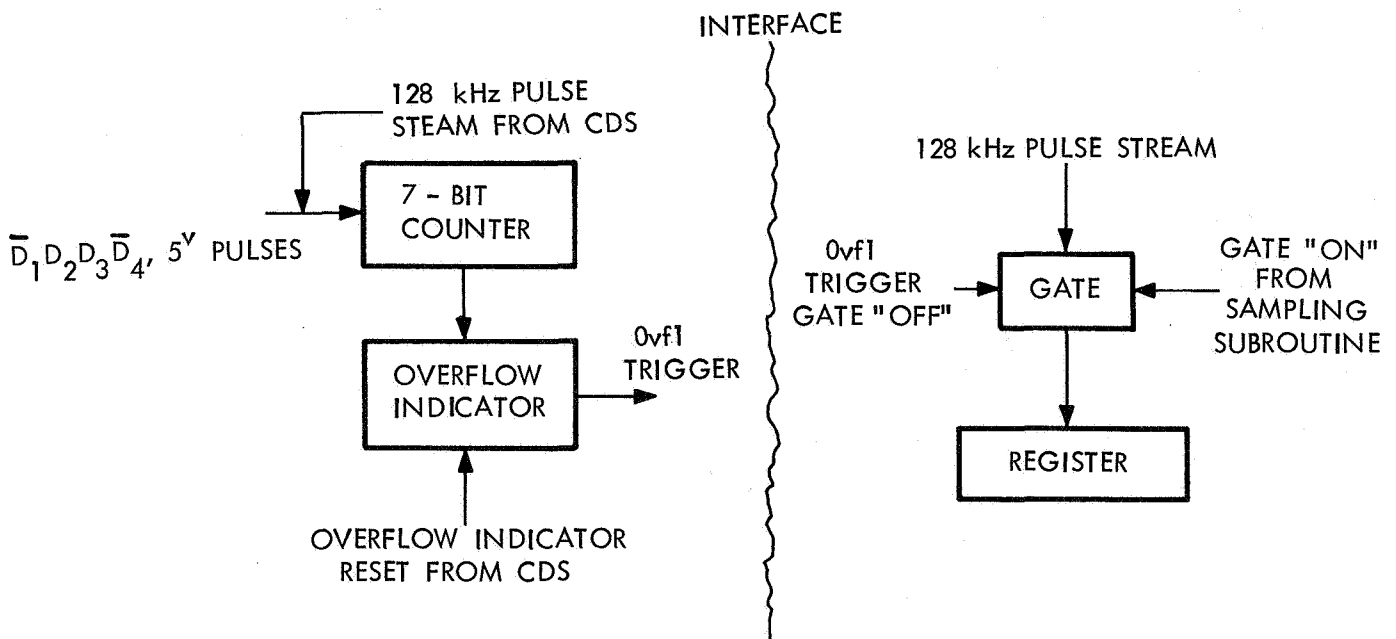


Figure IV-6. Moving Counter Contents to CDS

A sampling command simultaneously "opens" a gate and connects a 128-kHz pulse stream to the 7-bit counter in parallel with the normal input. When the 7-bit counter overflows, the gate in the CDS is "closed" by an overflow trigger. The accumulated count in the register is the complement of the count in

the instrument. The conversion time is approximately 1 millisecond for a 7-bit counter, 2 milliseconds for an 8-bit counter, etc.

A single flip-flop may replace a counter, for counting events, at a rate of 10^3 pulses/sec. The CDS samples the flip-flop 10^3 times a second and increments a CDS counter when necessary. Twenty-five memory cycles are required per service; a maximum of 25000/9 memory cycles/second are required for line $(D_1 \cdot D_2 \cdot \bar{D}_3 \cdot \bar{D}_4)$ assuming the 9 CR angular sectors are equal. Approximately 5.6 milliseconds maximum will be required to service line $(D_1 \cdot D_2 \cdot \bar{D}_3 \cdot \bar{D}_4)$ assuming a 2 microsecond memory cycle is employed by the CDS.

A similar situation exists for $D_1 D_2 D_3 \bar{D}_4$. It will require approximately 1/2 millisecond maximum to service the line designated $\bar{D}_1 D_2 D_3 \bar{D}_4$

The data lines designated $D_1 \bar{D}_2 \bar{D}_4$ and D_4 allow few conversion alternatives. If a counter is employed on the $D_4(17)$ line, the technique of Figure IV-6 leads to excessive conversion times. If the experimenter wants to transmit only selected bits, e.g., 6 out of 17, the 6-bit positions are sensed sequentially once each second. The counter is never reset in this mode of operation. The conversion time will be approximately 120 microseconds (20 microseconds per bit position) for a 2 microsecond memory cycle. If only 4 bits of counter $D_1 \bar{D}_2 \bar{D}_4(13)$ are to be read out, the conversion time is estimated to be 80 microseconds for a 2 microsecond memory cycle.

Wiring line D_4 directly into a register in the CDS is another technique. The register is directly addressable by the machine. This technique does not eliminate a counter, but does bypass the multiplexer since the register is dedicated to one function. There is some upper rate at which bits should move across the interface. We shall assume arbitrarily that this is 128 kHz.

A counter-shift register combination design simplifies the readout problem for the CDS when counters are employed within the instruments. When a 13-bit counter is read out, the contents of the counter are shifted into a CDS register by 13 pulses supplied by the CDS. The transfer time to the CDS is conservatively estimated at 100 microseconds.

Another technique connects the D_4 line to a CDS register for an interval of 100 microseconds every millisecond. The maximum effective sampling time is approximately 11 milliseconds/second. The $D_1\bar{D}_2\bar{D}_4$ line can be sampled the same way. The required sampling time will be approximately 11 millisecond/second. However, the availability of three parallel multiplexers allows the servicing of D_4 and $D_1\bar{D}_2\bar{D}_4$ to proceed simultaneously.

The sampling of engineering data or the execution of commands will, with high probability, not exceed 1 millisecond/second for the cosmic ray instrument. The conversion time required for the CR telescope is determined principally by the sampling required for the scientific data and the implementation of the A/D conversion.

In order to take a first cut at the timing requirements for the CR telescope, we shall assume that:

- 1) \hat{D}_1 and \hat{D}_3 employ peak detector and hold circuits which are reset by CDS command.
- 2) $D_1D_2\bar{D}_3\bar{D}_4$, $D_1D_2D_3\bar{D}_4$, and $\bar{D}_1D_2D_3\bar{D}_4$ employ flip-flops which are tested at a rate of 10^2 times a second for $\bar{D}_1D_2D_3\bar{D}_4$ and 10^3 times a second for $D_1D_2\bar{D}_3\bar{D}_4$ and $D_1D_2D_3\bar{D}_4$.
- 3) D_4 and $D_1\bar{D}_2\bar{D}_4$ are sampled at 100 microseconds per millisecond. When samples are scheduled 2 multiplexers are employed simultaneously.
- 4) All other operations for the CR instrument require less than 1 millisecond.

The total sampling and conversion time based on the above four assumptions are estimated to be:

$$\left\{ \begin{array}{l} 2 \text{ milliseconds for } \hat{D}_1; 1/2 \text{ millisecond for } \hat{D}_3; \\ 5.6 \text{ milliseconds for } D_1D_2\bar{D}_3\bar{D}_4; 5.6 \text{ milliseconds for } D_1D_2D_3\bar{D}_4; \\ 0.5 \text{ milliseconds for } \bar{D}_1D_2D_3\bar{D}_4; \\ 11 \text{ milliseconds for } D_4; 11 \text{ milliseconds for } D_1\bar{D}_2\bar{D}_4; \\ 1 \text{ millisecond for all other possible services.} \end{array} \right\}$$

The use of three multiplexers in parallel specifies a maximum sampling and conversion time for the CR telescope of approximately 20 milliseconds per second based on assumptions 1) through 4). Only one of 9 angular sectors is serviced per spacecraft revolution when a sample is scheduled.

IV.1.3 Plasma Probe

Figure IV-7 shows the input/output lines for the plasma probe.

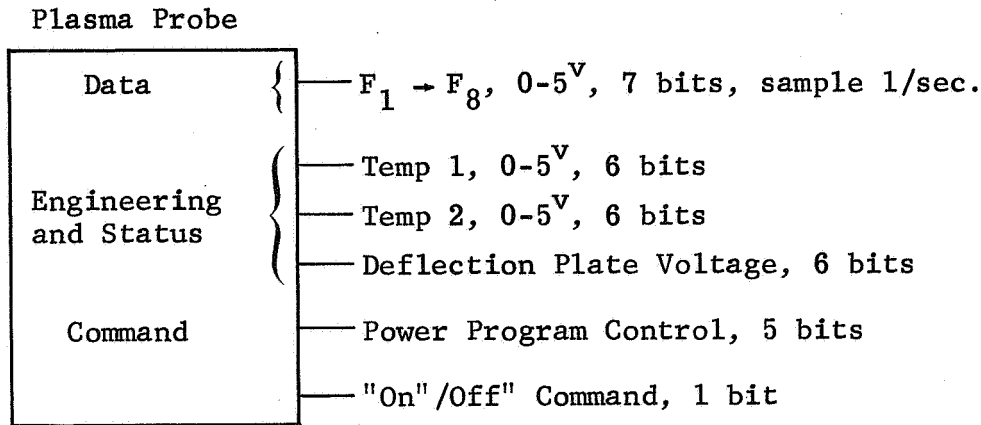


Figure IV-7. Plasma Probe Input/Output

We assume that each data line of Figure IV-7 employs an 8-bit A/PW converter. There is a valid argument for time sharing the A/PW converters. We assume the existence of CDS commands which allow time sharing. The conversion time for each data line is 2 milliseconds. The sampling requires 18 milliseconds if accomplished sequentially; 6 milliseconds if three multiplexers are employed simultaneously. The total time required for servicing the engineering, status, and command lines is less than 1 millisecond.

IV.1.4 Radio Propagation

Figure IV-8 shows the input/output lines for the radio propagation experiment.

Radio Propagation

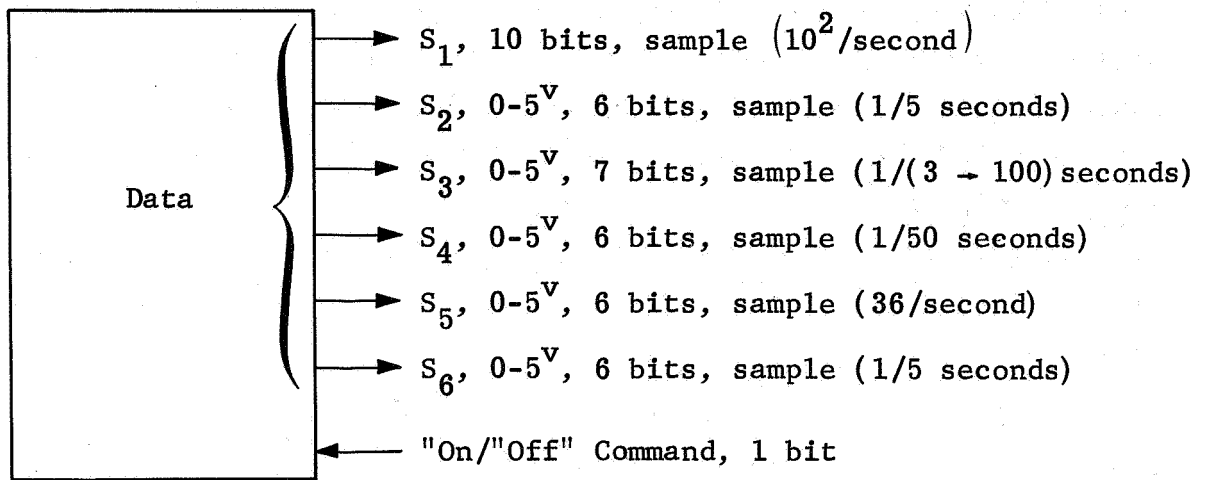


Figure IV-8. Radio Propagation Experiment Input/Output

We assume S₁ employs a flip-flop. The flip-flop is tested 100 times a second by the CDS. It requires 5 milliseconds per second to service the S₁ line and increment the relevant counter in the CDS when necessary. This assumes the multiplexer can be switched, the flip-flop tested, and the counter incremented in 25 CDS memory cycles per sample.

