NASA CR-73377
Available to the Public

SUMMARY FINAL REPORT

MULTIPAC, A MULTIPLE POOL PROCESSOR AND COMPUTER FOR A SPACECRAFT CENTRAL DATA SYSTEM

By T. Baker
G. Cummings
R. South

October 1969

Prepared under Contract No. NAS2-3255 by
APPLIED RESEARCH LABORATORY
SYLVANIA ELECTRONIC SYSTEMS
An Operating Group of Sylvania Electric Products, Inc.
40 Sylvan Road, Waltham, Massachusetts 02154

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
AMES RESEARCH CENTER
MOFFETT FIELD, CALIFORNIA 94035

FOREWORD

The study described herein was done at the Applied Research Laboratory of Sylvania Electronic Systems, under NASA Contract NAS2-3255. The work was done under the direction of Mr. Richard O. Fimmel, Systems Engineering Division, NASA-Ames Research Center.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A/D	Analog-to-digital
CDS	Central data system
CMD	Command module
D/A	Digital-to-analog
IC	Integrated circuit
I/O	Input or output
LSI	Large-scale integration
LSIC	Large-scale integrated circuit
MOS	Metal oxide semiconductor
MULTIPAC	Multiple Pool Processor and Computer
OPC	Operation code
R/M	Register or memory
SC	Shift clock
WS	Word strobe
TM	Telemetry module

SUMMARY FINAL REPORT

MULTIPAC, A MULTIPLE POOL PROCESSOR AND COMPUTER FOR A SPACECRAFT CENTRAL DATA SYSTEM

By T. Baker
G. Cummings
R. South

SUMMARY

MULTIPAC is a computer designed especially for use as an "off-the-shelf" central data system for deep space probes. This computer has the unusual characteristic that it may be repaired during flight through the command and telemetry link by reprogramming around the failed unit. This reprogramming is possible through a computer organization that uses pools of identical modules which the program organizes into one or more computers. The interaction of these modules is dynamically controlled by the program and not hardware. In the event of a failure, new programs are entered which reorganize the central data system. The only effect of such reorganization is to reduce the total processing capability aboard the spacecraft. Consequently, some low priority process may have to be eliminated, but data taking and transmission may continue.

As an example of one MULTIPAC configuration, a 16-watt system, including 12,288 words of memory, can act as a sophisticated data management system for a space probe with about 200 science and engineering input lines and 200 output lines. This MULTIPAC system could simultaneously schedule sampling of the experiments, perform needed analog-to-digital conversions, reduce the data using histograms or other data reduction techniques, perform some data processing for the experiments such as digital filtering, and then format the data for transmission by the telemetry subsystem. In addition, the system has all the flexibility of a computer by allowing wide variations in formatting, sampling schedule, etc. These program variations can occur under program control or be completely changed later in the flight from the ground after analysis of the data received.

1.0 INTRODUCTION

This report describes MULTIPAC, a spacecraft central processor, the concept of which was derived from the first year of this study. (The result of the first year study is reported in the final report for that part of the contract.[1]) MULTIPAC has modular organization which permits reprogramming around failed modules. Machine reorganization may be accomplished by program changes to utilize surviving modules optimally, thus affecting a gradual degradation of processing capability as additional

modules fail in the course of a long mission. The overall reliability is such that the probability is very high that at least some minimum mode of operating the spacecraft can be sustained throughout very long missions.

The MULTIPAC system is intended to replace the current technique of designing a new central data system for each probe with a standard "off-the-shelf" central data system which is programmed with software to perform as a flexible data management system. Some variation of flight-to-flight requirements are expected to be made up by differences in the number of modules carried and also with the possibility of the addition of one or two special modules.

As an example of one MULTIPAC configuration, a 16-watt system, including 12,288 words of memory, can act as a sophisticated data management system for a space probe with about 200 science and engineering input lines and 200 output lines. This MULTIPAC system could simultaneously schedule sampling of the experiments, perform needed analog-to-digital conversions, reduce the data using histograms or other data reduction techniques, perform some data processing for the experiments such as digital filtering, and then format the data for transmission by the telemetry subsystem. In addition, the system has all the flexibility of a computer by allowing wide variations in formatting, sampling schedule, etc. These program variations can occur under program control or be completely changed later in the flight from the ground after analysis of the data received. In contrast, today's fixed format central data system simply performs a fixed schedule of sampling followed by one of a few fixed formatting routines. Processing of the data is not possible, and the scheduling and the formatting is primarily variable by scaling to the telemetry rate.

The data formatting and data reduction studies in the first phase of the contract highlighted the need for stored program computer concepts for the design of the Central Data System. A centralized computer for a central data system leads to the problem of how to prevent failures from aborting the entire mission. Reliability becomes even more important when we realize it was recommended in phase one that many additional tasks normally performed in each experiment be taken over by this centralized computer. The solution arrived at was a multiple pool processor and computer (MULTIPAC) made up of a number of modules of a few types tied together by the program. The remainder of this study was devoted to the design of this MULTIPAC system.

