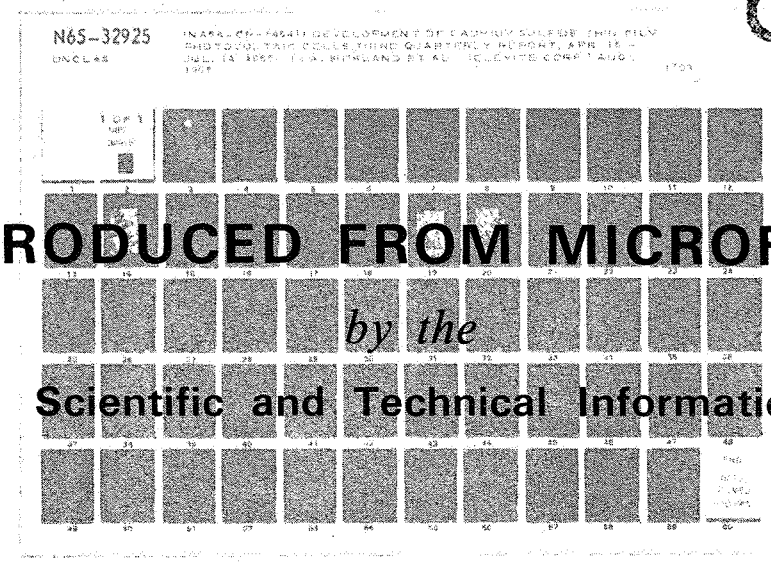
a NASA facsimile reproduction

OF

N69-10832

CASE FILE
COPY



REPRODUCED FROM MICROFICHE

by the

NASA Scientific and Technical Information Facility

Multiprocessing Techniques for Unmanned Multifunctional Satellites

FACILITY FORM 602
N69-10832 (ACCESSION NUMBER)
(THRU)
224 (PAGES)
1 (CODE)
CR-86113 (NASA CR OR TMX OR AD NUMBER)
31 (CATEGORY)

IBM Final Report

FSC 68-1002/2754

MULTIPROCESSING TECHNIQUES FOR UNMANNED MULTIFUNCTIONAL SATELLITES

PREPARED FOR

NASA ELECTRONICS RESEARCH CENTER
Cambridge, Massachusetts

6 September 1968

PREPARED UNDER CONTRACT NO.

NAS 12-507

Federal Systems Division
INTERNATIONAL BUSINESS MACHINES CORPORATION
Gaithersburg, Maryland

PRECEDING PAGE BLANK NOT FILMED.

FOREWORD

This Final Report describes the results of a study conducted under NASA contract NAS 12-507, "Multiprocessing for Multifunctional Satellites." It was performed by International Business Machines Corporation, Federal Systems Division, Communications and Engineering Sciences Center, in Gaithersburg, Maryland. The work was administered under the direction of the National Aeronautics and Space Administration, Electronics Research Center, Computer Research Laboratory, Cambridge, Massachusetts. Mr. G. Y. Wang and Mr. J. Roy provided the NASA Technical Direction of this study.

The investigators for this study were Dr. R. A. D'Antonio (Project Manager) and Mr. H. F. Najjar. Mr. G. Blum assisted during the simulation effort. Dr. H. L. Blasbalg provided overall direction.

CONTENTS

		Page
	INTRODUCTION	xv
Section 1	MULTIFUNCTIONAL SATELLITE SYSTEMS	1-1
Section 2	COMMUNICATIONS SUBSYSTEM	2-1
2.1	Pseudo-Noise (PN) Modulation	2-3
2.2	Adaptive PN Demodulator	2-5
2.3	Error Control	2-10
2.3.1	Principles of Error Control Subsystem	2-10
2.3.2	Variable Block Length, Fixed Redundancy	2-12
2.3.3	Fixed Block Length, Variable Redundancy	2-13
2.3.4	Variable Use of Redundancy	2-13
2.4	Data Compaction	2-14
2.4.1	Non-Adaptive Compression	2-14
2.4.2	Adaptive Compression	2-16
2.4.3	Adaptive Coding Penalty	2-19
2.5	Adaptive Multiplexing and Demultiplexing	2-19
2.5.1	Adaptive Multiplexer	2-21
2.5.2	Adaptive Demultiplexing	2-21
2.5.3	Decision Rules	2-25
2.6	Computer Control of the Antenna Systems	2-25
2.6.1	Antenna Configuration	2-26
2.6.2	Phase Shifters and Control Signals	2-30
2.6.3	Tracking Capability	2-30
2.6.4	Spin Stabilized Antennas	2-33
2.6.5	DIMUS (Digital Multibeam Steering)	2-35
2.7	Computer Requirements	2-35
Section 3	DATA COLLECTION AND SENSOR SUBSYSTEM	3-1
3.1	Link Capacities	3-1
3.2	On-Board Data Collection	3-4
3.2.1	Experiment Data Loads for AAP Missions	3-4
3.2.2	Jupiter Orbiter	3-8
3.3	Micrometeoroids	3-14
3.3.1	Background	3-14
3.3.2	Present Knowledge	3-15
3.3.3	Future Micrometeoroid Studies	3-16
3.3.4	Micrometeoroid Sensors	3-18

		Page
3.4	Solar Flares	3-23
3.4.1	Background	3-23
3.4.2	Present Knowledge	3-24
3.4.3	Future Solar Flare Studies	3-25
3.4.4	Solar Flare Sensors	3-26
3.5	Computer Requirements	3-28
Section 4	NAVIGATION AND GUIDANCE, CHECKOUT AND MONITORING, AND ENERGY MANAGEMENT SUBSYSTEMS	4-1
4.1	Guidance and Navigation Subsystem	4-1
4.2	Checkout Monitoring	4-2
4.3	Energy Management	4-3
Section 5	STATISTICAL ANALYSIS	5-1
5.1	Mean Value Calculation and Removal	5-2
5.2	Probability Density and Distribution Calculation	5-2
5.3	Auto and Cross-Correlation Calculation	5-3
5.4	Power Spectral Density	5-4
Section 6	COMPUTER SPECIFICATIONS	6-1
6.1	Data and Instruction Formats	6-2
6.2	Addressing	6-3
6.3	Processor Organization	6-3
6.4	Main Storage	6-4
6.5	Input/Output	6-5
6.6	Computer Packaging	6-5
6.7	Software Support	6-5
6.8	Instruction List	6-6
Section 7	MULTIPROCESSING AND SWITCHING SYSTEMS	7-1
7.1	Multiprocessing Concept	7-1
7.2	Multiprocessing State-of-the-Art	7-1
7.3	A Multiprocessing Configuration	7-3
7.4	Switching	7-8
7.4.1	Orthogonality Concept	7-8
7.4.2	Time Division—Switching and Orthogonal Addressing	7-9
7.4.3	Space Division Switching	7-17
Section 8	AN ADAPTIVE ANALOG-TO-DIGITAL CONVERSION SYSTEM	8-1
8.1	Adaptive A/D Converter	8-2
8.1.1	Gain Control	8-4
8.1.2	Sampling Rate Control	8-4
8.1.3	Quantization Control	8-4

		Page
8.2	Quantization	8-5
8.3	Quantizing Theorem	8-8
8.4	Results for Gaussian Signals	8-10
8.5	Statistical Analysis	8-10
Section 9	MICROPROGRAMMING APPLIED TO MULTIPROCESSING	9-1
9.1	Applications to Multiprocessing	9-1
9.1.1	Diagnostic and Corrective Programming	9-3
9.1.2	Instruction Modification	9-3
9.1.3	Modification of I/O Units	9-3
9.1.4	Instrumentation	9-4
9.1.5	Read-Only Memories	9-4
Section 10	ERROR DETECTION AND DIAGNOSTIC TECHNIQUES	10-1
10.1	Hardware	10-1
10.2	Software	10-3
10.2.1	Building Block Approach	10-4
10.2.2	Go/No-Go Approach	10-4
10.2.3	Multiple Clue Approach	10-5
10.2.4	Error-Environment Recording	10-5
10.2.5	Summary	10-5
10.2.6	Future Requirements	10-6
Section 11	ADVANCED TECHNOLOGIES	11-1
11.1	Advanced Logic Circuits	11-1
11.2	Large Scale Integration	11-3
11.2.1	Reliability	11-3
11.2.2	Limitations	11-5
11.2.3	Logic Design	11-5
11.2.4	Partitioning	11-6
11.2.5	Cost	11-8
11.2.6	Application	11-10
Section 12	ADVANCED MEMORY DEVICES	12-1
12.1	Advances in Memory Technology	12-1
12.1.1	Monolithic Ferrite Memory	12-2
12.1.2	Flat-Film Memory	12-3
12.1.3	Plated-Wire Memory	12-6
12.2	Aerospace Memory Requirements and Recommendations	12-12
Section 13	RELIABILITY AND AVAILABILITY CONSIDERATIONS	13-1
Section 14	SIMULATION AND EVALUATION	14-1
14.1	General-Purpose Systems Simulator (GPSS)	14-1
14.2	General Model for Simulation	14-1
14.3	Performance	14-5

		Page
Appendix A	GPSS ENTITIES	A-1
Appendix B	SIMULATION FLOW DIAGRAM FOR INITIAL SYSTEM	B-1
Appendix C	LARGE SYSTEM SIMULATION PROGRAM	C-1
Appendix D	MULTIPROCESSING IN MULTIFUNCTIONAL SATELLITES	D-1
	REFERENCES	R-1
	BIBLIOGRAPHY	Bi-1

ILLUSTRATIONS

Figure		Page
1-1	Satellite Communications System	1-2
1-2	Basic On-Board System	1-4
2-1	Communications Subsystem	2-2
2-2	Transmitter Modulation and Sending End Logic	2-6
2-3	Synchronization and Data-Extraction Subsystem	2-7
2-4	PN Synchronization Signals	2-8
2-5	RF Phase-Locked Receiver	2-9
2-6	Comparison of Various Techniques (Theoretical and Experimental Results)	2-11
2-7	Predictive Compression	2-15
2-8	Shannon-Fano Compressor	2-15
2-9	Adaptive Coder for Quasistationary Message	2-17
2-10	Excess Bits When Code is Derived from Probabilities P, and the True Probabilities are λ	2-20
2-11	The Adaptive Multiplexer	2-22
2-12	The Spacecraft and Earth Control	2-23
2-13	The Adaptive Demultiplexer	2-24
2-14	Antenna Array Geometry	2-27
2-15	Steerable Arrays and Coverage Range	2-29
2-16	Control System for Scanning in Two Angular Coordinates	2-31

Figure		Page
2-17	A Single Dimension Tracking System	2-32
2-18	Antenna Switching Arrangement for Spin Stabilized Vehicles	2-33
2-19	DIMUS Approach to Beamforming Using a Planar Array	2-36
3-1	Micrometeoroid Flux Near the Earth	3-17
3-2	A Multimode Micrometeoroid Detector with Variable Sensitivity and Data Rate (Ancillary Buffers, Blocks, and Reset Signals Not Shown)	3-22
7-1	A Centralized System	7-2
7-2	A Decentralized System	7-4
7-3	Decentralized Configuration of an Associative Logic Parallel System	7-5
7-4	A Typical Decentralized Multiprocessing System	7-6
7-5	Digital Matched Filter	7-11
7-6	Mixing Operation	7-15
7-7	Data Flow Between the CPUs and the Memory Modules	7-16
7-8	Overall System	7-18
7-9	Multiprocessing Configuration Utilizing Space Division Switching	7-19
7-10	Switching Matrix	7-21
8-1	An Adaptive Analog-to-Digital Conversion System	8-3
8-2	Quantizer Input (w(x)) and Output (w'(x)) Distributions	8-7
8-3	w'(x) From w(x) Through Convolution and Sampling	8-9
8-4	(a) Results of Convolution in the α Plane (b) α Spectra of w'(x)	8-11
9-1	Basic Microprogrammed Control Elements	9-2

Figure		Page
10-1	A Self-Checking Decoder	10-2
12-1	Flat-Film Memory	12-4
12-2	Measured Critical NDRO Curve	12-7
12-3	Plated-Wire Memory	12-9
12-4	Plated-Wire Operation	12-11
13-1	A Representative Decentralized Multiprocessing System	13-2
14-1	General Multiprocessing System Model for Simulation	14-2
14-2	Average Utilization of CPUs Using Six Memory Modules	14-6
14-3	Average Queue Time for CPU Using Six Memory Modules	14-7
14-4	Average Utilization of CPUs Using 10 Memory Modules	14-9
14-5	Average Queue Time for CPUs Using 10 Memory Modules	14-10
14-6	Average Utilization of CPUs Using 30 Memory Modules	14-12
14-7	Average Queue for CPU Usage Using 30 Memory Modules	14-13
14-8	Frequency Function of Data Blocks as a Function of the System Processing Time for 30 Memory Units and up to 30 CPU Units	14-15

TABLES

Table		Page
3-1	Estimated Data Rates and Channel Capacities	3-3
3-2	Complement an AAP Meteorological Mission Experiment	3-5
3-3	Solar Astronomy Experiment Data Estimates	3-7
3-4	Earth Survey Experiment Data	3-8
3-5	List of Instruments and Experiments for the Jupiter Orbiter Mission	3-9
4-1	System Power Requirements	4-6
6-1	Instruction List	6-7
7-1	Data and Code Mixing Results	7-13

INTRODUCTION

This report concludes a 9-month study to develop new multiprocessing concepts oriented toward the computational needs of future, long-lived, multifunctional space satellites.

The study's major objective was to investigate the subsystems on board a long-lived multifunctional satellite and determine how an on-board multiprocessor could service each subsystem and at the same time provide increased reliability, flexibility, and graceful degradation for the mission. The study was carried out in three phases.

In the first phase, the computational requirements of a representative future multifunctional satellite were defined. The computational requirements for servicing the various subsystems linked to the on-board computer system were specified. The subsystems that were investigated are: checkout, communication and telemetry, control and stabilization, energy management, sensors and data handling. The work accomplished in Phase 1 is described in sections 1 through 6.

In Section 1, the overall multisatellite-to-earth communications system is defined. The system uses a network consisting of multifunctional satellites in nonstationary orbits and relay satellites in stationary orbits. The mission objectives for an unmanned multifunctional satellite are also described. The satellite will be able to function in three areas: as a scientific vehicle for scientific experiments, as part of a weather prediction and data collection system, and as a component of an operational worldwide communications system. Although we have selected a particular mission and analyzed the multiprocessor requirements within the framework of that mission, the concepts developed are completely general. The reliability and flexibility obtained through a reconfigurable multiprocessing system can be useful in all types of spacecraft, both manned and unmanned.

The communications subsystem is defined in Section 2. A multiprocessor on board the spacecraft will perform or control all of the subfunctions related to communications, e.g., digital antenna multiple-beamforming, pseudo-noise modem processing, estimation of channel signal-to-noise ratio, electronics spin stabilization, error control, multiplexing, and data compaction. Each is thoroughly analyzed, and the computer requirements for tying these functions into a multiprocessing system are estimated. Pseudo-noise modulation is selected as the optimum modulation scheme and the principles of this technique are described. The cyclic algebraic codes are selected for the error control system, and various adaptive techniques are described, all using cyclic coding. The use of the computer to perform adaptive data compaction and multiplexing, and computer-controlled antenna beamforming and steering are discussed.

Section 3 describes the data collection and sensor subsystems. In these subsystems, the objective was to estimate the amount of on-board processing that will be required on any single mission during the next decade. Three representative missions were chosen and the amount of data that would be generated by the on-board experiments was estimated. The total data requirements of several past missions were obtained for comparison. Some of the experiments are described in detail because it was felt that the familiarity with the sensor outputs would be helpful in establishing the type of on-board processing required. The sensors used in two important areas of space research, micrometeorites and solar flares, are also described. They are typical of the type required to implement the proposed mission. The computational requirements for on-board preprocessing of these sensor outputs are also given.

Navigation and guidance, checkout and monitoring, and energy management are discussed in Section 4. All of these functions can be implemented by programming a general-purpose digital computer.

In Section 5, some basic statistical measures, which will be used often in the subsystems, are defined in a form suitable for computer implementation. In Section 6, a set of computer specifications, based on the requirements discussed in sections 1 through 5, is given. These specifications will adequately

meet the needs of any single subsystem. These specifications served as a guide for the second phase of the study and aided in estimating the computing capability required by each subsystem.

From the analysis of each of the subsystems on the spacecraft, we found that most of the functions on board could be effectively tied into an on-board multiprocessor computing system. In many cases much more sophisticated and intelligent instrumentation was developed by taking advantage of the on-board capability.

In Phase 2, one or more advanced multiprocessing configurations which satisfy the computational requirements of Phase 1 were formulated. Reliability, flexibility, adaptability, hardware realization, and the hierarchy of mission control had to be satisfied. Sections 7 through 12 describe the work accomplished during this phase of the study.

In Section 7, several multiprocessing configurations are defined in detail and described. These configurations use space, time, and code-division switching principles to solve the problem of interconnection and reconfiguration of the various computing, memory and I/O modules that make up the elements of the on-board processing system. The orthogonality concept is defined and applied to space and time-division switching systems.

In Section 8, a new adaptive analog-to-digital converter system that uses the on-board multiprocessing system is described. This system adapts itself to changes in the input signal by varying the sampling rate and quantization precision based on the statistics of the signal. Since the analog-to-digital conversion process is fundamental on board any spacecraft, it will be widely applicable.

Section 9 outlines some applications of microprogramming to satellite multiprocessing systems. Methods are described for applying these techniques to diagnosing and correcting machine failures; modifying instruction sets; modifying I/O unit operation; and automating the on-board sensors and instrumentation. Microprogramming (or microcontrol) is a relatively new concept that has not been exploited for space applications.

In Section 10, several error detection and diagnostic techniques are discussed. These techniques, when used in conjunction with reconfigurable computer systems, ensure even higher systems reliability. Hardware and software methods are described, and an example of a self-checking computer element is given. The future requirements of these types of systems are discussed.

Section 11 describes some of the advanced logic techniques such as Large-Scale Integration, Bipolar and Metal Oxide Silicon integrated circuits. All of these technologies have the advantage of low power consumption and high density packaging, both vital to spaceborne computer systems. Several advanced memory devices are described in Section 12. Special emphasis is placed on the plated-wire memories since they seem to offer most of the features required by spaceborne computers. Recommendations on aerospace memory requirements are made.

This study brought to light several interesting points. From the standpoint of reliability and flexibility, the polymorphic processing configurations, such as those described and analyzed in this report, are excellent. In addition, these configurations possess a parallel processing and reconfiguration capability that is important for the spacecraft of the future.

Finally, in Phase 3, the multiprocessing configuration developed during Phase 2 was evaluated. This evaluation delineated both the features and technical limitations of the configuration. A simulation program was used to analyze the performance of the configuration.

Section 13 contains an analysis of the reliability and availability aspects of the multiprocessing systems described in Section 7. Section 14 considers the simulation and evaluation phase of this study. A general-purpose system simulator, used to analyze, evaluate and compare the various multiprocessing systems, is described. Flow diagrams of the simulation programs and the results are given in the appendices and in this section, respectively. The behavior of the multiprocessor as elements fail is investigated, and the graceful degradation capability of these systems is demonstrated.

In the simulation effort we were looking for relationships within the multiprocessor structure, the entire processor and the utilization of the moving processor modules. Some of these relationships were found. Appendix D is an abstract of a paper based on this work.

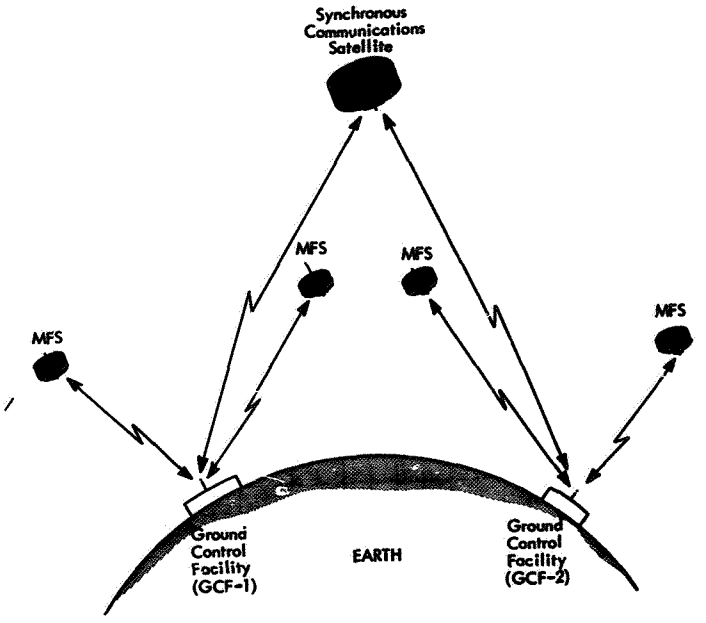
Based on this study the next logical step is to apply these concepts in the design of an applications-oriented satellite, the NIMBUS-E spacecraft. This study is now in progress. We feel that the modular multiprocessor approach developed and analyzed during this study is the answer to the reliability, flexibility, and parallel processing requirements of the data handling systems for these spacecraft.

Section 1

MULTIFUNCTIONAL SATELLITE SYSTEMS

Present satellite systems generate a large volume of data which is transmitted to earth, although only a small fraction is actually processed. With an increase in the sophistication of future satellite systems, there will be an increase in the amount of data collected, which will overload not only the communication and telemetry equipment, but also the data processing capabilities of the ground-based computers. One answer to this problem is a flexible, reliable multifunctional (MF) satellite system monitored and controlled by a ground control facility (GCF)—linked together by a synchronous communication satellite. Figure 1-1 shows the system we have selected using a single synchronous communications satellite and two GCFs. The GCF terminals are in contact with one another via the synchronous satellite repeater and each MF satellite is in contact with one or the other GCF terminals. Thus, a communication path can be established between any two MF satellites either via a single GCF station or via two GCF stations and the synchronous satellite repeater. In principle, this technique can be extended to cover the entire earth by providing three or more synchronous satellites in communication with many earth stations. In such a system, the entire MF satellite system complex can be used to optimize the collection of useful data, thereby reducing the load on the ground-based processing equipment, as well as on the communications and telemetry links.

A satellite continues to perform its prescribed and limited functions repeatedly for long periods of time only because it cannot be used for anything else. On the other hand, a multifunctional satellite can be reconfigured to perform other useful data collection and processing functions once a mission is completed.



Note: Multifunctional satellites are not synchronous.

Figure 1-1. Satellite Communications System

The goal of a satellite system, operational or scientific, is to transmit useful data to earth. Useful data is generally extracted from the raw sensor data by some form of processing. The raw sensor data can be transmitted down and processed in the ground-based processors, or it can be preprocessed in the vehicle so that only the important characteristics and parameters are transmitted down. In the latter system processing equipment is located in the vehicle; while in the former, processing equipment is located on the ground, provided a high-capacity communications link is available. Thus, a tradeoff exists between the capacity of the communications system and the computing power of the satellite-processing equipment.

A multifunctional satellite, if it is to be used efficiently, should have access to a large ground control facility from almost all points in the orbit, and vice versa. In present unmanned satellites, the data is stored in the vehicle and transmitted down during the relatively short period that the satellite is within view of the ground station. For each set of measurements, one would expect an optimum time at which to retrieve the data. A multifunctional satellite system can achieve efficient operation at all times if its communication and telemetry subsystem is connected to the GCF via a synchronous communication repeater satellite network. In fact, if there are multiple satellites in orbit, all can be connected via GCF stations and synchronous satellite repeaters. In this manner, the GCF can monitor the overall space system and optimize the combined missions.

Figure 1-2 shows the on-board system. Note that the multiprocessing system, defined in Section 7 of this report, interfaces with each of the on-board subsystems.

An earth-based ground control facility in communication with the satellite is also shown. This facility is composed of a large general-purpose computer and is used for ground control and backup. This computer will be programmed to request and accept data from the MF satellite. It will also be used to back up the on-board multiprocessor in case of failure.

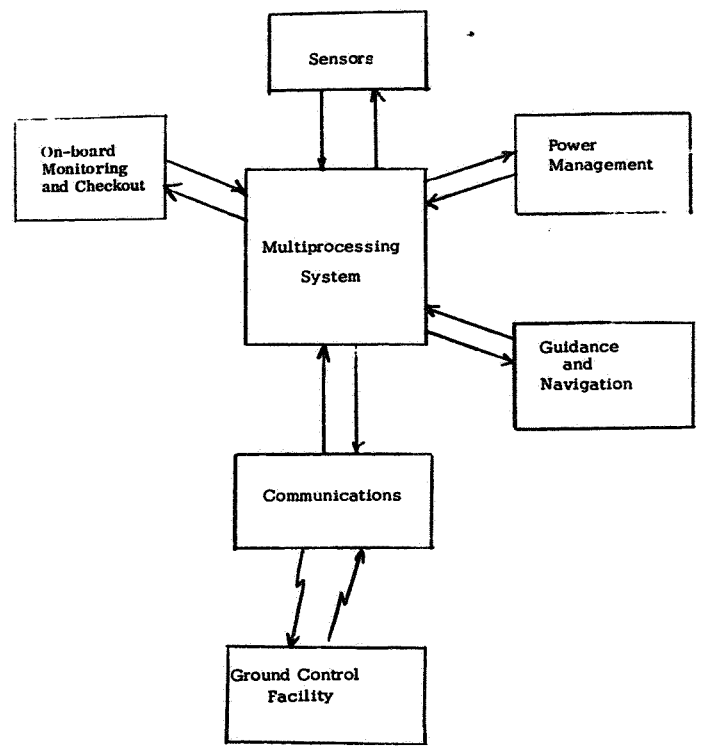


Figure 1-2. Basic On-Board System

The philosophy behind the approach given in Figure 1-2 is this: one of the main purposes of an on-board processing system is to reduce the amount of data transferred between the satellite and the earth. This is necessary because the amount of raw data gathered by a multipurpose satellite could far exceed the capacity of the satellite-earth communication link. Thus, a considerable amount of on-board preprocessing is required to reduce this raw data to fit the satellite-earth channel capacity. The key functions that any multipurpose satellite must perform are as follows: (1) communications, (2) power management, (3) guidance and navigation, (4) sensors, and (5) checkout and monitoring.

The Monitoring and Checkout subsystem periodically scans all of the subsystems and tests for failures or weaknesses. Both software and hardware techniques are used here where we have been concentrating on locating errors instantly before they have a chance to propagate.

The Sensor subsystem is the source of most of the data generated on board the spacecraft. In this report the processors assigned to cover this system control the analog-to-digital conversion equipment determining the sample rates and the number of bits/sample to use. The processors also monitor the data for decision making purposes and perform any necessary preprocessing

The Power Management subsystem allocates power to those subsystems in use, thus conserving the available power and ensuring its efficient use.

The processors assigned to the Guidance and Navigation subsystem execute the guidance algorithms and accept inputs from the earth stations.

Finally, the Communication processors implement or support error control, modulation, channel monitoring, and the multiplexing equipment. With a processor, these functions can be made to adapt to a changing situation. Note that the ability to execute programs in parallel is another obvious requirement of the on-board processor, since all five of the subsystems can operate at once. Also, each subsystem can be decomposed into many functions that also must act in parallel.

Section 2

COMMUNICATIONS SUBSYSTEM

A diagram of the communications subsystem is given in Figure 2-1. The various functions that this subsystem performs are all controlled by a multiprocessor.

The multiprocessor will perform or control all of the subfunctions related to communications. Some of these are: digital antenna multiple-beamforming, pseudo-noise modem processing, estimation of channel signal-to-noise ratio, electronic spin stabilization, error control, multiplexing and data compaction. A thorough understanding of each of these subfunctions is essential before one can tie these functions into a multiprocessing system and estimate the computer requirements.

This system uses pseudo-noise (PN) modulation. JPL has used pseudo-noise modulation successfully for interplanetary telemetry. This form of modulation, because of efficient power utilization and its inherent ability to effectively reject all forms of interference, makes it a prime candidate for use in multifunctional satellite systems. Subsection 2.2 shows how the mod-demod subsystem can be controlled by a processor in an adaptive communication system.

Since many satellites are likely to be in orbit and it is desirable to control them to optimize the missions, the communications problem from satellite to earth can be difficult. It is essential to understand the overall communications problem particularly since the ground control facility is certain to play a significant role in optimizing the space system configuration. One or more complex control stations can have the satellites in view for long periods of time if one or more synchronous communication satellites are used to repeat the transmitted

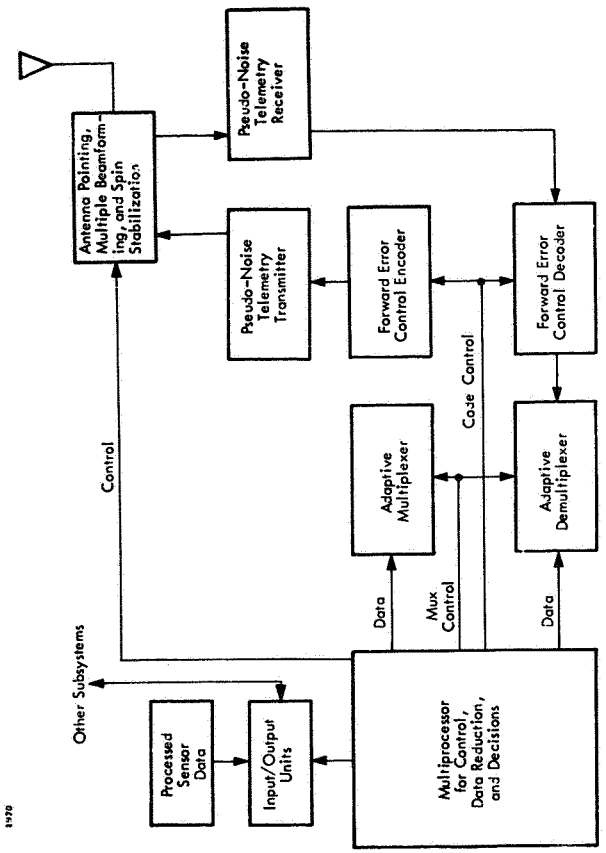


Figure 2-1. Communications Subsystem

messages down to earth; of course, the control signals from the ground will also be sent via communication satellites. In satellite communications, this is known as "the multiple-access problem" (i.e., many stations have access to a common repeater satellite). Pseudo-noise modulation techniques have been useful for this type of link because of the inherent ability of these signals to withstand severe amplitude distortion and all kinds of interference.

As stated, the ground control center monitoring the performance of a multitude of satellites in orbit can be used to make the overall data collection system more efficient. It can also be used to control the communications load transferred through the communication satellite link. As an example, the ground processor, being aware of the computing load of the satellites in orbit, can select the appropriate satellites at the appropriate time for unloading the processed data.

A large processing capability in orbit implies that only the essential data will be transmitted to Earth; as a result, this data must be received reliably. Efficient coding against channel disturbances is the insertion of a controlled amount of redundancy into a message to combat errors introduced by the medium. Adaptive coding matches the code to the type of channel disturbances, thereby optimizing performance. The coding and decoding operations can certainly be performed by a processor and once again this is a well defined problem mathematically. Error control is discussed in subsection 2.3 of this report.

2.1 PSEUDO-NOISE (PN) MODULATION

Satellite telemetry requires efficient utilization of the on-board power. Every db saved by using an efficient modulation scheme is a db saved in the on-board power.

An efficient modulation scheme for digital telemetry requires accurate synchronization. In a Gaussian-noise channel, the most efficient means of establishing synchronization is to use the information contained in the modulated signal. That is, it is essential to put all the power in the message and synchronization signal sidebands. Power splitting between message and sync is inefficient.

The required RF synchronization information can be extracted from a double-sideband, suppressed-carrier signal. However, in such a modulation scheme, bit, word, frame, etc., synchronization must be drived from the data transitions and special code words within the message format. When the data transitions are used, the timing jitter necessarily depends on the time distribution of the transitions, which may be a problem, particularly at slow data rates. A more efficient way of acquiring and maintaining synchronization is to combine modulo-2, the digital message, with a wideband, maximal length, pseudo-noise sequence from which all timing can be derived accurately. Here, the power is put into the message and sync signal sidebands as required. This efficiency is achieved at the expense of RF bandwidth. At first, this may appear as a severe penalty. However, careful study will show that by using spectrum-spreading, the power density (watts/cps) received on the ground can be maintained at a level low enough not to interfere with existing earth based requirements. If B is the RF bandwidth of the PN signal and B_0 the bandwidth of a conventional telemetry signal, then the interference caused by the PN signal in the band B_0 is simply B/B_0. If $B = 50 \times 10^6$ cps and $B_0 = 50 \times 10^3$ cps, then for the same received power, the spread-spectrum signal is 30 db below the narrowband signal in the band B_0. Equally important, the narrowband signal is suppressed 30 db in the PN receiver. The wideband telemetry system to be described next makes efficient use of satellite on-board power and at the same time minimizes the interference with conventional equipment. All future space telemetry systems should be designed with this in mind. A boundary should also be set on the power density received on the ground, since this is the only important measure of interference. This will limit the number of users of a common frequency band. Frequency allocation makes inefficient use of the overall spectrum, since, once allocated, the band is no longer available even when the channel is inactive. Clearly, this is a wasteful way of using the available spectrum, particularly for space telemetry.

Finally, since the PN signal rejects other interfering signals, several such signals can share a common broad band. This type of multiplexing is called "Code-Division Multiplexing" (CDM). If the RF bandwidth is approximately four

times the ratio of received signal power to noise power density, the CDM performance is, for all practical purposes, thermal-noise limited and not mutual-interference limited; hence, the overall behavior is essentially equivalent to, say, Time-Division Multiplexing" (TDM).

Figure 2-2 presents the PN modulator. Note that the clock, $2f_s$, is combined with PN and data as shown which, in turn, bi-phase modulates the RF carrier. In this manner, data and timing are combined so that full power is placed in message and sync sidebands.

2.2 ADAPTIVE PN DEMODULATOR

Figure 2-3 shows the PN (baseband) demodulator. Here, it is assumed that the RF carrier was extracted and used to demodulate PN coherently. The acquisition and tracking characteristics are shown in Figure 2-4. Note that these characteristics are precise; i.e., the timing accuracy is maintained well within a PN bit. The VCO is swept until PN and PN* are in phase. When this instant is detected, the search sweep is stopped and the loop locks up, since it is within the pull-in range. All timing is now available, and the message data can be recovered.

The purpose of having two arms in the baseband loop is to remove the data modulation from the clock. Without this operation, lock-up cannot be achieved. The channel throughput monitor selects the filter which matches the bandwidth of the existing data rate.

Figure 2-5 shows the RF carrier extraction and tracking circuit. When the PN reference is in phase with the incoming PN signal, the resulting bandwidth is equal to the message band. The IF filters are therefore matched to the data rate prior to the squaring operation, which extracts twice the carrier frequency. This frequency has a $(0, \pi)$ uncertainty, which is removed by sending a test signal of known phase. The recovered carrier is injected coherently into the received signal, which in turn yields the baseband PN signal as shown in the figure.

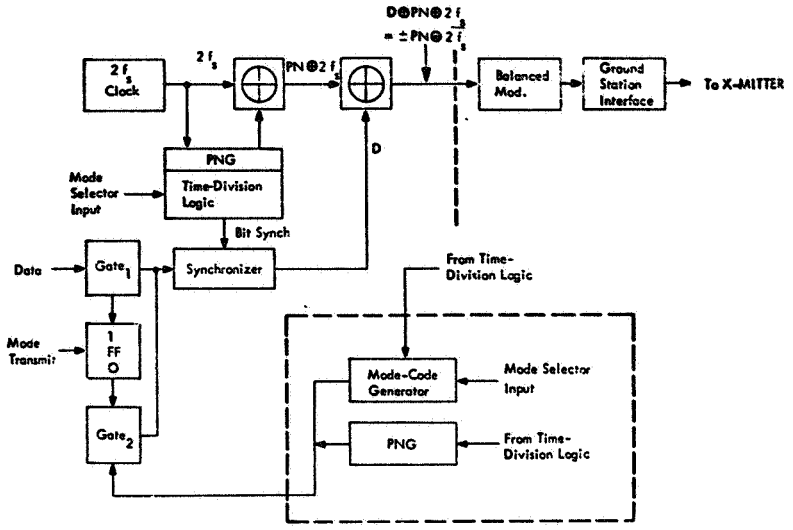


Figure 2-2. Transmitter Modulation and Sending End Logic

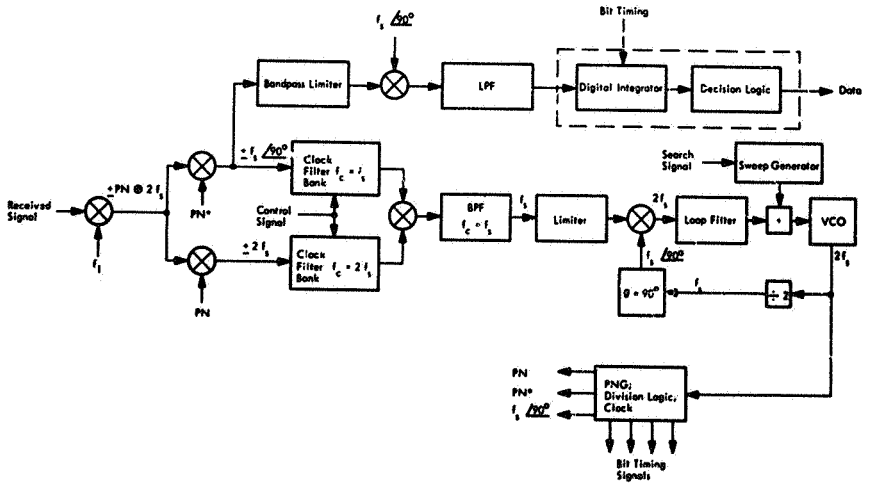


Figure 2-3. Synchronization and Data-Extraction Subsystem

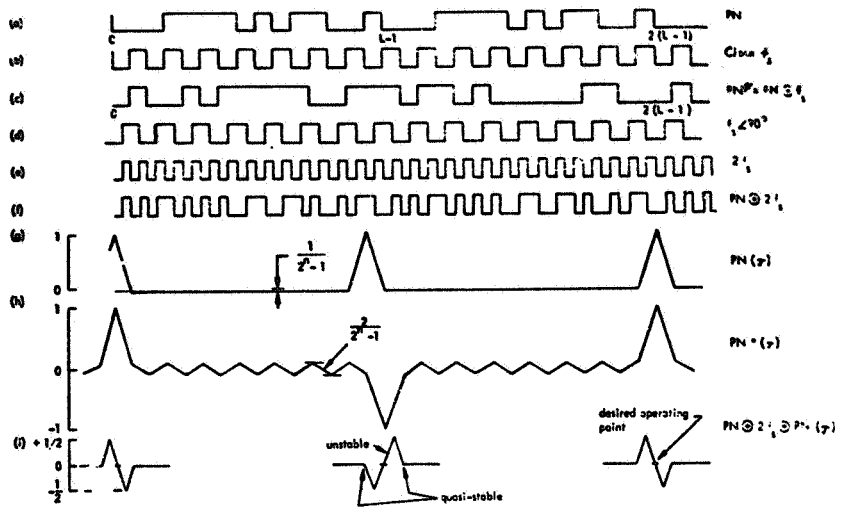


Figure 2-4. PN Synchronization Signals

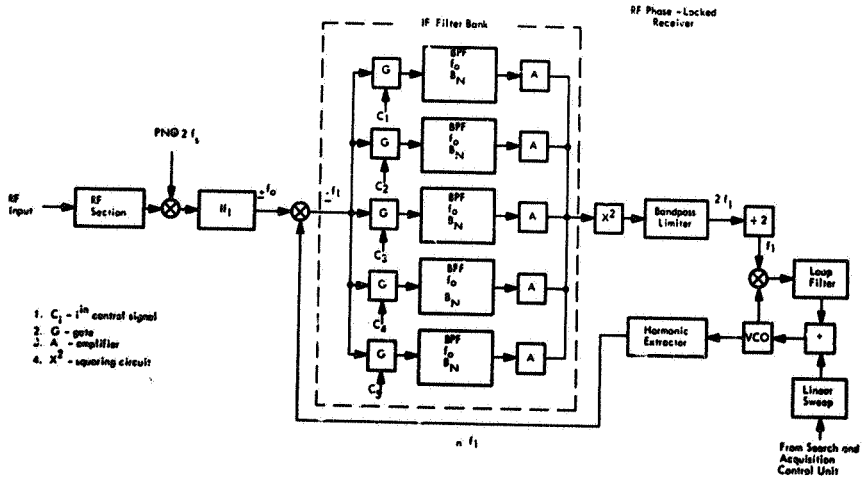


Figure 2-5. RF Phase-Locked Receiver

Figure 2-6 shows the performance of several digital modulation schemes, two of which are experimental and the others theoretical. The modem discussed here has a performance shown by curve number one which falls with 1/2 db of theoretical curve number three.