Lines S₂ → S₆ are assumed to be equipped with 7-bit A/PW converters. The S₅ line requires 36 milliseconds of servicing time. The average sampling and conversion time required for S₂ → S₆ excluding S₅ is 1 millisecond.

IV.1.5 Neutron Detector

Figure IV-9 promulgates the input/output lines for the neutron detector.

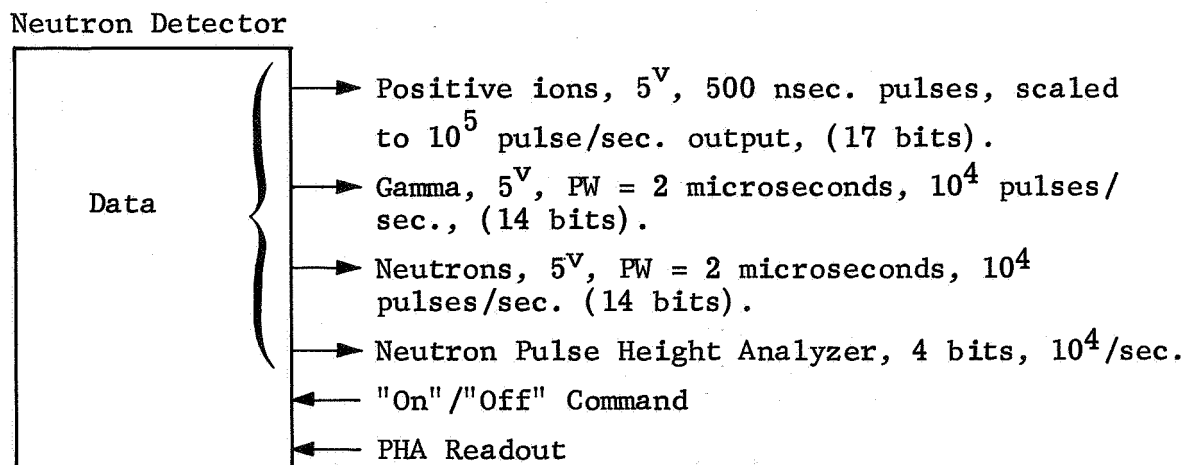


Figure IV-9. Neutron Detector Input/Output

Assume that the positive ions, gamma rays, and neutron events are counted in the CDS for 100 microseconds every millisecond, and three multiplexers are employed simultaneously. The servicing time (when sampling is scheduled) is 100 milliseconds/second. The neutron PHA line requires servicing every 100 microseconds. A compromise would be to sample the PHA line every 10 milliseconds. It is assumed that the 4-bit pulse height analyzer will be retained in the instrument. If the 4-bit register is read out 100 times a second, the conversion time will be approximately 10 milliseconds for this line.

If the experimenter is not satisfied with the suggested arrangement, the counters will remain in the instrument. The positive ion, gamma, and neutron accumulated counts will be shifted to CDS registers at a maximum rate of once/second. The counts can be moved into the CDS in approximately 100 microseconds per line. The 16 bucket counters of 10 bits each, can be moved to CDS registers in 0.1 milliseconds each. Total conversion time will be approximately 2 milliseconds if all counters are equipped with shift-register characteristics and retained in the instrument. Otherwise, the servicing requirement is greater than 100 milliseconds/second when samples are scheduled.

IV.1.6 VLF Experiment

Figure IV-10 promulgates the input/output lines for the VLF experiment.

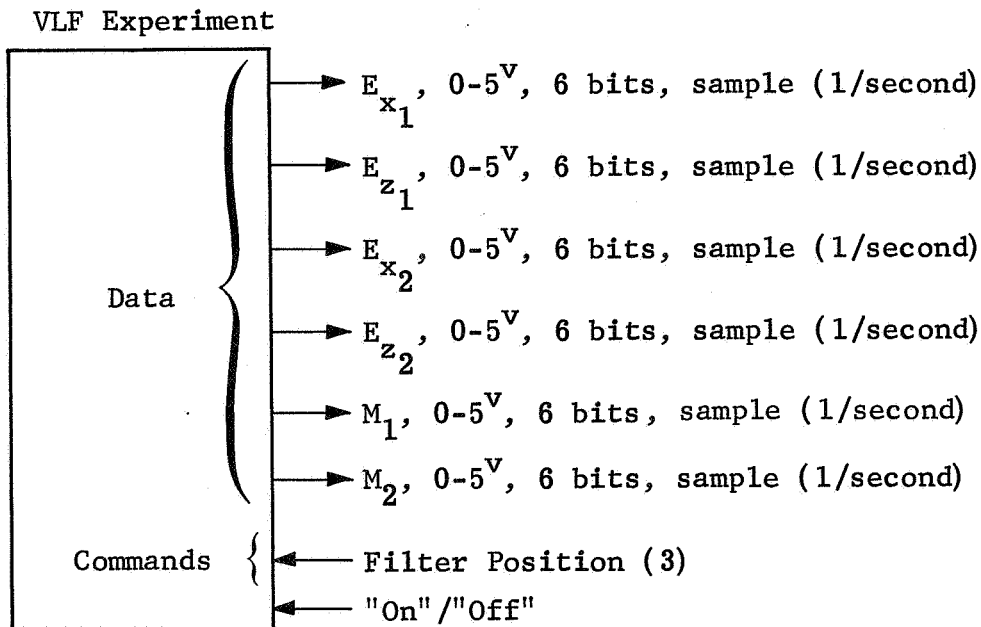


Figure IV-10. VLF Experiment Input/Output

It is assumed that each data line is equipped with an A/PW converter having an accuracy of 6 bits. The servicing time will not exceed 1 millisecond/second for each data line.

IV.1.7 Micrometeorite Detector

Figure IV-11 exhibits the input/output lines for the micrometeorite instrument.

Micrometeorite Detector

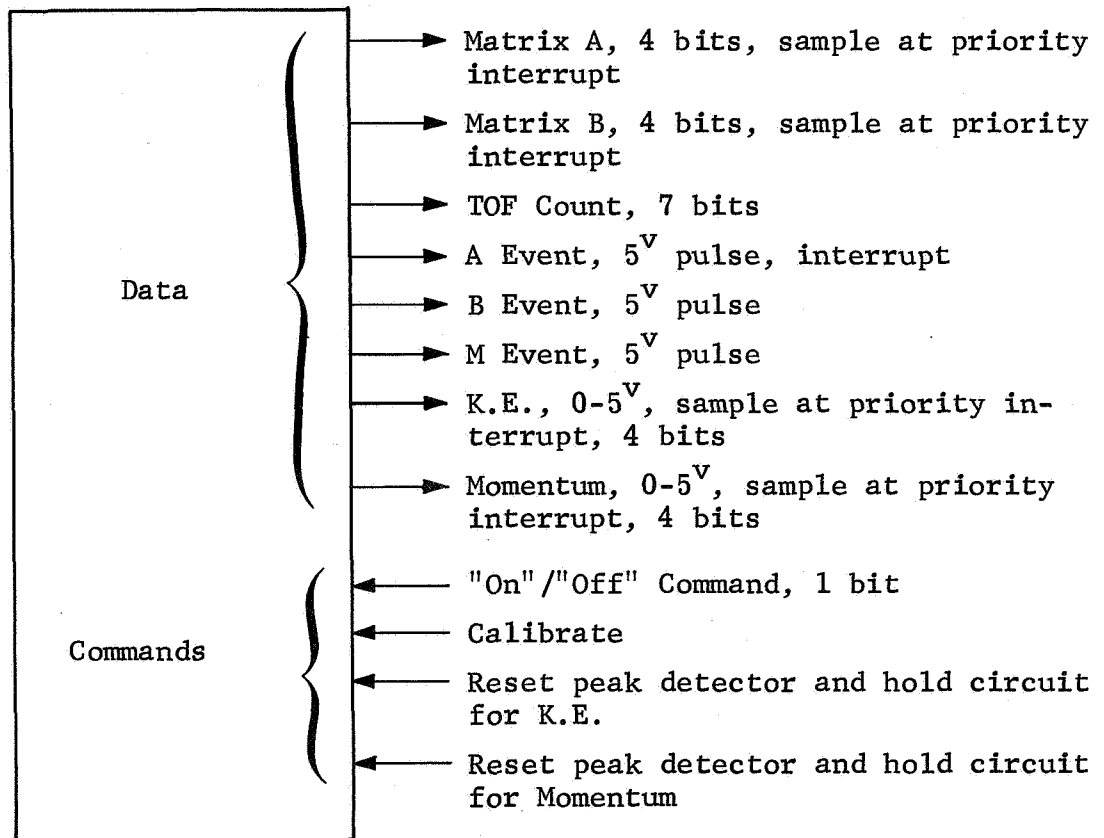


Figure IV-11. Micrometeorite Detector Input/Output

The A Event line will be wired directly to the CDS. An event will cause the CDS to branch (not interrupt) to a subroutine which services the micrometeorite experiment. It is assumed that the 4 bits describing Matrix A and Matrix B, respectively, will be shifted out of the instrument serially by the CDS. The TOF (Time of Flight) counter will be flushed out by paralleling

a CDS-generated pulse stream with the counter input for the readout. The conversion time for the TOF counter is 1 millisecond. The B Event and M Event lines each drive a flip-flop. The flip-flops are tested at interrupt plus 63.5 microseconds to validate the interrupt. These flip-flops are reset as necessary by CDS command. The K.E. (Kinetic Energy) and Momentum lines are converted to 4-bit digital numbers by the CDS and reset (grounded). The total servicing time is estimated to be 2 milliseconds per event. The anticipated frequency of events is of the order of 100 a day. The 4-bit conversion of the K.E. and M lines will be implemented with a standard counter-type converter. The servicing demands of the micrometeorite detector are negligible.

IV.1.8 Science Line Characteristics Summary

Table IV-1 summarizes the designations and line characteristics for the science instruments.

IV.1.9 Spacecraft Operations and Control Subsystems

IV.1.9.1 Orientation Subsystem

Figure IV-12 shows the input/output lines for the orientation subsystem.