The MULTIPAC system, as finally developed, is shown in Figure 1. It's most important module, the Logic Unit, controls the actions of all other module types. Each logic unit, using a few registers and one or two memories, acts as a computer. A typical system will have three logic units and enough registers and memories to act as three simultaneous computers, each performing one-third the overall processing tasks. This typical system will consume only 16 watts and use 173 LSI logic circuits, 768 memory store LSI circuits and six integrated circuits for special purposes (e.g., oscillator). The number of different LSI circuit types is 13 or 17, depending on whether or not a large discretionary wiring LSI type is used on the logic unit.

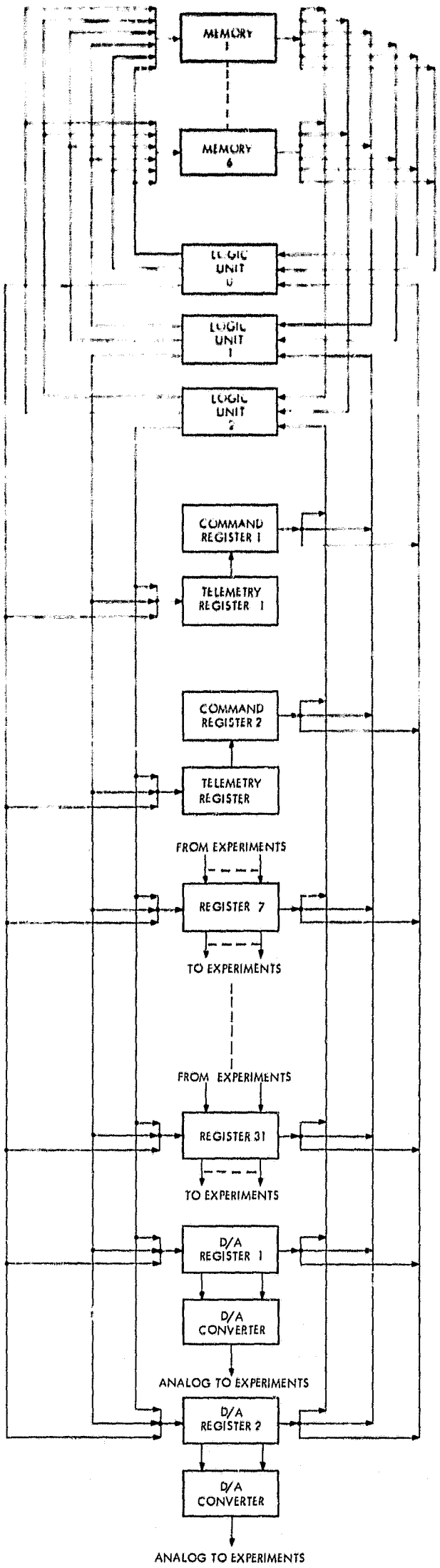


Figure 1. Block Diagram of Typical LSI MULTIPAC System

The registers are dual purpose. They act as index or scratch registers for normal processing and they are the input/output interface to the experiments. Each register contains a separate output buffer which holds output interface information. Thus, the register can be used as a scratch register without disturbing the output interface.

A few of the registers are special. One is used to produce an analog voltage of its digital contents to be used in analog-to-digital conversion for some of the experiments' signals. Another pair is used for the command and telemetry interface. The command register must have the ability to overtake the logic units by command in the event that the system does not respond properly to normal commands due to component failures.

The memories are passive devices which read or write data as commanded by a logic unit. The program counter is contained within the logic unit and instructions are requested from a memory selected as the program memory. Usually, a second memory is selected for data since the instruction rate for a separate data memory is faster than using the program memory for data.

All logic units can address all registers and all memories. In the event that two logic units address the same register or memory, any data transferring to this selected module will be ORed. The hardware normally associated with a multiprocessing system for handling such conflicts was purposely left out to keep the central data system small, light, and low power. Since all three processes are essentially working on three different tasks of the same problem and, therefore, know what the other processes are doing, these conflicts can be kept to an absolute minimum and those conflicts that do exist can be easily programmed around.

Overall, the system described here has a high likelihood of surviving a long mission with small minimal processing capability. A failure of a module will cause slightly degraded processing capability because some extra programming must be done to program around the failed module unless a space module exists. But even in the case of a failure of a logic unit, the worse that will happen is that the central data system will be able to perform only two-thirds of its processing load. For failure rates of 10^{-6} and 10^{-7} LSI circuits per LSIC-hour the probability of system survival for three years is 0.92 and 0.999 respectively. These failure rates are for a minimum operable configuration of one processor, 2048 words of memory and 83 percent of the input-output interface lines working properly. The processing capability of only one logic unit is more than enough to accomplish all the tasks done by the data system of Pioneer VI.

