Apr 3rd, 12:00 AM

Central Data System Design for Scientific Solar Probes

Stephen C. Bigelow
Sylvania Applied Research Laboratory, Waltham, Massachusetts

William F. Higgins
Sylvania Applied Research Laboratory, Waltham, Massachusetts

CENTRAL DATA SYSTEM DESIGN FOR SCIENTIFIC SOLAR PROBES

Stephen C. Bigelow and William F. Higgins
Sylvania Applied Research Laboratory
Waltham, Massachusetts

Introduction

Scientific solar probes (such as Pioneer) 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 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 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 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.

A system simulation model for the central data system of a scientific space probe has been designed and programmed. The goal is to define the basic and ancillary services that the central computer can provide the experimenters. Development of the simulation program has forced the designer 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 wave-particle experiment. Engineering data inputs are simulated and processed by the simulation program.

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, telemetry rates and formats, and the parameters which exercise the simulation model. The optional processing algorithms available in the simulation model include quantiles, histograms, spectral analysis, digital filtering, mean and variance, and zero order prediction subroutines.

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 instrument failure or shut-down 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 which are currently operational.

An analysis and evaluation of the simulation runs have provided keen insight into the computer requirements for storage capacity, machine word size, instruction word format, instruction list, and memory cycle time.

Problem Statement

The general problem considered here is that of designing a central data system 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 central data system must meet.

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 will be examined in some detail in the following sections.

Data Input Requirements

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

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 central data system 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 central data system must control the interface equipment to select both an input source and the time of its sampling.

The data input requirements which the central data system 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. While a detailed description of this model is beyond the scope of the present paper, it 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 central data system must satisfy in order to input the scientific data. This interface is shown in simplified form in Fig. 1.

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 central data system 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 central data system. This inputting function must be performed without delay whenever a ground command is received.

Data Output Requirements

Two classes of data must be outputted by the central data system: 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.

Central Data System Requirements

The central data system 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.

Design Objectives and Constraints

The central data system 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.

Approach

In view of the central data system 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 we split the CDS design study 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. Questions relating to machine capacity and speed were left unresolved during this phase.

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. Our purpose in doing this was twofold: First it forced us to examine with great care all of the CDS operations which would be essential or highly desirable in an actual system, and second it enabled us 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 central data system has proved to be quite effective. The results of the study are presented in the following sections.

Central Data System — Hardware

We have made use of a unique conceptual design[1] for a general purpose programmable central data system. A general purpose computer approach, which gives us enormous flexibility, is combined with a pooling technique to allow survival after a component failure. The survival is achieved by reporgramming 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 central data system.

CDS Logical Design

A computer using this pooling technique is shown in Fig. 2. 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 bussess 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 register associated with each logic unit. This logic 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.

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.

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.

The central data 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 central data system.

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, up-link commands will be made to change all required addresses. If a tape recorder is onboard 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, then we cannot change modes or "repair" the CDS by reprogramming.

Central Data System — Software

The CDS software or program requirements are discussed briefly in the following subsections 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, we have in this study 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 and 512 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 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 sections.

Fixed Engineering Format. One complete frame of this format consisting of 128-bit lines is shown in Fig. 3. 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 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 format is shown in Fig. 4. 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 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 Fig. 5. 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 Fig. 6. 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 nth 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 Fig. 6), 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 Fig. 6 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.

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 next section. 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.

Fig. 7 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. 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 up-dated. Instrument-operating modes are examined and the format assembly tables are up-dated when necessary. The rate at which the sampling format is to be executed is computed. The cut-off 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 up-dated. 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 up-dated.

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 up-dated. Instrument command and operational changes are executed. Sampling times are computed and stored. The digital filter cut-off 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.

Simulation Study Results

The simulation model program coding has been scrutinized in detail. It was written as a combination of FORTRAN and machine language instructions. The simulation program requires 11,800 storage locations in the CDS 3200. Twenty-four hundred storage locations are used for constants, data, and tables. Eight hundred and fifty locations are reserved for functions which exercise the simulation model. Five hundred locations are occupied by special call subroutines which would occupy virtually no locations if coded directly in machine language. Fifteen hundred locations are occupied by instructions for processing subroutines. The instruction list comprises 8900 locations. By more efficient flow charting and encoding, the instruction list may be reduced to 7000 locations. The transformation from FORTRAN to machine language for the type of instructions executed most frequently by the program is not efficient. Approximately 50 percent of the instructions are move instructions. Approximately 30 percent of the move instructions involve a test for zero. Approximately 20 percent of the move instructions involve a mask and pack. FORTRAN instructions generate 1 → 20 machine instructions. The 7000 instructions can be reduced to less than 2500 by rewriting the program in machine language. The 2400 locations for constants can be reduced to 1000 by consolidating tables and eliminating information which is not needed for a flight model. Approximately 20 percent of the instruction list is used for implementing processing algorithms. The number of locations required for servicing the input/output interface is estimated to be 500. The total storage required for a flight model of the CDS is, therefore, approximately 4000 locations.