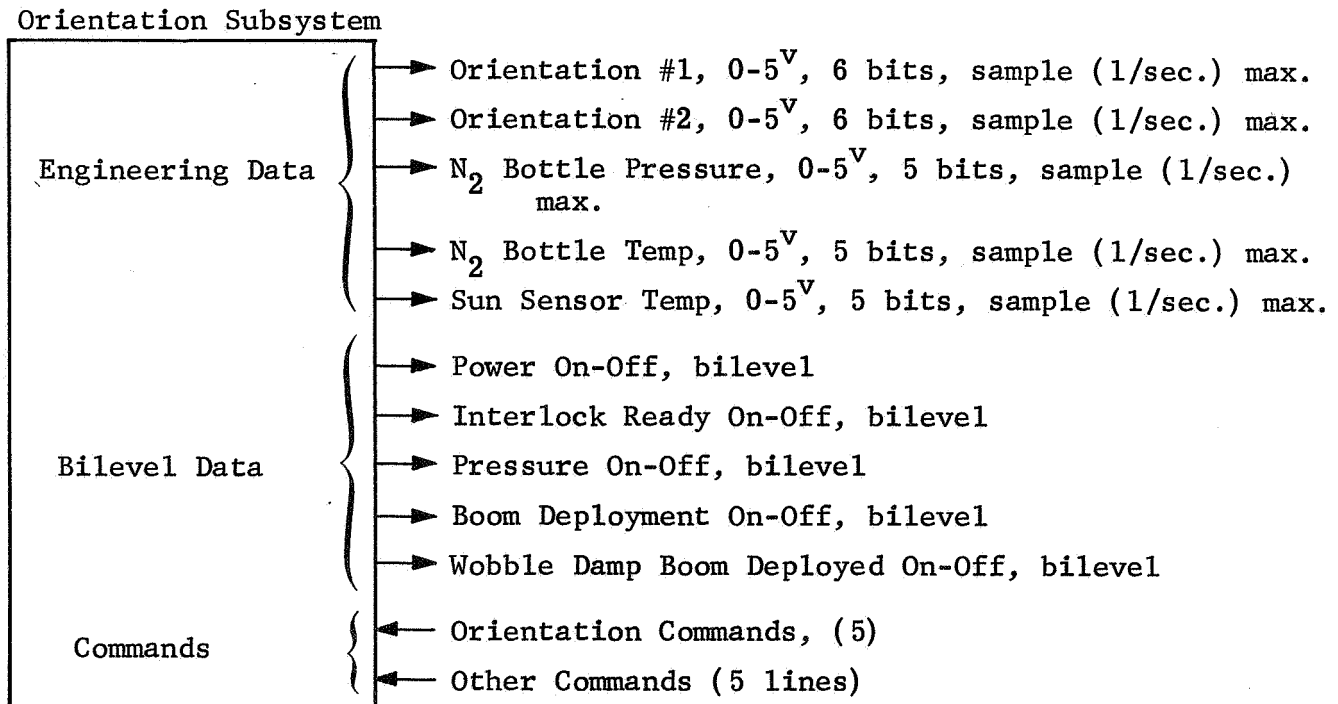


Figure IV-12. Orientation Subsystem Input/Output

TABLE IV-1

SCIENCE LINE DESIGNATIONS AND CHARACTERISTICS

Experiment	Science Line Designations and Characteristics
Magnetometer	Sensors A, B, and C; 0-5 ^V , 8 bits, sample 4/sec.
Cosmic Ray Telescope	<p>$D_1 D_2 D_3 \bar{D}_4$, $D_1 D_2 \bar{D}_3 \bar{D}_4$, 5^V pulses, 9 bits, sample 10³/sec.</p> <p>\hat{D}_1, \hat{D}_3; Peak Detector; 0-5^V; 7 and 5 bits respectively, sample 1/sec.</p> <p>$\bar{D}_1 D_2 D_3 \bar{D}_4$, 5^V pulses, 7 bits, sample 10²/sec.</p> <p>$D_1 \bar{D}_2 \bar{D}_4$, 5^V pulses, 13 bits, 10⁴ pulses/sec., sample 10³/sec. for 100 microseconds.</p> <p>D_4, 5^V pulses, 17 bits, 10⁵ pulses/sec., sample 10³/sec. for 100 microseconds.</p>
Plasma Probe	$F_1 \rightarrow F_8$, 0-5 ^V , 7 bits, sample 1/sec.
Radio Propagation	<p>S_1, 5^V pulse, 10 bits, sample 10²/sec.</p> <p>S_5, 0-5^V, 6 bits, sample 36/sec.</p> <p>S_2, S_3, S_4, S_6, 0-5^V, 6 bits, sample < 1/sec.</p>
Neutron Detector	<p>Ions, 5^V pulses, 500 nsec. duration, scaled to 10⁵ pulse/sec. output</p> <p>Gamma, 5^V pulses, 2 x 10⁻⁶ sec. duration, 10⁴ pulses/sec.</p> <p>Neutrons, 5^V pulses, 2 x 10⁻⁶ sec. duration, 4 bits, 10⁴ pulses/sec.</p> <p>Neutron Pulse-Height Analyzer, 16 channels, 4 bits, 10⁴ pulses/sec.</p>
VLF Wave-Particle	$E_{x_1}, E_{x_2}, E_{z_1}, E_{z_2}, M_1, M_2$, 0-5 ^V , 6 bits, sample 1/sec.
Micrometeorite Detector	<p>A, B, M Events, bilevel, service at interrupt</p> <p>Matrix A, B, 4 bits, service at interrupt</p> <p>Time of Flight count, 7 bits, service at interrupt</p> <p>K.E., Momentum, 0-5^V, 4 bits, service at interrupt</p>

The engineering data (top 5 lines) in Figure IV-12 are sampled at a maximum rate of approximately once a second when the engineering format is employed at high telemetry bit rates. A counter converter is employed for A/D conversion. The sampling and conversion process for the engineering data is estimated to require approximately 1 millisecond per line. This implies a maximum sampling and conversion time of 5 milliseconds per second for a high bit rate engineering format and of the order of 1/4 millisecond per second on the average when a high bit rate science format is employed assuming 5 percent of the telemetry capacity is reserved for engineering. Sample and decision time per bilevel line is expected to be 50 microseconds for a 2-microsecond, central-computer memory cycle; the total conversion time required for the orientation subsystem is approximately 6 milliseconds when a high bit rate telemetry engineering format is used and is negligible when a science format is employed.

IV.1.9.2 Power Subsystem

Figure IV-13 shows the input/output lines for the power subsystem.

Each analog data line of Figure IV-13 requires a maximum of 1 millisecond for sampling and conversion when a sample is scheduled. The bilevel command sampling and conversion time is estimated to be 50 microseconds per line, per service request. The command-line servicing times are negligible.

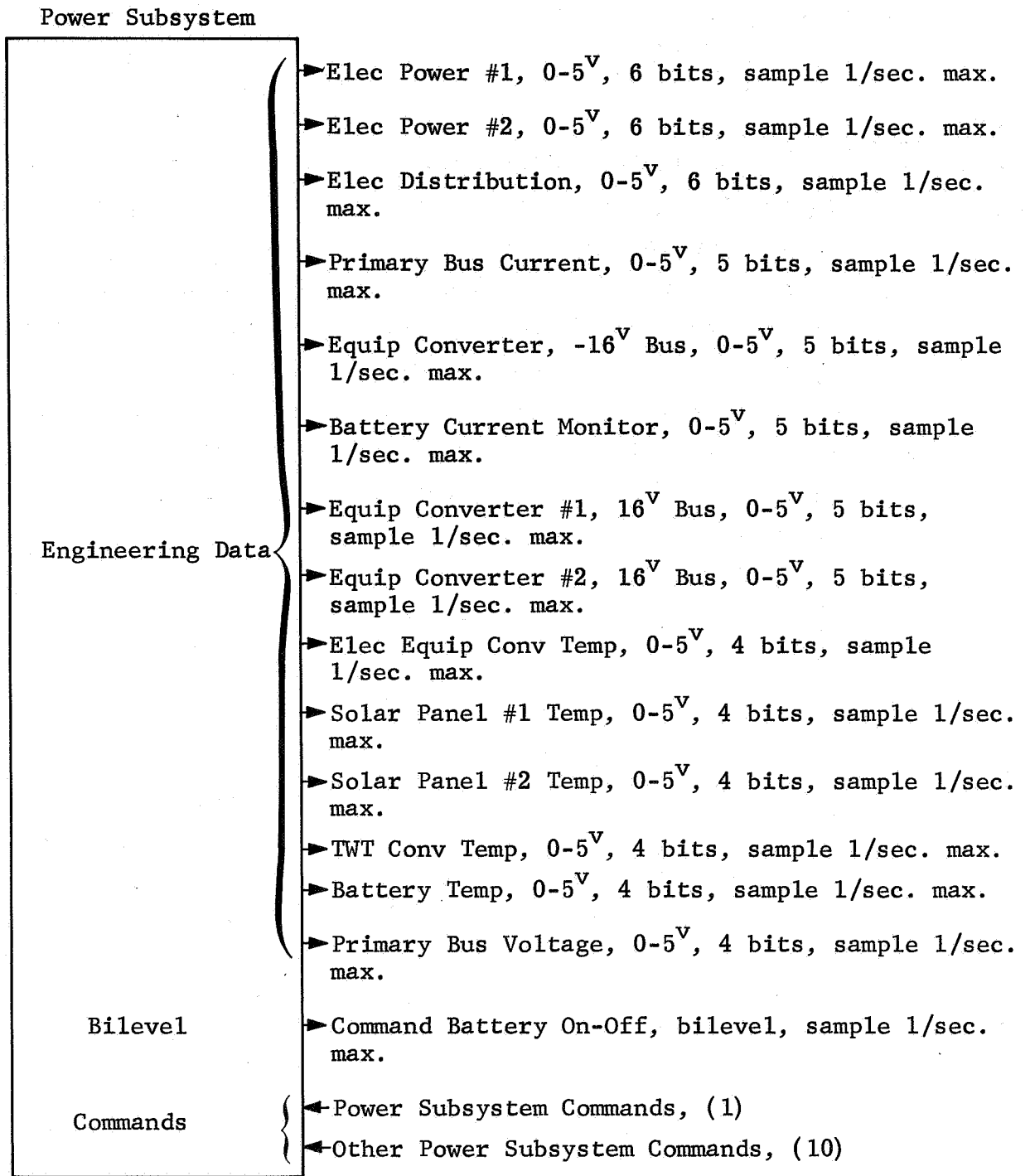


Figure IV-13. Power Subsystem Input/Output

IV.1.9.3 RF Sybsystem

Figure IV-14 shows the input/output lines for the RF subsystem.

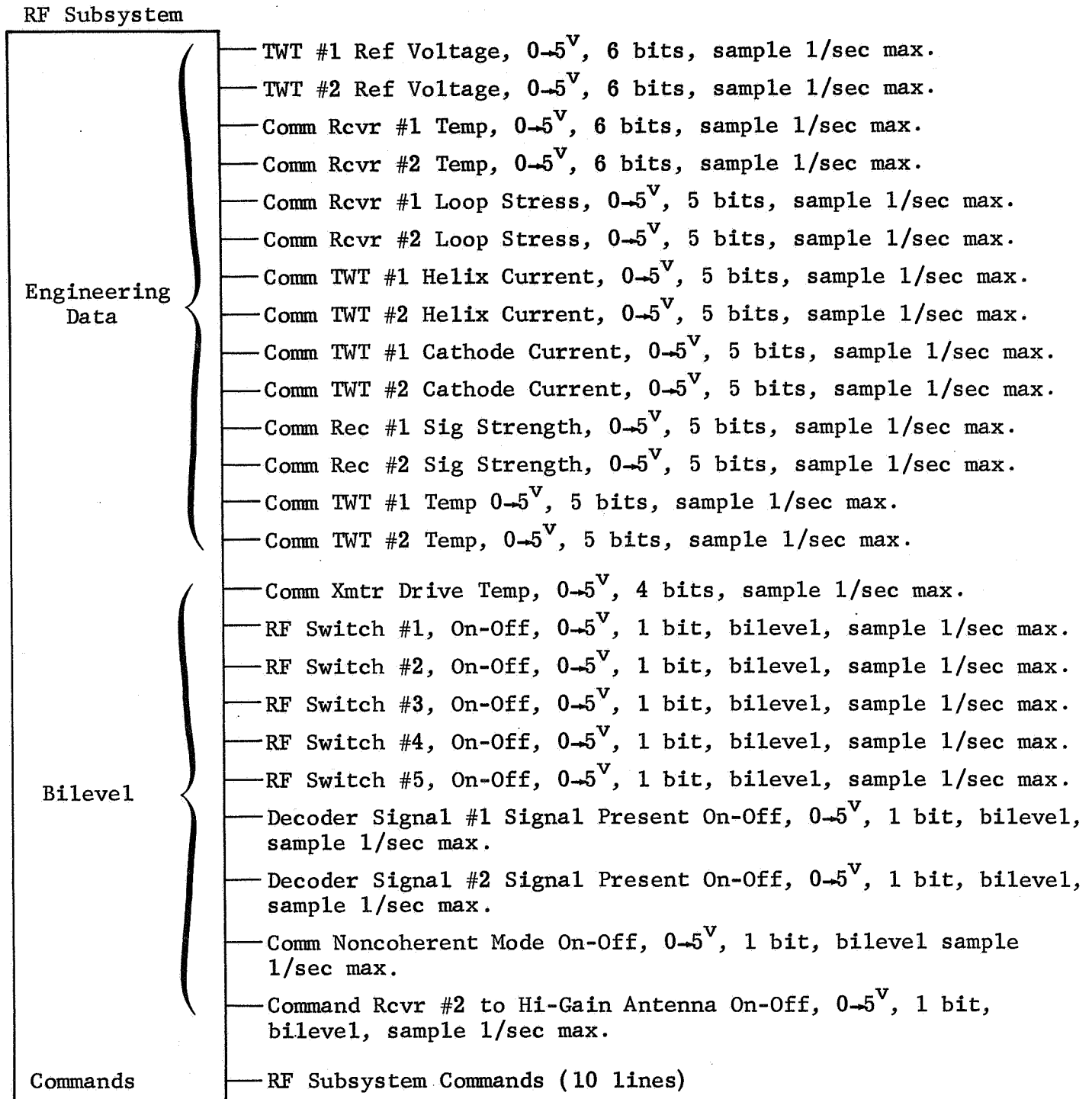


Figure IV-14. RF Subsystem Input/Output

Sampling and conversion servicing time for each analog data line of Figure IV-14 is estimated to be 1 millisecond. Less than 1 millisecond is required for servicing all bilevel lines. The command lines require roughly 500 microseconds for servicing. Total servicing time is approximately 15 milliseconds per second for engineering formats and less than 1 millisecond per second for science formats at high bit rates.