2.3 ERROR CONTROL

The inherent redundancy in a message prevents catastrophic degradation caused by channel errors. The message becomes more vulnerable to severe degradation caused by errors if more redundancy is removed, say by compression. The channel errors can be controlled, and hence, the fidelity, by inserting a controlled amount of redundancy in the form of coding. It is desirable either to adapt the codes to match the channel characteristics or to use codes that are efficient against a number of disturbances.

Coding requires a digital data transmission system. In addition, to code and decode economically, the data should be multiplexed (see Figure 2-1) into a single bit stream prior to modulation. In the latter case, only a single encoder-decoder combination is required. Thus, the coding requirement eliminates the use of analog modulation techniques and asynchronous multiplexing on economical grounds.

2.3.1 Principles of Error Control Subsystem

A digital data transmission link in a data processing system typically is more prone to error than other data handling media. Accordingly, all such links are protected by error-control subsystems.

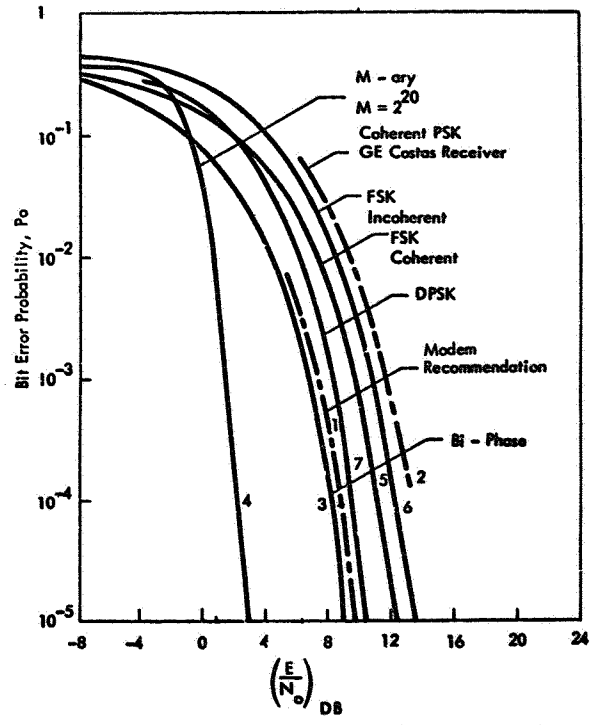
Advanced space systems will require advanced concepts, such as adaptive error control. This subsection considers the problems encountered in implementing three basic adaptive error control techniques:

a. Variable block length, fixed redundancy

b. Fixed block length, variable redundancy

c. Variable use of fixed redundancy with a fixed block length.

1975



NOTE: Dashed lines indicate experimental systems (measured results).

Figure 2-6. Comparison of Various Techniques (Theoretical and Experimental Results)

These techniques will be discussed with respect to feasibility of encoding, decoding, and maintaining synchronization. For the first two systems, it will be assumed that the control logic needed to maintain both encoder and decoder in the same mode is provided by the multiprocessor control computer as shown in Figure 2-1.

2.3.2 Variable Block Length, Fixed Redundancy

Designing a variable-block-length encoder and decoder by modifying a system designed to handle the largest desired block length is relatively simple. In a standard encoder, the redundancy is calculated over a number of information bits determined by sensing a value in a counter. To decrease this number, it is sufficient to incorporate controls so that the encoder will sense a specified smaller number in the counter when a mode change is made and return to sensing the original value when the mode is changed back again. Any combination of bit rate, block size, and coding structure constitutes a "mode" of operation. The problem in the decoder is only slightly more complex. Again, it is necessary to change the value sensed in a counter to indicate the length of the block and thereby indicate when the checkword or syndrome has been constructed. In all systems, except for a few error-detection-only systems, it is also necessary to store the information bits in the decoder until the checkword has been constructed and then retrieve the bits in the proper order. This requires modifying the controls on the storage element used. If the storage is a core memory, this is trivial. It is necessary to change only the sensing of a maximum address before resetting the address register to zero.

Synchronization for this technique presents some difficulties unless the timing of the mode changes at both the encoder and decoder can be precisely coordinated. One method for obtaining initial synchronization is to use the encoded structure of the message. The advantages of this technique are that it does not take additional channel space away from information bits, and whenever it is not received correctly the first time, it can be detected on the second, third, or subsequent transmissions. However, a resynchronization is more difficult.

The most desirable technique still appears to be use of the coded structure of the message, eliminating as much of the problem as possible by coordinating the time of mode change in the encoder and decoder.

2.3.3 Fixed Block Length, Variable Redundancy

A fixed-block-length system will be simpler to implement than a variable-block-length system. None of its problems will be as difficult as the synchronization problem described in the previous section. In fact, in a fixed-block-length system, there is no synchronization problem except for a startup procedure which is required even for a non-adaptive system.

The encoder and decoder for a fixed-block-length, variable redundancy system must handle a different number of information bits in each mode. Thus, they include the same problems with the same solutions described in the subsection on variable block length, fixed redundancy. In addition, the registers for constructing the redundancy in the encoder and the checkword in the decoder must be modified for each mode change. Since the capability to implement the mode with the most redundancy must be included, this becomes a simple change when using polynomial coding. It is merely a matter of changing feedback paths in a shift register with exclusive-or circuits and the gating to control which feedback path should be used. Modification of the feedback paths can easily be controlled by a multiprocessor computer.

2.3.4 Variable Use of Redundancy

A system incorporating variable use of fixed redundancy in a fixed block length is not faced with the synchronization problems described in the subsection on variable block length, fixed redundancy or with the variable storage and feedback problems of encoding and decoding. It must, however, include the correction circuitry for two types of correction, a complexity comparable to the other two systems. This system, however, is not faced with the problems of coordinating mode changes between the encoder and decoder, since the encoder operation remains constant.

All of the above techniques are readily implemented via cyclic codes as described by Peterson[1]. The computer requirements for implementing the coding techniques described above will be based on the use of cyclic codes.

2.4 DATA COMPACTION

This section develops information preserving (IP) compression and entropy reducing (ER) techniques. The latter involve on-board preprocessing and represent the largest potential payoff. ER operations will reduce the communication and telemetry channel requirements and, equally important, substantially reduce the vast amount of raw data received on the ground.

2.4.1 Non-Adaptive Compression[2]

A predictive compressor removes the redundancy in a message by exploiting the probability constraints within a message ensemble. The output sequence, encoded into fewer bits than the input, is completely random, i.e., all the redundancy has been removed. At the receiver, the rule for reconstructing the message is known; thus, the message can be decoded. In practice, it is essential to insert a controlled amount of redundancy in the form of error control to protect the message against channel errors which otherwise would cause catastrophic degradation.

Figure 2-7 shows a predictive compression which is useful when strong correlations between adjacent L-bit sequences exist. The present and previous L-bit sequences are compared in a modulo-2 adder and the resultant sequence is fed into a run-length encoder. If there are many bits in agreement, long strings of zeros will be generated which can be compressed by means of run-length coding. At the receiver, an exact inverse exists for reconstructing the message sequence.

Figure 2-8 uses Shannon-Fano coding to achieve compression. The sequences are divided into two groups each having equal probability. If the observed sequence falls into group one, the binary number one is generated; if in group two, binary zero is generated. If the decision is, say, $d_1 = 0$, then the second group of

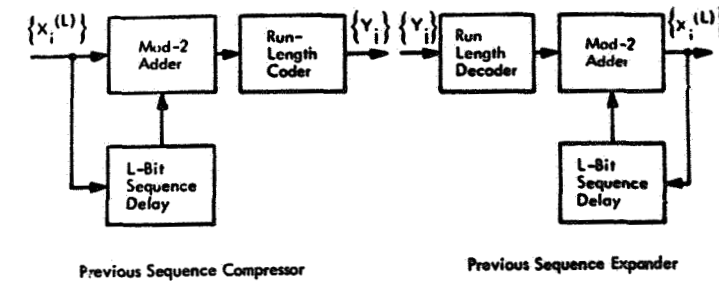


Figure 2-7. Predictive Compression

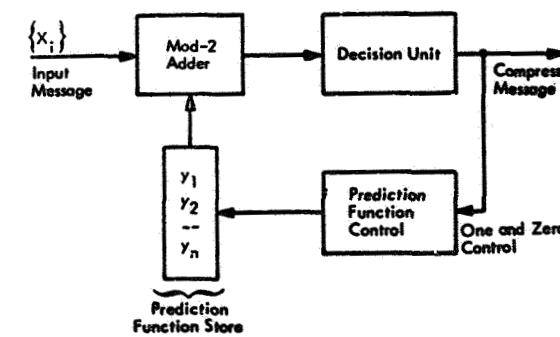


Figure 2-8. Shannon-Fano Compressor

sequences is divided into two groups of equal probability, and the procedure is repeated. If the decision is $d_2 = 1$, the binary number one is generated. This procedure is repeated until the sequence is finally classified. The coding for this case is, therefore, 01-------.

The code length in Shannon-Fano coding is variable but unique in that flags for word identification are unnecessary. The high probability sequences are assigned short codes and the low probability sequences use long codes such that the average code length is

$$H = -\sum_i p_i \log p_i \tag{2-1}$$

where (p_i) represents the message sequence probabilities.

A major difficulty with techniques of this type is that the higher-order message statistics must be known to achieve efficient compression. However, with an on-board processor, these statistics can be obtained from the data.

2.4.2 Adaptive Compression

Adaptive compression is a compromise between exact coding where the message statistics are known and straight transmission. The message statistics are measured and used to encode the subsequent message interval. While this procedure is in progress, the efficiency of the coding presently in use is monitored and new statistics are measured. If the compression efficiency falls below a given level, the newly measured statistics are used, i.e., the coding is updated. The adaptive procedure is effective provided the message statistics change slowly with time. As long as the rule for adapting is known at the receiver, the message can be reconstructed.

Figure 2-9 shows an adaptive compressor. The probability distribution analyzer measures the statistics (λ_i) during some prescribed time interval.

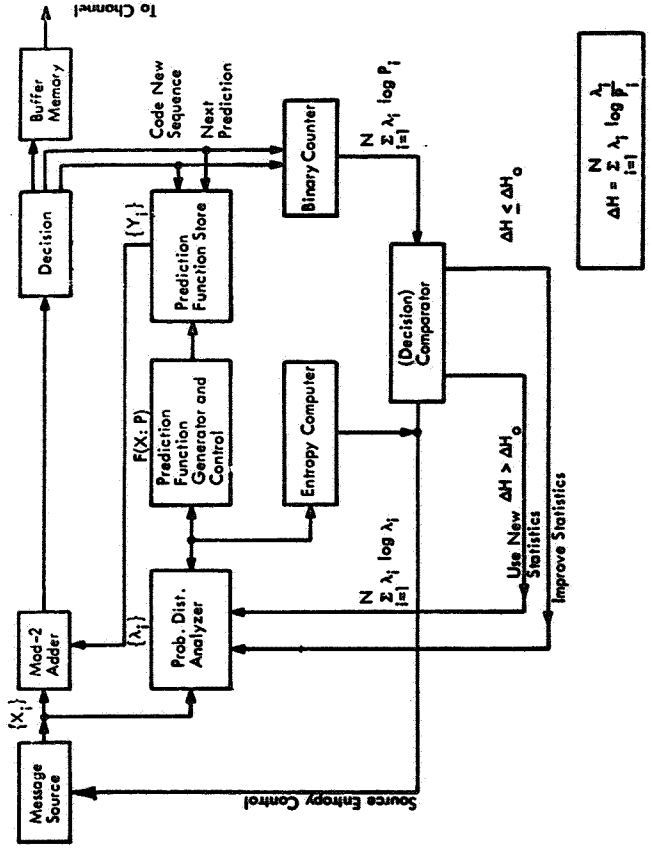


Figure 2-9. Adaptive Coder for Quasistationary Message

After measurement we call these (p_i). A Shannon-Fano prediction function (described previously) is generated and stored. The next message interval is encoded in accordance with the stored prediction function, and the coding efficiency is monitored.

If (p_i) is used and (λ_i) represents the newly measured statistics, then the average code length is

$$H_p = -\sum_{i=1}^{N} \lambda_i \log p_i \tag{2-2}$$

On the other hand if (λ_i) were used to encode the message, then the average code length would be

$$H_\lambda = -\sum_{i=1}^{N} \lambda_i \log \lambda_i \tag{2-3}$$

The figure of merit which is used to determine the coding effectiveness is

$$\Delta H = \sum_{i=1}^{N} \lambda_i \log \frac{\lambda_i}{p_i} \leq \Delta H_0 \tag{2-4}$$

where $\Delta H \geq 0$. The function ΔH gives the excess number of bits transmitted when (p) is used and (λ_i) represents the true statistics. If $\Delta H \leq \Delta H_0$, the coding continues with (p) and the statistics are improved; if $\Delta H > \Delta H_0$, the (λ_i) statistics are used to generate a new prediction function.

In all compression techniques, buffering is required since the input bit rate is variable while the output is uniform. To prevent buffer overflow and thus

catastrophic degradation, it is preferable to degrade the message fidelity gracefully and in a controlled manner. This can usually be accomplished by reducing the number of message bits per sample.

2.4.3 Adaptive Coding Penalty

Erroneous statistics used for coding can lead to message expansion, i.e., the encoded message contains more bits than the original message. This is shown for a binary message in Figure 2-10. When $\lambda = 0.5$, the most efficient coding transmits the message directly. However, if the message is encoded in accordance with $p = 0.1$, there is a 60 percent excess in the number of bits which can be translated into a 60 percent increase in channel bandwidth. This simple example shows the transmission rate penalty for coding in accordance with statistics which do not represent those of the source.

2.5 ADAPTIVE MULTIPLEXING AND DEMULTIPLEXING

It is very likely that situations will occur when the sampling rate of a particular sensor must be increased due to a sudden increase in data activity. The following are examples of sudden bursts of activity:

a. Passing through the auroral zone

b. Passing through the magnetospheric boundary

c. Solar radio bursts

d. Magnetic storms.

During these periods it will be necessary to change the multiplexing format in order to transmit this high density data to earth. What is required then is a multiplexer that can adapt to the sensor output rates. We have selected an adaptive multiplexer that will be controlled by the on-board multiprocessor. The computer will also make the decision, based upon measurements of the input data, to change the sampling rate of a particular sensor.

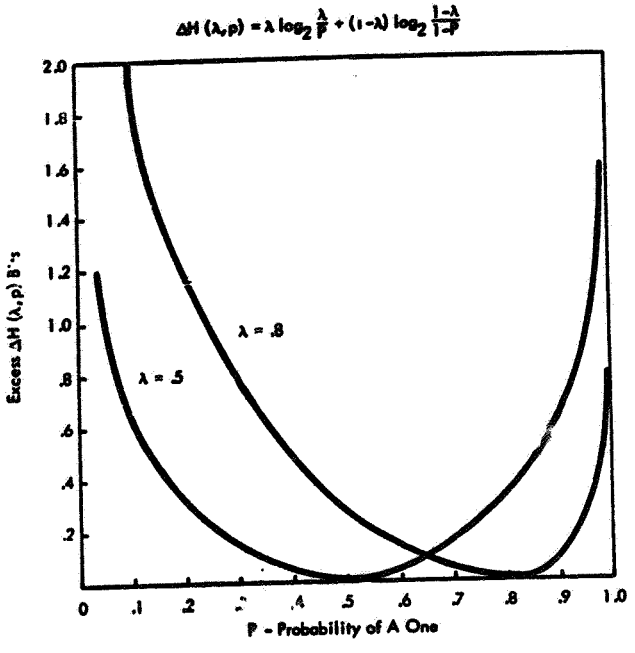


Figure 2-10. Excess Bits When Code is Derived from Probabilities P, and the True Probabilities Are λ

Figure 2-11 shows the overall adaptive telemetry system. At the transmitter there are three subsystems: multiplexer, processor, and pseudo-noise modem.

2.5.1 Adaptive Multiplexer[3]

Figure 2-12 presents the adaptive multiplexer. The sensor array feeds a conventional time division multiplexer which is under program control. The processor monitors the sensor outputs θ and the channel throughput H, and, for specified priority matrix P, computes the sensor scanning and control program $D(\theta, H, P)$. This program is fed into a memory driven by a clock derived from the PN modem. The PN modem clock rate is consistent with the channel throughput H.

The multiplexed message is fed into an A-D converter also under control of $D(\theta, H, P)$ and into the forward-error-control (FEC) unit. This message is then fed into the PN modem and transmitted down.

Command information is received in the vehicle via the PN demodulator. This information may contain the sensor control rule $D(\theta, H, P)$ which is stored. Thus, the ground station can generate the control program or it can be generated by the on-board multiprocessor.

2.5.2 Adaptive Demultiplexing[3]

Figure 2-13 shows the adaptive demultiplexer. The PN demodulator at the ground stations recovers the message bits which are corrected for errors in the FEC decoder. The processor generates the demultiplexing and sensor control rule $D(\theta, H, P)$ and controls the demultiplexer operation. The information transmitted up to the vehicle is control information which may include the multiplexing and sensor control program D.

Note that the program $D(\theta, H, P)$ depends on the measured sensor parameter point θ and on the estimated channel throughput H; the priority matrix P can, of course, be changed also. Based on measurement, a decision is made which

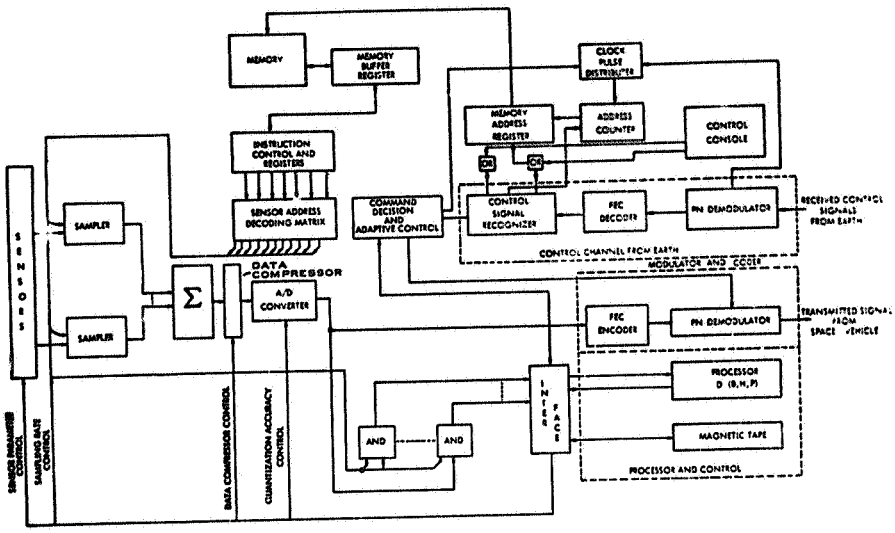


Figure 2-11. The Adaptive Multiplexer

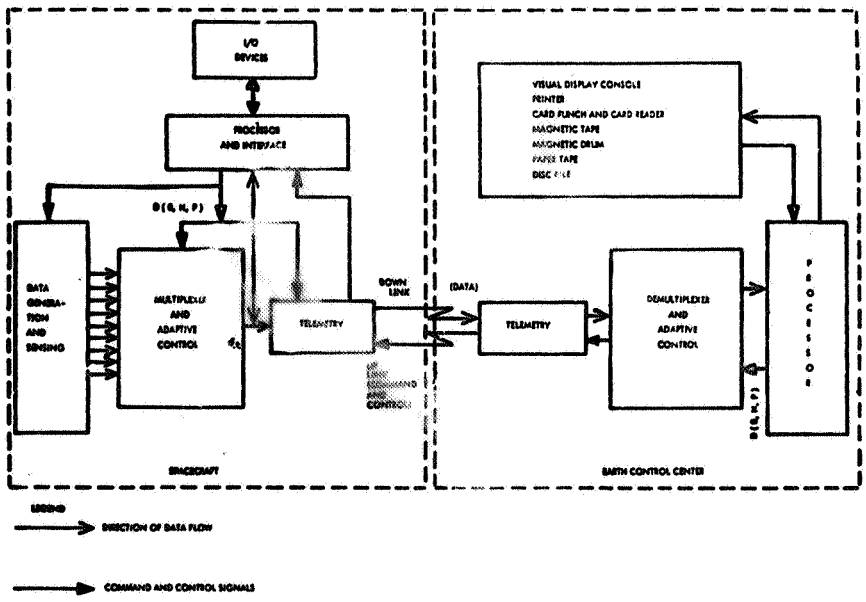


Figure 2-12. The Spacecraft and Earth Control

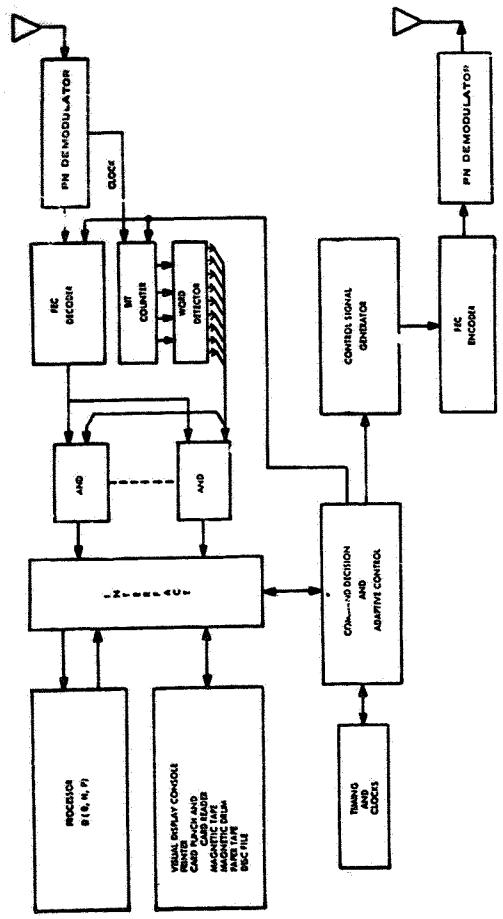


Figure 2-13. The Adaptive Demultiplexer

selects the program, say, D_i. The set of programs $D = \{D_i\}$ can be prestored, transmitted to the vehicle from the ground control center, and derived by means of a mathematical algorithm.

2.5.3 Decision Rules

In Section 4 of the proposal, a statistical decision theory was developed to monitor the sensors and adapt the sensor parameters according to the results of measurement. It is important that the structure of the decision mechanism be general enough to apply to any sensor configuration by inserting the appropriate programs and the appropriate information in memory.

2.6 COMPUTER CONTROL OF THE ANTENNA SYSTEMS

Detailed computer requirements for this aspect of the communications subsystem can be established only if the antenna configuration to be used is known. However, a general specification of the computational needs for a computer controlled antenna system can be obtained by investigating the characteristics, requirements, and some configurations of antenna systems that probably will be used.

The functions which must be performed by the on-board communications processor for the desired antenna operations will be determined by the antenna requirements, which in turn depend on the mission objectives, types of stabilization, and orbital parameters. The desired antenna systems may have the following characteristics:

a. Multiple beams may be required for transmission to and reception at different points on the earth, at the same frequency. One type of system that accomplishes this in a purely digital manner is the DIMUS (digital multibeam steering) system. This technique is described later in this section.

b. There are steerable beams which can be pointed to different areas upon command. This type of antenna can be used to effect spin stabilization or despinning. This problem is described later.

c. The antenna may have to track a source of power while there is relative motion between the vehicle and the source.

d. At different times in the mission it may be desirable to have a narrow beam to increase gain to illuminate small regions and receive as much signal power as possible, or it may be desirable to increase the illumination area or to receive from receiving locations distributed over a wide angular range.

e. Antenna beams must be switched on and off or from transmit to receive.

The above requirements can be met with multiple antennas or an antenna array in conjunction with an on-board processor which integrates the antenna requirements with the mission objectives and vehicle characteristics.

2.6.1 Antenna Configuration

Complex antenna requirements can be met better in a gravity gradient stabilized system than in a spin stabilized system. The antenna system can be characterized in terms of

a. The number of beams

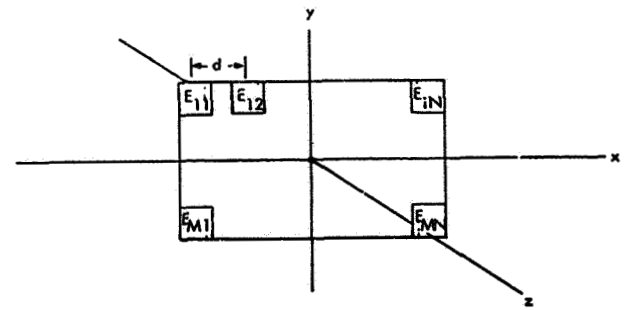
b. The total angular range over which each beam may be required to steer

c. The range of their beamwidth

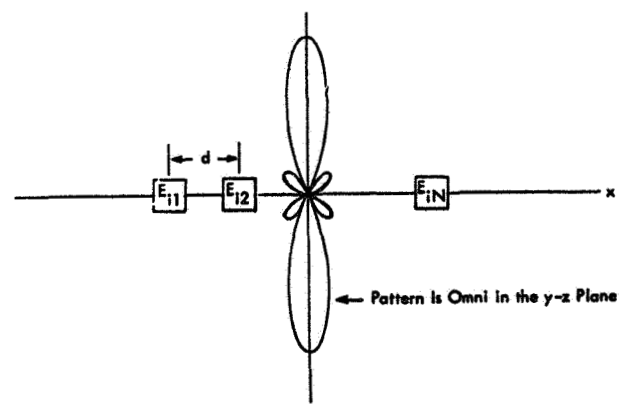
d. Whether the antenna must transmit or receive, or transmit and receive.

Figure 2-14a shows the geometry of a rectangular planner array where E_{ij} is the element in the ith row and jth column. If each element is omnidirectional, the radiation pattern of a single row will be omnidirectional in a plane perpendicular to that row if the elements are equally spaced as shown in Figure 2-14b.

The normalized radiation pattern is given by



a. Planar Array Geometry



b. Radiation Pattern from A Single Row

Figure 2-14. Antenna Array Geometry

$$G(\theta_R) = \frac{\sin^2 (N\pi(d/\lambda) \sin \theta)}{N^2 \sin^2 (\pi(d/\lambda) \sin \theta)} \qquad (2\text{-}5)$$

where θ_R is the angle from the X axis in the X-Z or x-y plane. For maximum gain, $d = \lambda/2$, the half-power beamwidth is $\theta_0 = 101.8/N$, and the gain is $G_0 = N$.

The above expression for $G(\theta_R)$ also applies to the case where a column of elements is considered a linear array, in which case, a pattern is formed which is orthogonal to the pattern that was formed from the row. It then follows that if all elements are considered, a narrow beam is generated along the perpendicular to the plane of the array.

If each element in the array is directional, the resulting pattern for the rectangular planar array is

$$G(\theta_R, \theta_C) = \frac{\sin^2 [N\pi(d/\lambda) \sin \theta_R] \sin^2 [M\pi(d/\lambda) \sin \theta_C]}{N^2 \sin^2 [\pi(d/\lambda) \sin \theta_R] M \sin^2 [\pi(d/\lambda) \sin \theta_C]} G_E \qquad (2\text{-}6)$$

where G_E is the gain of the individual element.

By controlling the electrical spacing between elements, the beam of the planar array can be offset from the perpendicular. If θ_0 is the offset desired, then the required phase spacing of adjacent elements is

$$\phi = 2\pi(d/\lambda) \sin \theta_0 \qquad (2\text{-}7)$$

Figure 2-15 illustrates the relative phasing of the individual element feeds to produce a pattern offset in a single angular dimension. By electronically adjusting the phase shifters, the pattern can be rapidly steered in a single angular coordinate; or for a planar array it may be steered in two coordinates. If a second source is used and feeds the individual elements through a second set of phase shifters, a second beam can be generated from the antenna. As many

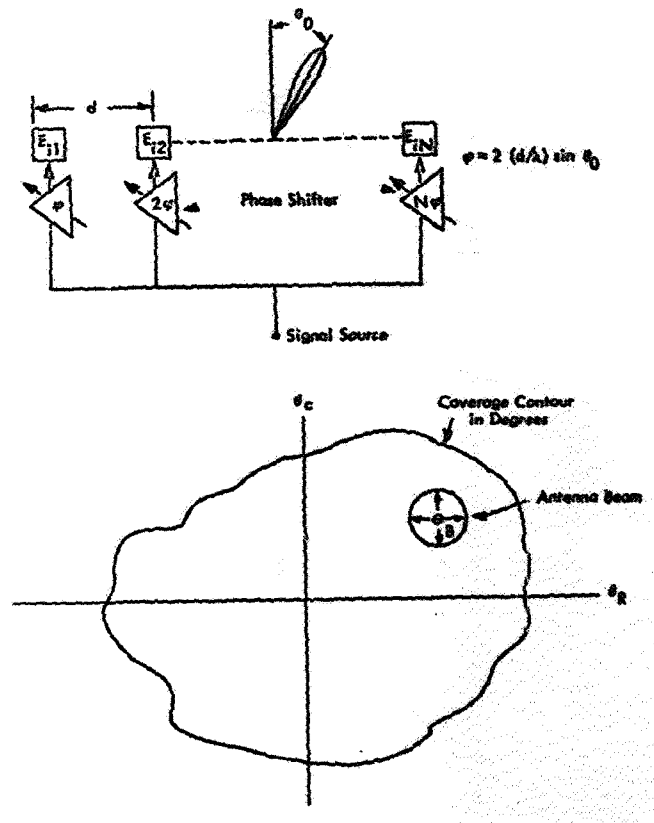


Figure 2-15. Steerable Arrays and Coverage Range

beams can be formed as desired as long as a separate set of phase shifters is used for each beam. The central control computer will designate the beams to be formed and furnish this information to the communications and telemetry processor which will steer the antenna.

The expressions for $G(\theta_R, \theta_c)$ and ϕ can be used to establish the architecture of the antenna system. The gain of the individual elements must be compatible with the overall coverage range. The array parameters are then selected to provide a steering capability within that coverage range and the required phase spacing can be determined.

The number of elements required will be determined by the ratio of the total coverage range to the desired beamwidth as illustrated in Figure 2-15.

2.6.2 Phase Shifters and Control Signals

The type of control signal furnished by the communications processor will depend on the characteristics of the electronically controllable phase shifters which are used. These include the use of ferrite materials, gaseous discharge tubes, traveling-wave tubes, transmission lines, varactors, etc. The major difference, as far as the computer is concerned, is whether the beam must be continuously steerable or discretely steerable.

If the beam is discretely steerable, the computer selects the correct phase shifters which are preset to give a particular beam direction. For a continuously steerable system, the computer must adapt the control signal to the phase characteristic of the device to achieve the desired beam direction.

Figure 2-16 illustrates how control signals are used in a two-dimensional scan system. The required accuracy of the control signals is determined by the sensitivity of the phase shifter to the control signal, the beamwidth of the array, and the stabilization accuracy of the vehicle.

2.6.3 Tracking Capability

A tracking system which employs a phased planar array and the communications and telemetry processor is illustrated in Figure 2-17. Although the tracking

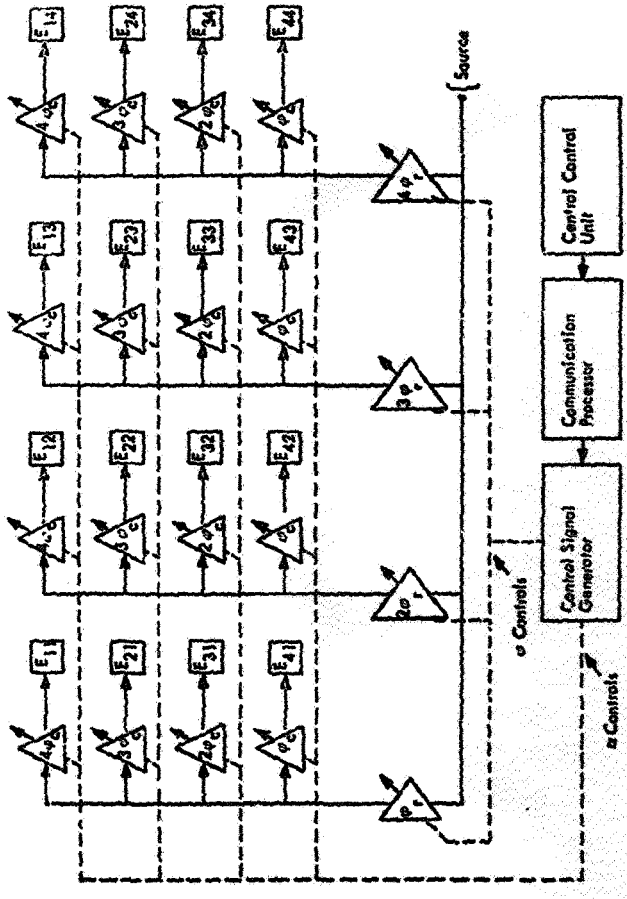


Figure 2-16. Control System for Scanning in Two Angular Coordinates

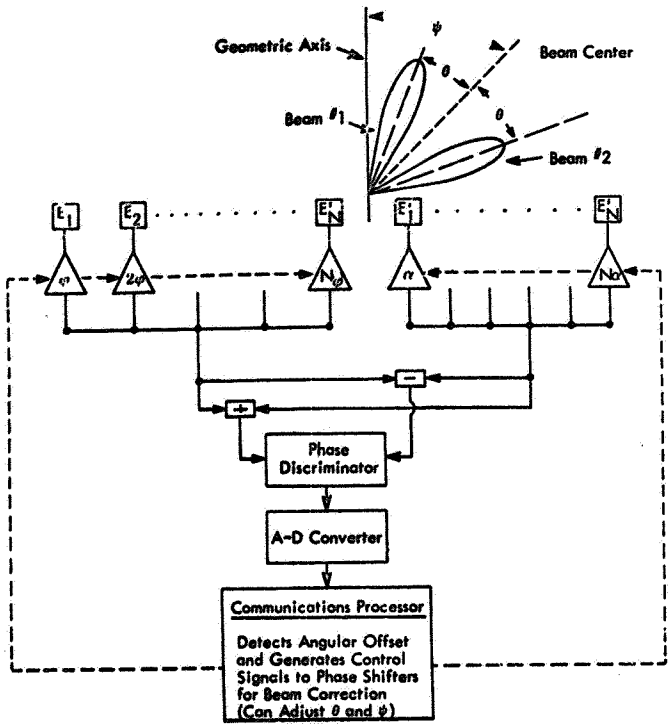


Figure 2-17. A Single Dimension Tracking System

capability is illustrated for a single angle only, the capability for volumetric tracking follows directly from Figure 2-16. The number of elements is divided into two sections, with each section forming a beam which is offset from the intended boresight. The sum of the two beams will be a symmetric beam about the boresight. The received signals from both sets of elements are added in one channel and subtracted in another. The phase difference between the sum and difference channels will determine where the source is located in relation to the boresight. The discrimination output will contain the error information. This will be processed by the communications processor, which will generate error correcting signals to the phase shifters which will correct the angular offset of the boresight.

2.6.4 Spin Stabilized Antennas

The beamforming techniques which have been discussed apply to a vehicle which is stabilized in three coordinates. It is assumed that the geometric center of the array is pointed at the same angle relative to the earth. If the vehicle is spin stabilized, complex antenna requirements are difficult to meet. If the array could be scanned at the rotational velocity of the vehicle, a narrow beam could be directed at a fixed point on earth. Power compensation would be employed to compensate for the variations in antenna gain.

A simpler system for spin stabilized vehicles is illustrated in Figure 2-18. Antennas are mounted on the periphery of the vehicle in the plane of rotation. The antenna beam, under computer control, would rotate at an angular velocity equal to but opposite to the rotation of the spacecraft. The antenna pattern closest to the desired direction of propagation would be selected, and as the vehicle rotates, the adjacent channel would be used.

To accomplish this task, information from the guidance and navigation processor would be processed by the central control computer and fed to the communication processor. This processor would control the antenna beam velocity in accordance with the position of the satellite as determined in the navigation and guidance processor.