By exercising the simulation model it has been possible to estimate the time required to execute all phases of the program. The time available to perform all programmed tasks for the simulation model is a function of telemetry rate and sampling schedule. For the high bit rate science format with all processing algorithms operating, the execution time for all operations exclusive of servicing the input interface is approximately 25 percent of the available time on the average. Fifteen percent of available machine time is required to service input/output interface. This implies that the machine uses 40 percent of its available capacity for the postulated interface and instrument set under the most demanding conditions. These estimates are based on a machine cycle time of 2 microseconds.

A machine word size of 16 bits is 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 4-bit operation code and a 12-bit address field are specified. Each 16-bit data word has 15 bits for data and 1 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, multiply, divide, move, shift left or right, test for zero and branch, mask, read and write; a special mask, move, and pack instruction is recommended since these operations occur frequently during format assembly.

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.

Conclusions

The results of the study reported here indicate that a stored program digital computer can be used to advantage as a central data system 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.

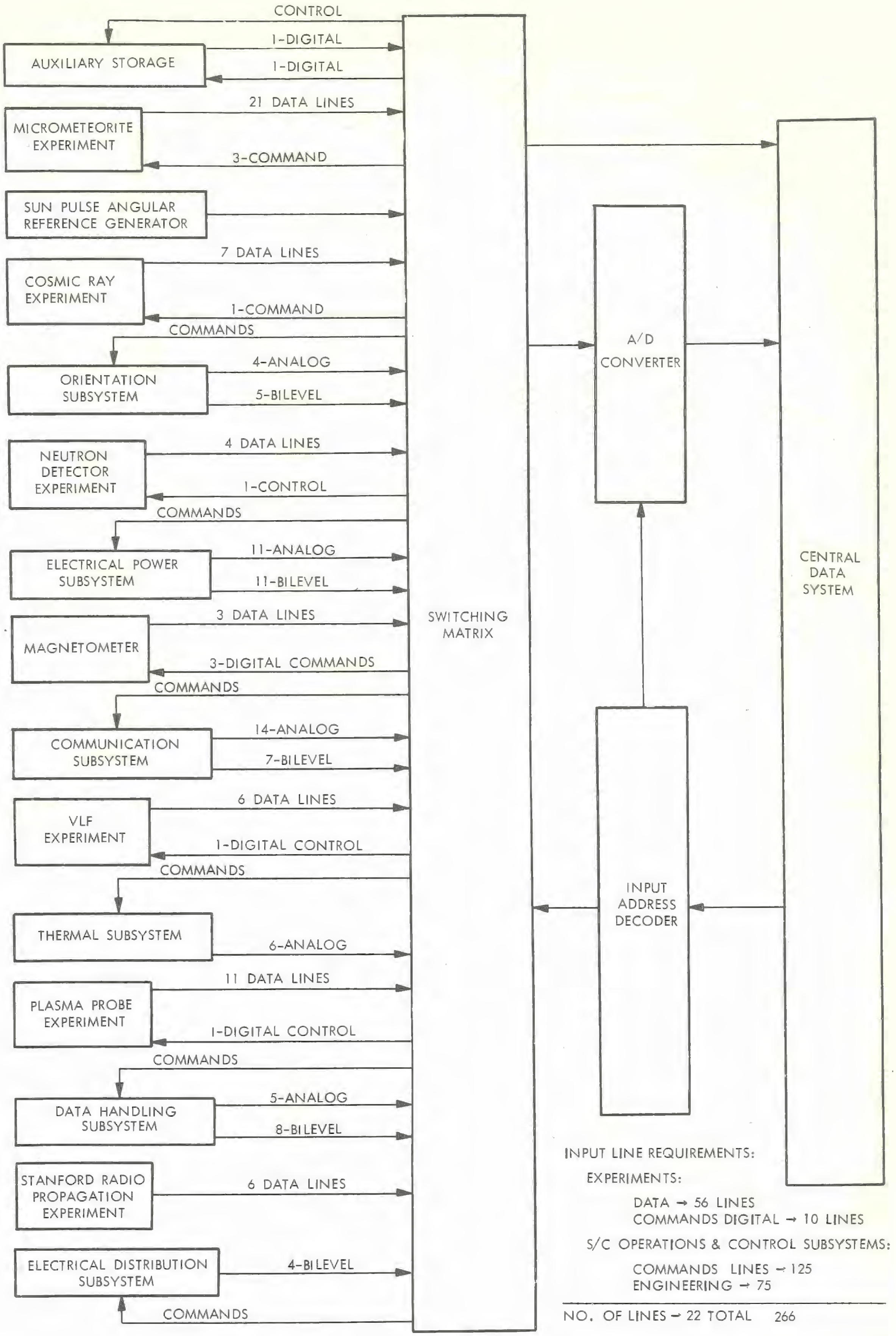
It has been demonstrated that a stored program CDS can provide more data processing and formatting capability than the present Pioneer data system without exceeding reasonable size, weight and power specifications. Furthermore, such a CDS makes possible efficient use of the telemetry channel capacity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