2.0 SYSTEM OPERATION

MULTIPAC is expandable and is comprised of seven module types as follows (see Figure 1 in Introduction):

Module	Min. No. Required	Number in Typical System	Number in Fully Expanded System
Logic Unit	1	3	5
Memory Unit	1	6	15
I/O Register	1	25	57
D/A Register	1	2	2
Command Unit	1	2	2
Telemetry Unit	1	2	2
Timing Generator	1	1	1

A processor is formed by software assignment of one logic unit, one or two memories, and several registers to operate in conjunction with one another. The I/O Registers serve the purpose of index or temporary storage registers and also provide I/O connections to the system. The most efficient use of memory units requires two per processor in order that program storage and data storage can be separate. The processor can also operate from a single memory unit but at a reduced computation rate. Communication with the command receiver and the telemetry transmitter are provided by the Command and Telemetry Units respectively, which are essentially specialized registers. Another specialized register is the D/A Register, which has a D/A ladder connected to its outputs so as to provide a reference signal for the successive approximation method of A/D conversion.

The MULTIPAC system is not limited to the module types listed above, but these are sufficient for our "typical" mission. Possible omissions are a magnetic tape (or other mass memory) interface, a real-time counter, a sample rate counter, and a television imaging system. Logic diagrams for a real-time counter and a sample rate counter are described in the Final Report.[2]

In the discussion which follows, the system is assumed to be configured with the quantity of modules listed in the table above under the heading "Number in Typical System."

2.1 Data Flow

Data flow takes place only under the control of one of the three logic units. Each of these communicate directly with six memories, 25 general purpose registers which also serve as I/O interfaces, two D/A registers, two command registers, and two telemetry registers.

Each memory is controlled by the logic unit addressing it. The logic unit can direct the memory to read and send the contents of a specified memory location to itself as either instructions or data, or to write the data which it supplies to the memory.

Each of the 25 general purpose registers has connections for 12 digital inputs and 12 digital outputs which may be accessed by I/O instructions. Thus, they provide 324 inputs (counting the D/A registers) and 300 outputs to the rest of the spacecraft and the instruments.

The D/A registers are very similar to the general purpose registers, having the same number of input channels, but the output channels are not present. Instead, a D/A ladder network is connected to provide an analog signal proportional to the arithmetic value of the register contents. This analog signal is used by the experiments as a reference voltage in the successive approximation A/D conversion process.

The telemetry and command registers share a common address. Instructions operating on such an address will connect one of the command registers to its input bus and/or one of the telemetry registers to its output bus.

The command module serves as the link between the command decoder and receiver and the MULTIPAC system. The module has three purposes: To transfer normal commands (e.g., turn on or off experiments, change mode), to allow special override commands to diagnose and reload new programs from the ground through the command link, and to allow loading of programs while on the ground before launch. The latter two use the ability of this module to have the instruction words it receives executed while inhibiting the normal program stream.

2.2 Transfer Timing

All data transfers between modules are serial. Synchronous machine timing is provided by two clock signals, the shift clock (SC) and the word strobe (WS). These are generated in the Timing Generator and distributed to all modules by triplicated signals driving majority voting gates at each module interface. Timing consists of 14 SC pulses followed by one WS pulse at equally spaced intervals of approximately 1 microsecond. The machine cycle is therefore about 15 microseconds.

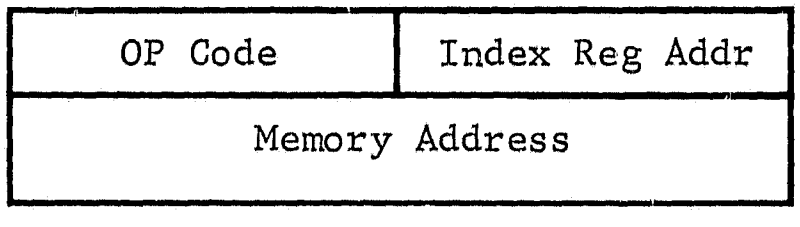
The actual 12-bit data transfer is preceded by the transfer of a 2-bit control code on the same line. It is by the transfer of this code that the logic unit controls the operations of the other modules with which it communicates.

2.3 Word Format

MULTIPAC uses a 12-bit word. The instruction format is as follows:

Single Word Instruction:

Double Word Instruction:



Data words are 12 bits in length and use two's complement arithmetic.

2.4 External Characteristics

2.4.1 Parts count.-- The general level of 100 gates and less than 50 pins per package was an assumption for the design. This level allows more than the selected vendor (Texas Instruments) to respond to the LSI circuit development program for this system. One LSIC where the use of Texas Instruments' capability for very large circuits could be used effectively is the control section of the Logic Unit. Two alternates to the partitioning of the Logic Unit are possible. Alternate 1 uses a large Texas Instruments' LSIC for all the control gating and alternate 2 uses 5 LSIC's for the same amount of logic to keep within the gate and pin limitation. Reliability estimates were made using alternate 2.

Table 1 lists all the LSI circuit types and their usage. Some (e.g., basic shift register) have large usage and others are used only two or three times. The total types needed are 16 (only 13 if the Texas Instruments' control circuit is used for the Logic Unit).

As much as possible, LSI circuits were reused rather than prolificate a new type. This is most apparent in the use of shift registers. The basic shift register without the parallel input gating could have been used in many places, but a new circuit type would have been required.