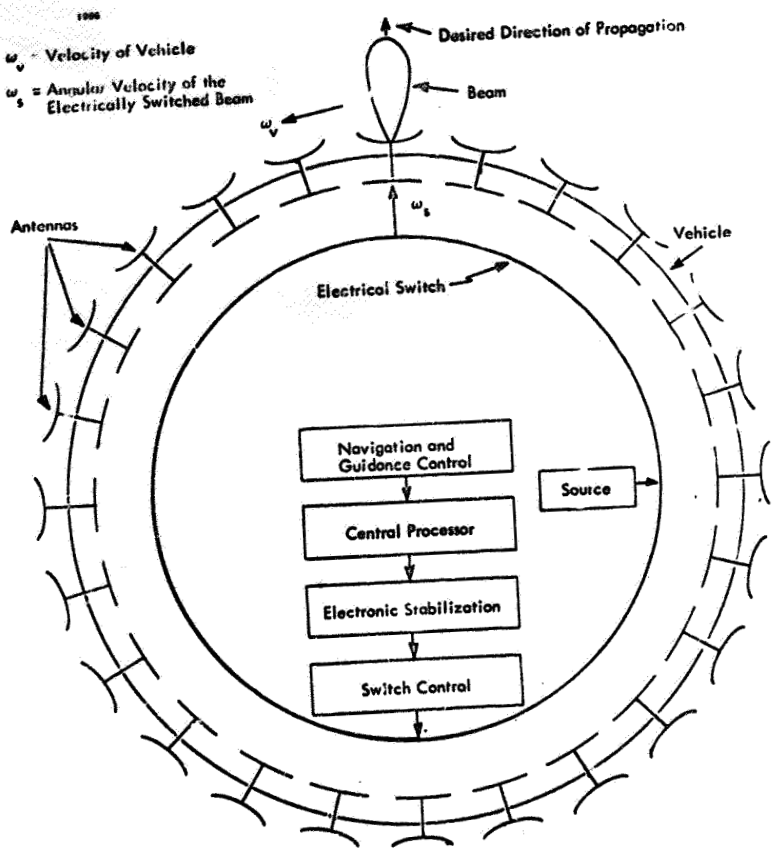


Figure 2-18. Antenna Switching Arrangemnnt for Spin Stabilized Vehicles

2.6.5 DIMUS

A convenient system which has been used for beamforming in sonar work is DIMUS (digital multibeam steering). This system, as it may be applied to the satellite system, is illustrated in Figure 2-19. The desired signals from the various elements of a planar array are selected by the array control. Each signal is then clipped and fed into a shift register or delay line. The desired beam is then formed by selecting the appropriate taps on the shift register or delay line. By shifting taps, the beam may be steered. Multiple beams are easy to form by increasing the number of tap connections.

The communication and telemetry processor would control the tap connections so they conform to the desired beams which are described by information from the central control unit, and would also control the shifting of the clipped signals.

2.7 COMPUTER REQUIREMENTS

As we can see from the above description, there are many functions to be performed in the communications subsystem. This subsection will describe the type of computation that is necessary for each of these functions.

2.7.1 Pseudo-Noise Modulation

Subsections 2.1 and 2.2 described the basic on-board modulation system and the reasons for choosing that system. The computer plays an important role in this system, in that it will monitor the modulation equipment and determine whether or not the demodulators are in phase lock. To perform this function, the computer must essentially go through a decision process and determine, based on the amplitude distribution of the signal, what the state of the system is. Thus, mainly statistical techniques will be used as described in Section 5, and the programs will be arithmetic in nature. The computer will also continually calculate channel signal-to-noise ratios based on the received signals and use these calculations to control the link data rate.

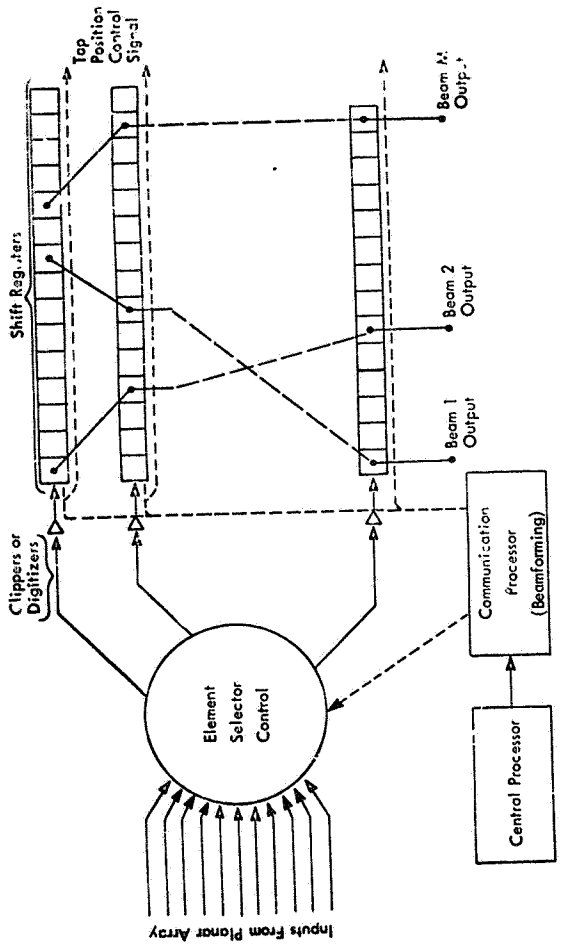


Figure 2-19. DIMUS Approach to Beamforming Using a Planar Array

2.7.2 Error Control

If advanced error correction techniques are employed, as described in subsection 2.3, a considerable amount of core storage will be required. In general, the storage required is equal to approximately twice the number of data bits in a code frame. The computer functions as a data manipulator in this case (shift, etc.) and no arithmetic operations are necessary. Logical instructions for performing MOD-2 addition of vectors will be required.

2.7.3 Data Compaction

The advanced adaptive data compaction techniques described in subsection 2.4 require a computer capable of manipulating data, performing logical operations on the data (such as MOD-2 addition) and computing statistical measures on the data. Thus arithmetic, logical, and data handling instructions are required.

2.7.4 Adaptive Multiplexing and Demultiplexing

In this area, the computer functions as a decision making device and controls the multiplexing format. It senses the data, calculates statistical measures and, along with error rates calculated from data received from the error control equipment, decides whether to change the sampling rate of a particular sensor output. Based on this decision, a new multiplexing format is devised and control signals are sent to the multiplexer.

2.7.5 Antenna Systems

The antenna system for a multifunctional satellite will probably be computer controlled and steerable; such systems are discussed in detail in subsection 2.6. Since the computer is strictly a control device in these systems, it will accept inputs from the antenna system or from a ground station and generate control signals to the antenna system. Only program storage is required for this function.

Section 3

DATA COLLECTION AND SENSOR SUBSYSTEM

3.1 LINK CAPACITIES

In current space activities and those planned for the future, considerable data handling (DH) is required: data transmission, signal processing, storage and retrieval, trajectory computation, and display. At the present time this data handling is largely ground-based, but already there is significant on-board DH. As we move deeper into space (and the communication channel to Earth narrows) and as we progress in producing light, reliable, and sophisticated spaceborne DH systems, the percentage of on-board data handling will greatly increase. In this section we shall look at the next decade, which will be dominated by Project Apollo, Earth-orbiting laboratories, and unmanned planetary probes. We shall consider the types of data to be handled and estimate total data requirements. We shall also estimate the capacities of the communication links and the quantity of data generated by experiments on several representative missions. It is evident that on-board data processing will become very important in the next decade. Clearly the difference between the channel capacity and the amount of raw data generated on board must be compensated by on-board data processing techniques.

The objective here is to estimate the amount of on-board processing that will be required in the next decade. To achieve this, two estimates are required—one for the channel data requirements and one for the amount of data generated on board the vehicle. To accomplish the latter, we have selected three representative missions and estimated the amount of data generated by the experiments scheduled for each of these missions. This approach yields a more realistic estimate of data generated in any single future mission.

Table 3-1 represents an effort to categorize space data in two directions: by its qualitative nature (such as scientific, graphic, physiological) and by the link over which the data is transmitted. Note that the distinctions are not sharp; "scientific" and "physiological" overlap, as do "scientific" and "graphic." The vertical categories are chosen to correspond to the various ground to space-vehicle distances which appear of interest during the 1966-1976 period. The table is a compilation of total bit requirements, average and peak bit rates, etc., for the various situations before the application of any on-board processing techniques, such as data compaction to reduce redundancy and lessen transmission requirements. We have also estimated channel capacities based on the current (mid-1966) state-of-the-art. Every effort has been made to show the most accurate data available; the data given has been gathered from many sources, carefully examined, and cross-checked. It should be clear from this table that unless there are tremendous breakthroughs in transmission technology, the channels available will not be able to handle satisfactorily (or, in some cases, handle at all) the data loads if we "send everything" as is common now. Moreover, even when the channel capacity is reasonably large, it is wasteful and expensive to transmit redundant material. Thus, attention is drawn to techniques of data compaction, which involve data processing of more or less complexity, depending on the type of data and the amount of compaction sought.

It should be pointed out that the present channel capacities are far from adequate to handle future data requirements. Granted that these capacities will be increased in the next 10 years, the amount of on-board data generated by the experiments will still far exceed these future capacities. Thus a large amount of on-board processing of data will be necessary.

In addition, these channel capacities will directly affect the computational requirements on board the space vehicle. Thus, the storage capacities of the memory modules will be a function not only of the data that is generated but also of the computational requirements. However despite that fact that communication technologies (lasers, directional antennas) will be improved substantially by

Table 2-1

ESTIMATED DATA RATES AND CHANNEL CAPACITIES

	Ground to Near-Space (to Satellites)	Near-Space to Near-Space (Sat to Sat)	Ground to Mid-Space (to Moon)	Mid-Space to Mid-Space (LEM and Mother)	Ground to Far-Space (to Mars)
Tracking and Control	100→200b/s (av) 1000b/s (peak)	10 → 20 b/s (av) 100b/s (peak)	100→200b/s (av) 1000b/s (peak)	1→2 b/s (av) 10 b/s (peak)
Scientific	$10^8 \rightarrow 10^9$ b/day $10^4 \rightarrow 10^5$ b/s (pea	$10^3 \rightarrow 10^4$ b/s (av)	$10^2 \rightarrow 10^3$ b/s (av) $10^7 \rightarrow 10^8$ b/day	$10^3 \rightarrow 10^4$ b/s (av) $10^6 \rightarrow 10^7$ b/day	10→100 b/s (av) $10^6 \rightarrow 10^7$ b/day
Physiological	$10^8 \rightarrow 10^9$ b/day 10^5 b/s (peak)	$10^6 \rightarrow 10^7$ b/day	$10^8 \rightarrow 10^9$ b/day 10^4 b/s (peak)	$10^8 \rightarrow 10^9$ b/day	$10^7 \rightarrow 10^8$ b/day
Voice	$10^4 \rightarrow 3 \times 10^4$ b/s (real time)	$10^4 \rightarrow 3 \times 10^4$ b/s (real time)	$10^3 \rightarrow 3 \times 10^3$ b/s (slow time)	$10^4 \rightarrow 3 \times 10^4$ b/s (real time)	100→300 b/s (slow time)
Graphic	$10^6 \rightarrow 10^7$ b/s (TV) $10^5 \rightarrow 10^6$ b/picture				
Channel Capacity for All Uses (present)	10^7 b/s	5 x 10^3 b/s (modem) 3 x 10^4 b/s (voice) 3 x 10^6 b/s (TV) per channel	$10^4 \rightarrow 10^5$ b/s	3 x 10^4 b/s (voice)	10 b/s

the 1975 era to accommodate higher bit rates, the need for more sophisticated on-board computation will actually increase to accommodate, implement, and support these esoteric communication systems. Thus, the amount of on-board processing that will be required is a key problem to be solved.

3.2 ON-BOARD DATA COLLECTION

3.2.1 Experiment Data Loads for AAP Missions

This subsection presents estimates of the quantity of data generated by the experiment hardware on two AAP missions—one related to astronomy/meteorology and the other to Earth survey/meteorology.

The analysis of experiment data outputs and processing requirements has been hampered by lack of definitive current data on mission experiment groupings and specific hardware characteristics. The information which follows was furnished to IBM by NASA for AAP analyses and is, therefore, considered to be the best available information.

For each experiment an identical or reasonably similar experiment was sought in the available experiment data. When no reasonable facsimile could be found, an experiment was synthesized from similar experiments of other programs, such as OAO or Nimbus.

3.2.1.1 Meteorological Experiments

The meteorological experiments and estimated daily data loads are listed in Table 3-2. The corollary experiment sources from which these estimates were derived are shown in the Notes column.

For record purposes, the raw data estimates used to compute the daily loads are presented here. The duty cycles (D.C.) indicated are the estimated percentage of total mission time (or daily average) that the experiment is in operation.

Table 3-2

COMPLEMENT AN AAP METEOROLOGICAL MISSION EXPERIMENT

Experiment Title	Notes	Digital Bits/Day
Millimeter Wave Propagation	4 frequencies; 2 words/sec/freq.; 7 bits/word; 20% D.C. Housekeeping data only, estimate. Based on priority missions.	9.7×10^5
Microwave Radiometer	2 systems; 2 words/sec/scan/system; 10 lateral scans/system 7 bits/word; add 50 bits/sec housekeeping; 25% D.C.	7.1×10^5
Near Infrared (IR) Spectrometer	7.5×10^4 bits/day/freq; 6 freq; 25% D.C. Based on MORL (Douglas) experiment.	9.0×10^4
Polarization Measurements	7.5×10^4 bits/day; 33% D.C. Based on MORL (Douglas) experiment.	2.5×10^4
UHF Sferics Detection	15 words/sec; 7 bits/words; 33% D.C. Based on Ariel Galactic Noise experiment.	3.0×10^6
IR Interferometer Spectrometer	1 sample/sec; 500 bits/sample; 33% D.C. Based on Nimbus experiment.	1.4×10^7
Micron Grating Spectrometer	1 sample/sec; 200 bits/sample; 33% D.C. Based on Nimbus experiment.	5.8×10^6
AAP Total		2.5×10^7
NIMBUS (unmanned)	1300 frames/day = 2.6×10^6 bits/frame.	3.4×10^9
TIROS (unmanned)	average 296 frames/day (over 6 flights) = 7.4×10^5 bits/frame.	2.2×10^8

For comparison, Table 3-2 also indicates the daily data loads for the Nimbus and TIROS unmanned meteorological satellites. These have much higher telemetry loads, but no film. The reason for the lower telemetry loads in the AAP mission appears to be that high-resolution image data is returned in the reentry vehicle in the form of film (and dielectric tape), whereas the telemetry is used mainly for non-image spectral measurements. TIROS and Nimbus return all image data via the telemetry link, however.

3.2.1.2 Astronomy Experiments

The astronomy experiments and estimates of the data generated are presented in Table 3-3. These are solar astronomy experiments. Descriptions of the experiments and assumptions made in estimating data quantities are given in the following paragraphs. The OSO and OAO satellite missions contained similar experiments and are shown for comparison.

The X-Ray Telescope experiment will collect and count X-ray photons from the stars. Data taken is a function of the number of stars observed, assumed to be three. It is estimated that one boresight photo will also be required for each star.

The Ultraviolet Spectrometer experiment measures stellar radiation, providing information similar to the orbiting astronomical observatory (OAO) unmanned satellite. Anticipated data rates range from 1042 to 50,000 bits/sec. An average of 25,000 bits/sec for a 3-hour observation period each day is assumed for this experiment.

3.2.1.3 Earth Survey Experiments

The feasibility of this grouping has not been tested from the standpoint of astronaut and vehicle constraints (time, weight, power, etc.). The grouping is, however, considered to be a good representation of the appropriate types of experimentation and, as such, appropriate for order-of-magnitude estimation of data loads.

Table 3-3

SOLAR ASTRONOMY EXPERIMENT DATA ESTIMATES

Experiment Title	Digital Bits/Day	Notes
X-Ray Telescope	1.4×10^8	4.8×10^7 bits/ star/day, 3 stars
Ultraviolet Spectrometer	2.7×10^8	assume rate = 25 kbs for 3 hr/day daily average
Total Per Day	4.1×10^8	
Unmanned Astronomy Satellites		
OSO	9.6×10^7	
OAO	2.4×10^8	

The results are presented in Table 3-4 along with the daily data outputs for the EGO and POGO satellites. The latter rely solely on the telemetry link for data transmission, whereas AAP (as configured here) telemeters housekeeping data but uses physical reentry/recovery for experiment films and analog data tapes. Accordingly, as indicated in Table 3-4, the AAP telemetry load for Earth resources experiments is negligible, but the physically returned data load is very heavy.

Tables 3-2, 3-3, and 3-4 each show a class of experiments. A typical mission would be made up of any two of these classes. For example, a set of meteorological experiments plus a set of astronomical experiments would constitute a mission.

Table 3-4

EARTH SURVEY EXPERIMENT DATA

Experiment Title	Bits/Day
IR Spectrometer	5.0×10^5
Multispectral Camera	5.0×10^5
Unmanned Satellites (Earth-Space Sciences)	
Eccentric Geophysical Observatory (EGO)	8.0×10^7
Polar Orbiting Geophysical Observatory (POGO)	3.2×10^8

3.2.2 Jupiter Orbiter

The aim of this subsection is to describe experiments and related instruments which will meet the scientific objectives for the Jupiter Orbiter.

Table 3-5 summarizes the instruments and experiments for the Jupiter Orbiter mission. In the first column the instrument is named. The second column shows what the instrument measures, including ranges where appropriate. The third column lists scientific objectives. It is assumed that the instrument will measure perfectly, in a sufficient number of locations, whatever it was designed to measure. The scientific objective shown will then be met more or less completely. (Some scientific objectives require more than one instrument.)

With the exception of number 3, the instruments listed (some of which have flown) have been designed for spacecraft. However, some would require modification to different ranges, sensitivity, etc. Therefore, it is possible to estimate weight and power requirements fairly accurately. Column 4 gives the estimated power of the experiment, and the last column gives an estimate of the data generated.

Table 3-5

LIST OF INSTRUMENTS AND EXPERIMENTS FOR THE JUPITER ORBITER MISSION

Instrument	Measured Quantity	Scientific Objective	Estimated Power (watts)	Data Generated
1. Magnetometer Pack	Magnitude and direction at magnetic field in the range 0.5×10^{-5} - 100 Gauss	Obtain magnetic moment. Map field configuration.	9	30 bits/ sample
2. Scintillation Detectors	Electron flux: 10-100 kev Proton flux: 0.1-10 mev	Understand electron flux mechanism.	1	50 bits/ sample
3. Not Known	Electron flux: 100 kev - 100 mev	Understand decameter radio emission.	2	50 bits/ sample
4. Plasma Probe	Fluxes of low energy electrons and protons	Understand interaction with solar wind.	3	10 bits/ sample
5. Cosmic Ray Spectrum Analyzer	Spectrum of proton fluxes; 1-100 mev	Find proton belt. Understand high-energy processes in magnetosphere.	1	70 bits/ sample
6. Radio Receiver	RF flux; several frequencies below 30 Mh	Understand mechanism for decameter bursts.	3	50 bits/ sample

Table 3-5 (Continued)

Instrument	Measured Quantity	Scientific Objective	Estimated Power (watts)	Data Generated
7. Moderate Resolution UV-Visible Spectrophotometer	Spectra 1000-7000 Å; Resolution of 5 Å	Identify constituents. Understand red spot features. Determine existence and composition of satellite atmosphere.	5	10,000 bits/scan
8. High Resolution UV Photometer	Intensity of ground state H, He, Ne	Obtain abundances of H, He, and Ne.	10	30 bits/ sample
9. IR Spectrophotometer	5-30 spectra; resolution of 30	Determine variations in abundances of CH_4 and NH_3.	7	5,000 bits/scan
10. Television (stereo)	High resolution pictures	Understand red spot features. Investigate general circulation. Note structure of satellite surfaces.	10	5×10^6 bits/ picture
11. Polarizer-Photometer	Amount of polarization in selected spectral ranges.	Determine nature of clouds	7	120 bits/ sample
12. Radiometer	Flux: 10-13μ	Obtain temperature well above cloud tops.	2	20 bits/ sample
13. Radiometer	Flux: 0.7 mm	Obtain temperature at or below cloud tops.	5	30 bits/ sample

Table 3-5 (Continued)

Instrument	Measured Quantity	Scientific Objective	Estimated Power (watts)	Data Generated
14. Radiometer	Flux: 3 cm & 10 cm	Determine temperatures. Understand decameter radio emission.	4	40 bits/sample
15. Radar Altimeter	Back scatter	Determine depth of atmosphere. Determine nature of surface.		
16. Tracking	Position velocity of S/C	Obtain accurate values of gravitational moments.	Telemetry	
17. Occultation Experiment	Amplitude and phase of telemetry	15, 16 Determine scale height of atmosphere and ionosphere.	Telemetry	

The first six instruments are designed to shed more light on the magnetosphere of Jupiter. The magnetometer package, scintillation detectors, plasma probe, and cosmic ray spectrum analyzer are all well-tested instruments. Experiments 3 and 6, coupled with experiment 2, are important to the understanding of decameter burst emission from Jupiter. Once understood, past records of decameter bursts can be used to provide more or less continuous monitoring of magnetospheric events over the past 10 years and in the future.

Instrument 3 has not been identified because no instrument designed to measure electron flux in the energy range shown has actually flown. However, the existing instrument referred to did accidentally measure electron flux within the range shown; therefore, the design of instrument 3 should present little difficulty. The radio receiver is designed to find exactly where in the Jovian magnetosphere the decameter burst emission is arising and should also provide more complete information about both the low frequency and weak radio emission from Jupiter.

The first six experiments, taken as a whole, will satisfy all of the scientific objectives relating to the Jovian magnetosphere. The next three experiments are designed to take spectral measurements of Jupiter and its satellites. The spectrophotometer of experiment 7 is not designed for better spectral resolution than is possible from Earth, but extends the range to the far ultraviolet and can probably obtain much better spatial resolution. Furthermore, the lighting conditions on Jupiter can be varied considerably; in particular, emission lines from unlit regions on the planet can be obtained. Gross composition of the satellite atmospheres might be obtainable.

The UV photometer of experiment 8 should be capable of satisfying one of the most important scientific objectives on the Jovian mission, determining the abundance of hydrogen, helium, and neon. On a very cold planet such as Jupiter, the ground state emission line should be the strongest, and this lies very far in the ultraviolet band. A mission to Jupiter would make it possible to get into this band and to view the dark side of Jupiter. For the most clear-cut results, a total eclipse condition would be best because it would cut down on background.

A subsidiary piece of knowledge is required to determine the abundance of hydrogen, helium, and neon from the intensity of a spectral line—namely, the solar flux at Jupiter. Since the interplanetary medium is selectively absorbing in the far ultraviolet, one should also observe the solar flux with the UV photometer as soon as possible before and after the Jovian measurement. The infrared spectra obtainable from an orbiter are an advance over Earth-based observations in that the presence of telluric lines does not occur, interpretations of the spectra should be considerably easier than from Earth, and better spatial resolution can be obtained.

A television system is included on board the proposed spacecraft to view atmospheric features and their motions and to note the structure of satellite surfaces. Ground-based observations lack both resolution and any vertical scale. A stereoscope television system can provide vertical scaling as well as vastly improved resolution, so that the meteorology of Jupiter and the origin of the red spot can be better understood.

The polarizer-photometer would be used to gain information of the size and dielectric properties of cloud particles. An orbiter mission could be a great advance over Earth-based observation because the range of phase angles is 0° to 180°, which is much wider than the 0° to 11° range possible from Earth. On the other hand, the interpretation of polarimetric data is difficult, and it is not clear that even a perfect experiment can yield data which will really meet the scientific objectives.

The next three experiments employ radiometers operating in various wavelengths. In the range of 10 to 13 microns there is strong absorption by the Jovian atmosphere, and so the region for which the temperature would be measured by the radiometer should lie very high in the Jovian atmosphere. The region around 0.7 mm is of very low absorption by the Jovian atmosphere. Measurements at 0.7 mm should obtain the temperature at or below the cloud tops. Finally, a microwave radiometer operating at 3 and 10 cm can obtain temperatures both thermal and nonthermal, and a better understanding of decameter radio emission, which is rather strong at 10 cm, might result. Some instrument development is probably required for the 0.7 mm radiometer.

The radar altimeter might yield information about the depth of the Jovian atmosphere and the structure of its surface, if one exists. In theory, a radar altimeter could determine the distance between the spacecraft and the surface, as well as information about the nature of that surface. The problem—and it is a severe one—is that so much atmosphere may be lying between the spacecraft and the surface that signals of high intensity will be required to receive a return signal. If a return signal is obtained, the scientific objectives will be met. If a return signal is not obtained, one can place some lower limit to the depth of the atmosphere through which the signal has passed.

The last two experiments shown do not necessarily require any special instruments. Tracking data should be adequate to determine orbit changes with time, thus yielding much improved values of the gravitational moments. The occultation experiment, similar to an occultation experiment on Mariner IV, should yield good values for atmospheric and ionospheric scale heights. This would probably lead to much better model atmospheres than are presently available.

A large number of sensors have been used in aerospace experiments, and some of these have been described in the preceding section. Many of the sensors have been designed for specific types of measurements, although a comprehensive experiment generally employs several sensors of various sensitivities to one or more source phenomena. In subsections 3.3 and 3.4 the sensor study is concluded with a discussion of two areas of space physics: micrometeoroids and solar flares.

3.3 MICROMETEOROIDS

3.3.1 Background

Small particles exist in the near-Earth, cis-lunar, and interplanetary regions. Particles smaller than 1 mm in diameter are commonly called micrometeoroids and, because of their high velocities, may represent a potential hazard to manned spaceflights. It is important, therefore, to expand present

knowledge about both spatial and temporal variations of particle mass distributions, velocity distributions, flux, and gradients. In addition, it is believed that increased understanding of micrometeoroids will extend our knowledge of the origins of the planets and, perhaps, the universe.

3.3.2 Present Knowledge

Micrometeoroids are generally less than 1 mm in diameter with the majority below 300 microns in diameter. Masses range downward from 10^{-3}g to below 10^{-16}g, with particle velocities up to 72 km/sec possible. It is suggested that most particles are of cometary origin and are of a "fluffy" nature with densities of the order of 0.05 g/cm^3. Other micrometeoroids may be of asteroidal origin and are of about 3.5 g/cm^3 density.

It has been suggested that micrometeoroids in interplanetary space are acted upon by the relativistic Poynting-Robertson effect, which causes a particle to fall in toward the sun. Calculations show that a 100-micron diameter particle in a 1AU heliocentric orbit would fall into the sun in about 4000 years. On the other hand, solar ablation of the micrometeoroids reduces their diameter to a critical value Dm for which solar radiation pressure becomes the predominant force, and the particles should, in fact, be blown out of the solar system. The critical diameter is given by the relation

$$Dm = \frac{0.6}{P} \qquad (3\text{-}1)$$

where P = particle specific gravity.[4]

If a density of 0.05 g/cm^3 is assumed for particles, these must have masses upward from a 4×10^{-10}g limit set by the solar radiation pressure. For materials of greater density, the lower limit on particle mass would be correspondingly smaller. A Venus Flytrap experiment captured particles near the Earth, which substantiates the predominance of the less dense particles.[5]

Perhaps the most important result of recent micrometeoroid experiments is the discovery of an asymmetrical dust cloud surrounding the Earth. A number

of orbital missions (Pioneer I, Explorer VIII, Sputnik III) have resulted in data on the mass distribution of particles near the Earth as shown in Figure 3-1. From this plot, it appears that the flux of particles of mass m present in the Earth dust cloud is given by ϕ (m):

$$\phi(m) = m^{-1.7} \times 10^{-17} \text{ particles/m}^2\text{/second} \qquad (3\text{-}2)$$

over the range 10^{-6} - 10^{-10}g. This data is relatively precise but subject to some criticism, since mass values are derived from measured particle momentums by the assumption of an average particle velocity of 25 km/sec. Mariner II verified the existence of such a cloud by finding a 2×10^{-4} difference between near-Earth and interplanetary micrometeoroid fluxes.[6] Whipple has suggested that the Earth dust cloud is primarily of lunar material ejected from the moon's surface by hypervelocity impacts of cometary and asteroidal bodies. A portion of the material is assumed captured by the lunar-Earth gravitational system.[7] An Earth dust accretion rate of 300 tons/day has been calculated. These particles are in the 20 to 100-micron diameter range.

There are interesting patterns of meteoroid flux documented, but apparent seasonal and daily variations are not understood. Most micrometeoroids are thought to be of sporadic radiant, concentrated in heliocentric orbits of which more than 50 percent are within 13 degrees of the ecliptic plane.

3.3.3 Future Micrometeoroid Studies

Although there has been much micrometeoroid study in recent years, the present knowledge is fragmentary. Future studies are justifiable simply on the basis of presumed dangers to manned spacecraft and the lack of knowledge about mass and flux distributions of particles in the 10^{-3} - 10^{-6}g range. There are many points of interest to the geophysicist and astrophysicist in relating micrometeoroids to Earth, solar, and universal origin. The following list compiles some areas of interest for future micrometeoroid experiments:

a. Determination of mass distributions, flux, and gradients of particles in the 10^{-3} to 10^{-6} range in the Earth dust cloud, near the moon, and in interplanetary space.

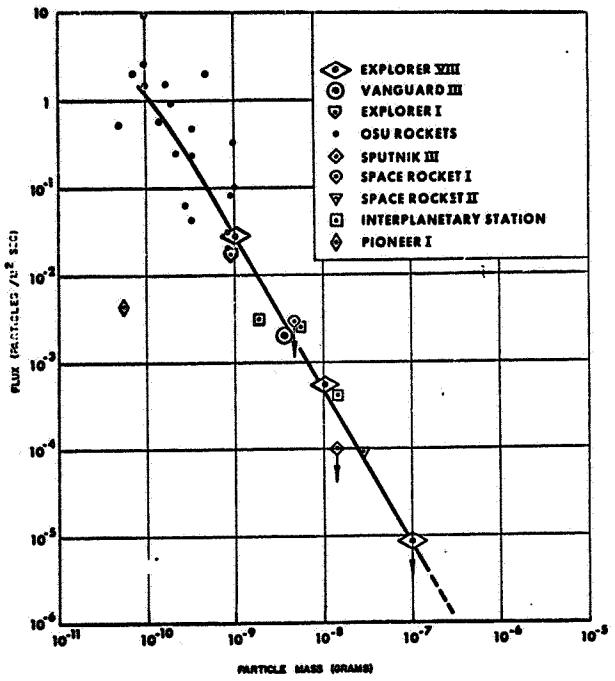


Figure 3-1. Micrometeoroid Flux Near the Earth

b. Micrometeoroid contributions of ions and ionizing particles in the ionosphere, effect on rainfall, noctilucent cloud formation, and zodiacal light.

c. Lunar, cometary, and asteroidal micrometeoroid replenishment mechanisms.

d. Relation of the solar F corona and micrometeoroids near the sun.

e. Verification of the Poynting-Robertson effect.

f. Interaction of the solar radiation pressure and micrometeoroids.

g. Physical and chemical composition of micrometeoroids and their growth in space.

h. Study of hypervelocity impact phenomenon using micrometeoroids, including study of secondary particles created by such impacts.

3.3.4 Micrometeoroid Sensors

Spacecraft micrometeoroid experiments attempt to measure a particle's velocity, energy or momentum, course direction, and the number of such particles encountered within a given time. Actual particle sizes are not easily measured, although some direct measurements have been made on "Venus Flytrap" captured particles. Particle mass is generally derived from velocity and momentum information. Particle density is inferred from mass and assumed size, and conversely. Common types of sensors and their characteristics are listed in the following subsections.

3.3.4.1 Microphonic Detector

A diaphragm acting as a sounding board is attached to a piezoelectric crystal. When a particle strikes the diaphragm, the crystal generates a pulse whose amplitude is related to the particle momentum. A pulse height analyzer is used to record the particle momentum data, and a counter is used to count the number of impacts per unit time. Spacecraft altitude information allows a rough guess as to the impacting particle radiant, since the location of the sensor on the spacecraft's exterior is known. Calibration of the pulse amplitude-particle

momentum relation is difficult on Earth, and this data must be extrapolated to the range of momentums encountered in space. Nevertheless, this type of instrument has been used to obtain 90 to 95 percent of all space micrometeoroid data presently available. Typical sensitivities are 10^{-8} to 10^{-11} kg-m/sec. This type of detector is useful on spacecraft only when there are no other acoustical sources present; motors and relays mask the measurements ([8], Chapter 8).

3.3.4.2 Impact Light Flash (Scintillators) and Impact Ionization Detectors

The impact of micrometeoroids on a suitable target results in a short light emission. The emission intensity is related to the incidence particle energy and may be analyzed by a photosensitive device and a pulse height analyzer-counter arrangement similar to that used with the microphonic detector. One arrangement was a 2000 Å thick target of aluminum evaporated onto the face of a photomultiplier tube. These devices also suffer from calibration difficulties but are extremely sensitive (10^{-16}g).

The impact ionization detector collects free charges ejected from the surface of a suitable target when impacted by a hypervelocity particle. Calibration is a difficulty; however, it appears that the charge Q_c collected per impact depends on the third power of the particle velocity and on a constant K related to the target material such that

$$Q_c = \frac{KE_p V}{A} \qquad (3\text{-}3)$$

where E_p = particle energy

V = particle velocity

A = particle atomic weight ([8], Chapter 8).

Clearly, some assumptions about an "average" Å are necessary when this detector is used in a space experiment. Impact velocity may be obtained by a simple analysis of the charge collected. It has been suggested that several such detectors placed in series could be used to determine the energy loss of an impacting particle during penetration of each of the successive stages. This

proposal suggests that by partitioning each detector stage into discrete areas and adaptively monitoring, micrometeoroid flux radiants could be determined quite accurately during periods of peak activity.

3.3.4.3 Covered Photocells

A thin opaque coating is deposited over the sensitive surface of a light sensor, such as a solar cell. Impacting micrometeoroids systematically expose the photosensitive surface to light from the sun or a standard source. The detector then measures the number and size of penetrations as they occur, provided the light increments at detector output are continuously monitored so that each penetration is observed.

At the conclusion of the micrometeoroid experiment, the same photomultiplier with the target covering replaced by suitable filters could be used for direct observations of the intensity and polarization of zodiacal light (4750 to 8500 Å). Airglow might also be examined with the proper filters being mechanically indexed to position in front of the photomultiplier as desired.

3.3.4.4 Erosion Gauges

Thin strips of resistive material may be placed on the outer surface of a spacecraft. An increasing resistance of the strip is a measure of material erosion due to impact of micrometeoroids. This type of instrument is sensitive to the long-term effects of very fine particles, and the output information need be sampled only infrequently.

3.3.4.5 Chemical Activity Detector

Although no practical detectors of this type are presently available, it appears possible to develop spacecraft systems that would capture interplanetary micrometeoroid material and analyze these samples. Some type of micro-qualitative chemical analysis or catalytic activity analysis could be an adjunct experiment to automated biological laboratories that have been proposed for

planetary exploration.[9] It would be sensible, in fact, to compare interplanetary dust with surface samples obtained on other planets.

3.3.4.6 Puncture Counters

There are several classes of puncture or particle penetration counters that have been used successfully to measure micrometeoroid impacts on spacecraft. Examples are pressurized cells, steel-covered wire grids, and copper-wire cards. Used alone, these devices give only gross knowledge about impacting particle energies, since the target penetration data on small mass-hypervelocity particles in space is extrapolated from the lower velocity-large mass particle penetration experiments used in earth calibration. It is important to note that each of these types of sensors is destroyed after one successful puncture so that a large number must be carried on each mission. This type of sensor would not be preferred if versatility or accuracy is a mission requirement.

The thin film capacitor is a more versatile puncture sensor, since it is self-healing and thus reusable. Sets of these sensors, each with a different penetrability, may be used as a simple micrometeoroid energy spectrum analyzer. Regardless of their inadequacies, puncture counters have been used advantageously in past experiments.

3.3.4.7 Multifunction Micrometeoroid Sensor

Figure 3-2 depicts an adaptive multiple-sensor system capable of measuring a number of important micrometeoroid characteristics. With a number of these sensors deployed about the spacecraft and with the craft attitude information, one can determine impacting particle velocity and velocity radiant, kinetic energy, mass, and charge. Spacecraft translation gives information on the spatial gradient of these quantities.

As depicted, a particle entering the detector passes through an induction coil and two thin-film ionization detectors. The voltage response of the induction coil is analyzed and gives information about the charge (if any) on the micrometeoroid. The ionization detector outputs are used to successively turn on and

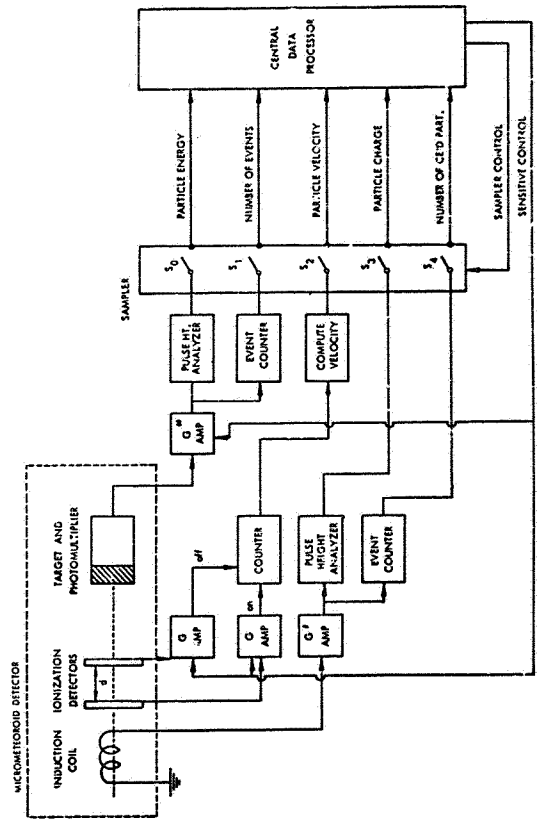


Figure 3-2. Multimode Micrometeoroid Detector with Variable Sensitivity and Data Rate (Ancillary Buffers, Clocks, and Reset Signals Not Shown)

turn off a time-of-flight counter from which the particle velocity is obtained. Finally, the particle strikes a fast response light flash detector whose output is analyzed to obtain an estimate of the particle kinetic energy. Particle mass is obtained by a simple calculation, since the velocity estimate is known. The sensitivity of the instrument can be controlled by amplifier gain settings. A calibration flash may be included in the package.