Reference

1. Fimmel, Richard O., and Thomas E. Baker, "Multipac, A New Concept for Spacecraft Data Management Systems," to be published.

This work was supported in part by the NASA Ames Research Center under Contract No. NAS2-3255.



INPUT LINE REQUIREMENTS:

EXPERIMENTS:

DATA → 56 LINES
COMMANDS DIGITAL → 10 LINES

S/C OPERATIONS & CONTROL SUBSYSTEMS:

COMMANDS LINES → 125
ENGINEERING → 75

NO. OF LINES → 22 TOTAL 266

Figure 1. Interface Between CDS and "On-Board" Experiments and S/C Support Equipment.

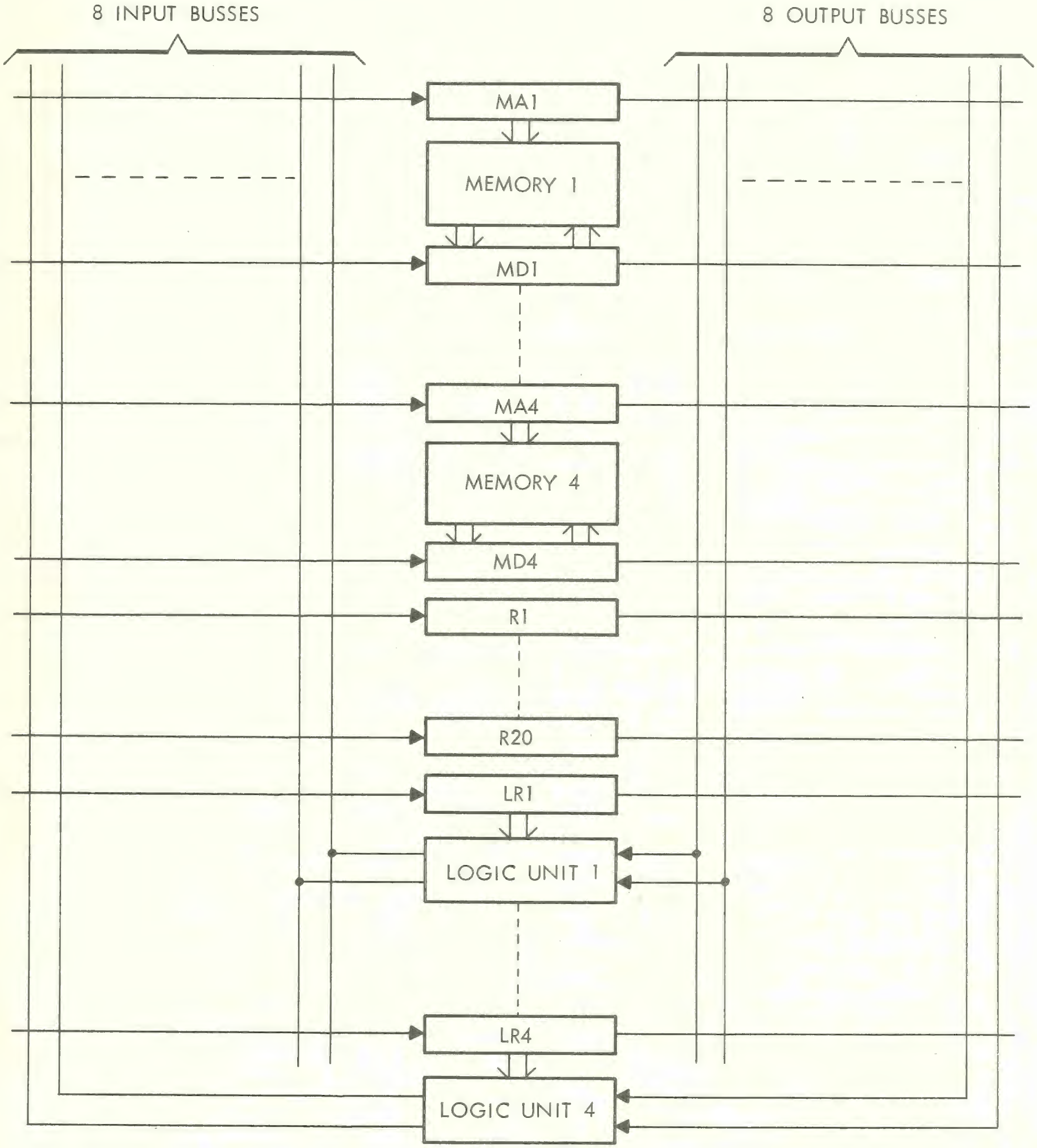


Figure 2. Ultra Reliable Computer.

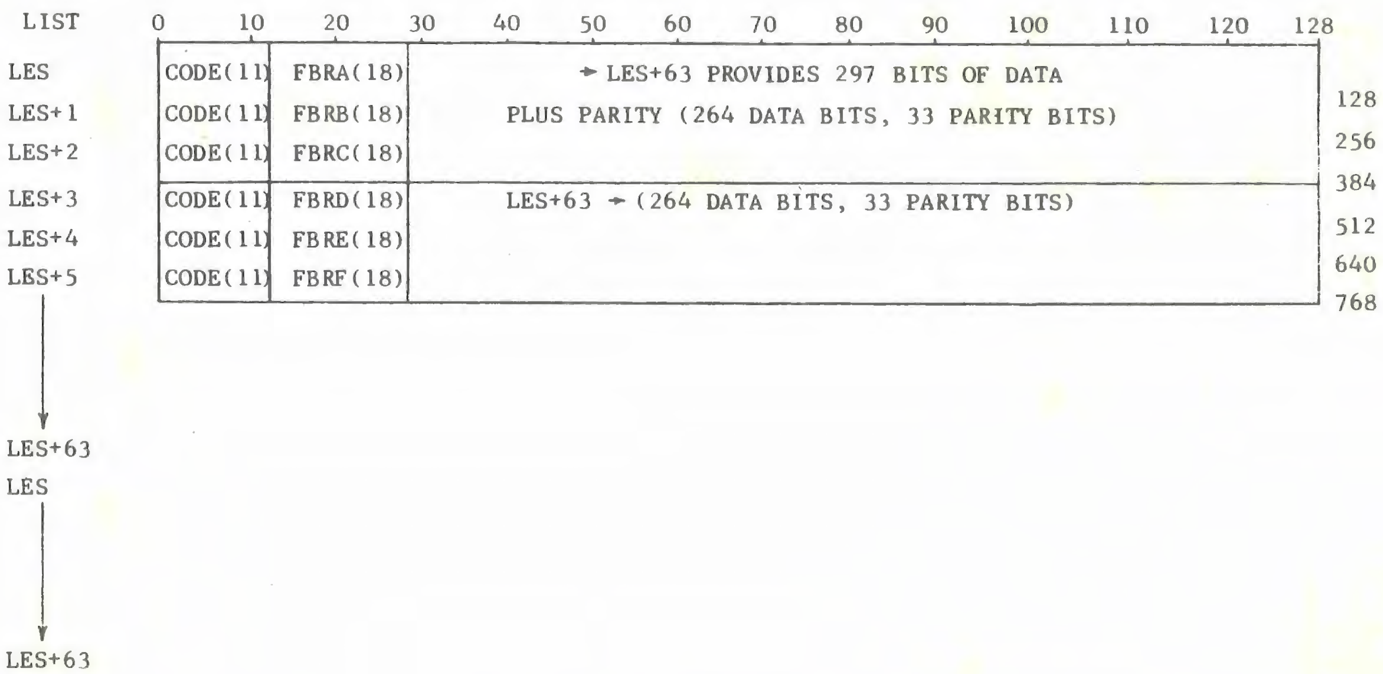
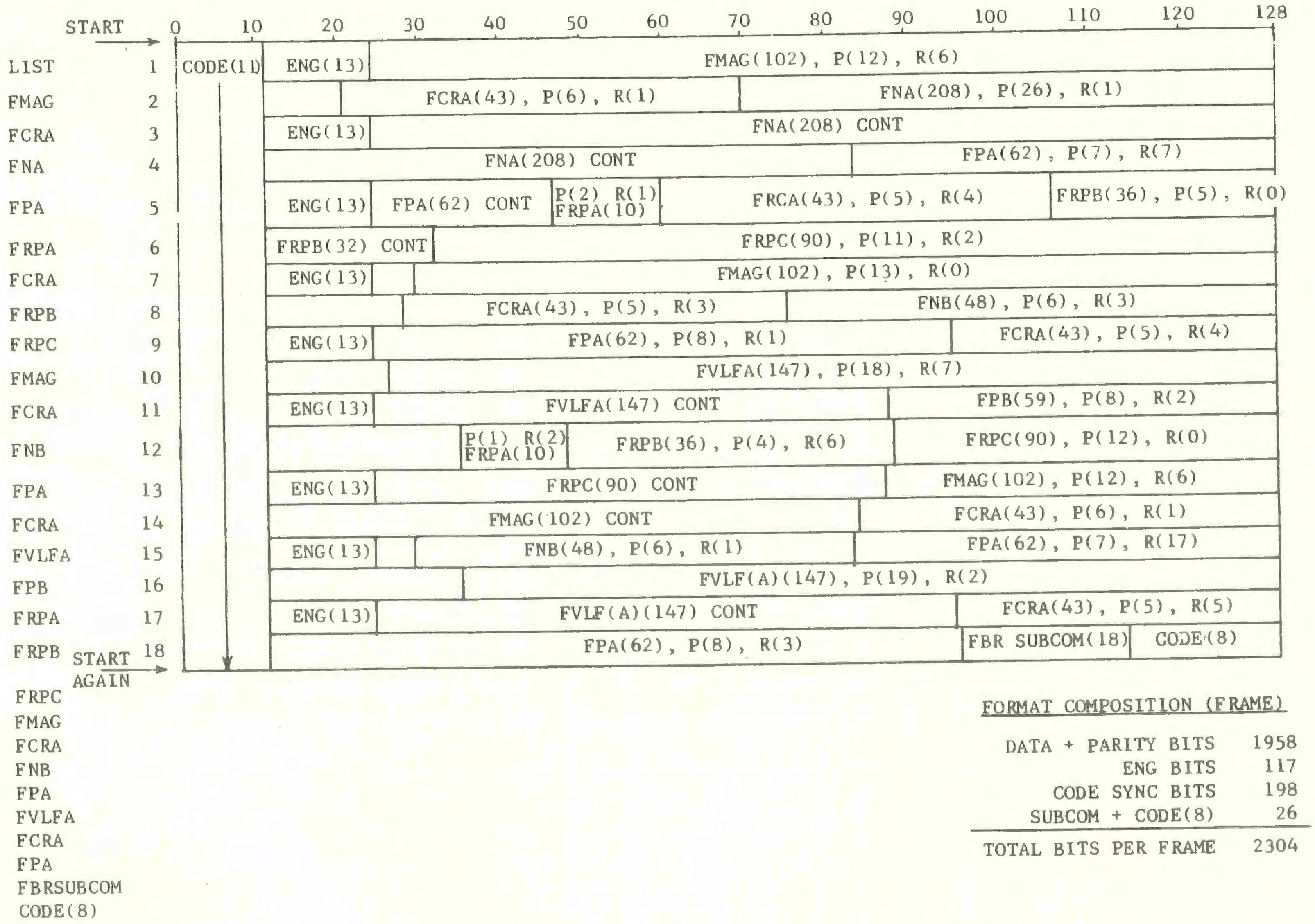