TABLE 1

QUANTITY OF CIRCUITS PER SYSTEM

Circuit	No. of Gates per LSIC	No. of Pins per LSIC	No. of LSIC's per Typ. Sys.	No. of LSIC's per Max. Sys.	Where Used
Basic Shift Register	96	42	56	107	Logic Unit, all Register Types, Memory
R/M Control	52	18	35	76	Memory, I/O Register, D/A Register, Telemetry Unit
Buffer Register	61	39	31	72	Memory, I/O Register
D/A Switches	8	18	2	2	D/A Register
TM Special	29	24	2	2	Telemetry Unit
CMD Control	115	27	2	2	Command Unit
Counter	34	8	3	3	Timing Generator
Memory Storage	1880 (MOS)	34	768	1920	Memory
Decoder	31	33	6	15	Memory
Bipolar-to-MOS Interface Circuits	14 (Special)	33	18	45	Memory
MOS-to-Bipolar Interface Circuits	12 (Special)	26	6	15	Memory
16-Way Switch	78	42	12	30	Logic Unit

TABLE 1.-- Continued

Circuit	No. of Gates per LSIC	No. of Pins per LSIC	No. of LSIC's per Typ. Sys.	No. of LSIC's per Max. Sys.	Where Used
Logic Unit Alternate 1:					
Complete Control	352	72	3	5	Logic Unit Alternate 1
Logic Unit Alternate 2:					
Control 1	83	34	3	5	Logic Unit Alternate 2
Control 2	88	40	3	5	Logic Unit Alternate 2
Control 3	83	36	3	5	Logic Unit Alternate 2
Control 4	33	40	3	5	Logic Unit Alternate 2
Control 5	65	61	3	5	Logic Unit Alternate 2
Integrated Circuits:					
Analog Amplifier	1	4	2	2	D/A Register
Oscillator	2	3	2	2	Timing Generator
Squaring Circuit	2	4	2	2	Timing Generator
Oscillator Switch	1	14	1	1	Timing Generator

2.4.2 Power consumption.-- Table 2 indicates the expected power consumption of about 16 watts for the typical system and about 32 watts for the fully expanded system. These figures are essentially dependent on two budgetary estimates: 1 milliwatt per logic gate and 10 milliwatts per interface circuit. The former figure is based upon the power consumption of the Fairchild LPDTμL logic used in the integrated circuit design and other low-power logic in the same general speed/power class. The latter estimate is based upon integrated circuit power levels generally and has yet to be verified by specific circuit design.

TABLE 2

ESTIMATED POWER CONSUMPTION

Typical System:

Module Type	No. of Logic Gates Per Module	No. of Modules Per System	No. of Logic Gates Per System
Logic Unit	1144	3	3432
Register	209	25	5225
Memory	240	6	1440
D/A Register	148	2	296
CMD Unit	307	2	614
TM Unit	273	2	446
Timing Generator	102	1	102
			11,755

Internal Power Budgets

Logic (1 mw/gate)	11.755 w
Oscillator and Squaring IC's	0.200 w
D/A Switches and Amplifiers	0.600 w
Memory quiescent power (100 nw/cell	0.015 w
Memory transient power	0.0XX w
Memory Interface circuits (10 mw/individual circuit)	3.240 w
Total ≈	15.8 watts

TABLE 2.-- Continued

Maximum System:

Module Type	No. of Logic Gates Per Module	No. of Modules Per System	No. of Logic Gates Per System
Logic Unit	1300	5	6500
Register	209	57	12369
Memory	240	15	3720
D/A Register	148	2	296
CMD Unit	307	2	614
TM Unit	273	2	446
Timing Generator	102	1	102
			23,371

Internal Power Budgets:

Logic (1 mw/gate)	23.371 w
Oscillator and Squaring IC's	0.200 w
D/A Switches and Amplifiers	0.600 w
Memory quiescent power (100 nw/cell)	0.035 w
Memory transient power	0.0XX w
Interface circuits and sence amplifiers (10 mw/individual circuit)	8.1 w
Total ≈	32.3 watts

2.4.3 Speed.-- The clock frequency of 1.0 MHz and the consequent instruction time of 15 microseconds are based on an anticipated propagation delay of about 50 nanoseconds for the LSI gates. The longest propagation path is 16 gate delays including the output delay of the transmitting flip-flop and the preset time of the flip-flop. At 50 nanoseconds per gate the signal requires 800 nanoseconds to propagate and has 200 nanoseconds to spare. This considers all gates to be identical. It may be feasible to include higher powered and, consequently, faster gates at critical points, which could further improve the delay margin.