The data rate of such a sensor may be inferred from the statistics of the most recent impacts. This calculation is based on a model in which the number of electrons n liberated within the photoelectric detector is Poisson-dependent on the micrometeoroid impact energy ϵ. If k is a constant of the detector design and $\overline{m}$ is the average number of liberated electrons, let k be defined by

$$\overline{m} = k \tag{3-4}$$

and

$$p\,(m/\epsilon) = \frac{(k\epsilon)^m}{m!} \exp(-k\epsilon) \tag{3-5}$$

Any sensor may be thought of as an input/output device in an information channel. In this case the distribution of energetic particles in space may be thought of as a memoryless source driving the photoelectric detector through which there is a resulting average information flow. The goal of an adaptive sensor system would be to optimize this flow of information by proper adjustment of parameters, such as sensitivity and sampling rate.

3.4 SOLAR FLARES

3.4.1 Background

Solar flares are explosive outbursts of electromagnetic radiation and involve violent ejection of ionized matter from the vicinity of sunspot groups. The understanding of these solar disturbances depends on one's ability to analyze the interaction of very intense solar magnetic fields with dense plasmas. The

radiation and energetic particles generated by flares produce significant hazards to space flight and serious interruptions of ionospheric propagation-dependent communications systems.

3.4.2 Present Knowledge

Solar flares are the primary source of transients in the natural flux of particles expelled by the sun. They are classified into five divisions according to the area of the sun's surface over which the flare occurs:

a. Subflares, area A 100 millionths of the sun's visible hemisphere

b. Importance 1 $100 < A < 250$ millionths of the sun's visible hemisphere

c. Importance 2 $250 < A < 600$ millionths of the sun's visible hemisphere

d. Importance 3 $600 < A < 1200$ millionths of the sun's visible hemisphere

e. Importance 3+ $A < 1200$ millionths of the sun's visible hemisphere.

Most flares impart nonrelativistic energies to the ejected particles. Typical proton energies are 10 to 40 mev, and flux densities of $10^4/cm^2$ sec have been observed by satellites. These intensities are not expected on Earth, since the atmosphere and the geomagnetic field provide an effective shield.

Approximately 10 intense flares producing relativistic particles have been observed since the first solar flare was recorded in 1942 ([8], page 613). The most intense of these flares was one detected February 23, 1956, when particles of 1 to 30 bev energies arrived at the Earth 19 minutes after optical detection of the flare and 8 minutes after the peak optical activity. Particles of this energy produce significant ionospheric disturbances and through upper atmospheric collisions produce secondary neutrons, μ-mesons, and auroral effects which may be detected at ground level.

Flares need not produce such energetic particles to create important effects on Earth. Less energetic particles (1 to 100 mev) penetrate into the geomagnetic field until the particle energy density approximately equals the energy density of the magnetic field. This condition is thought to occur at 10 to 15 Earth radii.

Some of the particles enter the Earth's trapped radiation belts and enhance the particle population there. If a large number of protons are trapped, the westward-directed component of the longitudinal current is enhanced. This "ring" current appears at about 5 R_e and contributes to the development of geomagnetic storms.

Almost all presently available information concerning solar flares comes from observational data. There have been no satisfactorily complete theoretical treatments of the flare phenomenon, since the solar plasma-magnetic field interactions are difficult to model. One reason for slow progress in this field has been a lack of data on solar ultraviolet emissions during major flares. The major observable changes in the sun's emission characteristics during a flare do, in fact, occur in the 1 to 1500 Å region of radiations filtered out by the atmosphere. This information was denied the physicist until recent satellite and space probe data became available.

3.4.3 Future Solar Flare Studies

Scientists are presently examining data from recent major solar flares. Although there are many observed effects of flares, comprehension of the mechanisms is only now allowing the formulation of a reasonably logical experimental attack. Areas of future solar flare studies may include some of the following:

a. Determination of a soft X-ray and of a UV source solar map

b. Variation of soft X-ray and UV rays with solar activity

c. Solar flare particle energy and mechanisms of energy transfer to the magnetosphere and atmosphere

d. Transient distortions of the magnetosphere with flare activity

e. X-ray, gamma ray, and positron energy spectra and temporal distribution in a flare

f. Distortion of interplanetary magnetic fields during solar flares

g. Cosmic rays and the Forbush effect

h. Radiation shielding criteria for design of manned spacecraft.

3.4.4 Solar Flare Sensors

As mentioned, the solar radiation during periods of flare activity is both electromagnetic and particulate. The electromagnetic radiation spectrum encompasses soft X-rays < 1 Å to the ultraviolet region at about 300 Å. High energy electrons, protons, and heavy nuclei are present in the flare plasma. To study solar flares comprehensively, it is necessary that a sensor system be capable of making both electromagnetic and particle measurements.

3.4.4.1 Charged Particle Detector

Several types of charged particle detector exist; however, the unifying principle of operation is that each measures the energy lost by a particle as it passes through the detector. Some of these are the Langmuir probe, ion trap probe, Faraday cup, and the soft electron spectrometer.

3.4.4.2 Neutron Detector

Solar flare radiations incident on upper atmospheric constitutents produce energetic neutrons. The neutrons possess no charge and have little difficulty in penetrating to the ground. Monitoring of neutron flux was an important method of solar activity study before the advent of satellites and space probes. The convenience of ground-based equipment makes the neutron monitor valuable for such purposes as obtaining time and space correlations of neutron fluxes and ionospheric or geomagnetic disturbances.

Since neutrons are not charged, they do not cause ionization in matter. However, neutrons can be detected through the reaction $B^{10} + n \rightarrow Li^7 + \alpha + Q$. In this equation the energy released (Q) is about 2.3 mev. This appears as kinetic energy of the alpha particle and the lithium nucleus, and since these are charged particles, they will produce ionization. The number of ionization pulses produced by this reaction will be proportional to the incident neutron flux.

In order to measure the neutron flux from the Earth, two ionization tubes filled with BF_3 gas are used. One tube is enriched with the B^{10} isotope of boron,

thus yielding a high counting rate for neutrons. The other tube, which has identical geometry, gas pressure, and collector voltage, is filled with BF_3 gas having a low concentration of the B^{10} isotope. Thus, while the background counting rate for all other types of radiation is the same for both tubes, they have different counting efficiencies for neutrons, and the neutron flux can be calculated from the difference in counting rates.

3.4.4.3 Ultraviolet Spectrometer

Ultraviolet spectrometers may be designed to measure radiations in the 50 to 1500 Å region. The same type of device is called a soft X-ray spectrometer when operated in the 1 to 16 Å region. In either case, the typical sensor consists of a device that produces an output signal proportional to the intensity of the incident radiation. Sensor signal conditioning, such as pulse height determination, is used to obtain a telemetry signal related to the sensed radiation intensity.

3.4.4.4 Ionization Chamber Ultraviolet Spectrometer

In this case, a gas-filled ionization chamber has a window of some special material which acts as a bandpass filter over a range of design frequencies. As an example, a beryllium window passes the soft X-rays of the 1 to 8 Å region while calcium fluoride passes the UV region 1225 to 1350 Å ([8], Chapter 2). Unfortunately, the ionization chamber can give ambiguous results, since it may also respond to incident charged particles. When operated in the proportional mode, the ionization chamber has an output current proportional to the radiation passing through the window. The band of frequencies studied at one time depends on the window characteristics.

3.4.4.5 Grazing Incidence Ultraviolet Spectrometer

The grazing spectrometer utilizes diffraction of a collimated sample of the incident radiation to maximize the resolution of incident stimuli. The resolved spectral lines are focused on a surface within the instrument where their positions

and intensities are sensed by arrays of photodetectors, by direct photography, or by ionization chambers. The type of detector used depends on the range of frequencies being studied.

In the case of photodetectors or ionization chambers, the usual signal conditioning equipment is required to translate measured currents into a useful form.

3.5 COMPUTER REQUIREMENTS

Micrometeoroid and solar studies require a very broad range of instruments which typically measure some specific particulate or electromagnetic radiation energy and flux intensity. Since both phenomena are poorly understood, a large number of related experiments would be of interest. If these experiments can be performed using unified but flexible sensor configurations, an important saving in equipment may be realized while more comprehensive and useful data is obtained. The concept of an adaptive multimode sensor system in conjunction with a multiprocessor is an approach to realizing this type of optimum spacecraft payload.

In general, the data processing that will be performed upon the sensor outputs will be statistical. Any data reduction process involves the calculation of statistical measures, and the parameters one wants to measure are usually statistical parameters. In most cases, the measures described in Section 5 of this report will apply in one form or another. Thus, the computer servicing this subsystem must be equipped with a high-speed arithmetic instruction set. The memory requirements will vary and will depend upon the on-board data processing requirements. Our approach will be to design a system with a modular memory so that the necessary capacity can be provided at any time. The instruction set, however, can remain constant. In general, a separate I/O channel will be allotted for each sensor.

Section 4

NAVIGATION AND GUIDANCE, CHECKOUT AND MONITORING, AND ENERGY MANAGEMENT SUBSYSTEMS

4.1 GUIDANCE AND NAVIGATION SUBSYSTEM

To determine the computer requirements of this subsystem, one must consider the navigation and guidance algorithm that will probably be used in the 1970's. The trend in this field is toward determining the optimum trajectory for a space vehicle, whether for navigating for a deep space probe or for determining the orbit injection point for an orbiter vehicle. This is done by continually recalculating the flight trajectory based on the craft's present position. If the calculation shows that the final end-point will be in error, then control signals are generated and a new course is set for the craft. This process is repeated continually until the craft has reached its destination.

The above guidance algorithm is implemented on board the space vehicle in the form of a computer program. This program is executed in a small, general-purpose, digital computer located on board the vehicle.

The computer requirements for implementing an advanced guidance algorithm such as this are as follows. The overall computer must be faster than the present day guidance computers by at least a factor of 5-10. This means that an instruction repertoire of approximately 25-50 instructions is required with special emphasis on instructions used in matrix manipulation. Approximately 60 percent of the guidance algorithm is devoted to matrix and vector calculations. The most frequently used instructions are ADD, subtraction, multiply, and load and store. Of these, the last two definitely require an indexing capability. An interesting fact is that generally the current guidance and control computers do not have an indexing feature.

The typical guidance program size, including data storage, is approximately 5000 words, assuming a word size of 16 bits. This estimate is for programs written in Fortran. For machine language programs, the size can be reduced by 20 percent.

The time required to calculate a new trajectory and close the guidance loop is termed the period of the algorithms. On present guidance computers, this period is about two seconds. Guidance algorithms, however, are becoming more and more complex, and it is estimated that by 1970, the speed, capability, and storage requirements of the guidance computer will be increased by at least an order of magnitude.

4.2 CHECKOUT AND MONITORING

Three levels of testing are required for conducting flight operations of an orbiter vehicle: monitoring—essentially a continuous check of certain system parameters; confidence testing—a detailed check of system parameters prior to the execution of a critical operation or event; and diagnostic testing—a detailed check of a suspected or actual faulty system.

All of the above checkout levels are required in the prelaunch and inflight phases of a mission. A programmable digital computer seems to be ideal and is the natural candidate for carrying out all of these checkout requirements. In fact, for an unmanned mission a computer is almost a requirement in order to maximize the system reliability. The number of test points on an unmanned orbiter will be large. Past experience shows that 2000-4000 points is normal. In the 1970 time frame, as missions become longer and the reliability requirements become more stringent, the number of test points will double or even triple.

Two broad classes of diagnostic approaches can be taken. The first is for diagnosing failures in digital systems; the methods used are well known and have been used for years. The second class concerns the diagnosis of analog failures; these techniques are less well known. For example, how does one detect that the output of an R.F. amplifier is slowly degrading. A straightforward approach to

this problem is to convert all analog signals to digital signals so that the computer can analyze them. This problem may be alleviated as time goes on since the trend is toward replacing analog devices with digital counterparts.

In general, diagnostic programs have few arithmetic operations, if any. Mostly they fall into the data manipulation class and the comparison class. Thus, the arithmetic class of instructions is not required. The totality of diagnostic programs will require a significant amount of bulk memory space, on the order of 10,000 words (16 bits/word) or greater. Of course, this estimate of memory size is a function of the tasks to be done and the way in which they are programmed. This variance points out the need for a system with modular memories so that the memory can be expanded easily if necessary. Section 6 of this report gives some general computer specifications which apply to this subsystem.

4.3 ENERGY MANAGEMENT

Nuclear fission, isotope decay, chemical fuels, and solar power are energy sources suitable for the operation of space power systems. Electric power is generated from these sources by different means such as energy converters. Such converters include photovoltaic, galvanic couple, thermionic, thermoelectric, and thermal-mechanical-electric types. Because of the power level requirements, the long mission durations, and the nuclear propulsion systems of the subject missions, the application of isotope and chemical energy sources and of the photovoltaic and galvanic couples is excluded. Strontium-90, polonium-210, cerium-144, promethium-147, curium-244, and cesium-137 have been evaluated on the basis of useful mission life, power mass density, shielding requirements, cost, and availability. None of these meet both the useful mission life and availability requirements. In the fuel-cell galvanic converter, the highest efficiency of all chemical converters and the minimum mass for long-period operation are exhibited. On the basis of a 420-day mission, a 20-kwe average load, and a 0.35-kilogram per kilowatt reactant consumption, the fuel alone would involve a mass of 70,000 kilograms.

Photovoltaic systems have been proposed for use in Earth orbital systems. In these systems, power in the 10- to 20-kwe range is required. In the vicinity of Mars, the solar intensity is approximately 43 percent of that in Earth space. The effect of the reduced solar intensity on cell output is partially offset by the fact that the cell operation occurs at a lower temperature. This cooler operation in turn produces a higher efficiency. The Jet Propulsion Laboratory, in connection with the Mariner program, determined the relative performances of the photovoltaic systems as follows for full sunlight in the vicinity of Earth and Mars, respectively:

a. Solar irradiation in mw/cm^2—135, 58.8

b. Operating temperature in degrees C—55, -10

c. Milliwatt output per 1 x 2 cm cell—21.5, 10

d. Cell efficiency—8.38, 8.96

e. Watts/ft^2—8.66, 4.0

f. Watts/meter^2—94, 43.4

g. Watts/pound—8.56, 3.96

h. Watts/kilogram—18.9, 8.75.

Consideration was given to the possible application of a solar photovoltaic system to power an austere type, minimum-crew-size mission. With power levels in the 5- to 10-kwe range, the photovoltaic system may be competitive with the nuclear systems. Tests of silicon solar cells were performed at various academic and industrial institutes such as U.C.L.A nuclear laboratory. Results of these tests demonstrated that the power output of the devices was degraded by 50 to 75 percent when they were exposed to an integrated neutron flux of 10^{14} nvt. Thus, any further consideration of the photovoltaic converters was not warranted. As a result of several analyses performed by these organizations, solar and nuclear energy sources remain as competitive systems to be evaluated for the specific missions. The thermoelectric, thermionic, and thermal-mechanical-electric converter systems have been selected for continued analysis.

The role of the on-board computer for this subsystem will be to check out various parts of the subsystem and monitor the loading which must be distributed as evenly as possible among various subsystems of equal priority. The monitoring and checking functions will constitute major tasks of the computational reliability requirements. Some of the power management and loading requirements for a typical orbiter appear in Table 4-1. This table summarizes some of the power requirements of various payload subsystems as projected for the missions previously indicated.

Table 4-1
SYSTEM POWER REQUIREMENTS

Payload Subsystems	Load in KWE		
	Orbiter		
	Average	Peak	Emergency
Navigation and Guidance	0.1	0.2	0.2
Attitude Command	0.1	0.3	0.2
Data Gathering: Optics	2.1	2.9	0
Data Handling	3.1	3.1	0
Communications	4.1	8.3	0.5
Vehicle Maintenance and Miscellaneous	0.9	1.3	0.6
SUB TOTAL	10.4	16.1	1.5
Estimate of Hydrogen Reliquification	15	30	10
Power Conditioning and Distribution Losses	3	5	1.5
TOTAL	28.4	51.1	13.0

Section 5

STATISTICAL ANALYSIS

The sensor output signals that we will be dealing with in this study will, in almost all cases, be random; we will assume they are stationary. In this section, several averages will be described, two of which, the probability density function and the autocorrelation function, will completely describe any random signal. Two other useful operations, the mean value calculation and removal and the Fourier transform of the autocorrelation function are also given. All of these functions are important and should be included in any on-board data processing library. The probability density function provides information concerning the amplitude variations of the signal such as its mean and rms values. Random signals are best described by the statistical distribution of their deviation from a certain mean level. However, this measurement gives no information whatsoever about the fluctuations themselves. It does not tell whether the peaks and troughs of the signal are close together or spread out in time, i.e., the rate of fluctuation. The Fourier transform of the autocorrelation function is the spectral density function. This function provides information about the frequency content and the randomness of the signal. A random variable can be considered as fully described statistically when its probability distribution function and autocorrelation function are known. These functions are calculated by converting the time continuous sensor output signals to digital formats using an analog-to-digital converter and by using the on-board digital computer to perform the necessary calculations. The equations for these calculations are presented in this section in parametric form and will be used in determining the computer requirements.

5.1 MEAN VALUE CALCULATION AND REMOVAL

The mean value, m_x, of a stationary signal, x(t), is given by

$$m_x = \lim_{T \to \infty} \frac{1}{2T} \int_{-T}^{T} x(t)\,dt \tag{5-1}$$

and the computer program approximates this by

$$m_x \approx \frac{1}{N} \sum_{i=1}^{N} x(i\Delta t) \tag{5-2}$$

where $x(i\Delta t)$ is the ith sample value, Δt is the time between samples, and N is the number of samples in the record. This mean value is subtracted from each sample value yielding new sample values, $\Psi(i\Delta t)$, where

$$\Psi(i\Delta t) = x(i\Delta t) - m_x \tag{5-3}$$

The discrete random variable $\Psi(i\Delta t)$ is thus described by a set of N samples with a zero mean.

5.2 PROBABILITY DENSITY AND DISTRIBUTION CALCULATION

The next function of interest is the probability density function p(x) given by

$$p(x) = \frac{d}{dx} \{P(x)\} \tag{5-4}$$

where P(x) is the probability distribution function. P(x) is in turn defined as

$$P(x) = \lim_{T \to \infty} \frac{T_x^+}{T} \tag{5-5}$$

where T_x^+ is the amount of time during the time interval T that the signal has a value greater than x. The computer program approximates Equation (5-4) as

$$P(k\Delta x) \approx \frac{N_k}{N} \quad 1/\Delta x \tag{5-6}$$

where N_k is the number of samples that have values greater than $\Delta x(k - 1/2)$ but not greater than $\Delta x(k + 1/2)$, and Δx is a convenient small increment of x. The increment Δx is chosen to be 0.1 percent of the range of x.

To evaluate the probability density function about the mean, Equation (5-6) is applied to a record of Ψ from Equation (5-3).

The probability distribution function given by Equation (5-5) is approximated in the computer program by

$$P(K\Delta x) = \sum_{k=1}^{K} p(k\Delta x) \tag{5-7}$$

As shown by Equation (5-7), the probability distribution function is obtained from the probability density function.

5.3 AUTO AND CROSS-CORRELATION CALCULATION

The cross-correlation, $R_{xy}(\tau)$, between two stationary time signals, x(t) and y(t), is defined as

$$R_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} x(t + \tau)\, y(t)\, dt \tag{5-8}$$

and is approximated in the computer by

$$R_{xy}(j\Delta\tau) \approx \frac{1}{N\Delta t - jg\Delta t} \sum_{i=1}^{N} x(i\Delta t + j\Delta\tau)\, y(i\Delta t)\, g\Delta t \tag{5-9}$$

where g is a positive integer that may be greater than unity to reduce computation time. The denominator term 2T, in Equation (5-8), is replaced by $N\Delta t - jg\Delta t$ because the shifting and multiplying evident in Equation (5-9) causes $|jg\Delta t|$ sample points of each signal to be multiplied by zeros introduced by terminating the data records at N sample points. The index j is not allowed to have values less than -N/10 nor greater than N/10 to ensure a good approximation to Equation (5-8) by Equation (5-9).[10]

To obtain an autocorrelation function, $R(j\Delta\tau)$, the same data record is used for the x and y variables in Equation (5-9). Since an autocorrelation function is an even function of τ, only positive values of j in Equation (5-9) are necessary.

5.4 POWER SPECTRAL DENSITY

The power spectral density function, S(f), is the Fourier transform of the autocorrelation function, thus given by

$$S(f) = \int_{-\infty}^{\infty} R(\tau)\, e^{-j2\pi f}\, d\tau \tag{5-10}$$

Since the autocorrelation function is an even function of τ, Equation (5-10) simplifies to

$$S(f) = 2\int_{0}^{\infty} R(\tau) \cos(2\pi f\tau)\, d\tau. \tag{5-11}$$

Equation (5-11) is approximated in the computer by

$$S(q\Delta f) = \left[R(0) + 2\sum_{j=1}^{J} R(j\Delta\tau)\cos(2\pi q\Delta f j\Delta\tau) \right] \Delta\tau \qquad (5\text{-}12)$$

where q is the index of the power spectral density sample points Δf apart and J is the largest value of the index j used in calculating the autocorrelation function from Equation (5-9).

Section 6

COMPUTER SPECIFICATIONS

Based on the requirements of the subsystems described in Sections 1 through 5, a set of computer specifications has been derived. These specifications are adequate, in our opinion, to service the needs of any single subsystem. They are also general enough to allow the instruction list to be modified where applicable. For example, if a subsystem requires that no arithmetic operations be performed, then the arithmetic instructions can be eliminated. The memory requirements may vary, but this is not critical since one of the requirements for the computer is that the memory be expandable in small increments or modules.

The general specifications outlined herein will serve as a guide during the second phase of the study and are intended only to give an idea of the computer capability required to service each of the various subsystems. Although the multiprocessor concept is organized differently, it will provide the same computing capability to each subsystem. Many of the features presented in our representative computer are included in the multiprocessor design.

The hardware requirements will be minimized by using a byte (8 bits) data transfer format and efficient use of hardware registers to perform multifunctions. This 8-bit byte will also simplify the data transfer and switching between modules.

The computer is subdivided into a memory, a processor, and an I/O section. All signals entering or leaving the computer are digital in nature, thus analog-to-digital and digital-to-analog converters must be provided in the input/output sections.

The computer is provided with a 1.0 microsecond, 8192-byte core memory which communicates with the processor section over a parallel 8-bit transfer path. The computer will be capable of addressing 65K bytes of main store, thus providing an expandable memory. The processor contains the facilities for addressing main storage, for reading or writing information, for arithmetic or logical processing of data, instruction sequencing, timing, and for initiating the communication between storage and external equipment.

Working registers are located both in hardware and in core memory storage. A total of 25 to 50 instructions will be required for each subsystem. Processing rates on the order of 100,000 operations per second will be attained by optimizing the machine organization, monolithic logic circuitry, and high-speed core memory modules.

6.1 DATA AND INSTRUCTION FORMATS

The computer transmits information between main storage and the processor in an 8-bit unit of information called a byte, the basic building block of all formats. The basic arithmetic operand is the 16-bit fixed point binary word (sign +15 bits) and is formed by two successive 8-bit memory words. Parity check on the memory byte is not used. Double precision arithmetic requires a 4-byte operand. The 4-byte operand is also used for products and dividends.

Three instruction word formats are used in the computer to provide ease of programming and to achieve good bit efficiency. These instruction formats are compatible with the 8-bit memory organization, being 8, 16, or 24 bits in length. All instructions of 16 or 24 bits cause a reference to main store. Some of the 8-bit instructions make no reference to main store.

The short instruction format is used for the most common instructions, thereby minimizing the program memory size. The longer (16-bit) instructions, however, provide greater flexibility and larger operand addressing. The 24-bit short format-immediate provides 16 bits of immediate data along with an 8-bit operation code. The first bit of each format is a flag bit which denotes a short

or long instruction format. The short format contains a 3-bit instruction code, and a 4-bit field which is used for operand addressing or for specifying the number of shifts.

6.2 ADDRESSING

Byte locations in storage are consecutively numbered starting with 0; each number is considered the address of the corresponding byte. A group of bytes may be addressed by the left-most byte of the group. Addressing capability permitting a maximum of 65,546 bytes is obtained by using a 16-bit binary address.

6.3 PROCESSOR ORGANIZATION

An 8-bit adder is used since it can be packaged as a module, thereby improving performance and cost packaging effectiveness. The adder executes all arithmetic and effective address update functions. In addition to the add function, the adder also has the capability to perform AND, OR, and exclusive OR functions.

The hardware registers of the processor section will also use modular packaging in order to obtain cost packaging effectiveness. All gating hardware will be included within the package. The following hardware registers will probably be included in the design:

a. Accumulator—This 16-bit register is the principal arithmetic register.

b. Multiplier-Quotient Register—This 16-bit register is a low-order extension of the accumulator.

c. Memory Buffer Register—This 8-bit register receives the output from the memory.

d. Instruction Register—A 14-bit register used to store the address of the next sequential instruction.

e. Operation Register—The operation register is an 8-bit register used to store the instruction code.

f. Memory Address Register—A 14-bit register used to store and select the proper memory address. The two least significant bits are configured as a counter.

g. Shift and Multiply Counter—This 5-bit counter is used to count iteration during shift, multiply, and divide instructions. The counter allows for a maximum of 32 shifts in either direction.

An interruption system will permit a change in state of the computer in response to conditions external to the system. The contemplated interrupts are

a. Interrupt No. 0—Major priority for such interrupts as power-off interrupts.

b. Interrupt No. 1—Noncritical priority interrupts that can be interrogated by program control. An example of this is a real-time clock interrupt.

An interrupt consists of storing the updated Instruction Counter (IC) and executing an automatic indirect branch. The IC is saved in a core storage location permanently assigned to the interrupt. This feature will probably be used extensively in the actual design of the multiprocessor system.

6.4 MAIN STORAGE

The computer serving a subsystem will use an 8192 word by 8-bit destructive readout core memory. The memory has a 1.0 microsecond read-write cycle time Each cycle consists of a read followed by a write operation. The machine cycle time is 1.5 microseconds (i.e., the computer initiates a start memory every 1.5 microseconds). A coincident current (3-D) selection scheme is used.

The memory uses external data buffer (memory buffer) and memory address lines (14) sent from the processor section. The system is organized such that the "X" half select driver is clocked first since it drives the long dimension. The "Y" current thus has time to reach full amplitude at a slower rate, thereby reducing driver supply voltage requirements.

6.5 INPUT/OUTPUT

The input/output section of the computer will perform the following functions:

a. Data Formatting—The I/O section provides for byte to serial data conversion to allow the processor to interface with other on-board systems. An analog-to-digital converter will also be provided to convert the analog sensor outputs to digital bytes. In some cases a digital-to-analog converter will also be provided for the purpose of generating analog control signals.

b. Real-Time Clock Generation—A real-time clock will be located in the I/O section and used to generate timed interrupts. The contents of the counter are periodically interrogated by the processor. Two low frequency clocks are also generated by the real-time clock.

c. Device Address Generation—The external device addresses are decoded in the I/O section.

d. Interrupt Control—The buffer and control logic associated with interrupts is a part of the I/O section.

6.6 COMPUTER PACKAGING

The computer will be packaged in a ruggedized aluminum structure and should occupy no more than 0.15 cubic feet or about half the size of the present day space computers. It is felt that this reduction in size is reasonable for the 1970 time frame. The logic will be assembled using Large Scale Integration techniques where upwards of 100 to 1000 basic NAND logic elements will be contained on a single chip.

6.7 SOFTWARE SUPPORT

The support programming system will assist the programmer in the production of operational software for the computer by providing:

a. Programming aids that will enable a program to be expressed in a language which is readily understood

b. Facilities for storing, modifying, and accessing precoded programs

c. A program checkout system to ensure operational program readiness

d. A technique for loading in new programs and data telemetered from earth stations

e. Effective documentation manuals.

In addition, an executive program system must be provided to handle the problems of reconfiguration and the control problems associated with multiprocessor systems.

6.8 INSTRUCTION LIST

A representative list of instructions and execution times is presented in Table 6-1. It is felt that these instructions are adequate to handle any of the subsystem functions; however, instructions may be added or deleted as the design progresses.

Table 6-1

INSTRUCTION LIST

OPERATION	FORMAT	EXECUTION TIME
ADD	Short	7
ADD	Long	9
ADD	Immediate	4.5
ADD DOUBLE	Long	12
AND	Immediate	4.5
AND	Long	9
ARITHMETIC COMPLEMENT	Short	1.5
BRANCH	Immediate	4.5
BRANCH	Long	6
BRANCH BACKWARD ON MINUS	Long	3
BRANCH BACKWARD ON PLUS	Long	3
BRANCH BACKWARD UNCONDITIONAL	Long	3
BRANCH BACKWARD ON ZERO	Long	3
BRANCH FORWARD ON MINUS	Long	3
BRANCH FORWARD ON PLUS	Long	3
BRANCH FORWARD UNCONDITIONAL	Long	3
BRANCH FORWARD ON ZERO	Long	3
BRANCH INDIRECT	Long	6
BRANCH INDIRECT AND STORE IC	Long	9
COMPARE	Immediate	4.5
COMPARE	Long	9
COMPARE DOUBLE	Long	12
DIVIDE	Long	27
EXCHANGE A AND Q	Short	1.5
EXCLUSIVE OR	Long	9
INPUT/OUTPUT CONTROL	Long	12
LOAD A	Short	7.5
LOAD A	Long	9
LOAD A	Immediate	4.5
MULTIPLY	Short	2[illegible]
MULTIPLY	Long	27
NO OPERATION	Short	1.5
OR	Long	9
SHIFT LEFT DOUBLE	Short	3-9
SHIFT RIGHT ARITHMETIC DOUBLE	Short	3-9
SKIP ON CARRY	Short	1.5
STORE A	Short	7.5
STORE A	Long	9
STORE DOUBLE	Long	12
STORE IC	Long	9
SUBTRACT	Short	7.5
SUBTRACT	Long	9
SUBTRACT DOUBLE	Long	12
TALLY	Long	6-12
ZERO Q	Short	1.5

Section 7

MULTIPROCESSING AND SWITCHING SYSTEMS

7.1 MULTIPROCESSING CONCEPT

The key requirement in a spaceborne computer system is the capability to assume different configurations when part of the system fails, and also when the mission requirements change. Another equally important feature is the capability to execute more than one program in parallel, or to execute several subroutines and sequence instructions within a program simultaneously. It is evident that the former capability ensures high reliability in the case of equipment failure, and high flexibility in the case of mission changes. The latter requirement provides the speed and accuracy required to solve complex real-time problems of supreme urgency. The emphasis on these requirements has led to the development of multiprocessing and multicomputer systems which emphasize the characteristics of reliability, efficiency, flexibility, and performance capability.

7.2 MULTIPROCESSING STATE-OF-THE-ART

There are two competing trends in modern data processing, the trend toward large, complex, centralized systems and the trend toward small, simple, decentralized systems. The centralized system, generally, uses a powerful computer to execute several tasks in a time shared manner. The reliability of such a system is poor since any catastrophic malfunction will cause a complete system breakdown. This makes the centralized system more prone to catastrophic failures and hence less desirable in space missions where reliability is of a great concern. Figure 7-1 is an example of such a configuration. Decentralized systems, generally more difficult to program, are highly reliable. A failure in any computing element simply

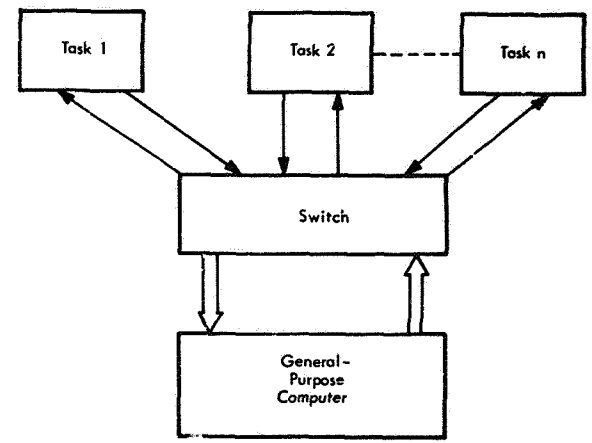


Figure 7-1. A Centralized System

reduces the overall system availability and hence reduces the computing rate; but never causes complete breakdown. This graceful degradation results from the fact that the system takes on different configurations as the units fail. It is because of this feature that such systems are sometimes called "polymorphic."

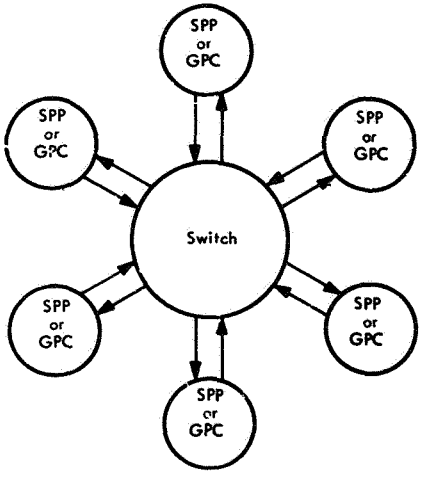
Previously proposed decentralized computer systems consisted either of many small special purpose computers each programmed to perform a specific task; or of many identical processing and computing elements interconnected via a switching network. In the latter case, each element could be a small general-purpose computer assigned to one or more tasks. If one computer should fail, or if the mission requirements change, the remaining elements can be assigned to perform that task or group of tasks. The inverse is also true, that is, any number of these identical computing elements may be assigned during any given time to service a task. An example of such a decentralized computer system appears in Figure 7-2.

Other configurations of a decentralized computer system are those proposed by Solomon and Holland. Solomon has proposed a system with central control, and Holland has proposed a machine with distributed control.

Other system organizations have been proposed by Unger, Hellerman, Slotnick, Seeber, Lindquist and many others. Most of these have been directed toward multiprogramming, where a single general-purpose computer coordinates the execution of several programs simultaneously to increase total system throughput. Seeber and Lindquist advocate the use of associative memories for data instructions and associative logic to control the parallel operations. Figure 7-3 shows a typical organization of an Associative Logic Parallel System.

7.3 A MULTIPROCESSING CONFIGURATION

One basic multiprocessing configuration that we have chosen for study is shown in Figure 7-4. The system is composed of "n" memory modules, "j" central processor modules and "i" input/output (I/O) modules. All modules are assumed to contain their own power supply. The memory modules, which are identical in size, store data in the form of 8-bit bytes. The central processor modules are all identical and each is assumed to contain an arithmetic element,



Code:

SPP = Special Purpose Processor

GPC = General Purpose Computer

Figure 7-2. A Decentralized System

Figure 7-3. Decentralized Configuration of an Associative Logic Parallel System

Figure 7-4. A Typical Decentralized Multiprocessing System

an instruction control element, and a control element. The I/O modules are also identical and merely form an interface between the many and varied on-board I/O devices such as the sensors, communications receivers, and data acquisition systems.

All of these modules communicate with one another via a computer-controlled automatic interconnection or switching system. Thus, all of the information flow between the elements passes through this switching mechanism. We propose that the switching system be used to partition the overall set of central processing, memory and I/O modules into a number of subsets of these elements where each subset may contain one or more of each of the three types of modules and is connected as a small general-purpose computer. Each of those small computers in turn is used to control and implement one of the several subsystems aboard the vehicle such as the guidance and control or the communications subsystem. An executive program controls the configuration and reconfiguration of the small general-purpose computers.

The prime reason for selecting the configuration shown in Figure 7-4 is its capability for satisfying all of the requirements of a reliable system. In the event that a module should fail, the switching system under executive control allows the machine to reconfigure itself and to switch in a new module in place of the faulty unit. When all of the space units have been utilized, the overall system capacity is reduced and the performance is degraded. The important thing is that the degradation is graceful rather than catastrophic.

As previously mentioned, the on-board computers are configured from a pool of computing, memory, and I/O modules. The key to the configuration and reconfiguration of these modules into many small general-purpose computers lies in the switching system that establishes the data-flow paths between elements.

There are three basic orthogonal switching techniques that can be used to implement such a computer switch: space division, time division, and code division. The following subsections will describe all three techniques.

7.4 SWITCHING

7.4.1 Orthogonality Concept

Two discrete N-dimensional vectors x, y are said to be orthogonal if their correlation coefficient $\rho(x, y)$ is zero. The correlation coefficient of such vectors is given in general by

$$\rho(x, y) = 1/N \sum_{i=1}^{N} X_i Y_i \qquad (7\text{-}1)$$

If x, y are vectors of 1's and -1's, then

$$\rho(x, y) = \cos \phi \qquad (7\text{-}2)$$

where ϕ is the argument between x, y. Thus, for x, y to be orthogonal $\phi = \pi/2$ and hence $\rho(x, y) = 0$. When the vectors x, y are binary codes, the correlation is given more generally by

$$\rho(x, y) = \frac{A-D}{N} \qquad (7\text{-}3)$$

where N is the length of the code; i.e., number of 1's plus number of 0's. A is the number of agreements of x with y, and D is the number of disagreements of x with y. Therefore, for the two codes to be orthogonal

$$A = B \qquad (7\text{-}4)$$

This result implies that the length "N" of each code must be an EVEN integer. If, however, the length is an ODD integer, then the condition of orthogonality becomes

$$\rho(x, y) = -1 \qquad (7\text{-}5)$$

This requires

$$D = A + 1 \qquad (7\text{-}6)$$

Equation (7-4) applies to both orthogonal and biorthogonal codes, while Equation (7-6) applies to transorthogonal codes.

Examples of orthogonal and biorthogonal codes are the first-order and higher-order Reed-Muller codes, respectively. These can be easily derived from the Hadamard matrix given as 1's and 0's by

$$a = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{7-7}$$

Any length code of this type can be constructed very easily from this matrix. If an 8-bit code is required, then a 4 x 4 matrix may be constructed as

$$\begin{pmatrix} a & a & a & a \\ a & \bar{a} & a & \bar{a} \\ a & a & \bar{a} & \bar{a} \\ a & \bar{a} & \bar{a} & a \end{pmatrix}$$

which will give rise to eight codes each eight bits long.

Examples of the transorthogonal codes, however, are the PN codes which are easily generated using maximal length linear shift register.

7.4.2 Time Division—Switching and Orthogonal Addressing

We have seen from the previous discussion that the problem of switching is common to all the multiprocessing system configurations. We have also seen that this problem is rather critical and that no elegant solution with high reliability exists at the present time. Our solution is based on the following:

a. Using orthogonal codes as address. Each code will represent a unique address of a module in the system.

b. Making both the data and the addresses orthogonal with respect to each other.