Figure 3. Fixed Engineering Format.


START
0
10
20
30
40
50
60
70
80
90
100
110
120
128
LIST 1 CODE(11) ENG(13) FMAG(102), P(12), R(6)
FMAG 2 FCRA(43), P(6), R(1) FNA(208), P(26), R(1)
FCRA 3 ENG(13) FNA(208) CONT
FNA 4 FNA(208) CONT FPA(62), P(7), R(7)
FPA 5 ENG(13) FPA(62) CONT P(2) R(1) FRPA(10) FRCA(43), P(5), R(4) FRPB(36), P(5), R(0)
FRPA 6 FRPB(32) CONT FRPC(90), P(11), R(2)
FCRA 7 ENG(13) FMAG(102), P(13), R(0)
FRPB 8 FCRA(43), P(5), R(3) FNB(48), P(6), R(3)
FRPC 9 ENG(13) FPA(62), P(8), R(1) FCRA(43), P(5), R(4)
FMAG 10 FVLFA(147), P(18), R(7)
FCRA 11 ENG(13) FVLFA(147) CONT FPB(59), P(8), R(2)
FNB 12 P(1) R(2) FRPA(10) FRPB(36), P(4), R(6) FRPC(90), P(12), R(0)
FPA 13 ENG(13) FRPC(90) CONT FMAG(102), P(12), R(6)
FCRA 14 FMAG(102) CONT FCRA(43), P(6), R(1)
FVLFA 15 ENG(13) FNB(48), P(6), R(1) FPA(62), P(7), R(17)
FPB 16 FVLF(A)(147), P(19), R(2)
FRPA 17 ENG(13) FVLF(A)(147) CONT FCRA(43), P(5), R(5)
FRPB 18 FPA(62), P(8), R(3) FBR SUBCOM(18) CODE(8)
START AGAIN
FRPC
FMAG
FCRA
FNB
FPA
FVLFA
FCRA
FPA
FBRSUBCOM
CODE(8)

FORMAT COMPOSITION (FRAME)

DATA + PARITY BITS	1958
ENG BITS	117
CODE SYNC BITS	198
SUBCOM + CODE(8)	26
TOTAL BITS PER FRAME	2304

Figure 4. FLBR Science Format with Parity.