2.4.4 Volume.-- The packaging of LSI circuits of this general size seems to require about four times the space of 14-lead flat packs. Therefore, using one-quarter the volumetric density (125 LSIC's per lb) as in the integrated circuit MULTIPAC, the volume can be estimated as follows:

$$\text{Typical System: } \frac{935 \text{ LSIC's}}{0.7 \text{ Spacd Utilization}} \times \frac{1 \text{ lb}}{125 \text{ LSIC's}} = 11 \text{ lbs}$$

$$\text{Maximum System: } \frac{2314 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ lb}}{125 \text{ LSIC's}} = 26 \text{ lbs}$$

2.4.5 Weight.-- Since the weight is largely a function of the packaging rather than of the circuit itself, it may be estimated similarly for one-quarter the density (5 LSIC's per cubic inch) of the IC model.

$$\text{Typical System: } \frac{935 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ in}^3}{5 \text{ LSIC's}} = 273 \text{ in}^3$$

$$\text{Maximum System: } \frac{2314 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ in}^3}{5 \text{ LSIC's}} = 668 \text{ in}^3$$

3.0 PROGRAMMING

3.1 Typical Data Reduction Techniques Possible with MULTIPAC

Data reduction programming will have to be done by the experimenter. To the experimenter various subroutines can be made available, such as histogram or fourier analysis. Data reduction techniques make some assumptions about the nature of the data. Consequently, it may be desirable to add data reduction techniques after the spaceprobe has gathered data. These programs can be checked out by the ground base computer and then transmitted to the spacecraft via the command link.

3.1.1 Histograms or quantiles.-- Histograms or quantiles take very little space for program storage, probably 100 to 300 words of program memory. Data storage, on the other hand, will most likely require 1000 words for each experimental line analyzed. Probability theory for normally distributed data and single quantiles state that the square of the mean deviation will be 1.57 divided by the number of samples, and for two quantiles, 0.767 divided by the number of samples. Thus, 1000 samples seems a reasonable amount to keep the error to a few percent. The determination of the optimal quantiles requires the 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.

3.1.2 Digital filters.-- Probably the best implementation of digital filters is to cascade recursive filter sections using different equations. Cascaded sections require the least accuracy of the data word and are the simplest to implement. The canonical form of difference equations is generally preferred in terms of ensuring against noise due

$$\text{Typical System: } \frac{935 \text{ LSIC's}}{0.7 \text{ Spacd Utilization}} \times \frac{1 \text{ lb}}{125 \text{ LSIC's}} = 11 \text{ lbs}$$

$$\text{Maximum System: } \frac{2314 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ lb}}{125 \text{ LSIC's}} = 26 \text{ lbs}$$

2.4.5 Weight.-- Since the weight is largely a function of the packaging rather than of the circuit itself, it may be estimated similarly for one-quarter the density (5 LSIC's per cubic inch) of the IC model.

$$\text{Typical System: } \frac{935 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ in}^3}{5 \text{ LSIC's}} = 273 \text{ in}^3$$

$$\text{Maximum System: } \frac{2314 \text{ LSIC's}}{0.7 \text{ Space Utilization}} \times \frac{1 \text{ in}^3}{5 \text{ LSIC's}} = 668 \text{ in}^3$$

3.0 PROGRAMMING

3.1 Typical Data Reduction Techniques Possible with MULTIPAC

Data reduction programming will have to be done by the experimenter. To the experimenter various subroutines can be made available, such as histogram or fourier analysis. Data reduction techniques make some assumptions about the nature of the data. Consequently, it may be desirable to add data reduction techniques after the spaceprobe has gathered data. These programs can be checked out by the ground base computer and then transmitted to the spacecraft via the command link.

3.1.1 Histograms or quantiles.-- Histograms or quantiles take very little space for program storage, probably 100 to 300 words of program memory. Data storage, on the other hand, will most likely require 1000 words for each experimental line analyzed. Probability theory for normally distributed data and single quantiles state that the square of the mean deviation will be 1.57 divided by the number of samples, and for two quantiles, 0.767 divided by the number of samples. Thus, 1000 samples seems a reasonable amount to keep the error to a few percent. The determination of the optimal quantiles requires the 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.

3.1.2 Digital filters.-- Probably the best implementation of digital filters is to cascade recursive filter sections using different equations. Cascaded sections require the least accuracy of the data word and are the simplest to implement. The canonical form of difference equations is generally preferred in terms of ensuring against noise due

to truncation and round off effects. Recursive filters require very little program storage and data storage. A very complicated cascaded filter can probably be implemented in a few hundred words of program storage and 10 to 20 data words.

3.1.3 Spectral analysis.-- The most generally useful program technique for spectral analysis is the fast fourier transform (FFT) which can be programmed in less than a thousand words of program space. The data space is a function of the size of the transform and is approximately twice the number of points in the transform. Half the data storage is for the data points themselves, a quarter for the cosine table and the remainder for miscellaneous constants. For example, a 512-point transform will take about a thousand words of data storage. A 512-point transform can be considered as 512 simple single-pole filters spread evenly across the bandwidth over a time span of 512 consecutive samples.

Reliable spectral estimates are possible only if the experimenter has a rough idea of the actual spectrum being estimated. Such knowledge will then enable an intelligent choice of the number of samples and the bandwidth of the bandpass filter preceding the transform. The sampling frequency must be high enough to minimize possible aliasing errors, a selection that demands the knowledge of the spectrum shape. Quite often the spacecraft instruments have a well-defined bandwidth so that the selection of the necessary sampling rate (Nyquist rate) is straightforward.