IV.1.9.4 Central Data Engineering Subsystem

Figure IV-15 shows the input/output lines for the CDS engineering subsystem.

Central Data Engineering Subsystem

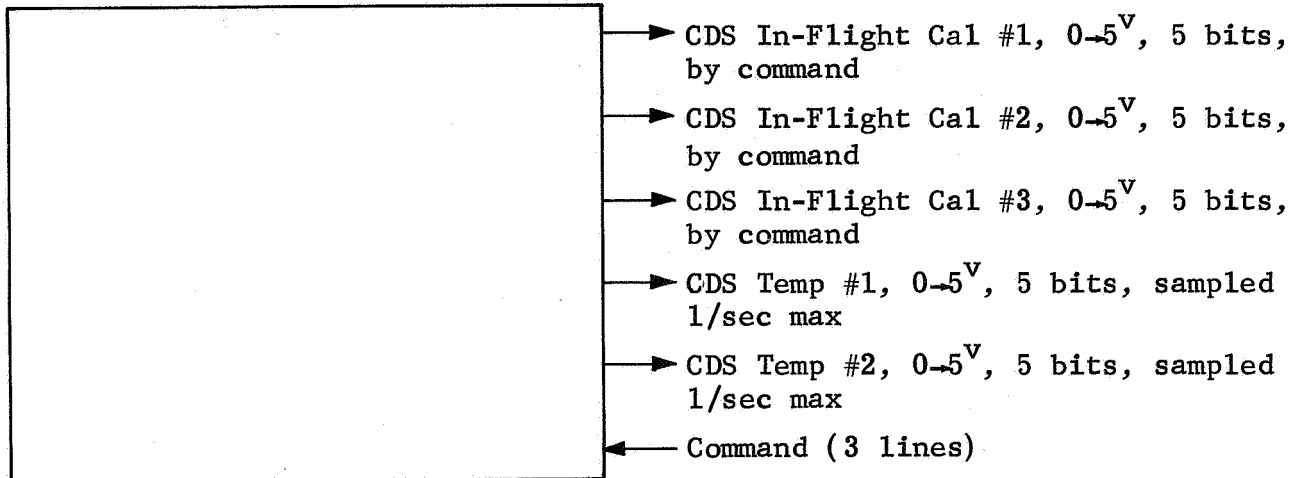


Figure IV-15. CDS Engineering Subsystem Input/Output

Sampling and conversion time will be 1 millisecond per second per line maximum. Command-line servicing times are negligible. Interface servicing time for this subsystem will be less than 5 milliseconds per second for the engineering format and less than 1 millisecond per second for a science format at high bit rates.

IV.1.9.5 S/C Support Subsystem

Figure IV-16 shows the input/output lines for the S/C support subsystem.

S/C Support Subsystem

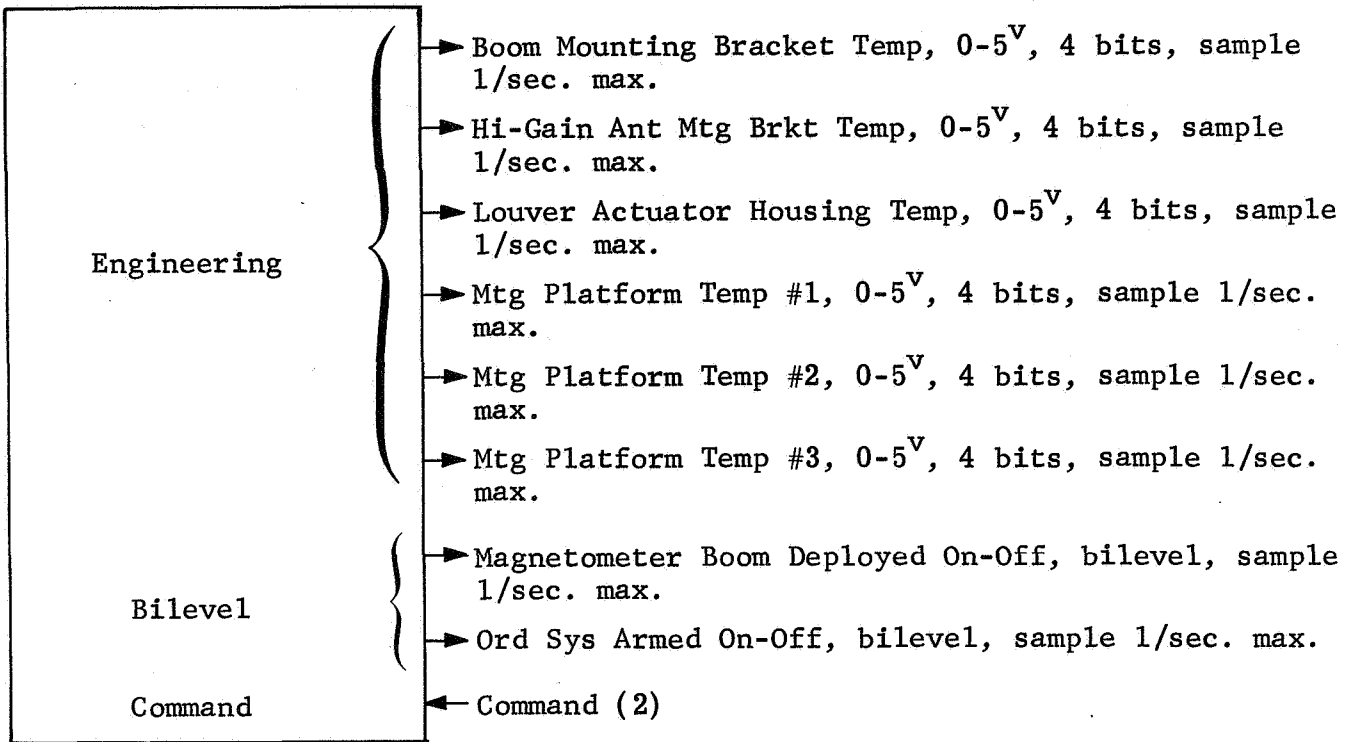


Figure IV-16. S/C Support Subsystem Input/Output

Maximum servicing time will be 1 millisecond per second per line for data and 50 microseconds per second per line for bilevel. Maximum servicing and conversion time will be less than 5 milliseconds per second for engineering and less than 1 millisecond per second for science formats.

IV.1.9.6 Other Subsystems

Figure IV-17 shows the input/output lines for the command, telemetry, sun sensor, and storage subsystems.

Other Subsystems

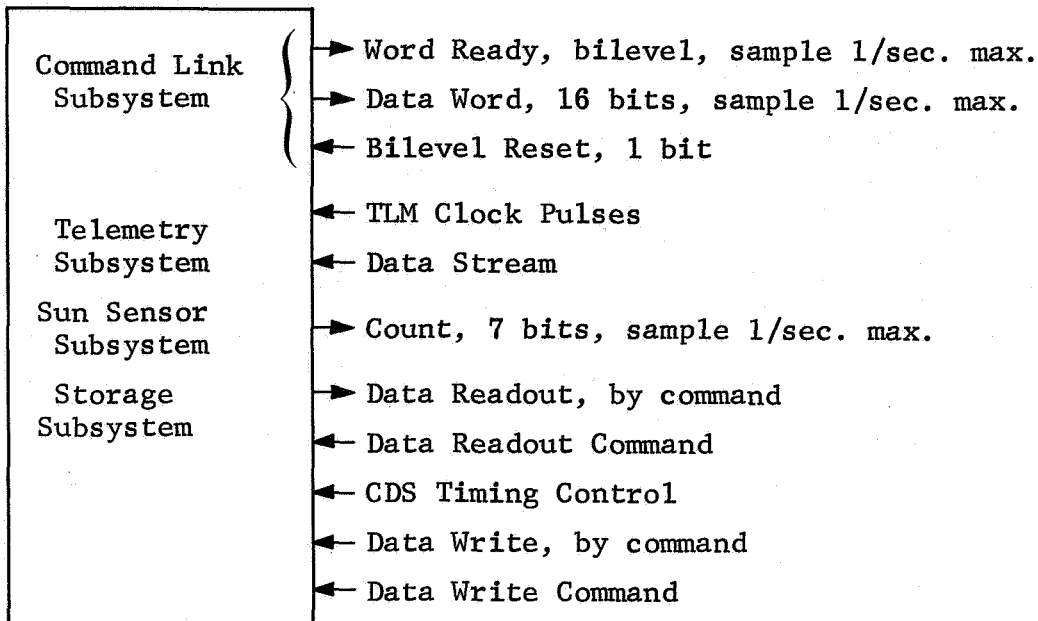


Figure IV-17. Other Subsystem Input/Output

The command link subsystem is interrogated each spacecraft revolution. It requires 50 microseconds to determine whether a command word is available. It takes approximately 125 microseconds to shift an up-command word into a CDS register when it is available.

The telemetry subsystem handles a maximum of 1024 data bits/second. The servicing time is expected to be less than 20 microseconds per data bit. Total servicing time will be less than 20 milliseconds/second.

Servicing the sun sensor requires approximately 100 microseconds each service request. Since the sun sensor will be serviced infrequently, once per 100 revolutions (e.g., the servicing time required is negligible).

The total servicing time for the lines of Figure IV-17 is estimated to be approximately 20 milliseconds/second maximum.

IV.1.10 Total Servicing Time Requirements for Experiments and Subsystems

The time required for servicing the input/output lines of the Spacecraft Operations and Control Subsystems previously described is less than 65

milliseconds per second. This total includes 20 milliseconds per second for moving bits to the TLM modulator. The estimates are based on using a counter converter employing a 128-kHz clock.

The time required for servicing the input/output lines of the experiments depends upon the assumed interface configuration, the sampling format, and the telemetry rate. The estimates promulgated for each experiment are based on reasonable estimates of the interface configuration, and the maximum rate at which each instrument may be sampled. This leads toward overestimating the time required for servicing the input/output for each experiment since the instruments are not sampled at the maximum rate when a telemetry channel capacity is limited. A first-order estimate of the servicing time required for the input/output lines for science formats is promulgated in Table IV-2.

TABLE IV-2
SCIENTIFIC INSTRUMENTS INPUT/OUTPUT (MAXIMUM TIME)
SERVICING REQUIREMENTS
(milliseconds/second)

<u>Instrument</u>	<u>1 Multiplexer</u>	<u>2 Multiplexers</u>	<u>3 Multiplexers</u>
MAG	32	32	16
CR	30	25	20
PP	12	8	6
RP	42	36	36
ND*	1	1	1
VLF	4	3	2
TLM**	20	20	20
TOTALS	141***	125	101

* Assumes neutron detector employs own counters.

** Assumes telemetry rate of 1024 bits/second and 20 microseconds to move each bit to the TLM modulator.

*** One-third of science data is unavailable assuming the multiplexer wiring provides for 50 percent redundancy.

IV.2 MULTIPLEXER CONFIGURATION

Three multiplexers are suggested for servicing data lines simultaneously and providing redundancy against failure. All science lines can be serviced with one multiplexer failure, and two-thirds of the science lines with two multiplexer failures. There are 57 science lines. Approximately 51 engineering and 21 status lines will be employed. The estimated number of command lines is 125. Table IV-3 shows the interface line assignment.

TABLE IV-3
INTERFACE LINES

Description	Science	Command	Engineering	Status
Magnetometer	3	10		
Cosmic Ray Telescope	7	9	2	1
Plasma Probe	8	14	2	1
Radio Propagation	6	7		
Neutron Detector	4/19	21		
VLF	6	10		
Micrometeorite	8	12		
RF Subsystem		10	14	10
Orientation Subsystem		10	5	5
Power Subsystem		11	14	1
CDS Engineering Subsystem		3	5	
S/C Support Subsystem		2	6	2
Other Subsystems		6	3	1
TOTALS	42/57	125	51	21

Triple redundancy could be provided for each science line. A practical compromise is to provide simultaneous sampling for 3 data lines for each instrument and full service with an altered sampling sequence if one multiplexer fails.