FMAGA(102)

DID(6)	$E_{x_1}(8)$	$E_{y_1}(8)$	$E_{z_1}(8)$	$E_{x_2}(8)$	$E_{y_2}(8)$	$E_{z_2}(8)$	$E_{x_3}(8)$	$E_{y_3}(8)$	$E_{z_3}(8)$	$E_{x_4}(8)$	$E_{y_4}(8)$	$E_{z_4}(8)$

DID(6) ⇒ DYNAMIC RANGE(2), BIAS(3), ORIENTATION(1)

FCRA(43)

DID(7)	PHA(7)	PHA(5)	$C_1(4)$	$C_2(3)$	$C_3(3)$	$C_4(3)$	$C_5(5)$	$C_6(6)$

$C_1 = D_1\bar{D}_2\bar{D}_4$, $C_2 = D_1D_2\bar{D}_3\bar{D}_4$, $C_3 = D_1D_2D_3\bar{D}_4$, $C_4 = \bar{D}_1D_2D_3\bar{D}_4$, $C_5 = C_2$, $C_6 = D_4$

DID(7) = SECTOR(4), QUADRANT(2), CALIBRATE(1)

FPA(62)

DID(6)	$F_1(7)$	$F_2(7)$	$F_3(7)$	$F_4(7)$	$F_5(7)$	$F_6(7)$	$F_7(7)$	$F_8(7)$

DID(6) ⇒ VOLTAGE LEVEL(5), SECTOR CYCLE FLAG(1)

FPB(59)

DID(3)	$F_M(7)$ CH(3) SECT(4)	$F_M(7)$ CH(3) SECT(4)	$F_M(7)$ CH(3) SECT(4)	$F_M(7)$ CH(3) SECT(4)

DID(3) = VOLTAGE CYCLE FLAG(1), POSITION AT WHICH FLAG SET(2)

FRPA(10)

$S_1(10)$

FRPB(36)

$S_2(6)$ $S_3(6)$ $S_6(6)$	$S_2(6)$ $S_3(6)$ $S_6(6)$

FRPC(90)

S_4 $\hat{S}_5$ $\check{S}_5$	S_4 $\hat{S}_5$ $\check{S}_5$	S_4 $\hat{S}_5$ $\check{S}_5$	S_4 $\hat{S}_5$ $\check{S}_5$	S_4 $\hat{S}_5$ $\check{S}_5$

$\hat{S}_5 = S_5$ MAX(6), $\check{S}_5 = S_5$ MIN(6)

FNA(208)

PHA 1	PHA 2	PHA 3	PHA 4	PHA 5	PHA 6	PHA 7	PHA 8	PHA 9	PHA 10	PHA 11	PHA 12	PHA 13	PHA 14	PHA 15	PHA 16	N-COUNT (14)	γ-COUNT (14)
PART COUNT (20)																	

PHA 1 ⇒ PHA CHANNEL 1(10)

FNB(48)

N-COUNT (14)	γ-COUNT (14)	PART COUNT (20)

FVLFA(147)