4.0 REPROGRAMMING AROUND FAILURES

The reliability of the MULTIPAC system is achieved through its ability to reprogram around failures. This section describes some of the techniques used to accomplish this reprogramming.

It does not seem feasible at this time to fly enough memory to perform all the diagnostics and automatic reprogramming. A more realistic approach is to diagnose the error through the command and telemetry links; to reassemble the program on a ground-based computer; and then transmit the new program to the spacecraft. In general, it will not be necessary to reprogram all of the memory.

A typical system will have three logic units and enough memories to have three complete processors operating simultaneously. A processor is defined here to mean enough programmable units to program one or more of the CDS tasks. The most likely division of work into these three processors will be: one responsible for inputting and outputting data; another responsible for telemetry buffering; and the third responsible for data reduction processing.

The discussion will be divided into failures of various units (e.g., registers, logic units, etc.).

4.1 Command Override Procedure

The uplink commands are 16 bits in length. The last four bits (the four most significant bits) are used to distinguish between normal commands and command override commands. When these 4 bits are all 0, a normal command is assumed. The other 15 combinations of these 4 bits are used for command override. Each of these 15 address 15 different locations in the MULTIPAC system: 3 to each of 5 logic units. When less than 5 logic units are in the MULTIPAC system, the unused commands will act like command override without performing any command override function. In other words, these unused commands will be treated like override commands, but will be sent to nowhere.

The 3 addresses within a logic unit addressed by the command override feature are the 2 accumulators and the instruction shift register. A command sent to the instruction shift register will override any other instruction entering the instruction shift register and this new instruction will be performed as if it came from the program memory.

When overtaking the MULTIPAC system, the first procedure, normally, is to turn off all logic units. A logic unit is turned off with an override command instruction to set the program memory page to "0". Since there is no program memory whose address is "0", this will effectively send zeros continuously to the instruction shift register. Zeros are treated as NOP instructions by the MULTIPAC logic unit.

The procedure to turn off all logic units is to send the instruction SPMP (set program memory page) "0" to each of the logic units in turn. SPMP "0" will set the program memory page of each logic unit to "0". After all logic units have been disabled, the procedure is either to enable some program stored in a known good memory or else to bootstrap in a program from the ground into a memory.

To start the program at some program memory N at address A, first the command to load Accumulator 1 with the Address A is sent. The instructions JMPR ACC1 are then sent to the instruction shift register. This jump through register instruction will set the program counter to the address in Accumulator 1. Since the program memory page is still "0", the program counter will not increment and the address A will remain in the program counter until the program memory page changes. Next, the instruction SPMP N is sent to the instruction shift register, and when this instruction is performed, the program memory page will switch to the requested memory and the instructions will begin to be performed from the address stored in the program counter.

To bootstrap in a program into memory, the data memory switch of a logic unit is set to the proper memory with an SDMP instruction and then the following three commands can be used to load each word into memory. First, the address is loaded into Accumulator 1, and second, the data is sent to Accumulator 2. The third command is the instruction STA2 indexed by Accumulator 1. Commands from command override portion of the second unit are sent to the instruction shift register as one word followed by

many words of all zeros. Thus, if a first word of a two-word instruction is sent to the instruction shift register, the second word will be all zeros. The STA2 indexed by Accumulator 1 instruction will have an address of "0" indexed by Accumulator 1, or, in other words, the address will be that of Accumulator 1. This will send Accumulator 2 to this address in memory. This procedure can be repeated over and over again until enough words are scored in memory to load programs with normal command words. Before this loading program is entered, the diagnostic procedures of the next section should be followed to determine what modules are working properly.

4.2 Reprogramming Methods

Reprogramming may be accomplished from the ground by simply sending up a new section of code. Given any failure, the next most efficient program usually involves a complete reorganization. This entails approximately 1000 words being sent up. At 10 bits per second, this requires less than one-half hour to transmit. Total reprogramming (~6000 words) would take only 3 hours to accomplish. On the other hand, reprogramming from within simply cannot cope with this kind of reorganization. Specifically, any reprogramming requires changing code, hence a reassembly.

A relabeling process is possible for register failures only. Any failure other than a register will require a total reorganization of some routine to prevent inefficiency of running time and memory storage utilization. There seems to be no real use of reprogramming which can be accomplished on-board so that the command link transmission speed is the factor which determines the amount of reprogramming done in a given time.

A very critical area is the problem of determining "what" has failed whenever it becomes apparent that "something" has failed. A great deal of the failure detection is going to occur on the ground. It does not seem practical to put sophisticated detection programs on-board, since these generally take a great amount of running time and memory. During the period when the spacecraft is not transmitting to ground, a reasonable amount of failure detecting can be run, but this does not detect failures that occur during the transmission period.