In order to comprehend the significance of these two concepts, let us consider a representative system organization such as the one shown in Figure 7-4. The multiprocessing system in this figure is a typical decentralized configuration with three diferent types of modules. There are "n" memory modules, "j" identical computing elements and "i" modules of input/output devices. Each module has a transmitter capable of addressing all other modules, and a receiver capable of recognizing its own address only. Each transmitter is essentially a code generator which can be set initially by the executive program to generate the code that will address the desired module. It is assumed that once a given configuration is selected by the executive program, the system will remain in that configuration for a definite time period. Thus, if a given computing element is assigned during a given time to one or more memory modules, it is possible that the same element will be assigned at some later time to other memory modules. These [illegible] assignments which will define a system configuration during any given time will depend on certain computational requirements. Such requirements will depend on the type of experiments to be performed and the type of parameter measurements to be taken during the specified time period.

The receiver in each module is a digital matched filter capable of correlating its locally stored address with the incoming one. When the two codes correlate, a pronounced spike whose amplitude corresponds to the correlated bits will be generated by the receiver indicating that the module has been addressed. Thus, any module may be asynchronously addressed without having to provide any interrupt feature. This relieves the executive program from additional bookkeeping regarding the queueing and priority problems.

The receiver in each module is basically a shift register and a Kirchoff adder such as the one shown in Figure 7-5. It is quite evident that the output of this circuit is

$$e_{out} = \sum_{k=1}^{n} \frac{g_k}{G} e_k; \quad G = \sum_{k=1}^{n} g_k \tag{7-8}$$

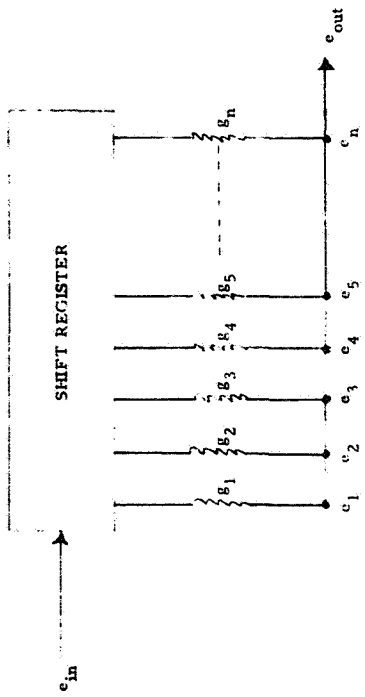


Figure 7-5. Digital Matched Filter

where e_k is the voltage at the kth bit position, and g_k is the conductance at the output of the kth bit ($g_k = 1/R_k$). Thus, by setting $g_k = GW_k$ the weighting network can be completely determined where W_k is the weighting function given by

$$\sum_{k=1}^{n} W_k = 1 \tag{7-9}$$

It follows from the above that the amplitude of the spike is proportional to the number of bits in the address code that are correlated. It is clear, then, that the receiver can be designed to tolerate errors up to a maximum of 50 percent and still be able to make correct decisions. Furthermore, since the addresses are chosen to be orthogonal with respect to each other, the probability of making erroneous decisions has been minimized. This is because no two addresses look alike even though 50 percent of their bits are in error. For this worst case, the digital matched filter of the appropriate module will still be able to pick up its own address. Therefore, the concept of graceful degradation is inherently built into the system for maximum reliability. To further minimize the probability of making erroneous decisions, both the data and the address codes are made orthogonal with respect to each other as well. This is highly desirable since the data is random; it could look like the address codes if the two were not orthogonal. The way this is accomplished is by modulating both the data and the address codes with two different carriers which are orthogonal to each other. The carriers are two clocks, cos $\phi_1(t)$ and cos $\phi_2(t)$, where $\phi_2(t) = 2\phi_1(t)$. The operation of mixing the data and the codes with these clocks is digital. The result of this mixing operation is expressed in the binary form as 1's and 0's, and is displayed in detail in Table 7-1. In this table an address code (derived from a Hadamard matrix) is mixed with cos $\phi_1(t)$ and the data (chosen at random) is mixed with cos $\phi_2(t)$. The two results appear in columns 3 and 6. Note that the number of agreements in these two columns equals the number of disagreements, i.e., A = 8 and D = 8. Thus

Table 7-1

DATA AND CODE MIXING RESULTS

C	$\cos \phi_1(t)$	$C \oplus \cos \phi_1(t) = C^*$	D	$\cos \phi_2(t)$	$D \oplus \cos \phi_2(t) = D^*$	$C^* \oplus D^*$
1	1	0	0	1	1	1
	1	0		0	0	0
0	0	0	1	1	0	0
	0	0		0	1	1
0	1	1	1	1	0	1
	1	1		0	1	0
1	0	1	0	1	1	0
	0	1		0	0	1
0	1	1	1	1	0	1
	1	1		0	1	0
1	0	1	0	1	1	0
	0	1		0	0	1
1	1	0	1	1	0	0
	1	0		0	1	1
0	0	0	1	1	0	0
	0	0		0	1	1

Space Division

Time Division

$$\rho = \frac{8-8}{16} = 0 \qquad (7\text{-}10)$$

which proves that the data in this case is orthogonal to the address code. The last column in this table shows this result. This means that despite the fact that data is random, it is ensured that any bit configuration within it will never look like that of any of the addresses. Therefore, even though the system is asynchronous and the receive registers in all the modules are widely open to the incoming data on the common bus, only the module which is being addressed will respond to the incoming data. The remaining modules will discard this data since they have not been addressed.

Figure 7-6 is the implementation of the mixing operation presented in Table 7-1. Note that the information will appear on the common bus after the mixing operation takes place. Data transfer on the bus will be in parallel. All modules will be scanned at very high speed; namely, within a memory cycle. This will allow sufficient time to transfer a complete word to each addressed module. The scanning operation is based on time division; the addressing operation on code division. Data transfer between the modules is based on space division since transmission and reception are in parallel.

It should be noted from the foregoing discussion that the principle of coding theory has been applied to maximize the information reliability. In addition, the modularity and flexibility which have been built into the system have maximized the system reliability. This reliability is ensured by the fact that the design provides the capability to take on different configurations when any of the modules fail. Thus, the proposed solution to the switching network ensures, among other things, both information and system reliability.

Figure 7-7 shows the data flow between the CPUs and the memory modules, and the data format as it will appear on the internal common bus. Note that each block of data (which could be a word) will be preceded by an address and terminated by the complement of that address. The complement of an orthogonal address is also orthogonal, and will have the same autocorrelation properties. As mentioned earlier, only the addressed module will respond to a given block of data even though

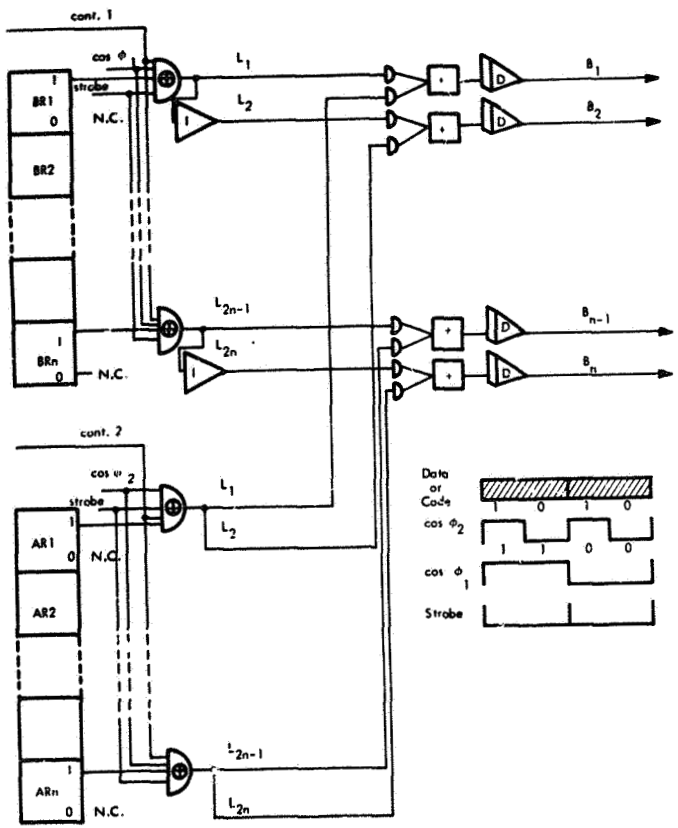


Figure 7-6. Mixing Operation

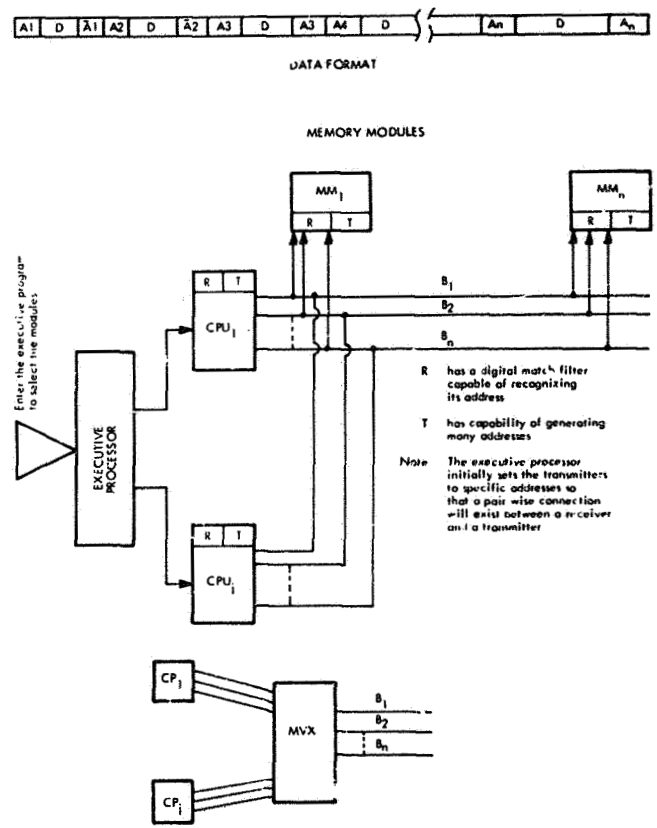


Figure 7-7. Data Flow Between the CPUs and the Memory Modules

some of the address bits might be received erroneously. This will not affect the receiver's selection of a correct decision, since the autocorrelation spike generated at this receiver will override any other signal that might be generated. Recognition of the complemented address will indicate end of word.

The executive processor in this diagram is no different from any other processor in the machine. Any processor in conjunction with one or more memory modules may be used to store the executive program.

Figure 7-8 is the overall system organization showing the data flow between the CPUs and memory modules on one end (via internal bus), and between the CPUs and I/Os on the other (via external bus). Of the three multiplexers shown in Figure 7-8 one is used to multiplex transmitter outputs of CPUs, and the others are used to multiplex the transmitter output of both the memory modules and the I/Os. Multiplexing of the transmitter outputs, in all three cases, is used to minimize the amount of wiring required.

Each module is expected to have an input trunk and an output trunk which have an equal number of wires. The output of the multiplexers constitutes the internal and the external buses. The buses are grouped into two pairs one of which corresponds to the internal input/output, and the other corresponds to the external input/output. The number of wires in each of these buses is the same as that of the module trunks. In this diagram, no direct path is shown between the I/O modules and the memory modules. It is felt that the data should transfer between these modules via the CPUs in order to minimize the wiring. If a direct path between these modules is required it can be easily provided.

7.4.3 Space Division Switching

This subsection describes the techniques for setting up paths between the various elements of the multiprocessing configuration shown in Figure 7-4. The space division switching organization to be used is shown in Figure 7-9. A separate path is provided for transferring information between the central processing units (CPU), the input/output (I/O) units, and the memory modules (MM). No

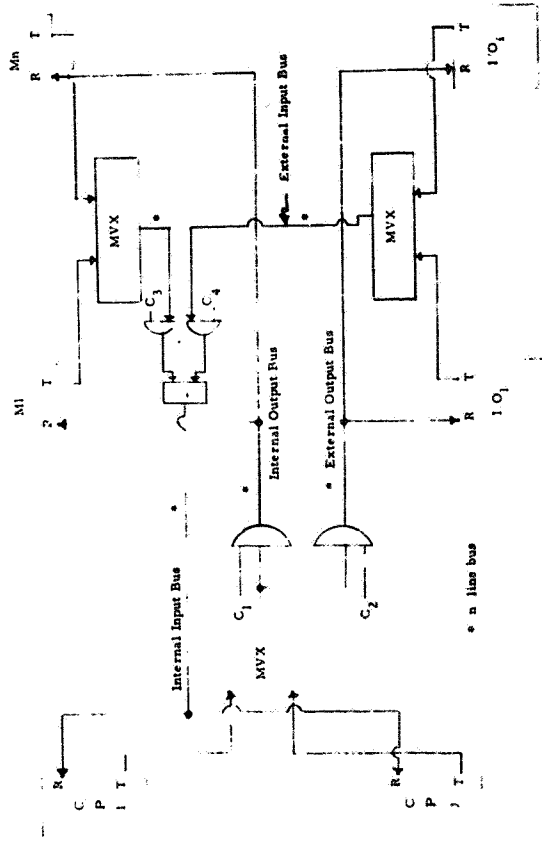


Figure 7-8. Overall System

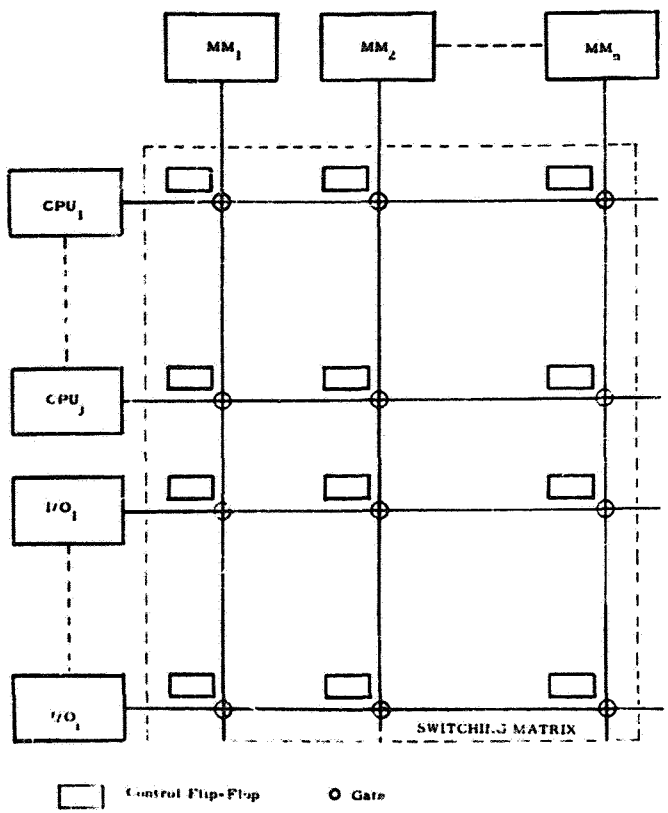


Figure 7-9. Multiprocessing Configuration Utilizing Space Division Switching

separate path is provided for exchanging information between the CPU and the I/O units, thus any data transfers between these two modules will be done via a memory module.

The switching matrix, shown enclosed in dotted lines, is composed of an array of gates and control flip-flops (FF). Each gate in the array provides a connection between either a MM and a CPU or an MM and an I/O unit. Each gate in turn is controlled by a separate control FF. The control FFs, controlled by the executive control program, establish different connections between elements thus forming various computer configurations. The executive program is executed by a computer configuration formed from modules connected by this same switching matrix.

The switching matrix shown in Figure 7-9 provides the means to transfer data in only one direction. An additional flip-flop and gate must be provided at each junction on a separate line in order to establish a data flow in the opposite direction. For the remainder of this discussion, we shall consider data transfers in one direction only. Data transfers may be performed either in parallel or serially. Based upon the computational requirements established in Section 6, all data transfers will be made in 8-bit bytes. For example, in the case of parallel transfer, eight gates controlled by one flip-flop will be required at each junction; whereas, in the serial transfer case, only a single gate and control flip-flop will be required, as shown in Figure 7-9.

Figure 7-10 shows in some detail the gating structure and the control flip-flop within the switching matrix. Note how each of the CPU and I/O units has an access to the memory buffer register of each memory module. In the case of parallel transfer, the block labeled "memory buffer" in this diagram represents a single stage of the 8-bit memory buffer register. In the case of serial transfer, the block represents the complete register and the incoming 8-bit byte is shifted in serially.

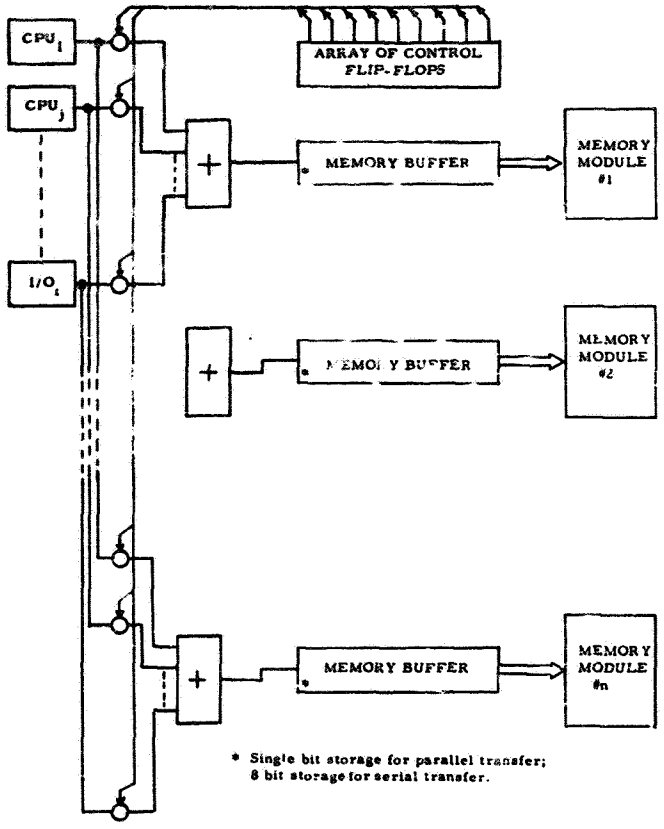


Figure 7-10. Switching Matrix

Section 8

AN ADAPTIVE ANALOG-TO-DIGITAL CONVERSION SYSTEM

This section discusses the Quantizing theorem used in the design of an adaptive analog-to-digital (A/D) converter. This theorem says, in effect, that if we quantize a signal at a sufficient fineness, then the statistics of the signal can be recovered. This theorem is analogous to the Sampling theorem which says, in effect, that if we sample a signal at a sufficient rate, then the signal itself can be recovered. Since the quantizing theorem guarantees that the statistics remain unchanged under the quantizing transformation, it follows that the entropy of the signal is also unchanged. This feature is important because much of the on-board processing of sensor outputs is statistical, and techniques for recovering the signal statistics are just as important as techniques for recovering the signal itself.

The analog-to-digital conversion process is fundamental on board the spacecraft, since all analog signals must eventually be converted into digital form. Currently, the sampling rate and quantization levels for each analog signal are predetermined by the designer, and they are usually designed for the worst-case condition. These systems, which operate in a suboptimum manner, do not take advantage of any changes in the statistics of the analog signals.

The A/D conversion process described here adapts itself to changes in the input signal statistics by varying the sampling rate as a function of the input signal bandwidth and by varying the number of levels into which the input signal is quantized as a function of the amplitude distribution of the input signal. Thus, the amount of data generated by the adaptive analog-to-digital conversion process is a function of two parameters of the input signal, the bandwidth and the amplitude probability density distribution. This amount will change as these parameters change.

8.1 ADAPTIVE A/D CONVERTER

Figure 8-1 is a diagram of the adaptive system. We assume that the sensor is the type that converts some physical input phenomenon into a time-varying voltage whose amplitude variations are proportional to the variations of the input physical phenomenon. It is assumed that the mean value of the output signal is zero. At this point in the system, the sensor output signal must be converted into an equivalent digital signal for further processing. This is true whether the raw data is to be telemetered straight to the earth station or transmitted to an on-board computer for processing.

The sensor output signal, v(t), is passed through a variable-gain amplifier whose gain is controlled by a digital computer. The function of the amplifier is to scale the input signal so that it encompasses the full dynamic range of the A/D converter input. This operation ensures that the system is utilizing the full conversion capability of the A/D converter. The amplifier gain is periodically, rather than continuously, adjusted with rather long intervals between adjustments. The gain setting is determined by calculating the amplitude probability density function of the sensor output signal using the general-purpose programmable computer. After enough samples of the signal have been taken to obtain a good distribution of the amplitude variations, a gain setting based upon this statistical distribution is established. This setting then remains constant while a new amplitude probability density function is calculated and a new gain setting is established. The value of the gain setting will be stored in the computer to be used later if necessary.

The output of the variable-gain amplifier is applied to the input terminal of the A/D converter. The A/D converter samples the analog output of the variable-gain amplifier at a rate of k samples/sec, which is determined by the digital computer. The converter then converts each sample into an n-bit byte representing a binary number equal to the amplitude of the input waveform. A sign bit representing the polarity of the input waveform at the sample point is also generated. It is assumed that the n-bit byte represents the maximum conversion capability of the A/D converter. In practice, n will be anywhere from 12 to 16. (Henceforth in this discussion, the sign bit will be ignored.)

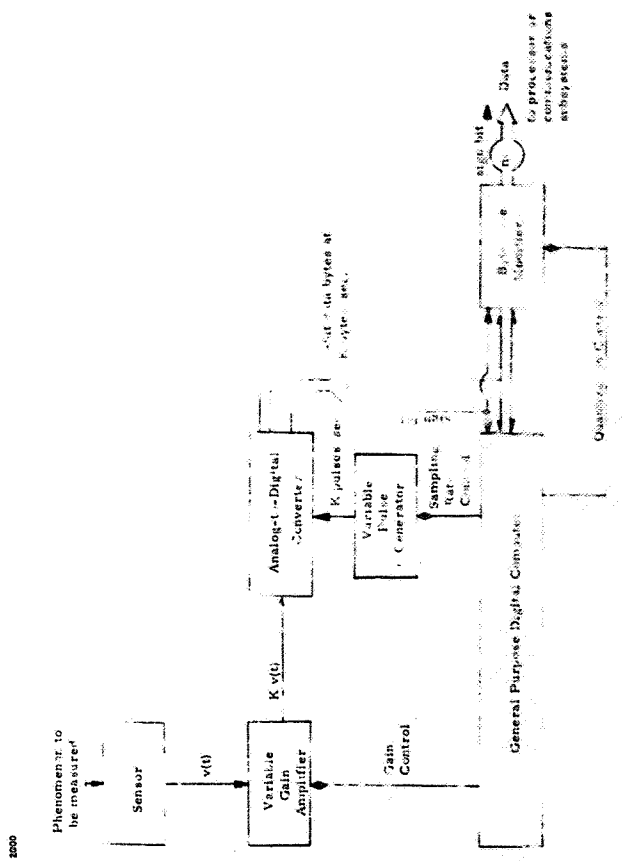


Figure 8-1. An Adaptive Analog-to-Digital Converter System

The n-bit bytes are transferred to the digital computer and a logic element called the Byte-Size Modifier. The computer computes certain statistical measures based upon the incoming data; based upon these measures, it also generates three control signals: gain control, sampling rate control, and quantization control.

8.1.1 Gain Control

This signal controls the gain of the variable-gain amplifiers and maintains the amplifier output signal at an amplitude such that the full dynamic range of the A/D converter is utilized. As described above, the value of this setting is based upon the amplitude probability density function of the sampled signal. The computer generates this function, decides on the dynamic range of the variable-gain amplifier output signal, and sets the gain control to modify this range, if necessary, to encompass the A/D converter input dynamic range.

8.1.2 Sampling Rate Control

This signal controls the output pulse repetition frequency of a variable rate sampling pulse generator. These pulses are fed to the A/D converter and used to sample the input waveform. The sampling rate is determined by calculating the power spectra of the input signal and setting the sampling rate equal to twice the highest frequency. The power spectrum of the input signal is calculated by using the digital computer. This computer also contains the decision-making programs for setting the sampling rate.

8.1.3 Quantization Control

There are two methods for achieving variable quantization. In the first method, the variable-gain amplifier gain control is used to scale the input signal so that it covers a particular portion of the full dynamic range at the input to the A/D converter. This operation effectively selects the number of levels into which the input signal will be quantized.

The second method, shown in Figure 8-1, accomplishes variable quantization of the input signal by a purely digital technique. This technique makes use of the fact that an n-bit binary number can be divided by two simply by shifting each bit of the number to the next lower-order position. The shifting operation is performed in the Byte-Size Modifier logic unit. This divide-by-two operation can just as easily be performed within the digital computer.

The technique for achieving variable quantization of the A/D converter input signal can best be explained by an example. Assume that the A/D converter is capable of 2^n quantization levels. Thus, each sample will be converted to a $\log_2 2^n = n$ bit byte, and transmitted to the computer and to the Byte-Size Modifier. Now suppose that the computer decides that 2^{n-1} quantization levels are sufficient for a particular input signal. (The mathematical basis for this decision will be discussed later.) This change in the number of quantization levels is accomplished in the Byte-Size Modifier element by shifting each bit of the input n-bit byte to the next lower-order position. The bit in the 2^0 position of the input byte is discarded. The output byte of the Byte-Size Modifier, containing only n-1 bits (plus a sign bit), now represents the value of the sample quantized to only n-1 levels. The number of quantization levels can be halved again simply by shifting each bit in the input byte twice toward the low-order bit positions.

Usually when an analog signal is to be converted into a digital form, two operations are involved: sampling and quantization. The rate at which a signal must be sampled is determined by the bandwidth of the signal according to Shannon's Sampling theorem. This theorem says that if $F(w) = 0$ for $|w| \geq \pi/T$, then $f(t)$ may be completely recovered from $f^*(t)$ where $f^*(t)$ is $f(t)$ sampled at the rate $1/T$ samples/sec. ($F(w)$ is the Fourier transform of $f(t)$.) $F(w)$ is calculated using the on-board computer and, based upon this calculation, the sampling rate is determined.

8.2 QUANTIZATION

At each sample point, the continuous range of the input signal is converted into a set of discrete levels, i.e., quantized. This process is extremely nonlinear, and

the precise effect of quantizing upon the waveform is not known. However, the effects upon the statistical properties of the signal are known.

A quantizer or A/D converter is a device which converts an analog input into an output having only discrete values. The amplitude of the input signal, f(t), is divided into a number of segments of length, q, the quantizing width. Whenever f(t) lies within one of these segments, the value at the midpoint of the segment is presented at the quantizer output as the value of the input signal at that sample point. Thus, while the amplitude probability density function (PDF), w(x), of the input signal f(t) is continuous, $w'(x)$, the PDF of the quantizer output, $f'(t)$, will be zero at all values of x except at $x = 0, \pm q, \pm 2q, \ldots$. Also, the value of $w'(x)$ at nq represents the probability that x will be in the interval $nq - q/2 \leq x \leq nq + q/2$. This probability is given by

$$p(x) = \int_{nq-q/2}^{nq+q/2} w(x)\,dx \qquad (8\text{-}1)$$

Figure 8-2 shows the two distributions w(x) and $w'(x)$ where w(x) is the distribution of the signal into the A/D converter and $w'(x)$ is the distribution of the signal out of the A/D converter. The height (or area) of each impulse at nq is equal to the area under the curve between $nq \pm q/2$. For this reason, the quantizing operation is often referred to as area sampling.

Since $w'(x)$ is composed of a sequence of samples, it might be possible to relate the known effects of sampling to investigate the effects of quantizing. In particular, we would like to investigate the problem of recovering w(x) from $w'(x)$. In the quantization problem, we are investigating ways of recovering the statistics of the signal rather than the signal itself which, from an information theory point of view, is highly desirable.

The key to applying previous results on sampling to the problem of quantizing is to change the area sampling (or quantizing) of w(x) into ordinary sampling of some related function. In Figure 8-2, the area of the impulse at nq is equal to the

2001

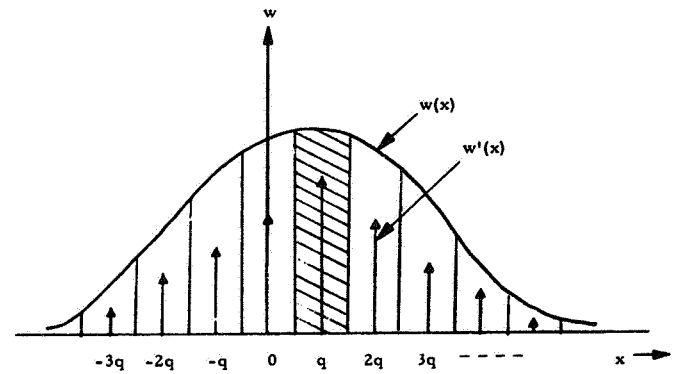


Figure 8-2. Quantizer Input (w(x)) and Output ($w'(x)$) Distributions

area of w(x) which lies inside a rectangle of height 1 and width q centered at nq. This area is also equal to the area under the product w(x) qf(nq-x) where

$$f(x) = \begin{cases} 1/q & -q/2 \le x \le q/2 \\ 0 & \text{elsewhere} \end{cases} \qquad (8\text{-}2)$$

The above suggests that the PDF of the quantizer output w'(x) may be obtained by first convolving w(x) with a rectangular distribution, f(x), and then sampling the resulting function at nq for n = 0, ± 1, ± 2, ... These relationships are shown in Figure 8-3.

8.3 QUANTIZING THEOREM

We have shown how the PDF quantizer output is obtained by convolving the input PDF with a rectangular PDF of width q, and then sampling this resultant function at nw for n = 0, ± 1, ± 2, ... The question now is: what should be the value of the sampling parameter q such that w(x) can be recovered from w'(x)? If w(x) is the PDF of f(t), the Fourier transform of w(x)

$$W(\alpha) = \int_{-\infty}^{\infty} w(x)\, e^{-j\alpha x}\, dx \qquad (8\text{-}3)$$

is called the characteristic function of w(x). Recall that multiplication of function transforms corresponds to the transform of the convolution of the original functions. Thus, the above operation of convolving w(x) with qf(x) corresponds to multiplying W(α) by qF(α), the transform of qf(x), which is

$$\begin{aligned} qF(\alpha) &= q\int_{-\infty}^{\infty} f(x)\, e^{-j\alpha x}\, dx \\ &= \frac{\sin \alpha\, q/2}{\alpha\, q/2} \end{aligned} \qquad (8\text{-}4)$$

2002

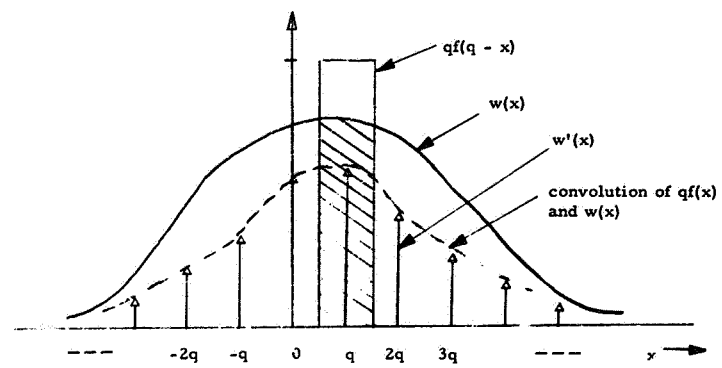


Figure 8-3. w'(x) From w(x) Through Convolution and Sampling

Since the operation of sampling results in the generation of a periodic spectrum, the resulting periodic spectrum of the sampled distribution is

$$F(\alpha)\, W(\alpha) \;=\; W(\alpha)\, \frac{\sin \alpha\, q/2}{\alpha\, q/2} \qquad (8\text{-}5)$$

repeated with a spacing of $2\,\pi/q$ as shown in Figure 8-4b. Figure 8-4a shows the various characteristic functions.

The quantization theorem may then be stated as follows: if $W(\alpha) = 0$ for $|\alpha| \geq \pi/q$, $w(x)$ may be completely recovered from $w'(x)$. As in sampling, if it is not true that $W(\alpha) = 0$ for $|\alpha| \geq \pi/q$, then aliasing will result, folding complementary signals from the range $|\alpha| \geq \pi/q$ onto signals in the range $|\alpha| \leq \pi/q$ giving erroneous results.

8.4 RESULTS FOR GAUSSIAN SIGNALS

An interesting result of the quantizing theorem for signals with a Gaussian PDF has been found. Since voice and many of the spacecraft sensor output signals are Gaussian-like, this class of signals is particularly interesting. It has been shown that if the quantizing width, q, is chosen to be equal to the standard deviation, σ, then the error in the mean square is 10^{-6} percent of the mean square of f(t) and the error in the mean fourth is 3×10^{-5} percent of the mean fourth of f(t). Also, by requiring that the dynamic range of the quantizer be 8q (corresponding to 3 bits/sample), we can be sure that the first few moments have acceptable accuracy. This is a remarkable result when one considers that most of the analog-to-digital converters on board present and planned spacecraft provide for 8, 9 and 10 bits/sample corresponding to 2^8, 2^9, and 2^{10} quantization levels.

8.5 STATISTICAL ANALYSIS

To implement an adaptive analog-to-digital converter of this type, two statistical measures must be calculated. The sampling rate is determined by calculating the spectral density of the input signal, and the quantizing fineness is

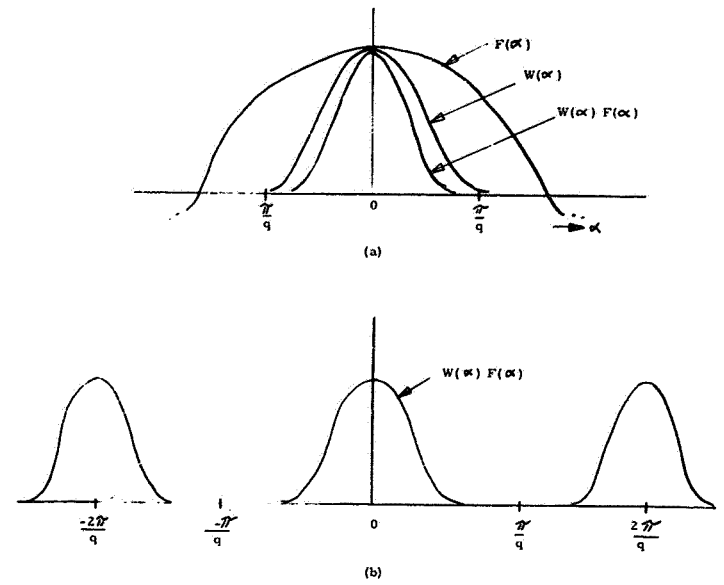


Figure 8-4. (a) Results of Convolution in the α Plane
(b) α Spectra of $w'(x)$

determined by calculating the probability density function. In Section 5 of this final report, the equations for performing these calculations on a computer are given in a form suitable for programming. In addition, any other statistical measures necessary to implement this adaptive system are given.

Section 9

MICROPROGRAMMING APPLIED TO MULTIPROCESSING

One of the more sophisticated advances in computer design during the past five years is the concept of microprogramming. The essential difference between a microinstruction and an ordinary computer machine instruction is that the microinstruction contains an order of magnitude more bits in its instruction code, thereby increasing the instruction repertory astronomically. Microinstructions are usually stored in a high-speed read-only memory and used to implement and execute ordinary machine instructions. In other words, each machine instruction is executed by means of a microprogram composed of microinstructions. Typically, the operation is as follows (see Figure 9-1). When a machine instruction is read out of the read-write core memory, it is sent to an instruction decoder and is decoded. The decoder, in turn, addresses the first word of a microprogram stored in a read-only memory. The microprogram is then executed, thus executing the machine instruction. The contents of the address register can be modified by the microprogram itself and by external conditions. Hence we see that the microprograms are a replacement for the usual instruction control logic.

9.1 APPLICATIONS TO MULTIPROCESSING

There are four areas in satellite multiprocessing systems, such as those described in Section 7 of this report, where microprogramming can be useful: diagnosing and correcting machine failures; providing the capability of modifying the instruction sets within each of the central processing units; providing the capability of modifying the operation of each of the various input/output modules; and automating many of the instruments and sensors on board the satellite.

2004

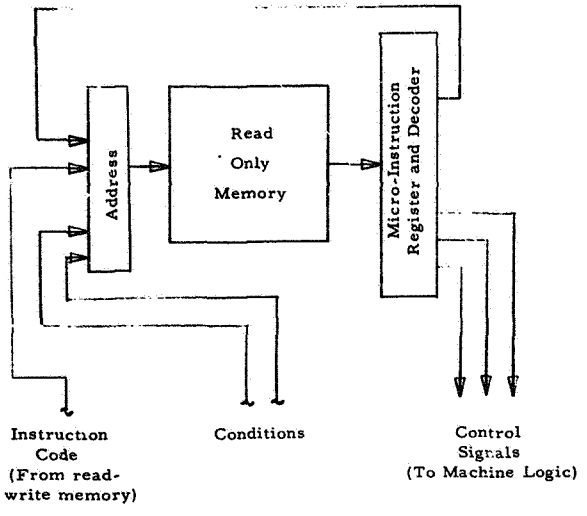


Figure 9-1. Basic Microprogrammed Control Elements

9.1.1 Diagnostic and Corrective Programming

The word-oriented logic of a microprogrammed machine can facilitate self checking and fault-location procedures. This is due to the fact that the microinstructions have access to the actual machine hardware such as the registers and counters. Of course, the machine must be designed with this in mind, e.g., there should be parts of the machine capable of performing logical operations independently to provide for mutual diagnosis; there should always be provided alternate data paths, so that access to certain regions will not become blocked. Provision must be made for the diagnostic microprogrammer to call out modifications to the machine design when he finds them necessary.

Once a failure has been diagnosed and the faulty element located, the corrective reconfiguration procedures outlined in Section 7 can be executed.

9.1.2 Instruction Modification

The several multiprocessing configurations that have been developed so far in this study have been designed for high-reliability and graceful degradation predicted on the long missions envisioned for multifunctional satellites. Thus, the organizations chosen have utilized the concept of providing many identical computing, memory, and I/O elements and forming many small computers from this pool of elements.

In some case, it might be desirable for a CPU element to have the ability to change its instruction set to fit a particular task. If an electronically changeable, read-only memory were available to store the microinstructions, microprogramming concepts could successfully be applied to accomplish this. In any event, microprograms will still be used to implement instruction sets and control within the CPU elements.