If two multiplexers fail, one-third of the science data is lost. Table IV-4 shows the line assignments for each multiplexer.

Seventy-two engineering and status lines remain to be assigned to the multiplexers. We assume that all engineering words associated with the instruments will be provided with 50 percent redundancy. There are 6 lines in this category. Assume 6 of 20 status lines are to be provided with 50 percent redundancy. The remaining 61 data lines are assigned to the multiplexers with no redundancy. Each multiplexer has 64 data lines assigned.

Each multiplexer is a solid state switch. Each multiplexer line has a machine address. An input subroutine is tailored to that address.

Figure IV-18 shows the input/output interface for multiplexer #1. The selection of converters for each data line is arbitrary but reasonable. Lines 1 - 19 of Figure IV-18 use A/PW converters of various resolutions. It is assumed that the accuracy is one bit less than the resolution, hence the assignment of 9-bit converters to 8-bit data lines, etc. Data lines 1 - 19 each require a command line for triggering a flip-flop which initiates A/PW conversion. Conversion times for lines 1 - 19 range from one-quarter to 4 milliseconds per line. The line switching time is 20 microseconds; this is the time required to multiplex a given data line to a converter. The 20-microsecond switching time includes switching transient settling time.

Lines 20, 21, and 23 of Figure IV-18 generate bilevel signals. The CDS samples these lines at a rate sufficient to count events. Lines 20 and 21 are sampled every millisecond during the time a particular angular sector is scheduled for sampling. Line 23 is sampled every 10 milliseconds. The idea is to count the number of bilevel changes during an observation period. Line 22 generates 5^V pulses at maximum rate of 10^5 /sec. Line 22 has a coaxial connector to minimize interference with other lines. The output shown in Figure IV-18 is designated X. This line drives a counter via a CDS controlled gate. The D_4 line will be duty cycled; the gate will be opened for a specified time, counts accumulated, and the gate shut-off. A suggested duty cycle is 100 microseconds every millisecond during an observation period.

TABLE IV-4

MULTIPLEXER LINE ASSIGNMENTS

Description	Multiplexer #1	Multiplexer #2	Multiplexer #3
Magnetometer	Sensor A		Sensor A
"	Sensor B	Sensor B	
"		Sensor C	Sensor C
Cosmic Ray Telescope	\hat{D}_1	\hat{D}_1	
"	\hat{D}_3		\hat{D}_3
"		$D_1 \bar{D}_2 \bar{D}_4$	$D_1 \bar{D}_2 \bar{D}_4$
"	$\bar{D}_1 D_2 D_3 \bar{D}_4$	$\bar{D}_1 D_2 D_3 \bar{D}_4$	
"	$D_1 D_2 D_3 \bar{D}_4$		$D_1 D_2 D_3 \bar{D}_4$
"		$D_1 D_2 \bar{D}_3 \bar{D}_4$	$D_1 D_2 \bar{D}_3 \bar{D}_4$
"	D_4	D_4	
Plasma Probe	F_1	F_1	
"	F_2		F_2
"		F_3	F_3
"	F_4	F_4	
"	F_5		F_5
"		F_6	F_6
"	F_7		F_7
"	F_8	F_8	
Radio Propagation	S_1		S_1
"		S_2	S_2
"	S_3	S_3	
"		S_4	S_4
"	S_5	S_5	
"	S_6		S_6

TABLE IV-4 (cont)

Description	Multiplexer #1	Multiplexer #2	Multiplexer #3
Neutron Detector	IONS		IONS
"	Gamma	Gamma	
"		Neutron	Neutron
"	PHA 1	PHA 1	
"	PHA 2		PHA 2
"		PHA 3	PHA 3
"	PHA 4	PHA 4	
"		PHA 5	PHA 5
"	PHA 6		PHA 6
"		PHA 7	PHA 7
"	PHA 8	PHA 8	
"		PHA (9)	PHA 9
"	PHA 10		PHA 10
"	PHA 11		PHA 11
"		PHA 12	PHA 12
"	PHA 13	PHA 13	
"	PHA 14	PHA 14	
"	PHA 15		PHA 15
"		PHA 16	PHA 16
VLF Experiment	E_{x_1}	E_{x_1}	
"	E_{x_2}	E_{x_2}	
"	E_{z_1}		E_{z_1}
"		E_{z_2}	E_{z_2}
"	M_1		M_1
"		M_2	M_2

TABLE IV-4 (cont)

Description	Multiplexer #1	Multiplexer #2	Multiplexer #3
Micrometeorite Detector	Matrix A	Matrix A	
" "		Matrix B	Matrix B
" "	TOF		TOF
" "	B Event	B Event	
" "		M Event	M Event
" "	KE		KE
" "	Momentum	Momentum	
TOTAL NO. OF LINES	38	38	36

It has been assumed that the neutron detector employs its own counters, and the counters are also shift registers. The shift takes place at 128 kHz. Conversion time on each line of the neutron detector will be 100 microseconds.

Lines 39 - 55 are serviced infrequently when science formats are employed. It is assumed, that a 6-bit counter converter is used to service these lines. The counter clock operates at 128 kHz. The conversion time will be approximately 1 millisecond per data line. The engineering bilevel lines (56-64) require approximately 50 microseconds/line for servicing.

Figure IV-18 exhibits the various types of inputs which must be handled. There are problems involved with isolation and reliability which require detailed consideration. The experimenters will ultimately specify an acceptable interface. Figure IV-18 is a tentative approach; other conversion possibilities have been considered.

Associated with the multiplexers are command switches which establish a path to a given instrument or subsystem, and through which flow pulses which trigger flip-flops or "flush-out" registers.

The command switches are bilateral. (Commands may be transmitted to the instruments and status can be determined over the command lines.) The number of command lines and the associated instrument or subsystem are exhibited in Table IV-5.

- 1 Mag Sensor A(8) A/PW(9) A/C Couple
- 2 Mag Sensor B(8) A/PW(9) A/C Couple
- 3 PP F1(7) A/PW(8) A/C Couple
- 4 PP F2(7) A/PW(8) A/C Couple
- 5 PP F4(7) A/PW(8) A/C Couple
- 6 PP F5(7) A/PW(8) A/C Couple
- 7 PP F7(7) A/PW(8) A/C Couple
- 8 PP F8(7) A/PW(8) A/C Couple
- 9 CR Telescope D1(7) A/PW(8) A/C Couple
- 10 CR Telescope D2(5) A/PW(6) A/C Couple
- 11 RP S3(6) A/PW(7) A/C Couple
- 12 RP S5(6) A/PW(7) A/C Couple
- 13 RP S6(6) A/PW(7) A/C Couple
- 14 Micro KE(4) A/PW(5) A/C Couple
- 15 Micro M(4) A/PW(5) A/C Couple
- 16 VLF Ex1 A/PW(6) A/C Couple
- 17 VLF Ex2 A/PW(6) A/C Couple
- 18 VLF Ez1 A/PW(6) A/C Couple
- 19 VLF M1 A/PW(6) A/C Couple
- 20 CR Telescope D1D2D3D4(9) Bilevel Tw Pr
- 21 CR Telescope D1D2D3D4(9) Bilevel Tw Pr
- 22 CR Telescope D4(17) 5V Pulse 10⁵/sec Coax
- 23 RP S1(10) Bilevel 10²/sec Tw Pr
- 24 ND IONS(17) Digital Tw Pr
- 25 ND GAMMA(14) Digital Tw Pr
- 26 ND PHA 1(10) Digital Tw Pr
- 27 ND PHA 2(10) Digital Tw Pr
- 28 ND PHA 4(10) Digital Tw Pr
- 29 ND PHA 6(10) Digital Tw Pr
- 30 ND PHA 8(10) Digital Tw Pr
- 31 ND PHA 10(10) Digital Tw Pr
- 32 ND PHA 11(10) Digital Tw Pr
- 33 ND PHA 13(10) Digital Tw Pr
- 34 ND PHA 14(10) Digital Tw Pr
- 35 ND PHA 15(10) Digital Tw Pr
- 36 Micro MATRIX A(4) Digital Tw Pr
- 37 Micro TOF(7) Digital Tw Pr
- 38 Micro B EVENT(1) Digital Tw Pr
- 39 CR Temperature(6) Analog Tw Pr
- 40 CR Bias(6) Analog Tw Pr
- 41 CR Calibrate Bilevel Analog Tw Pr
- 42 Engineering Data(6) Analog Tw Pr
- 43 Engineering Data(6) Analog Tw Pr
- 44 Engineering Data(6) Analog Tw Pr
- 45 Engineering Data(6) Analog Tw Pr
- 46 Engineering Data(6) Analog Tw Pr
- 47 Engineering Data(6) Analog Tw Pr
- 48 Engineering Data(6) Analog Tw Pr
- 49 Engineering Data(5) Analog Tw Pr
- 50 Engineering Data(5) Analog Tw Pr
- 51 Engineering Data(5) Analog Tw Pr
- 52 Engineering Data(5) Analog Tw Pr
- 53 Engineering Data(5) Analog Tw Pr
- 54 Engineering Data(5) Analog Tw Pr
- 55 Engineering Data(5) Analog Tw Pr
- 56 Engineering Data(1) Bilevel Tw Pr
- 57 Engineering Data(1) Bilevel Tw Pr
- 58 Engineering Data(1) Bilevel Tw Pr
- 59 Engineering Data(1) Bilevel Tw Pr
- 60 Engineering Data(1) Bilevel Tw Pr
- 61 Engineering Data(1) Bilevel Tw Pr
- 62 Engineering Data(1) Bilevel Tw Pr
- 63 Engineering Data(1) Bilevel Tw Pr
- 64 Engineering Data(1) Bilevel Tw Pr

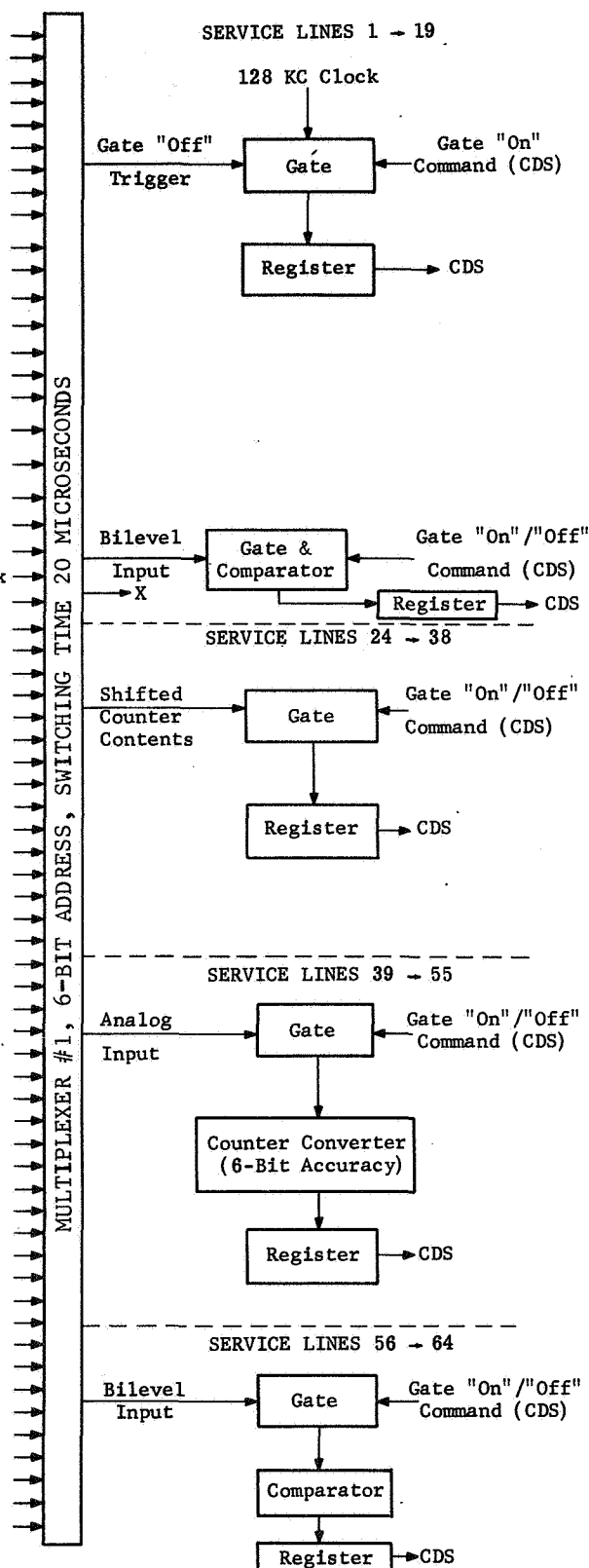


Figure IV-18. Multiplexer #1 Input/Output Interface

TABLE IV-5

COMMAND LINES AND PERTINENT INSTRUMENT/SYSTEM

Description	No. of Command Lines	Type of Commands
Magnetometer	7	on/off (1), orientation (1), bias (3), sensitivity (2)
Cosmic Ray Telescope	4	calibrate (1), \hat{D}_1 reset (1), \hat{D}_3 reset (1), on-off (1)
Plasma Probe	6	deflection plate voltage (5), on-off (1)
Radio Propagation	1	on-off (1)
Neutron Detector	2	on-off (1), readout (1)
VLF	4	on-off (1), filter center frequency (3)
Micrometeorite Detector	4	on-off (1), calibrate (1), \hat{P}_M reset, \hat{P}_{ke} reset
RF Subsystem	10	switch position
Orientation Subsystem	10	"
Power Subsystem	11	"
CDS Engineering Subsystem	3	"
Spacecraft Support Subsystem	2	"
Other Subsystems	6	"
Total	70	