4.3 Ground Software

Ground-based software must emphasize the ability to diagnose and reprogram in the shortest possible time. This primarily implies a large collection of preassembled routines using all the combinations of available hardware. This is, of course, impractical. What really is needed is: 1) a diagnostic generator which uses failure history to reduce its output, and 2) various organizations determined by sets of available units of the running software which are abstract in the actual units utilized. This

latter means assuming some subset of units being available and writing the most efficient code for the situation. Parameters to each such encoding would be the actual unit numbers (switch addresses). This ability implies an assembler of only moderate complexity with relatively simple macro features.

No time should be wasted on any kind of compiler. Code simply must be as efficient as possible, which means only machine language coding.

A computer must be available at all times for reassembling programs. Some on-board problems such as failures, will be solved only by having a programmer generate more code in real time. If possible, the computer should take care of transmission to the on-board system. Extremely desirable would be a diagnostic generator (as above) which checks out the system automatically when needed. There is probably no need of human analysis most of the time. This is more true of diagnostics than reprogramming.

5.0 CONCLUSIONS AND FUTURE RECOMMENDATIONS

This contract began as a study of data formatting and data system organizations for lightweight deep space probes. The first year of this study, which has been reported previously, recommended that the central data system use stored program computer concepts, that data formatting should be very flexible, and that data reduction algorithms should be used whenever possible. The initial contract was extended to develop the multiple pooled central data system concept described in this report.

MULTIPAC is a central data system using stored program computer concepts which would give future spacecraft extensive data processing capability for variable data formatting, sampling and converting analog information from experiments, and performing data reduction on experimental data to improve information transfer on a limited telemetry bandwidth. An organization consisting of pools of modules organized by the program is used to achieve an extremely high reliability for extended deep space probes. In the event of a failure, this multiple pooled organization allows reprogramming around the failed modules, permitting the surviving modules to be utilized optimally.

In addition to the ability to recover from failures, this multiple pool organization replaces the current technique of designing a new data system for each probe, with a standard "off the shelf" central data system, which is programmed by software to perform as a flexible data management system. One typical organization was used as an example throughout this report. This typical MULTIPAC configuration, a 16-watt system, including 12,288 words of memory, can handle about 200 science and engineering input lines and 200 output lines. This typical system could simultaneously schedule sampling of the experiments, perform needed analog-to-digital conversions, reduce data using histograms or other data reduction techniques, and then format the data for transmission by the telemetry subsystem. Since a computer organization is used, wide variations in formatting, sampling schedule, and other data management tasks are easily accomodated. These changes can be made later in flight from the ground after the data has been analyzed. It is at this time that data reduction techniques are quite powerful, since after the flight has been in progress for some time, enough may be known about the data to effectively perform data reductions on the raw data. In addition, if an experiment has failed, the part of the data format transmitted to earth from that failed experiment can be used by other experiments.

The above system has an extremely high probability of surviving 36 months (the longest mission considered). However, with the very pessimistic failure rate of 10^{-5} LSI circuits per LSIC-hour (1 percent per 1000 hours), the probability of system survival is 0.0001. For more realistic failure rates of 10^{-6} or 10^{-7}, the corresponding figures are 0.92 and 0.999, respectively. These failure rates are for a minimum operable configuration of one processor, 2048 words of memory and 83 percent of the input-output interface lines working properly. This configuration is more than enough to perform scheduling and sampling of the science and

5.0 CONCLUSIONS AND FUTURE RECOMMENDATIONS

This contract began as a study of data formatting and data system organizations for lightweight deep space probes. The first year of this study, which has been reported previously, recommended that the central data system use stored program computer concepts, that data formatting should be very flexible, and that data reduction algorithms should be used whenever possible. The initial contract was extended to develop the multiple pooled central data system concept described in this report.

MULTIPAC is a central data system using stored program computer concepts which would give future spacecraft extensive data processing capability for variable data formatting, sampling and converting analog information from experiments, and performing data reduction on experimental data to improve information transfer on a limited telemetry bandwidth. An organization consisting of pools of modules organized by the program is used to achieve an extremely high reliability for extended deep space probes. In the event of a failure, this multiple pooled organization allows reprogramming around the failed modules, permitting the surviving modules to be utilized optimally.

In addition to the ability to recover from failures, this multiple pool organization replaces the current technique of designing a new data system for each probe, with a standard "off the shelf" central data system, which is programmed by software to perform as a flexible data management system. One typical organization was used as an example throughout this report. This typical MULTIPAC configuration, a 16-watt system, including 12,288 words of memory, can handle about 200 science and engineering input lines and 200 output lines. This typical system could simultaneously schedule sampling of the experiments, perform needed analog-to-digital conversions, reduce data using histograms or other data reduction techniques, and then format the data for transmission by the telemetry subsystem. Since a computer organization is used, wide variations in formatting, sampling schedule, and other data management tasks are easily accomodated. These changes can be made later in flight from the ground after the data has been analyzed. It is at this time that data reduction techniques are quite powerful, since after the flight has been in progress for some time, enough may be known about the data to effectively perform data reductions on the raw data. In addition, if an experiment has failed, the part of the data format transmitted to earth from that failed experiment can be used by other experiments.