9.1.3 Modification of I/O Units

The function of an I/O unit in a multiprocessing system is to match a particular I/O device to the central processor. In many cases, a single type of I/O

unit can serve several different I/O devices. For those instances where small changes in the operation of the I/O unit are required, microprogramming can be used to provide this flexibility. Again, an electronically changeable read-only memory would be an asset; however, microprogramming will still be used to replace the control logic in the design of these elements.

9.1.4 Instrumentation

Microprogrammed elements have been successfully used to control the operation of digital computers. It seems reasonable, therefore, to apply similar microprogrammed control elements to other devices and processes of comparable complexity. The next generation of instruments will, in all probability, have some "intelligence" built into them. An example of this is the adaptive analog-to-digital converter described in Section 8 of this report. Microprogramming techniques can be used to provide these instruments with some of this capability. For example, an instrument can be microprogrammed to perform some preprocessing of the data it collects before transmitting it to the central computer. A microprogram can be provided with each instrument that will, upon computer command, check and calibrate he instrument, thus relieving the CPU of this task. Microprograms can be used for automating complex instruments so that they can perform a preprogrammed series of operations and report their findings to the CPU.

9.1.5 Read-Only Memories

The read-only storage memories presently used for storing microinstructions (such as in the IBM System/360's) are relatively bulky and do not fit the requirements for spaceborne or small machine usage. Also, these memories are not easy to modify.

For spaceborne application, a crushed ferrite core memory has been developed which solves the problem of size and weight. The core planes are wired as usual, but afterward the cores representing zeros are crushed and removed. The remaining cores, of course, represent ones. This memory must be replaced by another if modification is desired.

For spaceborne and mobile applications of microprogrammed controllers and computers, lightweight, high-capacity, fast, electronically modifiable, but inexpensive read-only memories are needed. Research and development programs have been initiated to develop such memories.

An example of an avenue of possible further development is the phototransistor memory built by Hagiwara. A phototransistor is a device which conducts electric current when activated by a beam of light. When the light is shut off, the device ceases to conduct.

Each bit of the memory is represented by a phototransistor. The phototransistors are arranged on a grid which can be covered by a punched card, the holes of which permit light to fall on the corresponding phototransistors. The memory contents can, therefore, be changed easily by changing the punched card. The author suggests that a machine might even change its own punched cards.

Section 10

ERROR DETECTION AND DIAGNOSTIC TECHNIQUES

10.1 HARDWARE

The on-board multiprocessing system will be inherently reliable due to the distributed polymorphic structure described in Section 7 of this report. This high reliability is achieved by detecting the machine errors as they occur and then reorganizing the computer architecture so that the faulty module is switched out of the system and replaced by a good unit. This section presents a technique for detecting and locating machine errors.

One of the redundancy techniques that IBM has successfully applied to increase the availability of the SAGE computer is self-checking logic. In the SAGE system, 30 percent of the logic was devoted to self-checking. This technique allows logic failures to be detected instantaneously so that errors will not be propagated. As soon as an error is detected, an error alarm logic interrupt element is signaled, causing all register contents to be frozen while the executive routine investigates the source and type of error and decides upon further action.

Figure 10-1, an example of a typical application of the self-checking logic technique, shows a decoding element with the error detection logic element enclosed within the dashed lines. The operation of the logic is as follows: The contents of Flip-Flops A and B are to be decoded causing a single pulse to appear on one of the output lines labeled 1 through 4. Assume that at time t = 0, data is loaded into Flip-Flops A and B. The Flip-Flop outputs are decoded by the four AND circuits and after the Flip-Flops have reached a steady state (all transients due to changing state have gone to zero), a check pulse samples each of the four AND gates in the decoder element. One of the four AND circuits will pass this sampling pulse, thus decoding the contents of the two Flip-Flops.

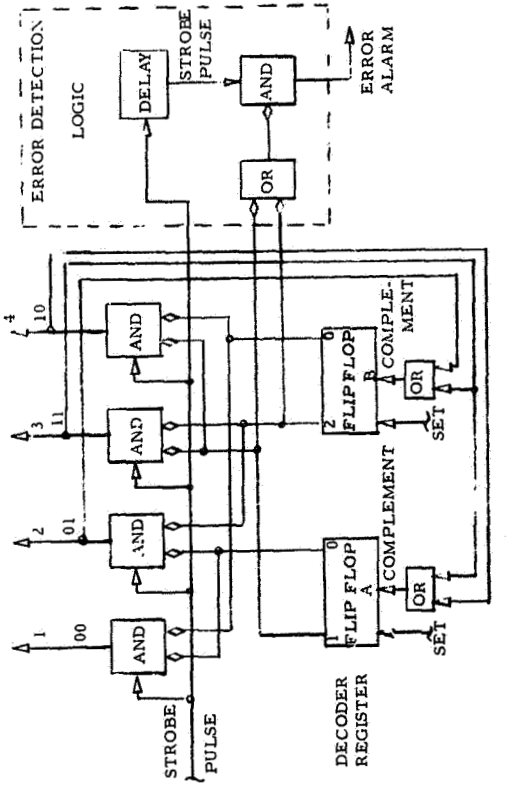


Figure 10-1. A Self-Checking Decoder

To accomplish self-checking, each of the decoder output pulses (except line 1) is fed back into the Flip-Flop register in fixed patterns causing certain stages of the register to be complemented (reversed). Each decoder output pulse is also fed into the error detection logic, passed through an OR circuit, delayed until the register has had time to settle down and then strobes an AND gate which is conditioned whenever the contents of the register are nonzero. An error alarm will be generated whenever the decoder incorrectly decodes the contents of the register. Note that the error detection logic shown in Figure 10-1 can accommodate an n-stage decoder register simply by providing additional inputs to all OR circuits.

Self-checking logic can be designed to check all of the elements within the CPU such as adders, multipliers, decoders, and instruction control. This technique, when applied to multiprocessors along with other error detection schemes such as coding, program diagnostics and marginal checking, ensures high machine reliability and availability. Microprogrammed diagnostic routines and marginal checking techniques are used to maintain and test the redundant error-detection logic.

10.2 SOFTWARE

Self-checking techniques are implemented in hardware and are one way in which errors can be instantaneously detected and isolated. Another technique which can be used in conjunction with the self-checking logic techniques is to use diagnostic programs. A diagnostic program is an automated procedure, defined in the form of computer instructions, for testing and/or analyzing results. Individual tests are arranged in a sequence, and each test result determines which point in the sequence will be executed next. Logical analysis is accomplished automatically in the same way. A predetermined decision tree is executed via coded instructions.

The term error refers here to any deviation from the expected operation or results, including component malfunctions and operator or program errors. Part of the diagnostic problem is to isolate and categorize each error.

For a multiprocessing system, unit diagnostics would be written for each input/output (I/O) unit and the various logical subdivisions of the processor modules. Typically, a unit diagnostic begins its test under a controlled set of conditions (that is, one I/O device at a time) in order to isolate errors more effectively. The testing sequence normally follows a logical structure. On the other hand, a system test program departs from the controlled environment of a unit diagnostic. It is designed to provide a comprehensive system test similar to an operational environment.

Diagnostic programs generally employ a mixture of testing and isolating techniques. The mix will depend upon the equipment being tested and the purpose of the test. Some of the basic techniques are given below.

10.2.1 *Building Block Approach*

This approach starts with the simplest test involving the smallest amount of circuitry possible. Each additional test includes a small additional amount of circuitry. Failure to pass a given test at any point in the test sequence indicates that the probable cause of error is the additional circuitry being tested. This technique is particularly effective when there are multiple machine malfunctions.

Since each error is repaired as soon as it is detected by switching in a new module, multiple failures should not confuse the diagnosis. Although this technique is disadvantageous in that it is time consuming, the time required is insignificant for the central processor because of the speed at which instructions are executed.

10.2.2 Go/No-Go Approach

In this approach, a large amount of logic is given a quick test. The strategy is to zero in on an error. The system program is designed to isolate errors to a small area of the system (that is, a specific unit or logic function). A more specialized diagnosis is then used for that specific area to further narrow down

the trouble. This technique provides the fastest test under successful conditions, and is often used to provide rapid checkout of a large system, and the isolation time is optimized. Multiple errors are difficult to handle with this approach.

10.2.3 Multiple Clue Approach

This technique uses a series of individual tests, saving the success or failure indications of each. Each test in the series will take a slightly different path through the maze of logic. At the end of the series, all the test results are analyzed and a diagnosis is made. For example, if paths 1, 3, 7, and 10 failed, any circuit common to these paths would be suspect. This technique is particularly useful for circuitry (logic) that cannot be adequately diagnosed by any one test; that is, the smallest amount of circuitry used by any single test is still too large.

10.2.4 Error-Environment Recording

This approach depends on special logic circuitry for detecting errors such as the self-checking logic described previously. Records are kept of the error and the machine's environment, that is, the status of indicators and contents of registers. Isolation of the error is then attempted by analyzing this recorded error environment. This technique, unlike the previous ones, is effective for diagnosing the intermittent error, particularly if the malfunction occurs in normal operation and cannot be recreated in a diagnostic test. The success of this one-shot diagnosis depends on the resolution of the error indicators, the exact environment saved, and the speed—relative to the speed of the computer—with which errors are detected and the environment captured for later analysis.

10.2.5 Summary

Earlier data processing systems used vacuum tubes and relied almost entirely on programs to detect and isolate malfunctions. Solid-state circuits, with continual reductions in size and cost, made it practical to include redundant circuits for the purpose of error detection. Today's systems have error-detection circuitry that will detect a majority of malfunctions. Isolation, for the most

part, is still performed by programs which are becoming increasingly complex. As the cost per logical circuit decreases, more of the isolation functions will probably be shifted from programs to logical circuitry.

The multiprocessing system error-management techniques described above can be accomplished through a combination of special equipment design (logical circuits) and special programming techniques. Possibilities considered in the design include:

a. Error-detection logic using some form of redundancy to quickly detect errors

b. Save mechanisms (logic) to preserve the system environment and error data for later analysis

c. An interrupt system (logic) designed as an integral part of the system to stop the operation in process and turn the control over to a diagnostic routine (program)

d. Diagnostic routines (program) to analyze the error data and environment information to determine the corrective course of action (for example, retry the operation in the case of an intermittent error and continue with the system operation at the point of error and, in the case of a solid error, reorganize the system to continue operating without the faulty unit)

e. Online test routines that are part of the operating system (program). These routines allow testing and servicing of portions of the system while the rest of the system is engaged in useful work.

10.2.6 Future Requirements

The state-of-the-art of logic circuitry, system organization, and application development has been advancing at a rapid pace. Some of the more pressing diagnostic needs for airborne multiprocessing systems are:

a. Total error management

b. Techniques that will approach 100 percent detection in the most efficient manner

c. Methods for the economical production of diagnostic routines that automatically isolate to the replaceable module

d. Diagnostic techniques for distinguishing between equipment errors and program errors.

Section 11

ADVANCED TECHNOLOGIES

11.1 ADVANCED LOGIC CIRCUITS

A brief review of monolithic integrated logic circuits for low power spaceborne computers is presented below. Of the available devices, large scale integration using technologies such as MOS (metal-oxide-silicon) field-effect and bipolar are clearly favorable because of their low power consumption, high reliability, low weight, small size, and economy due to batch fabrication.

Whether MOS or bipolar is more useful for large scale integration depends on the system requirements for a given application. For example, a single-polarity MOS inverter, using an MOS load resistor, exhibits twice the potential packing density when compared with the same circuit using bipolar transistors and a load resistor. This assumes similar process resolution capabilities. The MOS transistor's very high input resistance results in virtually unlimited dc fanout, and the low power requirements of MOS enhance packing density by easing power distribution and ground requirements.

The bipolar transistor, on the other hand, shows a clear superiority on a speed-power basis. This is a result of the bipolar transistor's excellent transconductance per unit area, which in turn is a measure of the device's ability to charge and discharge capacitance. In addition, bipolar arrays are compatible with today's existing integrated circuit signal levels, power, and logical organization. Bipolar arrays may be mixed with discrete integrated circuits and complex standard product integrated circuits for maximum cost effectiveness.

Thus, arrays using both technologies are required. Where the speed-power product and system compatibility permit, MOS is superior from a packing density

standpoint. Where speed or power capability, or both, are at a premimum, or compatibility with discrete integrated circuits or existing systems is required, bipolar should be chosen.

Since the energy required to execute different computational tasks simultaneously is the main concern in spaceborne multiprocessing computers, the selection of logic elements should be based on power consumption and high density packing. MOS technology seems to offer these characteristics and thus can be considered as the best candidate for use in the representative system. This section discusses the MOS, and Section 12 is devoted to discussion of large scale integration in general.

MOS field effect transistors and the integrated circuits containing them offer a number of significant advantages when compared to bipolar planar silicon transistors and integrated circuits. Advantages of MOS transistors include low operating power levels, very high input impedance, low noise level at higher frequencies, relative insensitivity to temperature, absence of carrier storage, absence of thermal run-away, simplicity of circuit design, and simple fabrication processes. In contrast to these advantages, initially available MOS transistors have revealed a number of limitations, including high threshold voltage, instability of threshold voltage, and sensitivity to gamma radiation.

A predicted advantage of MOS transistors was a lower sensitivity to radiation because they are majority carrier devices, but this prediction has not been borne out by experimental data. In fact, MOS transistors have proven more sensitive to radiation than bipolar transistors. Recently it has been shown that the effect of radiation on the oxide layer begins at a level well below that which directly affects the silicon. MOS transistors are at least an order of magnitude more sensitive to X-ray and gamma radiation than bipolar transistors. This problem is fundamental to MOS devices as they are now fabricated and a radical change in processes would appear to be required if MOS devices (which are less sensitive to radiation than bipolar devices) are to be produced.

Complex arrays of MOS integrated circuits have appeared earlier than those with bipolar transistors. This is due partially to the relatively small geometry

resulting in less loss due to imperfections on the silicon wafers, and possibly because fewer processing steps are required. Another factor that has undoubtedly forced the early emergence of complex arrays of MOS devices is that these circuits compare favorably to bipolar circuits only when a large number of MOS circuits are connected on the same chip. Due to the high output impedance of the devices, buffer circuits, which consume considerably more power than the logic gate analyzed and take up considerably more chip area, must be used to exit from the chip. Consequently, MOS integrated circuits look attractive in application where low clocking rates are employed or where the equipment must be turned on for long periods of time at relatively low activity rates.

11.2 LARGE SCALE INTEGRATION

There is no fixed definition of large scale integration when practical machine designs are considered because the level of integration offered to the machine designer varies as a function of circuit performance. Higher performance circuits are fabricated to closer tolerances so that the number of circuits that can be integrated at the chip level with reasonable yield will be lower than those designed for more modest performance. This is one of many tradeoffs to be considered in planning machine organization.

Although large scale integration promises to offer cost reduction, improved reliability, better performance, and less power requirements, the technology does impose some restrictions on the system designer. He is forced to design before he can thoroughly test the design.

11.2.1 Reliability

An important benefit to be expected from large scale integration is greater reliability. The number of circuit failures per 1000 hours has been decreasing by a factor of 10 every five years since 1950. Large scale integration may be the principal means for continuing this decrease. Although marked improvements in both system reliability and maintainability are expected, an exact quantitative measure of these advantages is not yet available.

The impact of large scale integration on reliability may be mainly in the reduction of interconnections. It is an accepted fact that the more interconnections integrated on the chip, the more reliable the system. If no new failure mechanisms intrinsic to large scale integration arise, then a considerable increase in system reliability will occur. This increase can be traded off in several ways. A given system will run longer, or if the system failure rate is adequate the system can be made larger. In some cases error checking and correcting circuits can be removed or extra circuits can be added to take over the functions of less reliable electromechanical equipment.

Other important and related benefits to be gained from large scale integration are lower power per circuit and higher speed. Much of the improvement will come from the decrease in the load imposed upon circuits by their environment as a function of the increase in circuit density.

As the number of circuits on a chip is increased, more circuits will have the load they are driving on the same chip. In these cases, the transmission delays or reactive loads will be decreased significantly—often more than an order of magnitude. This cut in circuit loading can be cashed in for either higher performance or lower power of some tradeoff between these factors.

Large scale integration introduces few new constraints to the system designer, but the limits can be more severe and impose a strong influence on design. Important among these are the potentially high costs and longer turnaround times associated with producing a new part or making changes These costs are sensitive to both quantity and performance. Without careful partitioning, intermediate levels of integration can also produce large numbers of unique parts. In addition, the proliferation of larger systems and more complex parts will increase the susceptibility to these factors.

Tradeoff relationships will reflect more significant interdependences among more diverse factors. For example, the yield (and cost) of a specific level of integration may not be known until chips have been produced in a manufacturing environment, yet by this time the system hardware design may be frozen to prevent product obsolescence.

11.2.2 Limitations

Limitations inherent in large scale integration must be circumvented throughout the computer system's design process before the technique can be applied extensively. Because large scale integration affects all the traditional steps in the design process and the interaction among them, its application demands a more unified approach to system design. A new set of ground rules is needed.

The system design process has always involved the following steps:

a. Drawing up overall specifications—including functions to be executed by the system, performance objectives, and cost projections.

b. Defining system architecture—allocating functions to hardware and software, and determining the organization of the system. At this point, machine instructions and formats for data and instruction are established, performance levels are set, circuits and equipment are selected, and packaging is specified.

c. Designing logic for the machine—developing detailed algorithms for machine functions, defining synchronization techniques, and developing microprograms to describe the execution of instructions. Detailed logic for the machine is expressed in terms of the selected logic circuits.

d. Partitioning machine logic and storage into packages—logic signals are allocated to package pins.

e. Assigning packaged logic and storage to locations within the machine framework.

f. Specifying wire routes that interconnect the packages for logic, storage, input/output equipment, and console.

In modifying the approach to this process to accommodate large scale integration, the designer must remember that large scale integration will increase the capability of logic and storage per unit cost in those parts of the system that can use it.

11.2.3 Logic Design

Current techniques of logic design are not adequate for large scale integration. The logical structure of a machine has traditionally been expressed as a

set of Boolean equations that has then been transformed to achieve minimum circuit count through such techniques as Karnaugh maps, Veitch charts, or Quine-McCluskey tables. Computer programs have been used to carry out the design process. These techniques were developed when discrete-component technology was common and they are still useful with unit-logic monolithic circuits.

With large scale integration, however, the development of a minimum-circuit system may not result in minimum product cost. Extra circuits may be justified to reduce pin requirements, to allow package testing, and to create part repeatability. In the past an attempt was made to design an optimum set of elementary logical building blocks and then to implement the logic in a straightforward manner. Now, however, there is the much larger task of designing the optimum hardware implementation of a complex logic structure (100 or more gates) while at the same time optimizing many other design criteria. Well defined systematic design techniques are not presently available to cope with this problem.

Large scale integration increases the amount of logic and/or storage in the elementary building blocks of a system. A single chip carries a significant fraction of the total logic in a small system or represents a major functional unit of a large system.

In machine organization, large scale integration forces the designer to consider hardware constraints to an unprecedented degree. Cost objectives, intended applications and production volume affect these considerations. For example, if a machine is being designed for high-volume and low-cost output, the organizational objective may be to minimize the total number of parts even though all the parts are different. On the other hand, if the machine's projected sales volume is low and minimum machine cost is still important, a considerable effort may be made to achieve part repeatability.

11.2.4 Partitioning

Selecting portions of a system (carried through the logic-design stage) capable of being implemented on large scale integration chips without violating hardware constraints is called partitioning.

The criteria to be met in partitioning are numerous, and the importance of each depends on the final application of the system. In partitioning of the chip level, a designer is usually given a ceiling on the number of circuits and on the possible values for the number of input/output pins. The criteria that must be optimized to achieve a minimum system cost with large scale integration include:

a. Number of first-level packages required (chips or parts)

b. Wiring on the second-level package; Part repeatability

c. Testing

d. Ease of maintenance

e. Reliability

f. Performance

g. Engineering change capability

h. Requirements for special applications.

A close examination of these factors indicates that some of them conflict and that compromises must be made. Although designers have gained some experience in satisfying the first three criteria and have some preliminary measuring tools, the techniques for achieving some of these criteria are not well developed.

There are two principal objectives in organizing a system for large scale integration:

a. Minimize the total number of chips and the interconnections between chips

b. Minimize the number of different chips to achieve a high degree of standardization.

The first objective must be met for any production of the system. If the second objective is met satisfactorily, a method must be found to successfully interconnect the various chips.

11.2.5 Cost

A method of increasing the volume of standard parts is through logic implementation. One technique involves the deliberate introduction of redundancy or superfluity on one chip to make it useful for several applications.

The amount of extra logic that can be added to reduce the number of different parts can be evaluated by the advantages of part repeatability within one particular system. The cost of a large scale integration part of a given type can be expressed as

$$C_P = C_M + C_T = \frac{C_G}{V} = C_{MF} \qquad (11\text{-}1)$$

where

C_M = Manufacturing cost per part (processing plus first-level packaging)

C_T = Test cost per part

C_G = Generating cost per part number (masks, test patterns, documentation)

V = Parts per part number (volume = useful volume + stock volume)

C_{MF} = Field maintenance cost per part.

The cost of a system is the summation of all the parts' costs, expressed as

$$C_S = \sum_{i=1}^{\alpha n_0} \left(C_{M_i} + C_{T_i} + \frac{C_{G_i}}{V_i} \right) + BC_{MF} + C_{PK} \qquad (11\text{-}2)$$

where

C_{PK} = Cost of the higher levels of packaging plus cooling plus power supply

B = The manner in which the maintenance cost varies as a function of the number of different part numbers, the number of systems in a given user location, stocking philosophy

N = The number of different part numbers in a system = αn_0

n_0 = The number of unique parts needed in the system (the ratio of total circuits in a system to the integration level)

$\alpha = \frac{1}{\text{Utilization factor}} \geq 1$ = increase in system parts to allow for a decrease in system part numbers

$1 - \frac{1}{\alpha}$ = Fraction of circuits on a chip that are extra or supernumerary.

Equation (11-2) can now be solved with C_S kept constant to find the relationship of α to N. The results shown above were determined under the following assumptions:

$$C_M = C_T = C_{PK}$$

C_{PK} varies directly with

$$\left[\begin{array}{l} C_{PK} = C_{PKO} \text{ for } \alpha = 1 \\ C_{PK2} = 2C_{PKO} \text{ for } \alpha = 1 \end{array}\right]$$

On-site stocking (11-3)

One machine per location

$$C_{MF} = 2\left[C_M + C_T + \frac{C_G}{V}\right]$$

increase of V due to $N \neq n_o$ has a second-order effect on C_M and C_T. Results applicable to any size machine as long as the same distribution of system parts in a part number is used.

In general, the analysis shows that:

a. The higher the values of C_M, C_{PK}, and C_T, the fewer the supernumerary circuits that can be added to achieve a given part-number reduction.

b. The larger the ratio of C_G/C_M, the larger the number of supernumerary circuits that can be added to achieve a given part-number reduction.

c. More supernumerary circuits could be added if C_M and C_T are reduced significantly with volume.

d. Fewer supernumerary circuits could be added if a more optimistic stocking and system distribution philosophy were used.

Even for the wide range of parameters considered, the maximum number of supernumerary circuits indicated by the analysis does not appear to allow a practical one-part-type system—all chips in a system alike—at least in terms of conventional thinking. However, further significant reduction is possible.

11.2.6 Application

One characteristic of multiprocessing computer systems is the capability to perform several tasks simultaneously, and to perform parallel operations within each task. These efforts require a high order of repeatability (so necessary for large scale integration) and thus more hardware is needed to improve total system capability. The increase in hardware can be used to substantially reduce the cost expected from large scale integration.

The standardization of chip types affects all facets of system organization—the arrangement of storage units, the manner of system control, the degree of sophistication in the central processing units (CPU), the variety and degree of autonomy of peripheral equipment (input/output), and the tradeoff between hardware and software for the overall system.

11.2.6.1 Main and Functional Storage

From the standpoint of system organization, a distinction can be made between the main storage units used in the classical manner (serial as well as random access) and functional storage units such as read-only stores (ROS) and content-addressed associative stores. In an associative storage system, data is addressed and retrieved from the main store by name rather than by reference to its physical location in the memory. An associative store is characteristically prodigal in the use of logic circuitry, and serves as a good example of how hardware can replace software.

The significance of large scale integration to storage-unit design is apparent in the small number of different chips required for memories. If they are properly designed, many replicas of two or three chip types can be fitted together to create large stores. Associative stores could be constructed out of standard chips to translate the names of data into physical location addresses.

11.2.6.2 Central Processors

The greatest logical complexity in most systems is found in the area of system and arithmetic processing. Included in the CPU are temporary storage registers (for instructions and addresses as well as alphamerical data), adders, counters, decoders, all accompanying traffic control switching circuitry, sizable collections of heterogeneous combinational switching-circuit networks of varying width and depth, and sequential circuits consisting of latches, triggers, and flip-flops interconnected in complicated ways.

Some of these, particularly the registers which often comprise 30 percent to 50 percent of the CPU circuits, can be designed as modules and made alike except for word length. The same is true for adders, with binary counters included as special cases. Much can be done in systematizing so-called control logic by deliberately adding redundancies and superfluities to the chips for the purpose of standardization. Doubtless the ROS master slice can gain a wider role in microporgrammed control devices and in the construction of combinational circuitry. For sequential circuits, the use of a general purpose logic master slice is attractive.

The interfaces between the main store and peripheral equipment (input/output channels) execute multiplexing, buffering, format control, and synchronization functions. These interfaces resemble the CPU in their organization. They differ mainly in the reduction of arithmetic functions, and it is possible that they can use standard chips in their design.

11.2.6.3 Peripheral Equipment

The present trend is to include as much or more digital circuitry in all the input/output units of a complete system as is built into the CPU. This trend will probably increase in the future, and it is expected that large scale integration will have extensive applications in peripheral equipment because of the standardization it allows and of the specialized data processing functions that can be incorporated.

Section 12

ADVANCED MEMORY DEVICES

This section presents advances in memory technology and recommends which type is best for aerospace systems. It is most difficult to investigate different varieties of devices which have been proposed for use as storage media. Fortunately, the task can be narrowed to include the most important discoveries which are relevant to the requirements specified by the study program. The monolithic ferrite, the flat-film, and plated-wire memories are some of the discoveries discussed in this section. Of these, the plated-wire memories seem to be the best for spaceborne computers. Therefore, a special emphasis will be placed on the plated-wire memories throughout this section.

12.1 ADVANCES IN MEMORY TECHNOLOGY

Three types of integrated superconducting memories are discussed here. These are monolithic ferrite, flat film, and plated wire.

The integrated semiconductor memories (nonmagnetic) which include MOS field effects were discussed in the previous section. Magnetic core memories are well known since they have been the conventional storage media in the computer industry since 1950; thus they are discarded from this discussion.

Some memories with very large storage capacities, such as magnetic drums and discs, are electromechanical devices with cycle time measured not in nanoseconds or microseconds but in milliseconds. These devices are known and have been used extensively in different applications and therefore will not be discussed here.

Other memories that are being studied make use of sound waves traveling in magnetic material and also electro-optical devices such as lasers. The feasibility of such devices as acceptable media of storage has yet to be proven.

12.1.1 Monolithic Ferrite Memory

The monolithic ferrite memory consists of three sheets of ferrite and an easily magnetized ceramic material. Two of the sheets contain conducting lines set at right angles to each other; these are separated by a third sheet of ferrite to form a sandwich. The intersections of the two sets of lines constitute the memory elements. This type of integrated magnetic memory is being considered because of its high speed, compactness, and low cost. The memory is made from the same type of material as the cores. The slurry is spread by a blade onto sheets. During this operation, conducting parallel lines of a refractory metal are also formed within the sheets. Two such sheets, with their conducting lines at right angles, face each other and are separated by a third sheet of ferrite to form a sandwich. The sandwich is then heated, pressed, and sintered to produce a monolithic structure. The conducting lines define the memory elements at their crossings. One set of lines is word windings; the other is digit windings. Because the lines are at right angles to each other, a current applied to a selected word conductor switches magnetic flux along a path that does not link the digit conductor. If coincident digit pulses are applied however, there will be flux common to the word lines and the digit lines at the corresponding crossings. The application of a word pulse in the opposite direction switches the mutual flux and induces sense voltages in the digit lines.

The monolithic integrated structure not only is easy to fabricate but also makes it possible to have very small elements. Experimental laminates about 0.005 inch thick have elements with an effective diameter of 0.003 inch. The close spacing of the lines (100 to the inch) corresponds to what can be achieved with an integrated row of diodes; fully integrated combinations of laminated ferrite and diodes have been made. One of these had a capacity of 65,416 bits and operated on a cycle of 400 nanoseconds. The most delicate step in the construction is the actual connection between the laminate and the diodes.

12.1.2 Flat-Film Memory

The fabrication of flat-film memory involves the use of nickel-iron alloys called permalloys rather than ferrites. One difficulty with metals is that Eddy current develops and slows down the switching speed. If the thickness of permalloy sheets is held to about 0.00025 inch, the slowing is negligible. If the thickness is further reduced until it is comparable to the wavelength of ultraviolet radiation—say less than 3000 angstrom units—entirely new properties appear. Switching can then be achieved by the rotation of magnetization, which is a much faster process than that used with magnetic cores. In cores, switching involves a movement of the walls of magnetic domains. Thin permalloy films can be made to exhibit strong differences in magnetic properties along different directions of magnetization. In a hard direction there is practically no hysteresi but in an easy direction at right angles to it there is an almost perfect hysteresis loop. Furthermore, thin films are readily deposited by evaporation or electroplating, which lend themselves to integrated fabrication.

In one approach permalloy is evaporated in a vacuum onto glass or metal to a thickness of between 1000 and 2000 angstroms in the presence of a direct-current magnetic field, which produces the desired direction of magnetization. Separate spots of permalloy are obtained either by evaporating through a mask or by etching a continuous film. Mylar sheets with photographically formed copper lines are laid on the film to provide the word and digit lines as shown in Figure 12.1. The word lines run in the easy direction of magnetization. When a spot is storing a pulse, it is in one of the easy states of magnetization. In the read cycle a current pulse is applied to the selected word line; the corresponding spots are forcibly magnetized in the hard direction and thus induce sense voltages on the digit lines in a direction corresponding to their state. In the write or rewrite cycle currents are applied to the digit lines while the word current is still on; the combined currents tilt the magnetization from the hard direction and thus establish the easy direction at the termination of the pulses.

2008

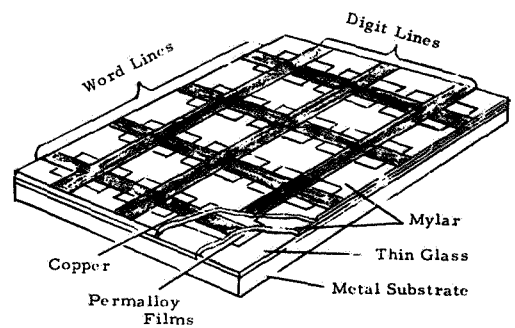


Figure 12-1. Flat-Film Memory

This arrangement is attractive in that a small digit current suffices to trigger the spot into the desired state. In practice the directions of magnetization are not perfectly aligned and vary from element to element, so that a larger digit current is required to flip the flux in the desired direction in spite of irregularities haphazardly favoring one direction or another. When these imperfections are severe, the minimum digit current can be so large that it can cause elements of unselected word lines to creep from one magnetic state to the other. This state of affairs is particularly troublesome in lines near the selected line, which are subjected to its stray word field. The field strays because lines of magnetic flux can extend between storage elements through the air adjacent to the sheet. A worse result of having the elements open in this way is a stray field that can demagnetize the elements altogether. To keep such fields from diluting the desired sharp magnetic characteristics too much, it is necessary that the length of the storage element in the easy direction be about 10,000 times its thickness, that is, more than a millimeter; this sets a limit on compactness. The elements can be smaller, however, in the hard direction. To minimize creep and demagnetization some memories have a flat ferrite keeper plate on top of the sheet of storage elements.

To STORE digital information, the two thin films are in one of two possible equilibrium states where the magnetizations are antiparallel in the easy direction. The magnetic flux is closed within the films and this leads to excellent stability against demagnetization. The closing flux penetrates the unslotted digit line.

To READ the stored information, a word field perpendicular to the easy axis must be applied. The magnetization of both films tends to align with it. The magnetization rotation of the lower film, which is imbedded between the metallic substrate and the unslotted digit line, is retarded by eddy currents in these conductors. The thickness of the digit line determines the duration of this eddy current braking, which is chosen to be longer than the reading word pulse. This means that at the end of a sufficiently short word pulse, the lower film magnetization has only been rotated by a small angle out of its original position. This film is called STORE film. The magnetization of the upper

film rotates fast, i.e., within the risetime of the word pulse into the hard direction, thus inducing a voltage in the sense line. This film is called READ film. Its fast rotation is possible since the word and sense lines are slotted and, therefore, this READ film is prevented from being eddy current braked. After the word pulse, the magnetic stray field of the STORE film forces the READ film back into its original position; the READ operation is, therefore, nondestructive.

The word pulse for WRITE is so large and long that eddy current braking of the STORE film is surmounted and its magnetization can be rotated nearly to the hard direction. The digit field coincident with the trailing edge of the word pulse forces the STORE film into the desired position. The READ film enters the antiparallel position.

Every digit pulse is preceded by a pulse having the same shape but opposite polarity. These bipolar digit pulses cause much less creeping because the initiation of a domain wall motion by the first part of the pulse is essentially canceled by the second part of the pulse. A further advantage of bipolar digit pulses is the avoidance of pumped up currents in the metallic ground-plate, especially at high speed memory operation.

Some experimental results demonstrate the properties of this memory element. Figure 12-2 shows the critical word pulse width T_W and amplitude I_W up to which memory element can be read out nondestructively. More than 10^9 READ operations have been found to have negligible influence on the stored information.

It has been demonstrated that a 140,000 bit thin-film memory can be built, which can be read out nondestructively with 20 nanosecond READ cycle time and 30 nanosecond access time, and which requires 65 nanoseconds for the WRITE operation.

12.1.3 Plated-Wire Memory

The plated-wire memory is fabricated by electroplating a beryllium-copper wire with permalloy. Straight parallel lengths of this wire are digit lines;

2007

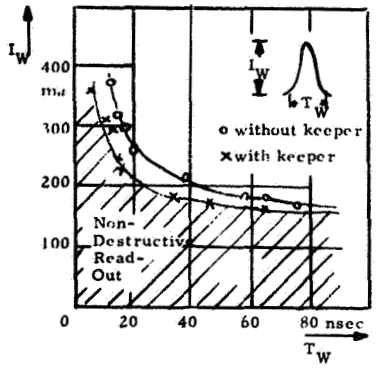


Figure 12-2. Measured Critical NDRO Curve. (Word pulses with amplitude I_W and width T_W in the shaded area do not destruct the stored information.)

conductive ribbons strapped on at right angles are word lines as shown in Figure 12-3. Storage is accomplished by magnetization around the circumference of the plated film, which is in the easy direction. That direction is established during plating by passing a direct current through the wire. The word current flips the fields toward the axis of the wire, which is the hard direction. This induces sense voltages in the digit lines, and small digit currents flip the fields to the selected easy directions as they do in the flat-film memory.

The plated-wire memory represents several fortunate compromises. Plating can be regarded as a one-dimensional integration. It is amenable to a continuous process, which should be more economical than the hand-threading of individual cores; and it is simpler to control than the lamination of ferrites or the deposition of flat films. Moreover, the plated-wire storage element has magnetic flux lines that are closed within the magnetic material in one direction, again a compromise between cores and flat films. The resulting reduction of demagnetizing effects makes it possible for the film to be comparatively thick: from 15,000 to 20,000 angstroms. This thickness of permalloy provides just the right amount of flux—more flux than the necessarily thin flat films and less flux than the difficult-to-miniaturize ferrite cores.

In spite of these attractions, it took years to develop plating techniques that avoided unfavorable magnetic effects. In addition, the wire on which the permalloy is plated is susceptible to strains that tend to distort the magnetic characteristics of the films. Such effects can be minimized by careful choice of materials, and today plated-wire memories are a technical success. This is in spite of the fact that the volt-second product of their sense output signals is very low due to their being magnetostrictive devices. The plated-wire memory is, perhaps, the first practical NDRO film memory element available for high speed operation. It operates with equal read and write word currents and NDRO operation, which allows it to be organized in a form of 2-1/2 D organization in which a long word line services 16 bytes in one read operation. A significant milestone in plated-wire technology is its commitment, in volume, to a low-cost commercial product line. The pricing is such that the plated-wire memory is significantly competitive to a core memory of equivalent performance.

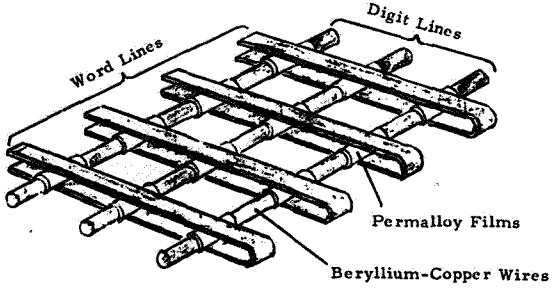


Figure 12-3. Plated-Wire Memory

Another application of plated wire is the mass direct access memory. A research contract under Air Force sponsorship is directed at a 100 megabit memory with a one microsecond cycle time, and projected costs of less than one cent per bit. In addition, other development activities encompass magnetic bit steering, for use in content addressed memories, shift registers, and magnetic logic. Plated-wire memories are indeed a major future contender for main frame memories, competing very strongly with core memories to an increasing degree in the future. The 600 nanosecond memory is a relatively early development. Memories for future machines will be faster and it appears possible, with existing techniques, that large memories down to 200 nanosecond cycle times are economically feasible.

The basic principle of operation can be visualized by referring to Figure 12-4. In writing, the word drive current rotates the magnetization in the film from the easy toward the hard direction. Digit drive current in the plated wire tilts the magnetization vector away from the hard direction. When the word current is removed, the magnetization vector rotates to the easy axis in the direction determined by the polarity of the digit drive current. The two binary storage states consist of the two possible directions of the easy-axis magnetization. The memory is interrogated by pulsing the word lines. This rotates the magnetization in the plated wire to the hard direction, inducing a voltage in the plated wire which now serves as the sense line. The polarity of the sense voltage depends upon whether the original easy-axis magnetization was clockwise or counterclockwise. In NDRO operation, the magnetization vector rotation is limited so that it falls back to its original orientation upon conclusion of the interrogate pulse.