In addition to the command lines listed in Table IV-5, a trigger must be provided for each A/PW converter listed in Figure IV-18. Nineteen command lines are required for servicing these A/PW converters. There should be a command line for setting or resetting the flip-flops on lines 20, 21, and 23 of Figure IV-18. Lines 24 through 38 should have command lines for shifting

the counter contents through the multiplexers. Thus a total of 37 additional command lines are required for implementing conversions, shifting out registers, or setting and resetting switches for multiplexer #1.

IV.3 COMMAND SWITCH CONFIGURATION

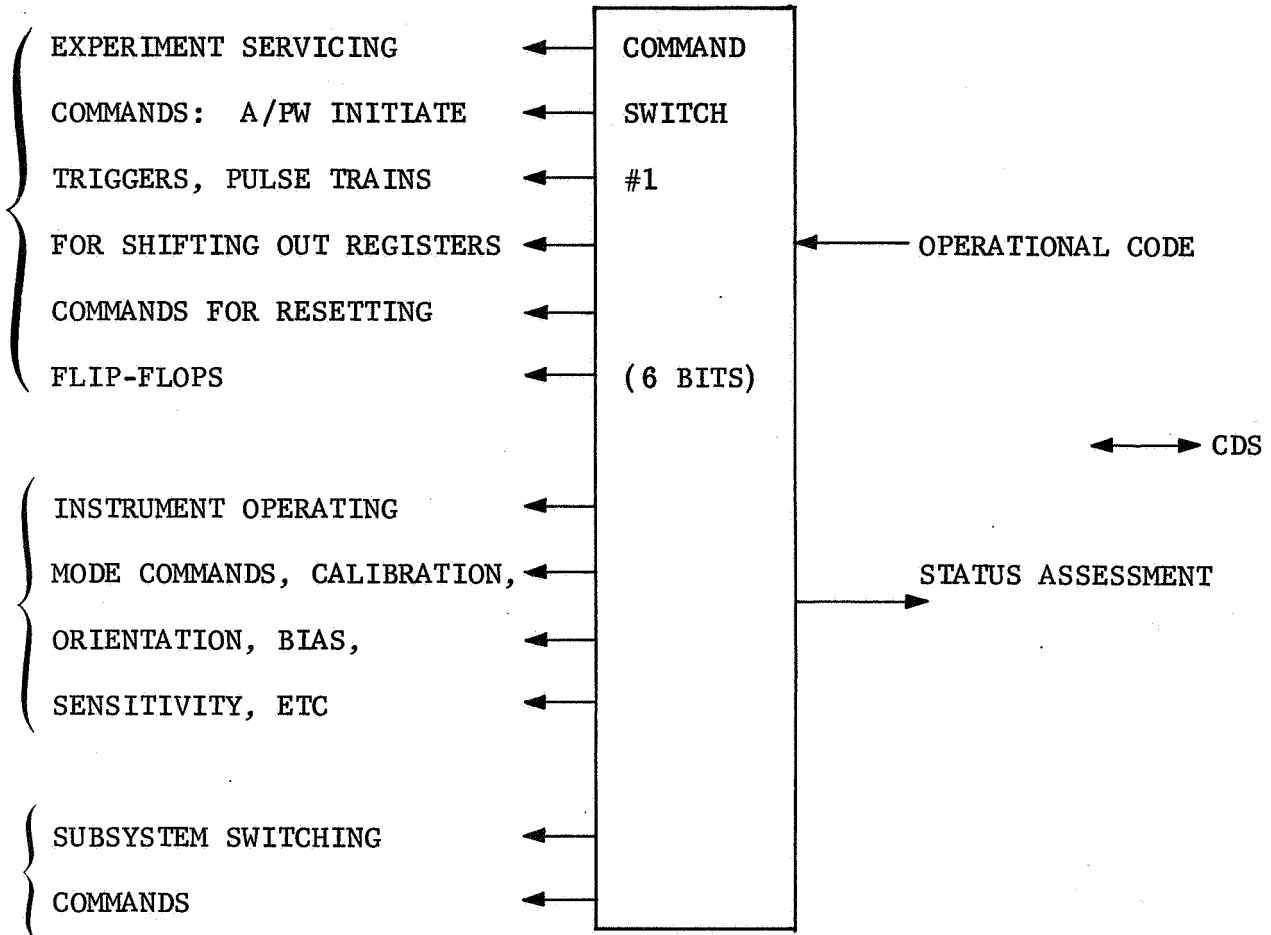


Figure IV-19. Command Switch #1

Figure IV-19 portrays the function of a command switch. Each command line has an address. Once an address has been established (the switch set), the operational code triggers a pulse train to initiate A/D conversion and move data across the multiplexer interface. The command switch is bilateral so that the status of flip-flops can be assessed. The address field of command switch #1 is specified by 6 bits; bit rates across the interface will not

exceed 128 kHz. The command switch is an intrinsic part of the computer. The registers in the experiments and subsystems are CDS registers which are physically displaced from the central computer.

Figure IV-20 shows the functional arrangement of the interface.

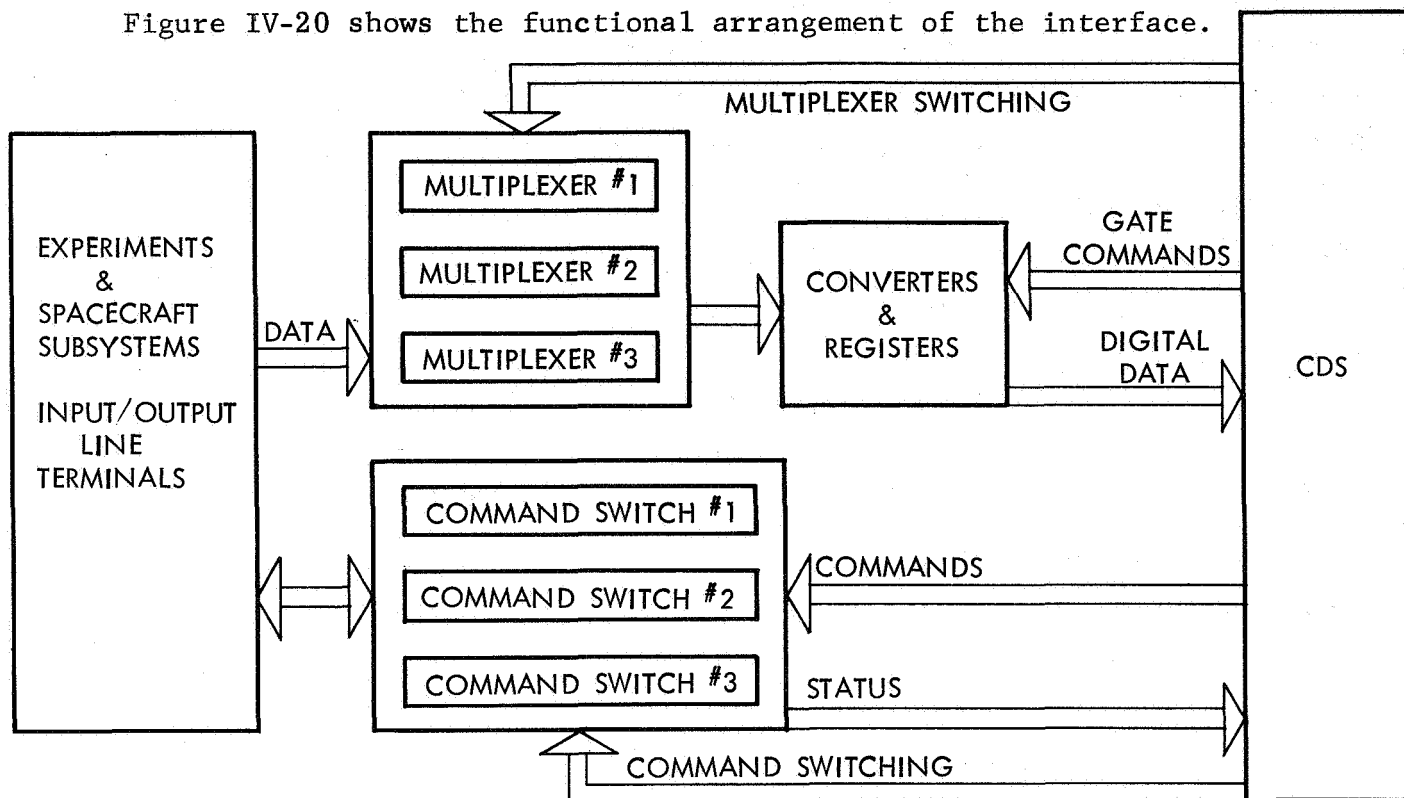


Figure IV-20. Overall Interface

Assume MAG Sensor A, line 1 of Multiplexer #1 is scheduled to be sampled. The sampling instant is specified by a software program which continually interrogates a S/C clock which is synchronized to the rotation period of the spacecraft. The S/C clock counts modulo 128 per spacecraft rotation. At the sampling instant, line 1 of multiplexer #1 is connected to the converters and registers block of Figure IV-20, and the appropriate command line is connected, via a command switch, to the set line of the MAG Sensor A, A/PW converter. Immediately, a pulse triggers the A/PW flip-flop to set and "opens" an interface gate; 128 kHz clock pulses are counted. At the instant the data input and sweep voltage of the A/PW converter are equal, a trigger is generated which turns off the interface gate. The count accumulated is the digital representation of the input analog voltage. The software program specifies the next sample to be taken. If the content of a counter is to be moved to the

CDS, the multiplexer and command switching is executed first. The command line conveys a sequence of pulses to the appropriate register. At the time that the pulse train is being initiated, a gate is turned on in the converters and registers block. The pulse train flushes the relevant register contents from an experimental counter through the gate to an input register. At the instant the pulse train ends, the gate is turned off. When the conversion has been completed, the relevant register contents are moved into CDS storage.

The multiplexer and command switches operate as a team moving data toward the CDS. In the interest of interface simplicity, it seems reasonable to provide each command bit with a line. For example, bias (3) is assigned three lines, eliminating the need for a switching matrix internal to the instrument.

IV.4 MICROSTRUCTURE OF SCHEDULING AND SAMPLING SUBROUTINE

The sampling algorithm developed for the system simulation program specifies samples at specific spacecraft revolutions. The sample times during a spacecraft revolution are specified by the data-generation subroutine. A more detailed sampling algorithm is required to specify the detailed timing requirements for a computer flight model.

Samples are taken at spacecraft revolution increments defined by:

$$\Delta S = \frac{(\text{Bits/Line})(\text{Lines/TLM Frame})}{12 \cdot \text{TLMR}}$$

where: (Bits/Line) specifies the number of bits in one line of a TLM format.