The above system has an extremely high probability of surviving 36 months (the longest mission considered). However, with the very pessimistic failure rate of 10^{-5} LSI circuits per LSIC-hour (1 percent per 1000 hours), the probability of system survival is 0.0001. For more realistic failure rates of 10^{-6} or 10^{-7}, the corresponding figures are 0.92 and 0.999, respectively. These failure rates are for a minimum operable configuration of one processor, 2048 words of memory and 83 percent of the input-output interface lines working properly. This configuration is more than enough to perform scheduling and sampling of the science and

engineering lines and data formatting which is the capability of present-day fixed format central data systems.

The present design is expandable to five processors and 32,768 words of storage. Each of these processors will act as a computer with an instruction rate of 15 microseconds. These limits are arbitrary and simple changes to the system design can be made if greater memory storage and/or computers are needed. This fully expanded MULTIPAC system will require 32 watts of power, and will handle an extensive input-output interface to the experiments, many times greater than that of present day space probes. The generality of the design is such that it can easily handle input/output devices not included in this design with existing modules or with the addition of new modules. These new modules are easy to interface, require very few interconnections, and may be directly addressable by the programs.

As this design moves into hardware implementation, it is probable that some changes will be made due to further analyses of the system's requirements. Before final implementation, it is recommended that a typical mission be programmed, and that diagnostics be written to determine whether they ought to be transmitted from the ground (the most likely) or stored in memory. These programming tasks may result in recommendations for some changes in the overall design. It is expected that such design changes will be limited to change of instruction repertoire, in which case only the design of the logic unit need be affected. The register and memory can remain exactly the same, and the LSIC's of these modules may be released before programming is done. Programming a typical mission will give a closer estimate of memory requirements and the amount of data reduction processing capability available to the experimenter.

These programming tasks may be accomplished while a breadboard is being built. Breadboarding costs about the same (assuming integrated circuits are used in place of the LSI circuits) as performing a computer simulation and is a far more accurate representation of the final system.

Without a mass storage device aboard the spacecraft, reprogramming from the ground will require a command link capability of at least 10 bits per second. Even at that rate it is possible that a failure could put a spacecraft out of contact with earth for one to two hours. If this is an unacceptable delay, it may be desirable to add a simple commutator under control of the command decoder which will bypass the central data system. This bypass, which would be used while reprogramming, could simply transmit the raw data with frame syncs and parity in a fixed sampling sequence.

In conclusion, a very low power, extremely flexible central data system has been described which can be reprogrammed from the ground to either change its characteristics or to program around failed components. This design can be used for all (or most) future deep space probes replacing the present data systems which are specifically designed for each flight.

engineering lines and data formatting which is the capability of present-day fixed format central data systems.

The present design is expandable to five processors and 32,768 words of storage. Each of these processors will act as a computer with an instruction rate of 15 microseconds. These limits are arbitrary and simple changes to the system design can be made if greater memory storage and/or computers are needed. This fully expanded MULTIPAC system will require 32 watts of power, and will handle an extensive input-output interface to the experiments, many times greater than that of present day space probes. The generality of the design is such that it can easily handle input/output devices not included in this design with existing modules or with the addition of new modules. These new modules are easy to interface, require very few interconnections, and may be directly addressable by the programs.

As this design moves into hardware implementation, it is probable that some changes will be made due to further analyses of the system's requirements. Before final implementation, it is recommended that a typical mission be programmed, and that diagnostics be written to determine whether they ought to be transmitted from the ground (the most likely) or stored in memory. These programming tasks may result in recommendations for some changes in the overall design. It is expected that such design changes will be limited to change of instruction repertoire, in which case only the design of the logic unit need be affected. The register and memory can remain exactly the same, and the LSIC's of these modules may be released before programming is done. Programming a typical mission will give a closer estimate of memory requirements and the amount of data reduction processing capability available to the experimenter.

These programming tasks may be accomplished while a breadboard is being built. Breadboarding costs about the same (assuming integrated circuits are used in place of the LSI circuits) as performing a computer simulation and is a far more accurate representation of the final system.

Without a mass storage device aboard the spacecraft, reprogramming from the ground will require a command link capability of at least 10 bits per second. Even at that rate it is possible that a failure could put a spacecraft out of contact with earth for one to two hours. If this is an unacceptable delay, it may be desirable to add a simple commutator under control of the command decoder which will bypass the central data system. This bypass, which would be used while reprogramming, could simply transmit the raw data with frame syncs and parity in a fixed sampling sequence.

In conclusion, a very low power, extremely flexible central data system has been described which can be reprogrammed from the ground to either change its characteristics or to program around failed components. This design can be used for all (or most) future deep space probes replacing the present data systems which are specifically designed for each flight.

REFERENCES

1. A Study to Determine an Efficient Data Format and Data System for a Lightweight, Deep Space Probe. Final Report No. NASA 73211, Contract No. NAS2-3255; February 1968.

2. MULTIPAC, A Multiple Pool Processor and Computer for a Spacecraft Central Data System. Final Report No. NASA CR-73348, Contract No. NAS2-3255; October 1969.