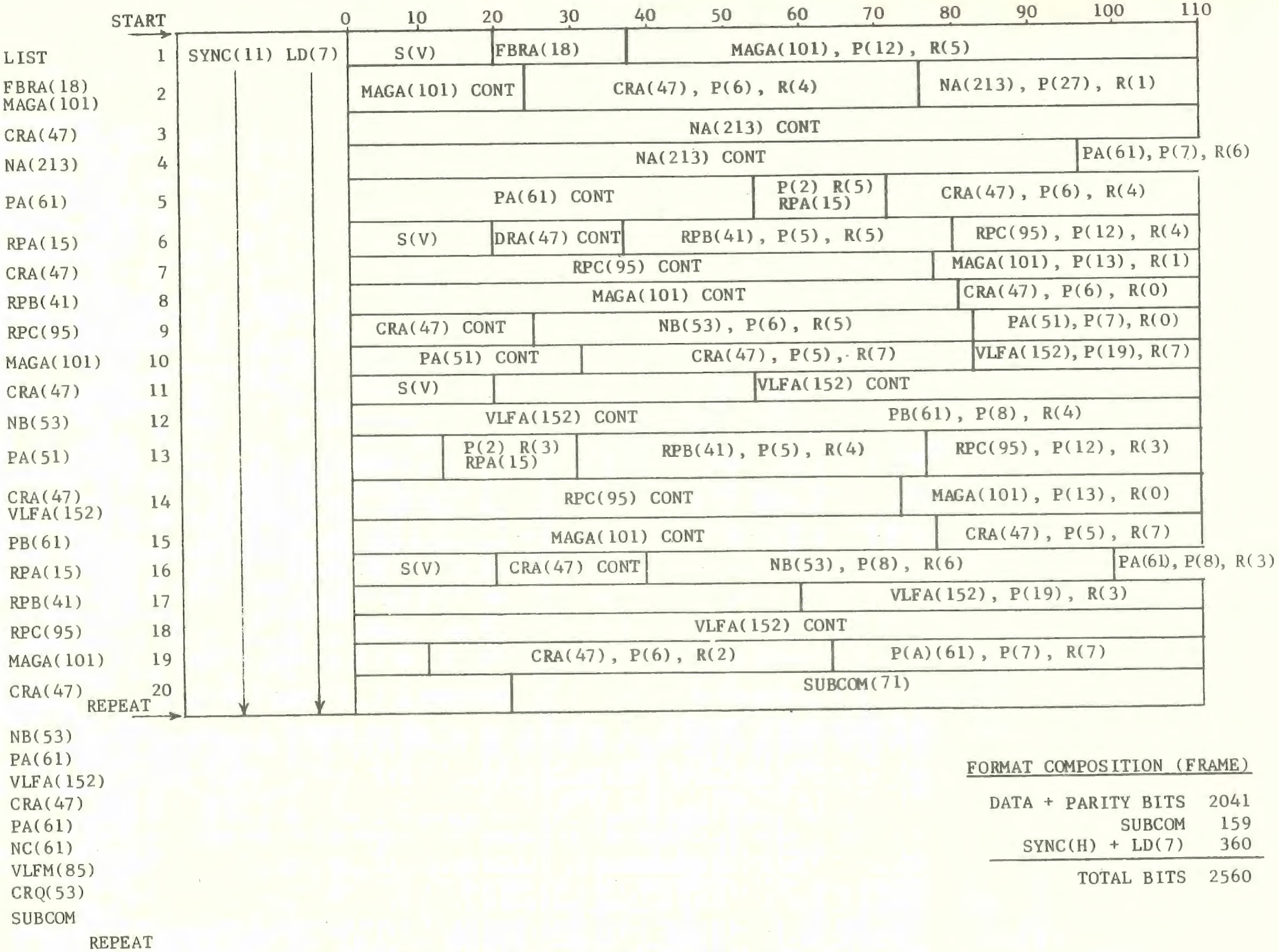
DID(3)	E_{x_1} E_{z_1} E_{x_2} E_{z_2} M_1 M_2	E_{x_1} E_{z_1} E_{x_2} E_{z_2} M_1 M_2	ETC	ETC

DID(3) ⇒ FILTER SELECTION(3)

ENG(13)

DATA IDENT(7)E	DATA(6)

Figure 5. Block Formats for Fixed Bit Rate Telemetry.



FORMAT COMPOSITION (FRAME)

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

Figure 6. Variable LBR Telemetry Format with Parity.