(Lines/TLM Frame) specifies the number of lines per TLM frame.

TLMR is the telemetry rate in bits/second.

For the TLM formats, (Bits/Line) = 128. A factor, δ , converts ΔS from units of seconds to spacecraft revolutions; δ is expressed in units of S/C revolutions per second. To accommodate instrument "shutdown," $\left(\frac{N-\Sigma}{N}\right)$ scales ΔS . N is the number of science bits in a sampling frame; Σ is the number of science bits unavailable because specific instruments are "off". Thus,

$$\Delta S = \frac{(\text{Bits/Line})(\text{Lines/TLM Frame})}{12 \cdot \text{TLMR}} \cdot \left(\frac{N-\Sigma}{N}\right) \cdot \delta$$

Table IV-6 shows some typical values of ΔS (assuming $\delta = 1$).

TABLE IV-6
TYPICAL VALUES OF ΔS (ASSUMING $\delta=1$)

	TLMR (BITS/ SEC)	ΔS ($\Delta S \geq 1$ by Definition), ($\delta=1$)					
		$\Sigma = 0$	$\Sigma = 302$	$\Sigma = 606$	$\Sigma = 909$	$\Sigma = 1815$	$\Sigma = 4818$
VLBRF (N = 1815)	8	26.67	22.30	17.80	13.30	No Sample	
	16	13.335	11.15	8.90	6.65	No Sample	
	32	6.67	5.58	4.45	3.33	No Sample	
	64	3.33	2.79	2.23	1.66	No Sample	
VHBRF (N = 4818)	128	4.50	4.20	3.94			No Sample
	256	2.25	2.10	1.97			No Sample
	512	1.13	1.05	1.00			No Sample
	1024	1.00	1.00	1.00			No Sample
FLBRF (N = 1815)	8	24.00					
	16	12.00					
	32	16.00					
	64	3.00					
FHBRF (N = 4818)	128	4.00					
	256	2.00					
	512	1.00					
	1024	1.00					

Generally, ΔS is noninteger. In the simulation model sampling occurs at specific integer values of spacecraft rotations. The science sampling format recycles every 12 ΔS spacecraft revolutions. A clock synchronized to the spin rate of the spacecraft provides the basic timing for the sampling subroutine.

We are concerned with the microstructure of the sampling during a spacecraft revolution. The correspondence between the sun-pulse clock sunline and instrument sampling times is given in Table IV-7.

TABLE IV-7
SUNLINE CORRESPONDING TO SUN-PULSE CLOCK

	Magneto- meter Sunline	Radio Prop Sunline	Neutron Detector Sunline	VLF Sunline	CR Sunline	Plasma Probe Sunline
Sun-Pulse Clock (SPCL) Modulo 128	0	107	85	64	43	72

At those S/C revolutions at which samples are scheduled, the samples promulgated in Figure IV-21 are taken. If the high bit-rate sampling format is employed and $\Delta S = 1$, magnetometer data is sampled at SPCL = 0, 32, 64, 96, modulo 128 each S/C revolution. The radio propagation data is sampled at SPCL = 107 when it is scheduled. The neutron detector is sampled at SPCL = 85. The VLF, CR, and PP experiments are sampled at different angles relative to the sunline, each spacecraft revolution; the sampling pattern is repetitive. For example, the VLF is sampled when scheduled at SPCL = 64, 96, 128, 32, etc.; only one sample set of the VLF output is taken during a particular spacecraft revolution. The cosmic ray experiment has nine sectors. It is sampled at $SPCL = \left[43 + n_9 \frac{128}{9} \right]$ nearest integer where $n_9 = 0, \dots, 8$ on successive samples. The sunline is assumed to be specified at the edge of the zone to be sampled. The sectors are not necessarily equal as specified for the CR telescope. Samples for the plasma probe are taken at SPCL = 72, 74, 76, 78, 80, 96, 112, 0, 16, 32, 48, 64, 66, 68, 70, modulo 128. The sectors running from SPCL = 64 - 80 are $\left(5 + \frac{5}{8} \right)$ degrees; all other sectors are 45 degrees in angular width.

The time resolution for the SPCL is $\left(\frac{1}{128} \right)$ seconds or approximately 8 milliseconds. It is convenient to move samples scheduled at a particular value of SPCL into the CDS before the SPCL count changes. This is not possible for

the cosmic ray telescope which must accumulate counts over an angular sector. Sampling must be controlled by a software program which determines if samples are to be accumulated during the current spacecraft revolution. If samples are scheduled, the program interrogates the SPCL and commands the necessary sampling procedure. At those spacecraft revolutions when samples are scheduled, the magnetometer is sampled at SPCL = 0, 32, 64, 96. Each magnetometer sample is converted and moved into the CDS in 4 milliseconds. When SPCL = 107, lines S_2 , S_3 , S_4 , and S_6 of the radio propagation experiment are A/D converted and moved into the CDS in approximately 2 milliseconds. RP (S_1) is sampled at SPCL = 0, 1, ... 127 when a sample is scheduled. Approximate time to sample, convert, and move data into the CDS from the RP (S_1) line is 50 microseconds. RP (S_5) is sampled at SPCL = 0, 4, 8, ... 124. The A/D conversion time is 1 millisecond per sample. The 19 counters of the neutron detector are sampled at SPCL = 85. The register contents are moved from the neutron detector into the CDS in approximately 1 milliseconds. It requires 100 microseconds to shift each of the 19 registers into the CDS. The VLF is scheduled for sampling at SPCL = 64, 96, 128, 32, etc. The servicing time required is 2 milliseconds. The plasma probe is sampled at SPCL = 72, 74, 76, 78, 80, 96, 112, 0, 16, 32, 48, 64, 66, 68, and 70. It requires 1 millisecond to sample each data line; it will take 3 milliseconds to sample, convert, and move data from the 8 data lines into the CDS, assuming that three multiplexers are used. At each scheduled sampling time, \hat{D}_1 and \hat{D}_3 must of the cosmic ray telescope be A/D converted and reset. Estimated time is 1 millisecond, assuming use of two multiplexers. It is assumed that the three lines of the cosmic ray telescope designated $D_1 D_2 \bar{D}_3 \bar{D}_4$, $D_1 D_2 D_3 \bar{D}_4$, and $\bar{D}_1 D_2 D_3 \bar{D}_4$ are terminated in flip-flops in the instrument. Each flip-flop is tested by the program at a rate of 1000/second for the former two lines and 100/second for the latter line. Each test operation on a flip-flop takes approximately 50 microseconds. The number of tests scheduled depends upon the angular width of each individual sector. Only one sector is scheduled for sampling per spacecraft revolution for the cosmic ray telescope for $\Delta S = 1$. Lines $D_1 \bar{D}_2 \bar{D}_4$ and D_4 are connected to CDS registers for an interval of 100 microseconds approximately every millisecond when a sector is scheduled for sampling. The sector dwell time is approximately one-ninth second assuming 9 equal angular sectors.

(128 → 512) BIT/REV SAMPLING FORMAT

ENTRY	MAG	COSMIC RAY	PLASMA PROBE	RADIO PROPAGATION					VLF
				RPA DATA	RPB (S ₂ , S ₃ , S ₆)	RPC (S ₄ , S ₅ , S ₅)	ND		
1	E _{x1} ...E _{z4}	ALL DATA (CRA)	ALL DATA (PPA)	NONE	NONE	RPC (S ₄ , S ₅ , S ₅)	NA DATA	(E _{x1} , E _{x2} , E _{z1} , E _{z2} , M ₁ , M ₂)	
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									

REPEATS →

30 82 190 304

302 304

732 564 1212 304

SUBTOTAL TOTAL = 4818

(8 → 64) B/S TLM SAMPLING FORMAT

ENTRY	MAG	CR	PPA	PPB	(S ₁) RPA	(S ₂ , S ₃ , S ₆) RPB	(S ₄ , S ₅ , S ₅) RPC	NA	NB	VLF
1	(E _{x1} , E _{y1} , E _{z1})	SECTOR 1	✓	✓	✓	✓	✓	✓	NONE	✓
2	(E _{x2} , E _{y2} , E _{z2})	NONE	NONE	NONE	NONE	NONE	✓	✓	✓	✓
3	(E _{x3} , E _{y3} , E _{z3})	SECTOR 2	✓	✓	✓	✓	✓	✓	✓	✓
4	(E _{x4} , E _{y4} , E _{z4})	NONE	✓	✓	✓	✓	✓	✓	✓	✓
5	(E _{x1} , E _{y1} , E _{z1})	SECTOR 3	NONE	NONE	✓	✓	✓	✓	✓	NONE
6	(E _{x2} , E _{y2} , E _{z2})	NONE	✓	✓	✓	✓	✓	✓	✓	✓
7	(E _{x3} , E _{y3} , E _{z3})	SECTOR 4	✓	✓	✓	✓	✓	✓	✓	✓
8	(E _{x4} , E _{y4} , E _{z4})	NONE	NONE	NONE	✓	✓	✓	✓	✓	✓
9	(E _{x1} , E _{y1} , E _{z1})	SECTOR 5	✓	✓	✓	✓	✓	✓	✓	✓
10	(E _{x2} , E _{y2} , E _{z2})	NONE	✓	✓	✓	✓	✓	✓	✓	✓
11	(E _{x3} , E _{y3} , E _{z3})	SECTOR 6	NONE	NONE	✓	✓	✓	✓	✓	✓
12	(E _{x4} , E _{y4} , E _{z4})	NONE	✓	✓	✓	✓	✓	✓	✓	✓

12Δ S → REPEAT

244 61 30 82 190 213 106 304

305 302 319

SUBTOTAL 282 303 304

TOTAL = 1815

Figure IV-21. (128 → 512) and (8 → 64) Bit/Rev Sampling Formats

Priority for the postulated sampling algorithm, must be specified since two instruments may be scheduled for sampling at the same value of SPCL. The priority list is promulgated in Table IV-8.

TABLE IV-8
SAMPLING PRIORITY

1	RP (S_1)
2	RP (S_5)
3	Magnetometer
4	RP (S_2, S_3, S_4, S_6)
5	Neutron Detector
6	VLF
7	Cosmic Ray
8	Plasma Probe

The problem of scheduling nonoverlapping sampling can be determined from Table IV-9.

Some quick fixes can be made. For example, the VLF and magnetometer samples can be separated by simple rescheduling. The schedule for RP (S_5) can be modified to avoid conflicts with the magnetometer or VLF sample schedules; the RP (S_1) line poses no problem. The sampling schedule including the plasma-probe line of Table IV-9 can be rearranged to minimize overlap. The cosmic-ray instrument must be sampled over an angular sector. For low bit-rate formats, the problem can be avoided by changing the sampling format of Figure IV-21. At high bit rates, the problem cannot be avoided.

The real-time program provides the scheduling function. It specifies what is to be done next and when, and also, assigns task priorities. The available telemetry rates range from 8 - 2048 bits/second. In one method of

TABLE IV-9

SAMPLING SCHEDULE

Instrument	Sampling Schedule (SPCL Units)	Time Required
Magnetometer	0,32,64,96	A/D conversion time → 4 milli-seconds/sample.
RP (S_1)	0,1,2,3,4,...127	Bilevel test time → 50 micro-seconds/sample.
RP (S_5)	0,4,8,...124	A/D conversion time → 1 milli-second/sample.
ND	85	Shift register contents to CDS → 1 millisecond.
VLF	64,96,0,32	A/D conversion → 2 milliseconds/sample.
Plasma Probe	72,74,76,78,80,96, 112,0,16,32,48,64, 66,68,70	A/D conversion → 6 milliseconds/sample.
CR (\hat{D}_1, \hat{D}_3)	43,57,71,86,100, 114,0,15,29	A/D conversion → 1 millisecond/sample.
CR ($D_1 D_2 \bar{D}_3 \bar{D}_4, D_1 D_2 D_3 \bar{D}_4$)	43,57,71,86,100, 114,0,15,29	50 microseconds/test and 1 test/millisecond.
CR ($\bar{D}_1 D_2 D_3 \bar{D}_4$)	43,57,71,86,100, 114,0,15,29	50 microseconds/test and 1 test/10 milliseconds.
CR ($D_1 \bar{D}_2 \bar{D}_4, D_4$)	43,57,71,86,100, 114,0,15,29	100 microseconds/sample, 1 sample/millisecond.

providing bits for the telemetry modulator at the highest bit rate, the program scheduling subroutine must be entered every $\left(\frac{1}{211}\right)$ seconds which is approximately every (1/2) millisecond. Each time the program-scheduling subroutine is entered, the following kinds of questions must be asked:

- 1) Is it time to provide a bit for the telemetry modulator?
- 2) Are instrument samples scheduled this S/C revolution?
- 3) What is SPCL and what sample subroutine is scheduled?
- 4) What task should be done if samples are not scheduled?
- 5) Do we have time for executing a portion of a processing subroutine?