The storage element is a ring with a closed magnetic path. Thus it is less susceptible to disturbance from external magnetic fields and gives much better coupling of the sense signal to the sense line than planar flat-film configurations. However, the sense signal has a very small volt-second value (minimum 3-millivolt signal, 50-nanoseconds wide). Reliably detecting this sense signal in the presence of external fields with low-power consumption sense electro-

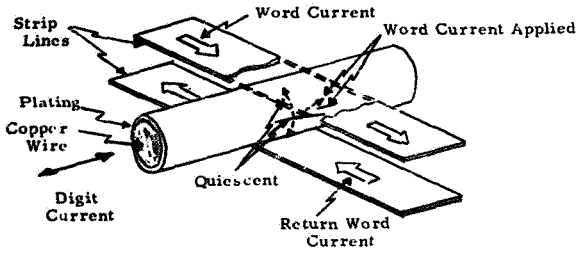


Figure 12-4. Plated-Wire Operation

nics is the most formidable problem to be overcome in successfully designing a plated-wire memory system.

The specifications of the presently available NDRO plated-wire memory planes which are most suitable for application in low power computers are as follows:

Word write current	= 200 ma
Word read current	= 200 ma
Word read rise time	= 50 ns
Digit current	= 55 ma
Minimum output voltage	= ± 3 mv

Permissible current tolerances are about ± 10 percent

Planes with unequal read and write currents or planes that are specified for only destructive readout are also available, but the system operating advantages of the planes described above more than compensate for their added cost.

These planes will operate satisfactorily over the full temperature range of -55°C to +100°C. The memory is most sensitive to external magnetic fields applied along the axis of the wire. External fields above approximately 0.5 oersted can cause faulty operation.

12.2 AEROSPACE MEMORY REQUIREMENTS AND RECOMMENDATIONS

Typical memory requirements for a spaceborne computer are 32K words, 24 to 32 bits, modular in increments of four and eight K; NDRO operation, half microsecond read cycle, one microsecond write cycle; multiword readout of 10 to 16 words per word access; operating temperature from 0 to 100°C and storage temperatures in excess of that in both directions. In addition, an ambient magnetic field of several oersteds is usually present. However, the most severe problem is the requirement for recovery from the burst of radiation. In addition, minimized volume is required, perhaps one cubic foot, nonvolatile operation, with 100,000 hours MTBF. If this was not difficult to achieve, the entire system should have low power dissipation, 20 watts being an acceptable figure, and an

objective cost of five cents a bit. Circuitry serving the stack should be minimized and some organization similar to coincident current operation is required.

Furthermore, it would be desirable to have equal word read and write currents in the low 100 milliampere range, bit currents of 10 to 20 milliamperes, without any temperature compensation required. Line impedance thus should be less than 50Ω, and delay of the line should be less than 20 nanoseconds for several thousand bit lines. Sense output should be greater than five millivolts. This implies a bit density of two to three thousand bits-per-square-inch. In particular, the radiation hardness is the most difficult because radiation pulses tend to liberate uncontrolled current from semiconductors. It is this requirement that motivates designers to specify nondestructive readout, so that when circuits go wild during this temporary environment, the system can be designed to prevent erroneous readout and still allow information to remain with integrity. One technique commonly used is to use balanced push-pull circuits so that transient currents flow from one circuit to another rather than through the stack, this tending to eliminate current in the memory array. Active components should be minimized to help achieve the reliability goal.

It follows from the above requirements that plated-wire memory is the candidate for space applications.

Section 13

RELIABILITY AND AVAILABILITY CONSIDERATIONS

This is a discussion of some reliability aspects of multiprocessing systems with emphasis on time and space-sharing of various modules, and on the interconnection between these modules provided by the switching network described earlier.

Let us consider the representative system configuration shown in Figure 13-1. It is clear from this diagram that the system is characterized by a multiplicity of modules at each functional level and the existence of all possible interconnections between modules via a distributed switching network. This configuration ensures that only failed modules are removed from system use. As units fail and are removed and repaired, the system takes on a different configuration. To assess the information capacity of the system, we must assign information capacity for each possible system configuration. Suppose the system has N levels ($N = 3$ in the example of Figure 13-1) with n_i units at the ith level. We assume that each unit at the ith level has a capacity c_i and that if k_i units are "up" (unfailed and available) the capacity of the level is $k_i c_i$. Thus, the capacity C of the total system is

$$C = \sum_{i=1}^{N} k_i c_i \qquad (13\text{-}1)$$

We have assumed in the above expression that capacities are uniformly assigned to the units at each level and that the surplus of units at one level cannot substitute for a lack of units at another. This is obvious since one cannot substitute, for example, CPUs for I/O devices or memory modules.

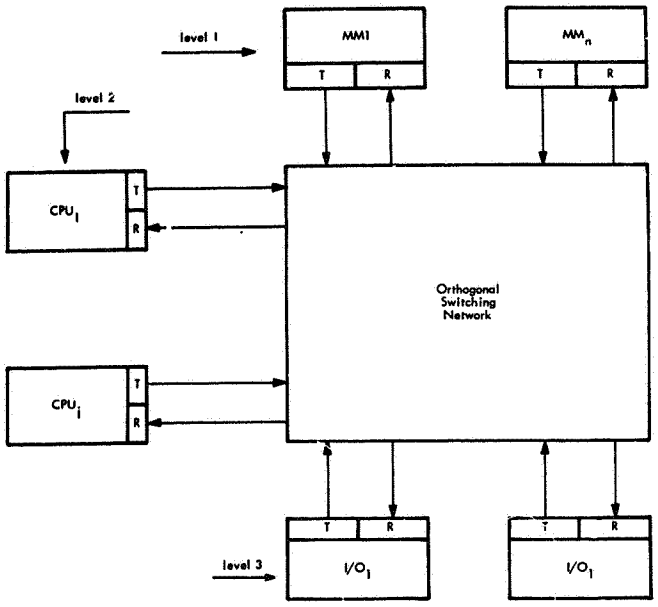


Figure 13-1. A Representative Decentralized Multiprocessing System

Let us at this point introduce some reliability parameters. For example, suppose a unit has a mean-time-to-failure of f and a mean-time-to-repair of r, then the availability (a) of that unit is given by

$$a = \frac{f}{r + f} \tag{13-2}$$

This represents the probability that the unit is in operable condition. Similarly, the unavailability (u) of the unit is given by

$$u = 1 - a = \frac{r}{r + f} \tag{13-3}$$

This represents the probability that the unit is inoperable or is undergoing repair. For any functional level, we define the availability function

$$A_i(x) = P(C_i \geq x) \tag{13-4}$$

as the probability that the capacity is greater than or equal to some variable quantity x. Similarly, the unavailability function is given by

$$U_i(x) = P(C_i < x) = 1 - A_i(x) \tag{13-5}$$

The availability function, A(x), and the unavailability function, U(x), are defined similarly for the entire system. In this case, the capacities of the level and that of the system are random variables; thus the availability and unavailability functions describe their probability distributions.

Consider a task performed by a system of n units, each unit having a capacity c. Suppose that when k units are operating, the system has a capacity C = kc. Let each of the units have an unavailability u. Then for any integer j (j=1, 2,---n) the unavailability of the capacity jc is given by

$$U(jc) = P(C < jc) = \sum_{k=n-j+1}^{n} \binom{n}{k} u^k (1 - u)^{n-k} \tag{13-6}$$

This means the probability that n-j+1 or more units are inoperable is the sum of the right side.

Now, for any integer m, the following inequalities hold:

$$\binom{n}{m} u^m (1-u)^{n-m} < \sum_{k=m}^{n} \binom{n}{k} u^k (1-u)^{n-k} < \binom{n}{m} u^m \qquad (13\text{-}7)$$

The first inequality is obvious since the quantity on the left is one of the terms in the sum on the right. The second inequality can be proven easily. In the case of small digital modules, the unavailability is very small since these units are inherently reliable; therefore, it is reasonable to assume that powers of $(1-u)$ are approximately one. Hence,

$$\sum_{k=m}^{n} \binom{n}{k} u^k (1-u)^{n-k} \approx \binom{n}{m} u^m \qquad (13\text{-}8)$$

Using the above inequalities and replacing m by n-j+1 we have:

$$\binom{n}{n-j+1} u^{n-j+1} (1-u)^{j-1} < U(jc) = \sum_{k=n-j+1}^{n} \binom{n}{k} u^k (1-u)^{n-k} < \binom{n}{n-j+1} u^{n-j+1} \qquad (1$$

Using the same approximation as above, we get

$$U(jc) \approx \binom{n}{n-j+1} u^{n-j+1} \qquad (13\text{-}10)$$

This is a good approximation for calculating system unavailability and hence can be used safely to evaluate the unavailability for various load sharing configuration For example, if we know the total number of units (n), the number of units that are available (j), and the unavailability of each unit (u) from the mean-time-to-failure and mean-time-to-repair, then overall system availability, which is a measure of its probability of survival, can be determined for a given capacity.

Consider now a general system configuration with N levels and n_i units of capacity c_i at the ith level. For any x, let $j_i(x)$ be the smallest integer greater than or equal to x/c_i. Then, for c_i the availability function is given by

$$A_i(x) = P(C_i \geq x) = \sum_{k=j_i(x)}^{n_i} \binom{n_i}{k} (1-u_i)^k (u_i)^{n-k} \qquad (13\text{-}11)$$

and the system availability function is given by

$$A(x) = P(c \geq x) = \sum_{i=1}^{N} A_i(x) \qquad (13\text{-}12)$$

From these two expressions the exact availability for various levels and for the entire system can be calculated.

For the above situation, the unavailability is given by

$$U_i(x) = P(C_i < x) \qquad (13\text{-}13)$$

As was seen earlier, the approximation for this can be written as

$$U_i(x) \approx \binom{n_i}{n_i - j_i(x) + 1} u_i^{\,n_i - j_i(x) + 1} \qquad (13\text{-}14)$$

The above two approximations for $U_i(x)$ are upper bounds; hence, the final approximation is an upper bound. Thus, upper-bound approximations to the unavailability curves can be obtained and their sum is a good upper-bound approximation to the unavailability curve for the entire system.

These observations can serve as a guideline for synthesizing a system with a given unavailability function. If one has an N level system, a desired unavailability function U(x) can be achieved by keeping

$$U_i(x) \leq U(x)/N \quad \text{for all } i \qquad (13\text{-}15)$$

Section 14

SIMULATION AND EVALUATION

14.1 GENERAL-PURPOSE SYSTEMS SIMULATOR (GPSS)

GPSS is an IBM simulation language which is useful in simulating those types of systems that can be represented as flow diagrams. In the third quarter of the study, we used the GPSS to analyze, evaluate, and compare various multiprocessing systems. The following describes the GPSS and illustrates its use in analyzing a simple network.

A flow diagram consists of a series of blocks, each describing some action of the system. The blocks are connected by lines describing the flow through the system. Whatever is flowing through the system is called a transaction. Transactions are generated in GPSS by means of a generate block. Each transaction has a group of parameters (P_j) associated with it. It also has a priority and a time. The priority determines its place on the current events chain. The time is used statistically to determine the amount of time it takes for the transaction to flow through the system.

GPSS keeps track of time in an absolute manner. The meaning of one unit of time depends on the simulation. A list and description of the GPSS entities used in our simulation programs are given in Appendix A.

14.2 GENERAL MODEL FOR SIMULATION

Figure 14-1 shows a general model of the multiprocessing system that was used in the simulation effort. In the following description certain constants and probability distributions are defined. In later simulations most of these constants and functions were under program control and were treated as variables.

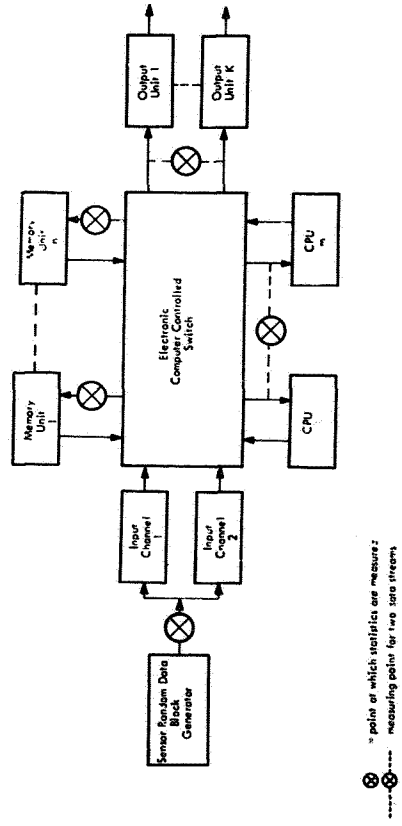


Figure 14-1. General Multiprocessing System Model for Simulation

The size of the multiprocessing system was varied under program control so that the statistical effect of adding or deleting a unit could be measured. This technique was applied in order to study the graceful degradation of a system by gradually deleting units from the system. The following describes the operation of the general model shown in Figure 14-1.

14.2.1 Data Generator

Data is generated at random time intervals and is assigned to a particular memory. (That is, the time between data bytes is a random variable and each time increment is selected with equal probability.) The data is generated in blocks of 10 bytes where each byte is assumed to contain 8 bits. Each block of data seizes the first available input channel. If the memory assigned to a particular block is not available, then a queue will be formed at the output of the data generator. This block will be stored in the input channel buffer and will hold this channel until its assigned memory is available. Statistics on the queue size at the input channels will be measured. This is denoted in Figure 14-1 by a small circle in the data-flow line. Statistics will be taken whenever a queue can be formed in this model. (Note that data enters the memory units in blocks of 10 bytes.) However, all further data transfers between the memory, CPU, and output channels are performed one byte at a time. The random data block generator simulates data generated by input sensors.

14.2.2 Memory

Data blocks in memory are processed on a serial first-in, first-out basis. Queue statistics are measured only on the memory-input lines. Queues formed by memory output data are measured at the inputs to the output channels and the central processing units.

The memory size in this model is assumed to be 1000 bytes. Statistics on memory utilization, i.e., the percentage of available memory storage, will also be gathered. The input queues to memory will measure memory overflow.

14.2.3 Central Processing Units

Data from memory will be addressed to one of the m CPU units with a probability of 0.98. This percentage was selected as being representative of a typical operation. The first available CPU will seize the memory. Queue statistics will be measured on data addressed to any CPU.

The amount of time each CPU requires to process each byte of data is a random function generated by a random number generator.

14.2.4 Output Units

The cumulative queue is measured at the input to both output units. It is assumed that memory will address one of the two output units with a probability of 0.02, since the CPU units are addressed with probability 0.98. In this model the first available output channel is seized by a memory unit. Since the probability of a memory unit addressing an output unit is low, queues should not form in the output channel unless more than two memory units address the output units within the channel transmission time.

14.2.5 Initial Simulation and Flow Diagram

The program flow diagram given in Appendix B is the result of our initial simulation effort. A modest system was selected, namely, in Figure 14-1, $m = 2$, $n = 4$, and $K = 2$. This model was used to gather basic statistics and to familiarize ourselves with this type of simulation. Once this simple simulation was working, an expanded general version was written and executed.

Data blocks are randomly generated with equiprobable time intervals (Appendix B) (1) and are assigned to a memory unit. The data then seizes the first input unit available (2). Once in the input unit, the data is buffered in temporary storage until its assigned memory is available (3). At that time the data enters memory (4) and is put on a processing chain (5). There is one chain for each memory and data is processed on a first-come, first-served basis. The data is then transferred to either a CPU unit or an output unit with certain

probability (6). If data goes to a CPU it seizes the first CPU available (7) and is processed. The processing time for each byte of data is assumed to be random (8), i.e., any increment of time selected is as probable as any other increment of time. If data goes to an output unit it is processed the same way (10). When the data has been processed, its memory is released and transit time through the system is tabulated (9).

14.3 PERFORMANCE

14.3.1 Small Systems

The program flow diagram given in Appendix B was expanded to accommodate up to six CPU's and six memory modules. The program was debugged and run on the System/360 Model 40, and the results are shown in figures 14-2 and 14-3. In these curves the number of memory modules used in the model was constant and was equal to six; the number of CPU modules was varied.

14.3.2 Results

Appendix B shows the average utilization of CPU modules as a function of the number of CPU's available in the model. Note that, as expected, the utilization of these modules decreases as the number is increased. In fact, for the case of six memory modules, a model containing only six CPU's is the most efficient. Thus, it would appear that for six memory modules, only three CPU's need be provided. However, the curve shown in Figure 14-3 shows that the waiting time for use of a CPU module decreases as the number of CPU modules increases, with the highest waiting time occurring when three modules are used. Thus, while three CPU's is optimum as far as utilization is concerned, it is the most inefficient in terms of the operational system since the average waiting time is reflected into the average program running time. Figure 14-3 shows that no significant improvement in the average waiting time is obtained by going from five to six CPU models, and in addition, from Figure 14-2, a decrease of 12 percent in utilization is obtained by this change.

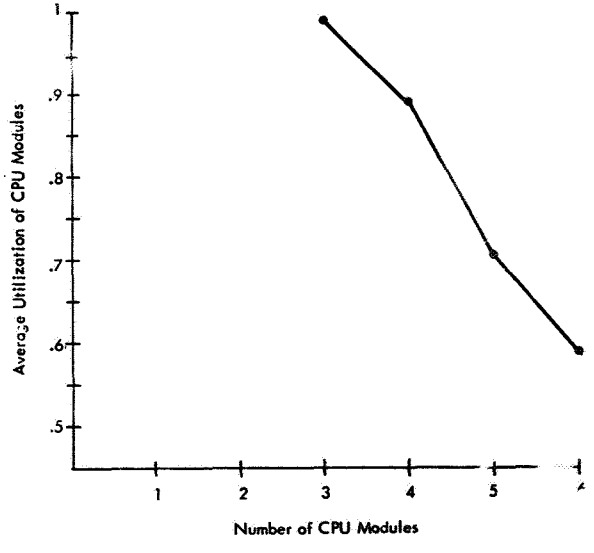


Figure 14-2. Average Utilization of CPUs Using Six Memory Modules

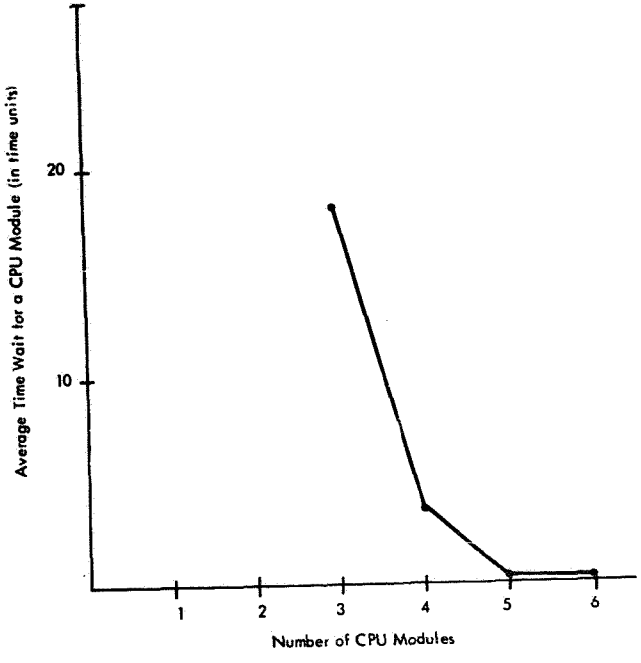


Figure 14-3. Average Queue Time for CPU Using Six Memory Modules

Taking the two curves together, we can conclude that for the parameters used in this simulation, the number of CPU models should be not less than three nor more than five.

14.3.3 Large Systems

Based upon the experience gained in writing, running, and debugging the initial simulation program described in subsection 14.2, a larger and more versatile simulation program was designed and written. In this program the number of memory modules per simulation is easily selectable by changing a single card in the program. The number of CPU modules used in the multiprocessor model is varied under program control. The multiprocessor models were simulated. In the first, 10 memory modules (fixed), 10 CPU's (variable), and two output units were included in the general model shown in Figure 14-1. In the second simulation, 50 memory modules (fixed), 30 CPU's (variable) and six output units were included.

14.3.4 Results

Figures 14-4 and 14-5 give the results of these computer runs using 10 memory modules and up to 10 CPU modules. Data blocks are generated every g time units. For the above parameters, generation times of 60, 75, and 90 time units were used. This corresponds to data block rates of 1/60, 1/75, and 1/90 blocks/unit time. Thus, the lower the generating time, the higher the transaction generation rate. Each transaction (or data block) represents 10 bytes of data. Different generation times were used to investigate the effect of different data rates upon the average CPU utilization and the average queue waiting times.

As can be seen from Figure 14-4, the utilization of CPU modules for a given number of CPU's is highest for g = 60. The reason is that at the high input data rates, fewer CPU modules are idle at any given time. Thus, it may be advantageous in a real system to multiplex the outputs of several sensors that generate data at low speeds and combine them into a single high-speed channel for entry into the computer system.

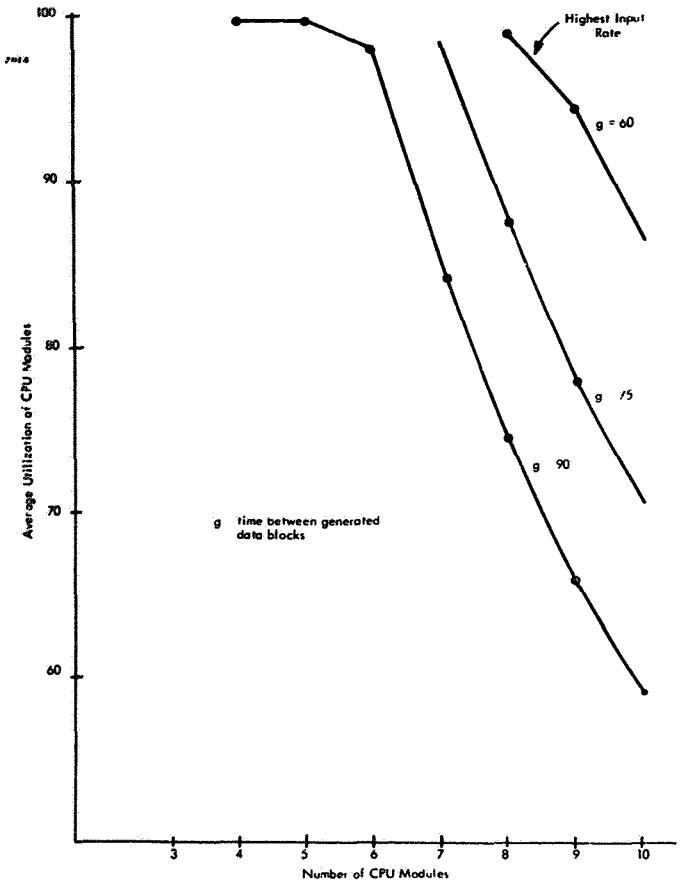


Figure 14-4. Average Utilization of CPUs Using 10 Memory Modules

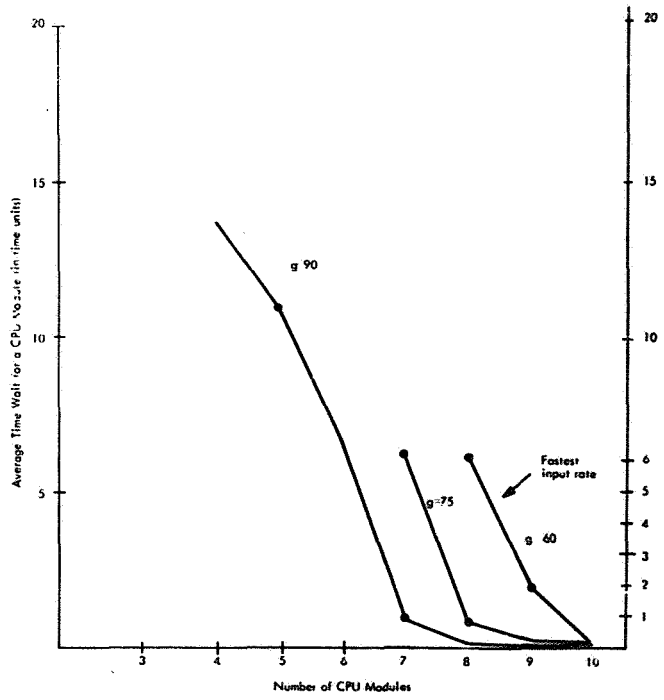


Figure 14-5. Average Queue Time for CPUs Using 10 Memory Modules

From Figure 14-5 however, it can be seen that the waiting time for a CPU module, as the system degrades, is worse for g = 60 than for g = 90. For example, consider the case where the number of CPU modules is degraded from nine to eight. Note that with nine CPU modules the average waiting time in the CPU queue is 1.85, 0.15, and 0.04 time units for g equal to 60, 75, and 90 time units, respectively, while for eight CPU modules the average waiting times are 6.1, 0.75, and 0.1, respectively. It can be seen that the greatest change in queue waiting time occurs for g = 60. High utilization of CPU's and low queue waiting times can be obtained by varying the generation time as a function of the number of CPU modules in service. This is a function that can be performed by the executive controllers in an on-board multiprocessor system.

The utilization and waiting times can be optimized if the data generation rates can be controlled. One way to accomplish optimization is to control the sensor sampling rates. That is, certain sensor inputs could be dropped from the system depending upon priorities. It is assumed that the sampling rates and quantizing resolution have already been optimized by means of the adaptive A/D converter system described in the Second Quarterly Report.

For given data rates, the curves in figures 14-5 and 14-6 specify the number of CPU models which must be provided to achieve a certain average CPU utilization and an average CPU queue waiting time. In a real system, simulations of this type can be performed for all expected input data rates. The resultant curves, such as those given in figures 14-4 and 14-5, can be stored and used by the executive controller in making decisions for reconfiguration during the flight.

Figures 14-6 and 14-7 show average CPU utilization and waiting times when 30 memory modules are used and up to 30 CPU modules are available for use. In both these curves, an input data generation rate of 1/20 words/unit time was used. This rate is triple the highest rate of 1/60 that we used in the curves given in figures 14-4 and 14-5. This high rate was used because the system capacity was essentially tripled.

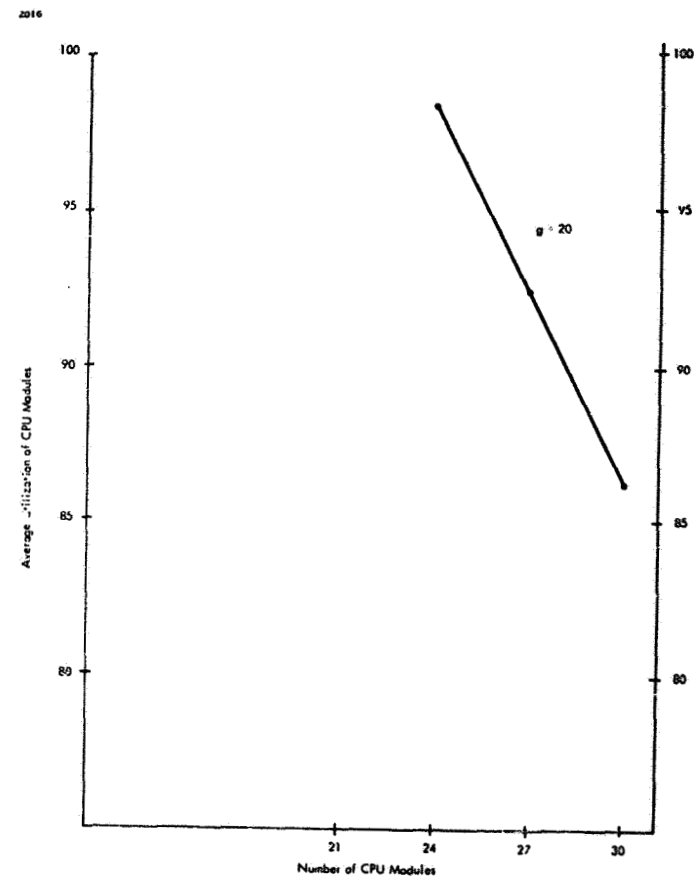


Figure 14-6. Average Utilization of CPUs Using 30 Memory Modules

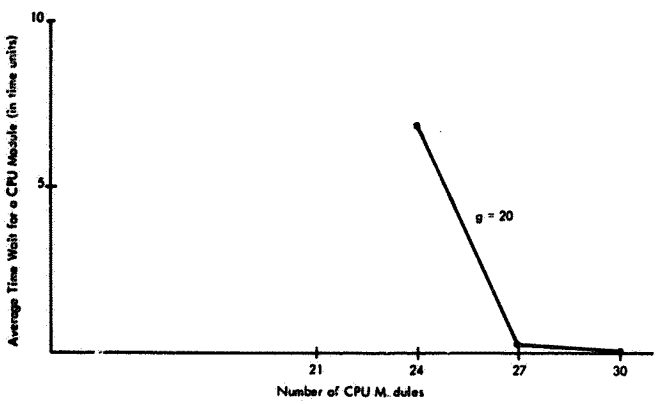


Figure 14-7. Average Queue for CPU Usage Using 30 Memory Modules

Since we have tripled the rate and tripled the system capacity, one would expect that the average CPU utilization and the average CPU queue waiting times would remain constant. If we compare the curve labeled g = 60 in Figure 14-4, between 8 and 10 CPU modules, and the curve for g = 20 in Figure 14-6, between 24 and 30 CPU modules, we see that the shapes of the two curves are essentially the same in the case of CPU utilization. Also, the average CPU utilizations for 8, 9, and 10 CPU's at g = 60 (Figure 14-4) are approximately equal to the average CPU utilization for 24, 27, and 30 CPU's, respectively, at g = 20 (Figure 14-6).

With regard to the CPU queue waiting times, it appears that while there is little change in the average CPU queue waiting time for 27 CPU's as opposed to 30 CPU's (Figure 14-7), there is a significant change when the data rate is cut by one-third and we go from 9 CPU's to 10 CPU's, as shown in the curve labeled g = 60 in Figure 14-5.

Figure 14-8 gives the frequency functions (equivalent to the probability density function except for scaling) of the transactions or data blocks as a function of the system processing time. From this curve it can be seen that the most likely processing time for a block of data is approximately 600 units of time. The mean processing time is 2017.9 time units. Data for this curve was taken for 30 memory units and up to 30 CPU units.

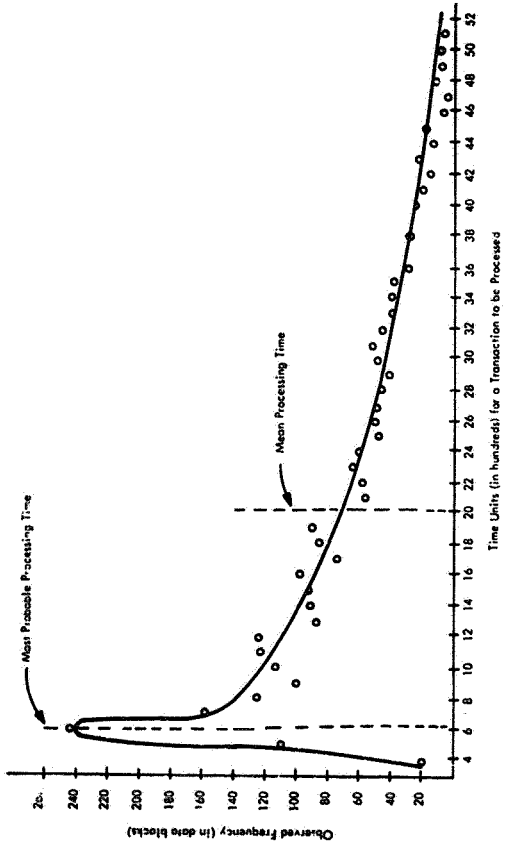


Figure 14-8. Frequency Function of Data Blocks as a Function of the System Processing Time for 30 Memory Units and up to 30 CPU Units

Appendix A

GPSS ENTITIES

The following is a list and explanation of the GPSS entities. The symbols are used in drawing simulation flow diagrams. The letters A B C D E and F are parameters of the block.

Basic Entities/Generate Block

A generate block (Figure A-1a) generates a transaction at a time depending on A, B, and C. A is the mean time B the spread, and C the offset time. C is only referenced for the first transaction generated. D is the creation limit, E the priority, and F the number of parameters of the transaction.

Thus, in the example shown in Figure A-1b, transactions would be created every 100 $\pm$ 50 time units beginning at time 10. In this case 100 transactions would be created. Their priority would be three and each transaction would have 20 parameters.

Priority Block

A priority block (Figure A-1c) changes the priority of the transaction to A. If BUFFER is stated, GPSS starts at the beginning of the current events chain.

Assign Block

An assign block (Figure A-1d) assigns B to transaction parameter A.

Equipment Entities

Facilities—a piece of equipment that can be used by only one transaction at a time. It can be preempted by another transaction. Usage statistics of all

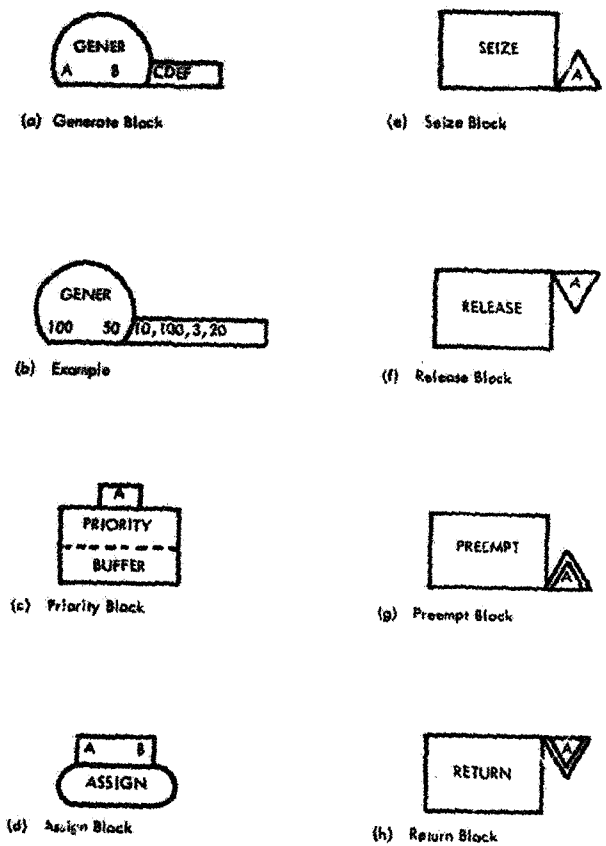


Figure A-1. Entities Blocks

facilities are automatically kept by GPSS. The blocks associated with facilities are illustrated by figures A-1e through A-1g. Figure A-1e is a seize block which seizes facility A. Figure A-1f is a release block which releases facility A. Figure A-1g is a preempt block which preempts facility A. If a transaction has already seized A, that transaction waits until A is returned. Figure A-1h is a return block which returns facility A to whatever it was doing when preempted.

Storage—has a quantity, the total amount of storage, associated with it. The two blocks used with storages are illustrated by figures A-2a and A-2b, enter and leave blocks. A transaction enters or leaves storage number A and uses or releases B amount of store. If B is not specified, 1 is assumed.

Logic Switch—is used as a binary switch. The block associated with a logic switch is illustrated by Figure A-2c where S = set (on), R = reset (off), and I = invert. All logic switches are initially in the reset (off) condition.

Computational Entities

Variables—an arithmetic variable V_j looks almost like a Fortran statement. Permissible operations are: + addition, - subtraction, * multiplication, / division, (module division. Parentheses are illegal.

As an example, in V1 = P1 + P3 / P2, variable 1 would equal parameter 3 divided by parameter 2 and added to parameter 1. To have (P1 + P3) / P2, two variable statements must be used: V1 = P1 + P3 and V2 = V1 / P2.

Functions—are defined by a Function card and Function Follower cards. There are five types of GPSS III functions:

Numerical Valued:

1.	Continuous	Cn
2.	Discrete	Dn
3.	List	Ln

Attribute Valued:

4.	Discrete	En
5.	List	Mn

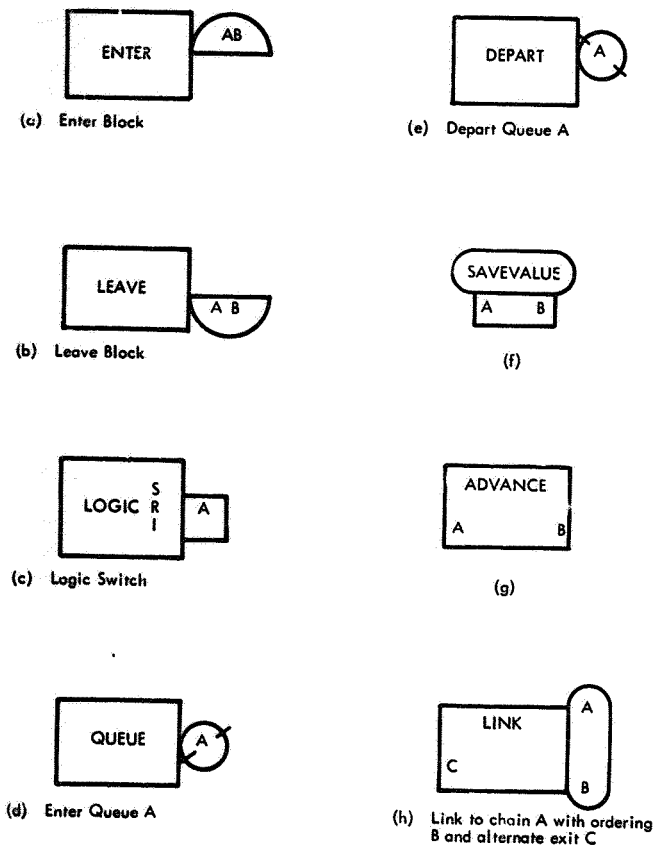


Figure A-2. Entities Blocks

A Function card has two arguments, A and B. The A field defines the independent variable of the function. The B field defines the type of function and the number of points on the function follower cards. If RN 1 is specified in the A field, the independent variable is a random number $0 \leq RN\ 1 \leq 1.0$. RN 1 has a maximum value of 0.99999. As a nonfunction argument, Rn 1 has values 000 RN 1 999, that is, each of the numbers has an equal probability 1/1000 of occurring. A list function is a function whose independent variable takes on the values 1, 2, 3, . . . n.

Statistical Entities

Queues—can be referenced for the purpose of gathering statistics at any point in the model where blocking delays occur. Statistics on queues are automatically kept by the GPSS program. Two blocks associated with queues are illustrated by figures A-2d and A-2e, Queue and Depart.

Reference Entity

Savevalue—since transactions cannot reference each other, a parameter value of one transaction can be saved in a savevalue (Figure A-2f) and then referenced by another transaction. A is the savevalue number, B the numerical attribute saved.