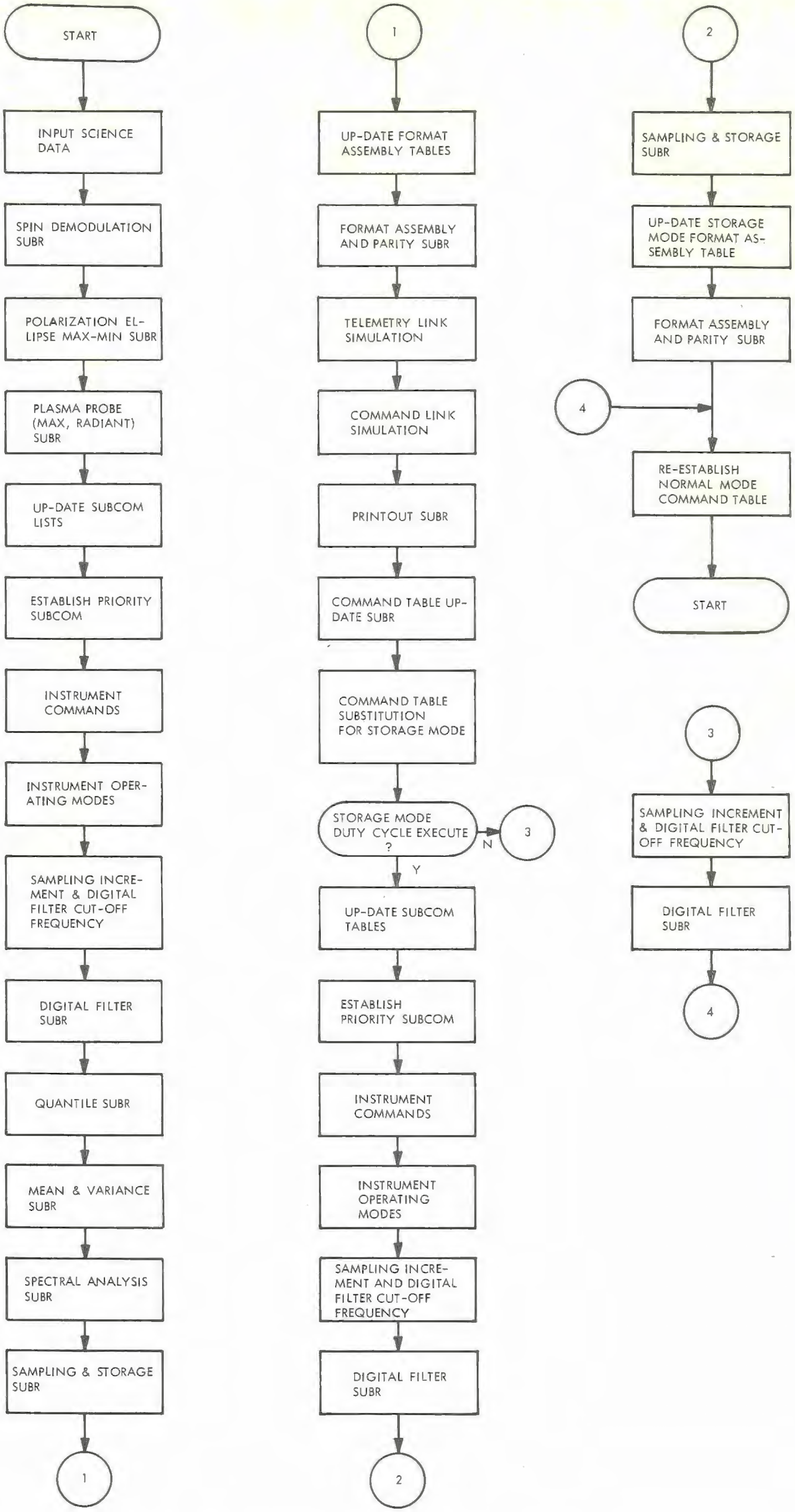

START
INPUT SCIENCE DATA
SPIN DEMODULATION SUBR
POLARIZATION ELLIPSE MAX-MIN SUBR
PLASMA PROBE (MAX, RADIANT) SUBR
UP-DATE SUBCOM LISTS
ESTABLISH PRIORITY SUBCOM
INSTRUMENT COMMANDS
INSTRUMENT OPERATING MODES
SAMPLING INCREMENT & DIGITAL FILTER CUT-OFF FREQUENCY
DIGITAL FILTER SUBR
QUANTILE SUBR
MEAN & VARIANCE SUBR
SPECTRAL ANALYSIS SUBR
SAMPLING & STORAGE SUBR
1
1
UP-DATE FORMAT ASSEMBLY TABLES
FORMAT ASSEMBLY AND PARITY SUBR
TELEMETRY LINK SIMULATION
COMMAND LINK SIMULATION
PRINTOUT SUBR
COMMAND TABLE UP-DATE SUBR
COMMAND TABLE SUBSTITUTION FOR STORAGE MODE
STORAGE MODE DUTY CYCLE EXECUTE ?
N
3
Y
UP-DATE SUBCOM TABLES
ESTABLISH PRIORITY SUBCOM
INSTRUMENT COMMANDS
INSTRUMENT OPERATING MODES
SAMPLING INCREMENT AND DIGITAL FILTER CUT-OFF FREQUENCY
DIGITAL FILTER SUBR
2
2
SAMPLING & STORAGE SUBR
UP-DATE STORAGE MODE FORMAT ASSEMBLY TABLE
FORMAT ASSEMBLY AND PARITY SUBR
4
RE-ESTABLISH NORMAL MODE COMMAND TABLE
START
3
SAMPLING INCREMENT & DIGITAL FILTER CUT-OFF FREQUENCY
DIGITAL FILTER SUBR
4

Figure 7. Program Routine Executed Each Spacecraft Revolution.