Question 2) needs to be affirmed at the time SPCL = 0 (approximately once a second). If an input sample is scheduled, a data sampling schedule subroutine is entered.

Assume the system is operating at 512 bits/second and science samples are scheduled. The high bit-rate sampling format of Figure IV-21 is to be executed. Assume the samples are defined by Entry 1 of the high bit-rate sampling format of Figure IV-21. Assume a bit has just been provided the TLM modulator. At SPCL = 0, the first operation is to sample the contents of RP (S_1) for bilevel. It is assumed the CDS memory-cycle time is 2 microseconds and it takes 20 microseconds to establish the connection and 20 microseconds to test and move the results into storage as necessary. The flip-flop is not reset. It is compared to its previous condition, and a counter is incremented only when the flip-flop changes state. The program reenters (under memory cycle clock control) the data sampling subroutine approximately 50 microseconds from the previous service request. The next sample is RP (S_5) which requires 1 millisecond. The time available before the next entry into the scheduling routine is determined. If greater than 100 microseconds is available, RP (S_5) is connected through the multiplexer to a CDS register. The program triggers the A/PW converter via the command switch. The program continually tests for moving a bit to the TLM modulator, and executes the bit transfer as necessary. The program reads the appropriate register (contents of RP (S_5)) into the machine at the end of 1 millisecond. The elapsed time is

roughly 1070 microseconds. The program rebranches to the data sampling schedule subroutine where a request for magnetometer data is scheduled. If greater than 100 microseconds is available, the three data lines of the multiplexer are connected to three multiplexers, respectively. The A/PW converters are triggered and conversion starts. The conversion time is 4 milliseconds per line. The program continually interrogates to determine when a bit is to be moved to the TLM modulator. At the end of 4 milliseconds, the respective registers are moved into CDS memory. The remaining possibility for sampling at SPCL = 0 is the VLF. It is sampled as necessary and moved into the CDS in 2 milliseconds. The total elapsed time is 7100 microseconds. The SPCL changes count every 7800 microseconds so that the schedule is rather crowded for SPCL = 0 at high-bit rates. The program has been completely devoted to sampling and moving data into memory and satisfying the TLM requirements. There has been no time for doing anything else. It is assumed that the time lag is not serious, or put in another way, that angular resolution of one part in 128 is adequate for the experimenters. The preceding example suggests that not only must the master scheduling subroutine be entered frequently but that operations should not be performed unless there is adequate time. The questions are: "What is the next task? Is there time to perform it and still service the TLM modulator?" These questions must be answered when the real-time program is written.

One of the busiest traffic situations occurs at SPCL = 72. The RP (S_1), RP (S_5), and the plasma probe samples are scheduled, and cosmic ray data is to be accumulated over a sector starting at SPCL = 71 and ending at SPCL = 86. At SPCL = 72, RP (S_1) and RP (S_5) are sampled. This requires roughly 1050 microseconds. Lines $D_1 D_2 D_3 \bar{D}_4$ and $D_1 D_2 \bar{D}_3 \bar{D}_4$ of the cosmic ray telescope are scheduled for sampling every millisecond. Line $\bar{D}_1 D_2 D_3 \bar{D}_4$ is sampled every 10 milliseconds. The three lines are sampled simultaneously on three different multiplexers. Assume 50 microseconds is necessary to complete the switching, testing, and incrementing of associated flip-flops. After this operation, the TLM link is serviced if necessary. (Assume this requires another 20 microseconds), we now have 1120 microseconds of elapsed time since SPCL = 72 initially. The plasma-probe data lines are scheduled for sampling. There are eight data lines. Each conversion requires 2 milliseconds. Three lines

may be converted simultaneously. Each line of the plasma probe is equipped with its own eight-bit A/PW converter. During the time this conversion is going on, no other data line can be serviced. During the 2 milliseconds, the program can perform other functions; it checks the master scheduling subroutine every $\left(\frac{1}{211}\right)$ seconds to determine what the next task will be. At the end of 2 milliseconds or at a total elapsed times of 3120 microseconds, cosmic ray lines $D_1 D_2 D_3 \bar{D}_4$ and $D_1 D_2 \bar{D}_3 \bar{D}_4$ are serviced for 50 microseconds. Cosmic ray lines $(D_1 \bar{D}_2 \bar{D}_4$ and $D_4)$ are serviced for 100 microseconds. Total elapsed time is now 3290 microseconds. Plasma-probe lines $(F_4, F_5,$ and $F_6)$ are serviced for 2 milliseconds. Total elapsed time is 5310 microseconds. Cosmic ray lines $D_1 D_2 D_3 \bar{D}_4$, $D_1 D_2 \bar{D}_3 \bar{D}_4$, and cosmic ray lines D_4 and $D_1 \bar{D}_2 \bar{D}_4$ are serviced in sequence. Total elapsed time is 5480 microseconds. Plasma probe lines $(F_7$ and $F_8)$ are serviced for 2 milliseconds. Total elapsed time is 7500 microseconds. This leaves roughly 300 microseconds during which additional tasks can be handled. The cosmic ray continues to be sampled approximately every millisecond until the SPCL = 86. Another sector of cosmic ray sampling is not scheduled for at least another spacecraft revolution. We have demonstrated for reasonable conversion and servicing times, the inputting function can be implemented. It may not be possible to sample each cosmic-ray line every millisecond, but a sampling schedule can be written to guarantee that a specific number of samples can be executed for each sector.

We have considered two values of SPCL and a telemetry rate such that virtually all the available time is utilized servicing the input and output of the system. From an examination of Table IV-10 and Figure IV-21, the input/output servicing time is 106 milliseconds/second on the average. This means that for approximately 90 percent of the time, the machine is available for doing other tasks. If one multiplexer fails, we estimate that 15 percent of the available time is required for input/output servicing time, leaving 85 percent available for doing other tasks. If two multiplexers fail, we estimate that 20 percent of the machine time is used servicing input/output lines. Table IV-10 summarizes the input/output servicing requirements for high bit rates for the assumed interface.

At the low telemetry rates, the situation improves since the frequency at which the input is sampled is inversely proportional to the telemetry rate. Consequently, the time the CDS is available for processing is inversely proportional to the telemetry rate. At eight bits a second, the input/output service time requirements are never more than 5 percent of the available time on the average.

The forgoing conclusions are based on a machine having a two-microsecond memory cycle and a specific interface configuration. If either the memory cycle or interface configuration changes, the numbers must be modified.

The software program re-enters the schedule and sampling subroutine approximately 2000 times a second. If it requires 50 microseconds to decide the next task, 100 milliseconds/second is tied up with this function. Some of this time should be charged against the input/output interface. The rough estimates arrived at after a detailed consideration of the problem can be refined by a real time flow chart of the scheduling and sampling subroutine. This must be done taking into account various interface configurations, memory cycle times, and various degrees of spin rate and telemetry rate asynchronism.

TABLE IV-10

SERVICING TIME REQUIREMENTS FOR INPUT/OUTPUT
AT TLMR \geq 512 IN PERCENT OF TOTAL TIME

<u>1 Multiplexer</u>	<u>2 Multiplexers</u>	<u>3 Multiplexers</u>
20% (some data loss)	15%	10%

IV.5 INPUT/OUTPUT INTERFACE POSSIBILITIES

A/PW conversions require a large percentage of the input/output servicing time. A 128-kHz clock takes 4 milliseconds to count to 512 (9-bit resolution, 8-bit accuracy). If the clock frequency is increased by a factor of 4

and the sweep generator of the A/PW converter modified to sweep the range 0-5^V in 1 millisecond, the total conversion time is 1 millisecond. Conversion accuracy is a function of the linearity of the sweep generator and the shape of the pulse which triggers a control gate to "off" in the CDS. The trigger is a/c coupled to the CDS via a multiplexer. If the pulse is well defined, 8-bit accuracy is possible. If the pulse is not reasonably well shaped, 8-bit accuracy may not be possible. With high quality transformers and well designed multiplexers, a factor of four or more decrease in A/PW servicing time over that promulgated in Table IV-1 appears to be feasible for the analog science data lines.

Three A/PW converters can service the 8 data lines of the plasma probe. Commands to assign A/PW converters to particular data lines of the plasma probe can be generated by the on-board program. One A/PW converter could be assigned as a spare. Similar provisions could be made for the VLF experiment.

A/PW converters have been suggested for 26 science-data lines. An equal number of science-data lines require counters. The neutron-detector data lines place a heavy demand on interface servicing time. The gamma and neutron events are characterized by two-microsecond pulses and rates of 10⁴ events/second. To service these lines directly during the heaviest traffic situations, each line must be sampled every 100 microseconds. The lines in this case would be terminated in flip-flops; the estimated servicing time is 50 microseconds per sample. This implies the CDS spends 50 percent of available time servicing these two lines (assuming they are serviced simultaneously). If the cognizant experimenter is willing to sample his experiment every millisecond for 100 microseconds, the service demand is more reasonable (10 percent of available time). The experimenter may elect to scale his counter in the instrument and consequently reduce the rate at which the gamma and neutron lines have to be sampled. The experimenter may have other ideas. One point is clear--the experimenter will want data if one of his counters fails. To provide for such a contingency, the CDS should be capable of providing a counting service to the experimenter. The counting for 100 microseconds every millisecond when a sample is scheduled seems like a reasonable service. The ion line of the neutron detector could be serviced in the same way. The 16-channel, pulse-height analyzer presents a more difficult problem. To perform

this function in the CDS requires performing a 4-bit A/D conversion at a maximum rate of 10^4 times a second. It would take at least 50 microseconds to make the conversion. This implies a servicing time of 1/2 second/second during busiest traffic conditions. Clearly, this line cannot be serviced during busy traffic conditions. However, when the bit rates are 64 bits/second or less, this line can be serviced. The scheduling and sampling subroutine must employ telemetry rate as a parameter. The 8 - 64 bit/second telemetry format of Figure IV-21 requires modification to allow a greater flexibility in the selection of the S/C rotations at which the various data lines are serviced. Table IV-5 shows the ΔS ranges from 26.67 to 3.33 for all instruments "on" for the VLBRF. This means that these are spacecraft rotations at which no samples are scheduled. A more flexible sampling format can employ time during these spacecraft revolutions for servicing data lines which require a large percentage of available time for input operations.

Attention has been directed toward establishing the interface configuration for the busy traffic (high telemetry bit rate) situation. The interface configuration can change for the low bit rate science format so the CDS performs more of the functions currently performed within the instrument black boxes. This is the type of back-up an instrument designer wants. If his counter fails, the CDS can perform the counting function completely at low bit rates and partially at high bit rates. The CDS should have access to the input and output of all instrument counters. For analog science lines, the analog voltage and the A/PW output can be made available to the CDS. The relevant data line should be selectable by command. This allows a fluid interface and provides insurance against equipment failure. It provides each experimenter the option of employing CDS furnished counters or his own. The present sampling mode of the simulation program can be altered to provide more flexibility in the sampling format at low bit rates. This is only a partial solution. The next approach is to map out the real-time logic which must be employed by the flight model.

The input/output time budget is heavily weighted by the assumption that the CDS holds and waits which an A/PW conversion takes place. This is obviously a waste of time since the CDS can execute other instructions during

the time required for the A/PW conversion to take place. Thus the time estimated for input/output operations is extremely conservative. However, the amount by which it is too high cannot be resolved further until the task scheduling executive routine has been completely defined.

The marriage between the software and hardware (necessary for future Pioneer central data system design) recommends that the logical design effort and a software phase be done concurrently.