Other Entities

Delays are caused in GPSS by an advance block (Figure A-2g). The transaction is delayed a length of time A modified by B as in the generate block. GPSS keeps transactions in chains. Transactions presently moving through the system are on the Current Events Chain. The Current Events Chain is a First-In, First-Out Priority arranged chain. Transactions having higher priority are higher on the chain. GPSS moves only one transaction at a time starting at the top of the chain. The transaction is moved until it is blocked or placed on another chain. If a transaction is blocked, GPSS moves the next transaction on the chain. When a transaction reaches an advance block, the amount of time it is to be delayed is calculated and it is put on a future events chain.

At a generate block each time, a transaction leaves the block and a new generate time is calculated and put on the future events chain. Transactions which are delayed trying to seize a facility are put on a delay chain.

There are also User Chains which can be used in three models: FIFO, LIFO, or using an action parameter as a priority level. They are useful in speeding up machine time. There are two blocks associated with chains, Link and Unlink.

As an example of using Link and Unlink, the following may occur: (see sample program listed in Figure A-3a). Transactions waiting in queue 1 would be on a delay chain. When facility 1 is released, all the transactions would be placed on the current events chain. The first would seize facility 1 and the rest would be put on the delay chain again.

A solution to this problem is given in Figure A-3b. The link block has an indicator associated with it. When the chain is empty, it is off. The first transaction would go through the link block and turn the indicator on. It would then go through the blocks. When it reaches the unlink block (Figure A-4a), only one transaction is released from the chain. If the chain were empty, the link indicator would be turned off. Otherwise, it remains on.

Standard Numerical Attributes

The following is a table of standard numerical attributes that may be referenced by the programmer. They may be used as parameters of blocks.

Changing the flow of the model—the flow of the model may be changed in one of three ways:

1. Change block (Figure A-4b)—change block number A to block number B. Block B is unaffected.
2. Execute block (Figure A-4c)—block A is executed as if it were in the place of the executive block.
3. Gate blocks (Figure A-4d through g)—there are four different gate blocks: (1) Gate on Logic switch A (Figure A-4d). B field contains

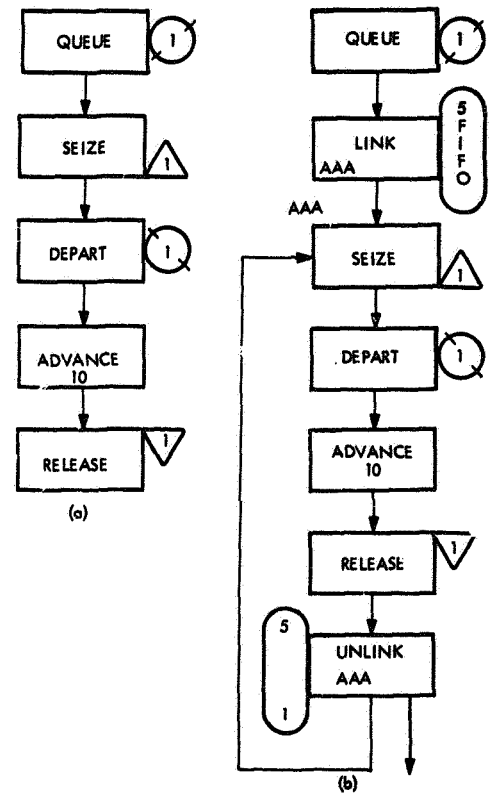


Figure A-3. Examples of GPSS Programs

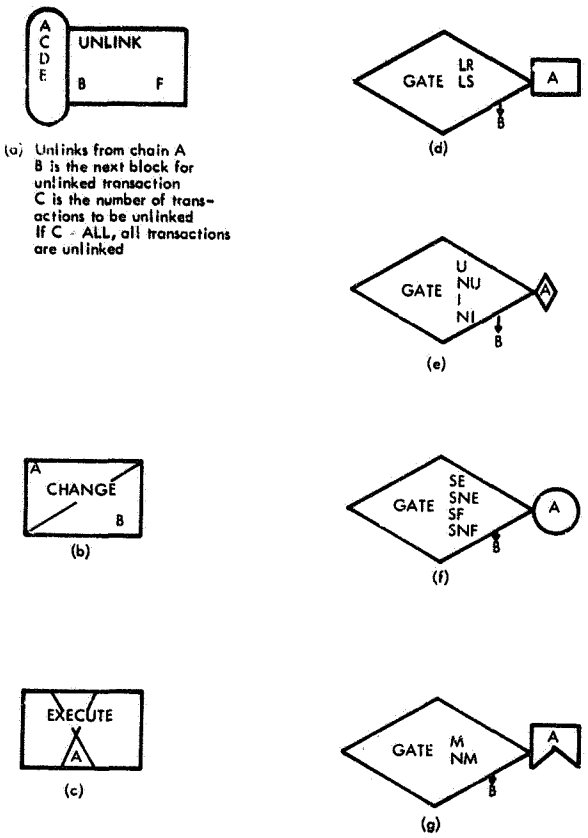


Figure A-4. GPSS Entities Blocks

next block number if Gate condition is false. (2) Gate on facility A (Figure A-4e). Condition U is true if facility is preempted or seized. NU if neither, I if seized, NI if not seized. B is next block number if Gate condition is false. If B is not specified, transaction is placed on a delay chain until Gate condition becomes true. (3) Gate on Storage number A (Figure A-4f). SE = storage empty, SNE = storage not empty, SF = storage full, SNF = storage not full, B is the same as for a facility. (4) Gate on block number A where A is a match block (Figure A-4g). M is true is a transaction and is at the match block. NM is otherwise.

A test block (Figure A-5a) will test Standard Numerical Attribute A against Standard Numerical Attribute B. If true, the transaction proceeds to the next block. If false, to block number C. L = less than, LE = less than or equal to, G = greater than, GE = greater than or equal to, E = equal to, NE = not equal to.

Splitting Transactions

Sometimes it is useful to split a transaction into a number of copies each of which follows different paths. In GPSS this is done by means of a split block (Figure A-5b). This splits the incoming transaction in A number of copies; the copies go to block B. The original and copies may be serially numbered by specifying a parameter number C. D is the number of parameters each copy is to have. An assemble block assembles copies of transactions. When A copies of a transaction reach an assemble block (Figure A-5c), they are assembled into a one transaction and continue to next block. A gather block (Figure A-5d) is similar to an assemble block except that when A copies of a transaction reach a gather block they all proceed to the next block. A match block (Figure A-5e) is used to coordinate two copies of a transaction. When a transaction reaches a match block,it checks to see if a copy is at the other block. If there is a copy at the other block, both transactions continue; if not, it waits.

Other Blocks in GPSS

Loop block—(Figure A-5f)—if parameter number A is greater than zero, go to block B and decrement A by one.

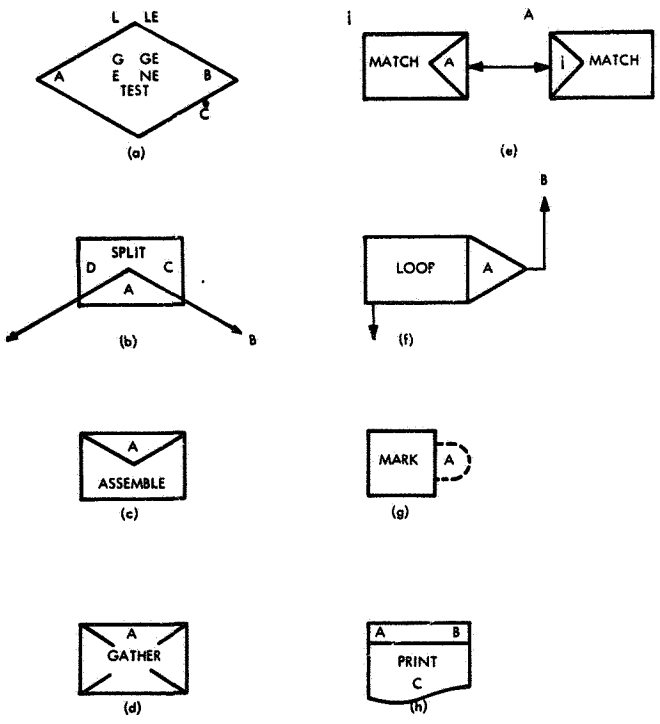


Figure A-5. GPSS Entities Blocks

Mark block—(Figure A-5g)—put present clock time C1 into parameter A.

Print block—(Figure A-5h)—print statistic C from lower limit A to upper limit B. Used to print items such as Current Events Chain.

Tabulate—(Figure A-6a)—is used along with a table card. A is the number of the table card. The table card specifies what argument of the transaction is to be tabulated.

Terminate—(Figure A-6b)—terminates transactions from the simulation. In a simulation run the number of transactions desired to go through the model can be specified. When that amount is terminated the simulation will stop. A is the number of transactions to be terminated.

Index—(Figure A-6c)—index parameter A by B.

Transfer—(Figure A-6d)—has nine models which are specified as the Mnemonic Code in A. B and C are possible next block numbers. The nine modes used are: Blank, Fractional (.), Both, All, Pick, Function (FN), Parameter (P), Subroutine (SBR), Simultaneous (SIM).

Bank Mode:

Transaction will proceed to block B.

Fractional: (.)

Selection between B and C will be done on a three-digit number basis, thus 0.33 would send 33.3 percent of the transactions to block C and 66.7 percent to block B.

Both:

The transaction will try to go to the B block. If it cannot it will try the C block. If it cannot go to either one it will remain in the transfer block until it finds an exit.

All:

The transaction will try to go to the B block. If it cannot it will try the B + D, B + 2D, ... C blocks. The transaction remains in the transfer block until it finds an exit.

Pick:

A single next block will be chosen from B, B + 1, ... C on a random basis. Transaction will remain in the transfer block until it can exit.

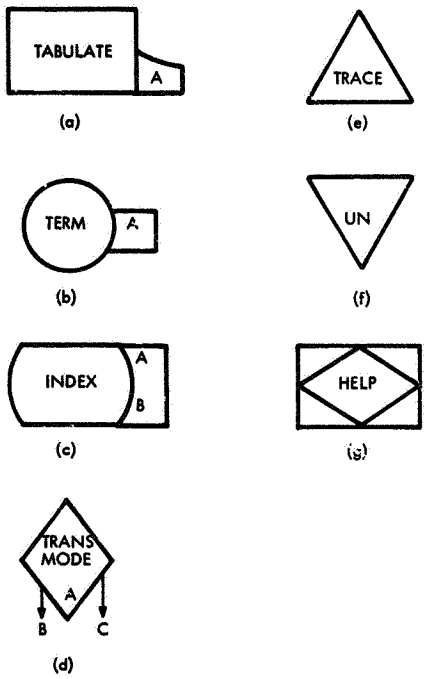


Figure A-6. GPSS Entities Blocks

Function, Subroutine, and Simultaneous Transfer—a function transfer interprets B as a function number. The function is evaluated and C is added to it giving the transfer block number. A subroutine transfer goes to block B and puts the block number of the transfer block in parameter C. This allows the transaction to return later. A simultaneous transfer goes to block B unless the transaction has been delayed, in which case it goes to block C. The delay indicator is reset to zero each time it is returned to the Current Events Chain from the Future Events Chain.

Trace and Untrace—(Figure A-6e, A-6f)—are useful in debugging. For each transaction that passes through a trace block, its path through the model will be printed out. Print out stops at the untrace block.

Help block—(Figure A-6g)—allows a programmer to call a machine language subroutine. The program returns to the next sequential block.

EXAMPLE

The above is a brief description of the language. The following is a simple example of what can be done using GPSS (see Figure A-7). We have a message processing center with messages arriving from five different terminals. The messages arrive at a Poisson arrival rate with each terminal equally likely. The message is inputted to a CPU and processed, and the outgoing message is sent to the terminal. We wish to find out how long a queue builds up at the terminals and the transit time from message generation to termination. Messages are assumed to be of 10 characters in length, answers 15 characters in length.

The generate block(1) generates the arriving messages according to an arrival rate of 500 modified by function number 1. A terminal number is then assigned(2) to the transaction by function number 2. This number is placed in parameter number 12. The transaction then enters a queue(3) for that terminal. It then tries to seize the terminal and line(4). When it succeeds it departs the queue(5) and seizes the input section of the channel.(6) If we were interested in how long it waits to do this we could put another queue here. The message is then transmitted to the channel at 5 ms per character.(7) It then enters the CPU

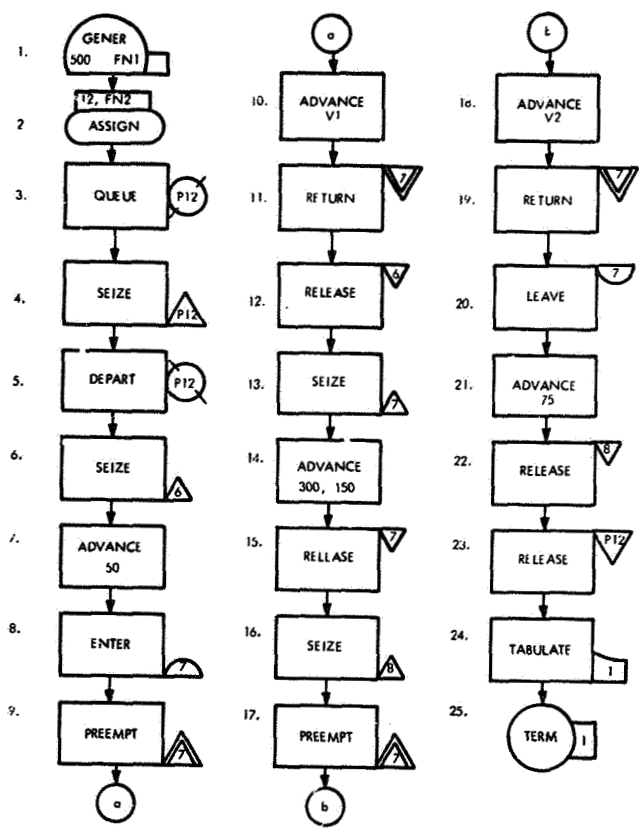
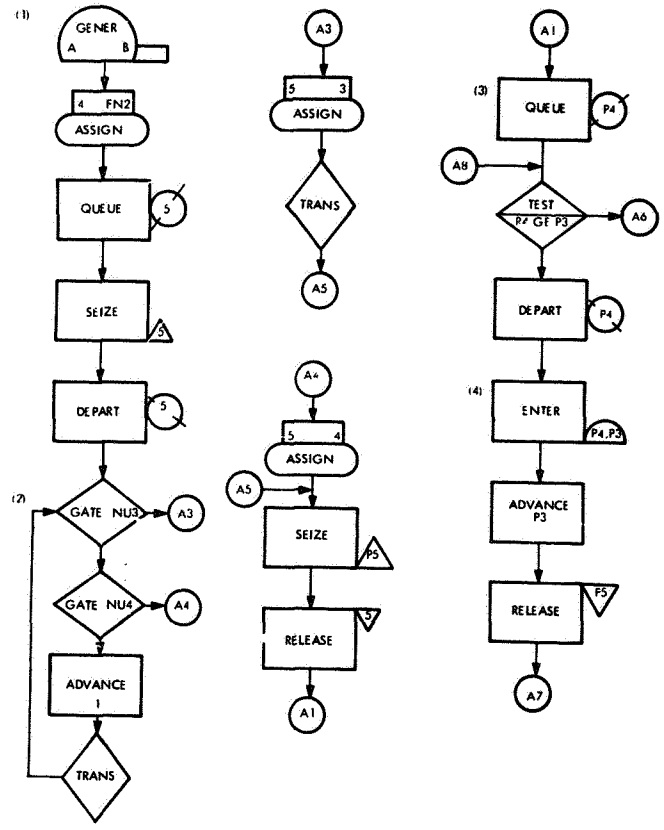


Figure A-7. Program Example

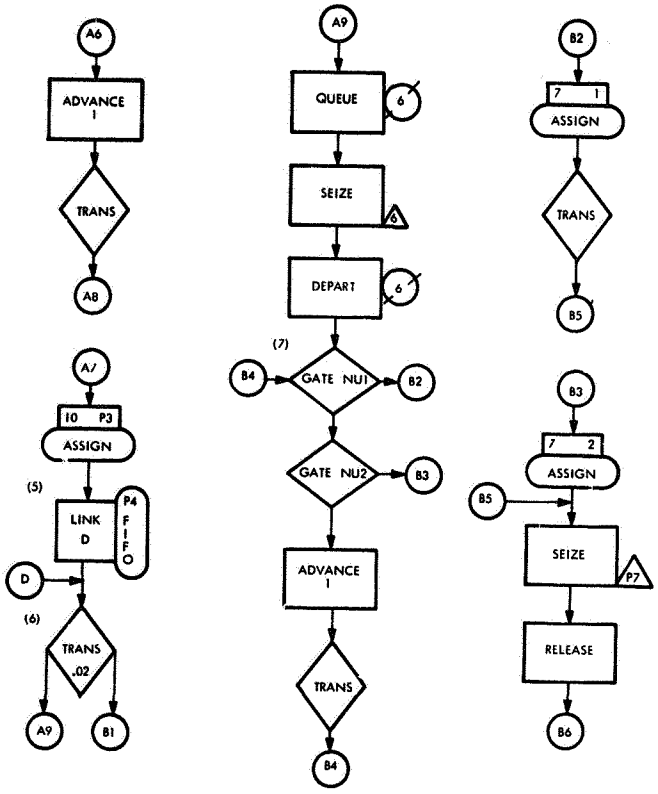
to input the message(9). The message enters the CPU from the channel at 1 ms per character(10). The CPU is then returned(11) and the input channel released.(12) The CPU is now seized to process the message(13) and the processing takes 300 ± 150 ms(14). When processing is complete, the CPU is released(15) and the output channel seized(16). The CPU is again preempted(17) and the message transmitted to the channel(18). Now the CPU is returned(19) and storage occupied by the message is freed.(20) The message is transmitted to the terminal(21) at 5 ms per character and the channel released(22). The terminal and line are now released(23) and the transit time is tabulated.(24) The transaction is then removed from the system(25) and the termination counter decremented by 1.

Appendix B

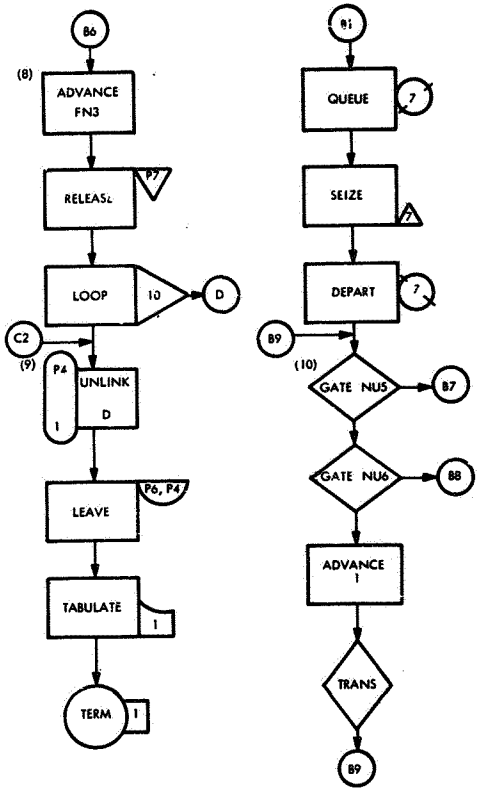
SIMULATION FLOW DIAGRAM
FOR INITIAL SYSTEM



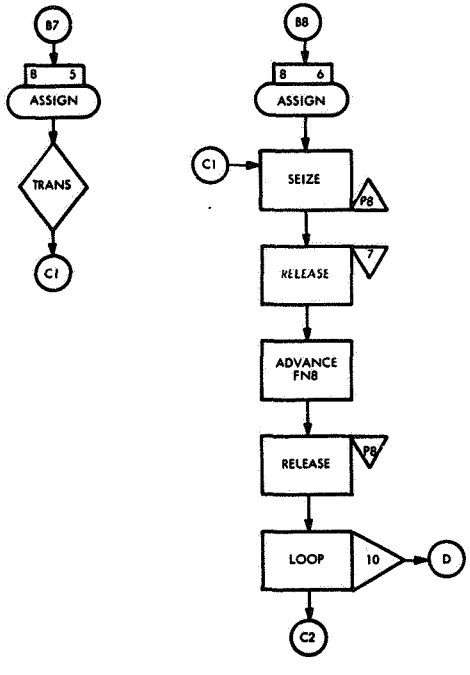
(Sheet 1 of 4)



(Sheet 2 of 4)



(Sheet 3 of 4)



(Sheet 4 of 4)

Appendix C

LARGE SYSTEM SIMULATION PROGRAM

Figure C-1 gives the flow diagram of the large system simulation program. A new block type was used in this program to replace numerous gate blocks used in the small system programs. This block allows the number of CPU modules to be varied under program control.

The new block available in the IBM System/360 version of GPSS is called the SELECT block and has the following format

SELECT X A, B, C, D, E, F

where

- X is a conditional operator, logical operator, or (max, min)
- A is a parameter in which to place entity number that meets condition
- B is the lower limit of entity class to be examined
- C is the upper limit of entity class to be examined
- D is a comparison value if conditional operator is specified
- E is an entity attribute to be examined
- F is an alternate exit if no entity meets specified condition

In the program X is NU (facility not in use), and the action of the select block is to store in Parameter A the entity number of the first available facility between the specified limits. The upper limit is a save value location which is initialized before the run. By doing this, it is possible to change the number of CPU models being simulated without having to recompute the model.

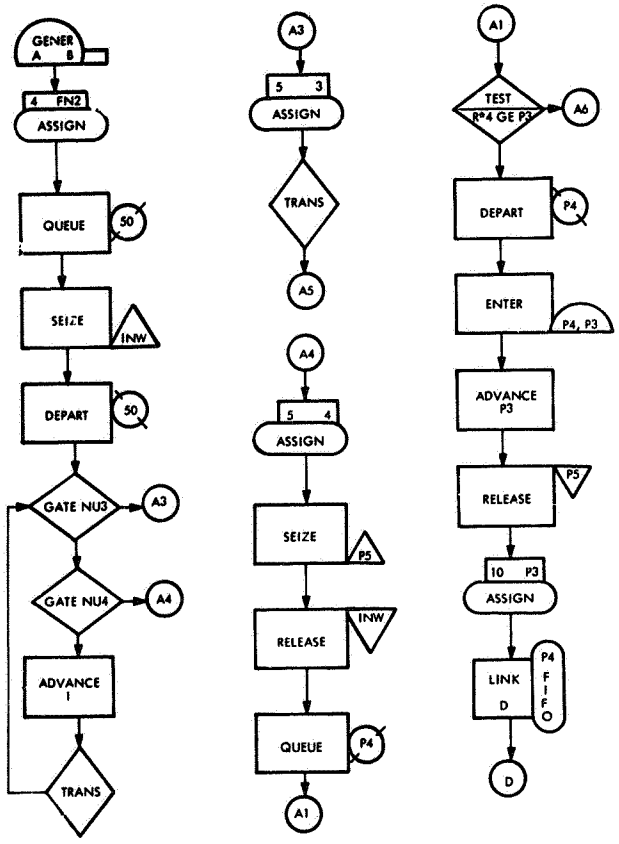


Figure C-1. Large System Simulation Program (Sheet 1 of 3)

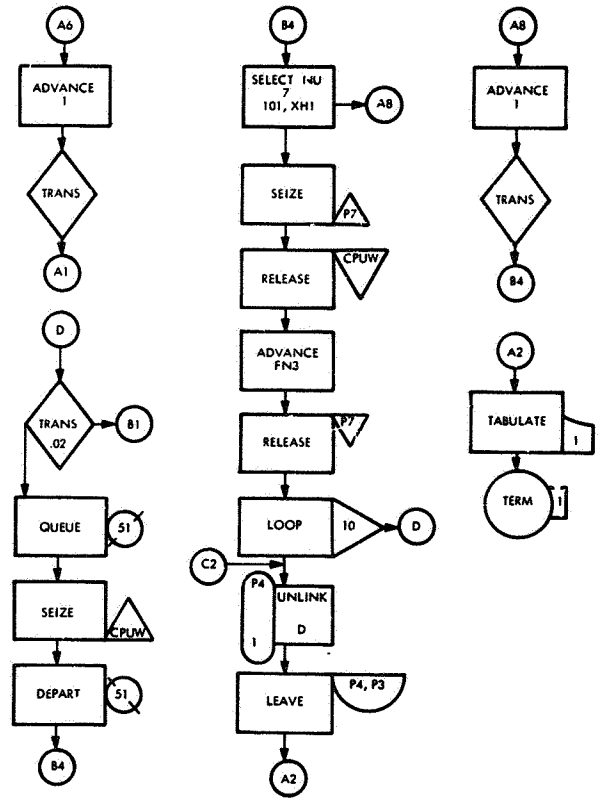


Figure C-1. (Sheet 2 of 3)

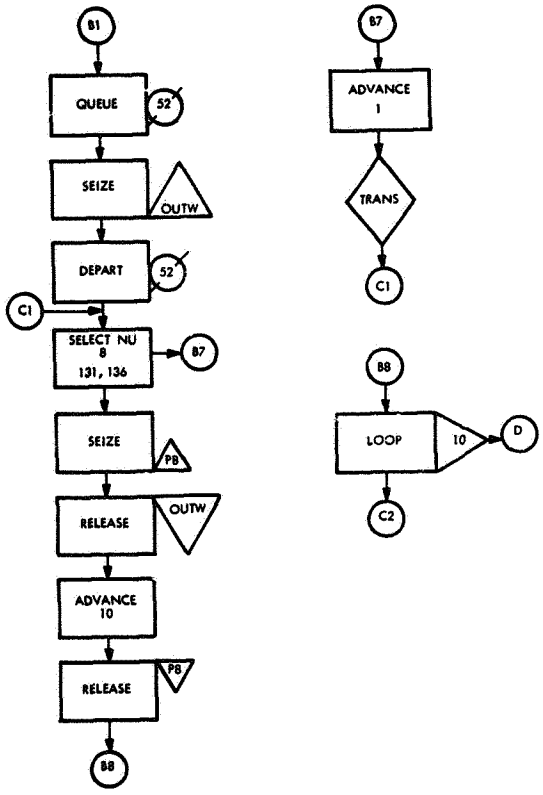


Figure C-1. (Sheet 3 of 3)

Appendix D

MULTIPROCESSING IN MULTIFUNCTIONAL SATELLITES

H.L. BLASBALG*
R.A. D'ANTONIO**
H.F. NAJJAR*
G.Y. WANG***

The following abstract was published in the August 1967 bulletin of the American Institute of Aeronautics and Astronautics (AIAA). The paper was presented by Dr. D'Antonio at the AIAA conference, "Spacecraft Issues for Missions of the 70's," held on August 28-30, 1967, in Seattle, Washington.

This paper describes the results of research to develop multiprocessing techniques for processing the diverse requirements of a multifunctional satellite. The objective is to develop new concepts of multiprocessing, oriented toward the computational needs of future long-lived satellites and to provide a sound technological base for further development of general onboard computer and data management systems. The system we have selected is composed of a large pool of memory, control processor, and input/output modules interconnected to form several general purpose computers, each assigned to a specific task. The key problem here is in the design of the computer-controlled electronic switching network which establishes the data paths between the various elements. This problem is directly analogous to the multiple access problem that exists in a satellite communication repeater. Thus, by applying the communication techniques used in solving the multiple access problem, a solution to the computer element switching problem has been found. The solution utilizes space, time, and code division switching techniques.

---

*IBM Communications and Engineering Sciences Center, Gaithersburg, Md.
**IBM Communications and Engineering Sciences Center, Cambridge, Mass.
*** NASA Electronics Research Center, Cambridge, Mass.

REFERENCES

1. N. Peterson, Error Correcting Codes, M.I.T. Press and John Wiley and Sons, Inc., 1961.

2. H. Blasbalg and R. Van Blerkom, "Message Compression," IRE Transactions of the Professional Group on Space Electronics and Telemetry, Vol. Set-8, No. 3, September 1962.

3. J. A. Buegler, Jr., and IBM Staff Members, "Adaptive Digital Satellite Transmission Ground Terminal Design Considerations," AIAA Communications Satellite Systems Conference, Washington, D. C., May 2-4, 1966.

4. "The Space Environment," Engine Operating Problems in Space, Final Report, Vol. I, NASA Publication N64-26370, p14.

5. F. Whipple, "Dust and Meteorites," Space Flight Report to the Nation, ARS Paper 2253, Oct. 9-15, 1961.

6. "Mariner-Venus 1962, Final Project Report," Jet Propulsion Laboratory, California Institute of Technology, NASA Publication NASA-SP-59, 1965, p327.

7. F. Whipple, "The Dust Cloud About the Earth," Nature, Vol. 189, Jan. 14, 1961.

8. D. P. LeGalley and A. Rosen, editors, Space Physics, John Wiley and Sons, New York, 1964.

9. T. W. Neumann, "Automated Laboratories for Scientific Exploration of Mars," Stepping Stones to Mars, AIAA/ARS Conference.

10. R. B. Blackman, and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, Inc., New York, 1958.

BIBLIOGRAPHY

Section 2

Athans and Falb, Optimal Control, McGraw-Hill Book Co., Inc., 1965.

Balakrichman, A. V., Editor, Advances in Communications Systems, Theory and Applications, Academic Press, Vol. 1, 1965, pp 108-112.

Blasbalg, H., et al., "Adaptive Satellite Data Transmission System," Presented in Chicago, Illinois, at the National Electronics Conference, October 3-5, 1966.

Brown, K. R. and G. W. Johnson, "Optimal Guidance for Orbital Transfer," Marshall Space Flight Center, Huntsville, Alabama.

Davidon, W., "Variable Metric Method for Minimization," Atomic Energy Commission R. and D. Report, Argonne National Laboratory, ANL—5990, 1959.

Fletcher, R., and M. U. D. Powell, "A Rapidly Convergent Method for Minimization," The Computer Journal, London, July 1963.

Golomb, S., et al., Digital Communications, Prentice-Hall, Inc., 1964.

Hillsley, R., "IBM Programs for Translunar Trajectories," FSD, Space Guidance Center, Owego, N. Y., IBM No. 64-512-007, August 21, 1964.

Kovasznay, L. S. G. and H. M. Joseph, "Image Processing," IRE, May 1955, pp 560-570.

Kunz, K. S., Numerical Analysis, Chapter 11, McGraw-Hill Book Co., Inc., 1957.

Lee, R. C. K., "Optimal Estimation Identification and Control," Research Monograph No. 28, the MIT Press, Cambridge, Mass., 1964.

LeSchack, A. R., "Step Size in Numerical Integration," IBM/FSD Independent Research and Development Final Report, Task No. 0494, October 28, 1964.

Nillsen, K. L., Methods in Numerical Analysis, MacMillan Company, 1956.

Southwell, R. V., Relaxation Methods in Theoretical Physics, Chapters 1 and 2, Oxford Press, 1946.

Winfield, D., "Optimum Orbital Transfer," IBM/FSD Independent Research and Development Report, Task 0232-H, April 1965.

Section 3

"ORL Experiment Program," Vol. B, Parts I, II, III, Final Report, Contract NASw-1215, February 1966.

"Small Standard Satellite (S^3) Feasibility Study," NASA, Goddard Space Flight Center, March 1966.

Stepping Stones to Mars, AIAA/ARS Proceedings, March 28-30, 1966.

"World Weather System Cost/Performance Analysis Study," U.S. Department of Commerce, Weather Bureau, Washington, D. C., Contract No. Cwb-11014, April 1965.

"Mariner 1964 Spacecraft Functional Description," Engineering Planning Document No. 277, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Calif., February 1966.

Space Technology, Vol. II, publications of Goddard Space Flight Center, 1963.

Space Sciences, Vol. I, publications of Goddard Space Flight Center, 1964.

Section 4

"A Study of Mission Requirements for Manned Mars and Venus Exploration," Technical Report by General Dynamics, Fort Worth, May 30, 1965.

Gruman, E. L. and P. S. Schaenman, "Functional Requirements of Spaceborne Computers on Advanced Manned Missions," NASA/ERC Spaceborne Multiprocessing Seminar, Boston, Massachusetts, October 31, 1966.

Purser, P. E., M. A. Faget, and N. T. Smith, Manned Spacecraft Engineering Design and Operation, Fairchild Publications, Inc., New York, 1965.

Section 7

Anderson, J. P., et al., "D 825, Multiple Computer System for Command and Control, "Fall Joint Computer Conference, 1962.

G. A. Blaauw, "The Operation of System/360," Multiple Organized IBM System, Journal 3, No. 2, Part V, p 181, 1964.

Burke, T. E., and G. Y. Wang, "Spaceborne Multiprocessing Organizations," WESCON, August 23-26, 1966.

Comfort, W. T., "A Computing System Design for User Service," Proceedings of Fall Joint Computer Conference, 1965, pp 611-626.

Critchlow, A. J., "Generalized Multi-Processing and Multi-Programming Systems," Proceedings of Fall Joint Computer Conference, 1963.

Forgie, J. W., "A Time-and Memory-Sharing Executive Program for Quick-Response On-Line Applications," Proceedings of Fall Joint Computer Conference, 1965, p 599.

Lichtenberger, W. W., et al., "A Facility for Experimentation in Man-Machine Interaction," Proceedings of Fall Joint Computer Conference, 1965, p 589.

McCullough, et al., "A Design for a Multiple User Multiprocessing System," Proceedings of Fall Joint Computer Conference, 1965, pp 611-617.

Seeber, R. R. and A. B. Lindquist, "Associative Logic for Highly Parallel Systems," Proceedings of Fall Joint Computer Conference, 1963, pp 489-493.

Section 8

Analog-to-Digital Conversion Handbook, Digital Equipment Corporation, 1964.

Bennett, W. R., "Spectra of Quantized Signals," Bell System Technical Journal, Vol. 27, No. 3, July 1948.

Blackman, R. B. and J. W. Tukey, The Measurement of Power Spectra, Dover Publications, Inc., New York, 1958.

Susskind, A. K., editor, Notes on Analog-Digital Conversion Techniques, the MIT Press, 1959.

Widrow, B., "A Study of Rough Amplitude Quantization by Means of Nyquist Sampling Theory," Trans. IRE on Circuit Theory, Vol. PGCT-3, No. 4, December 1965.

Section 9

Gilmore, J. T., Jr. and H. P. Peterson, "A Functional Description of the TX-O Computer," Memorandum 6-M-4789, Lincoln Laboratories, MIT, October 3, 1958.

Grabbe, E. M., S. Ramo, and D. E. Woodbridge, editors, Handbook of Automation Computation and Control, Vol. 2, John Wiley and Sons, 1959, pp 2-251 through 2-257.

Hagiwara, H., "The KT Pilot Computer—A Micro-Programmed Computer with a Phototransistor Fixed Memory," in IPIP 62 Proceedings of the International Federation of Information Processing, North Holland, Amsterdam, 1963, pp 684-689.

Wilkes, M. V. "Microprogramming," Proceedings of Eastern Joint Computer Conference, 1958, pp 18-20.

———, "The Best Way to Design an Automatic Calculating Machine," Manchester University Computer Inaugural Conference, Manchester, England, July 1951, pp 16-18.

Wilkes, M. V., and J. B. Stringer, "Microprogramming and the Design of Control Circuits in an Electronic Digital Computer," Proceedings, Cambridge Philosophical Society, Vol. 49, Part 2, April 1953, pp 230-238.

Wilkes, M. V., W. Renwick, and D. J. Wheeler, "The Design of the Control Unit of an Electronic Digital Computer," Proceedings IEE, Vol. 105, Part B, March 1958, pp 121-128.

Section 10

Dent, J., "Diagnostic Engineering," IEEE Spectrum, Vol. 4, No. 7, July 1967.

Lehman, M., "A Survey of Problems and Preliminary Results Concerning Parallel Processing and Parallel Processors," Proceedings of IEEE, Vol. 54, No. 12, December 1966.

Roder, J. and A. F. Rosene, "Memory Protection in Multiprocessing Systems," IEEE Trans. on Electronic Computers, Vol. EC-16, No. 3, June 1967.

Wald, R., "Utilization of a Multiprocessor in Command and Control," Proceedings of IEEE, Vol. 54, No. 12, December 1966.

Section 11

Hughes, H. L., and R. R. Girox, "Space Radiation Affects MOS FET's," Electronics, Vol. 37, No. 32, 1964.

Marvin, C. E., and R. M. Walker, "Customizing by Interconnection," Electronics, Vol. 40, No. 4, 1967.

Mitchell, M. M. and N. H. Dirtrick, "Stability Effects in MOS Enhancement Transistors," Solid State Design, November 1965.

Notz, et al., "Benefitting the System Designer," Electronics, Vol. 40, No. 4, 1967.

Schegel, E. S., "Design, Development, Fabrication, and Delivery of Improved MOS Transistors," 1st, 2nd, and 3rd Quarterly Progress Reports, Philco Corporation, Lansdale, Pennsylvania, 1965-1966.

"System Study for Low Power Spaceborne Computer," Final Report, Digital Computer Laboratory, A. L. Electronics Division, El Segundo, California, November 1966.

Section 12

Fedde, G. A., "Design of a 1.5-Million-Bit Plated-Wire Memory," Journal of Applied Physics, Vol. 37, 1966, p 1373.

Fedde, G. A., "Plated-Wire Memories," Electronics, Vol. 40, May 1967, p 109.

Kaufman, B., "Advances in Memory Technology," IEEE Computer Group News, July 1967, p 28.

Kohn, G., et al., "A Very High-Speed Nondestructive Read Magnetic Film Memory," IBM Zurich Research Laboratory, Ruschlikon, Switzerland.

Rajchman, J. A., "Integrated Computer Memories," Scientific American, July 1967, p 18.

Smith, J. L., "Waffle Iron —A New Memory Structure," Journal of Applied Physics, Vol. 34, Part 2, p 1167.

"System Study for Low Power Spaceborne Computer," General Motors Corporation, El Segundo, California.

Section 13

Bazovsky, I., Reliability Theory and Practice, Prentice-Hall, Inc., New Jersey, 1961.

Dixon, W. J. and F. J. Massey, Introduction to Statistical Analysis, McGraw-Hill Book Co., Inc., New York.

Hoel, P. G., Introduction to Mathematical Statistics, John Wiley and Sons, New York, 1963.

Morse, P. M., Queues, Inventories, and Maintenance, John Wiley and Sons, New York, 1962.

Roberts, N. H., Mathematical Methods in Reliability Engineering, McGraw-Hill Book Co., Inc., New York, 1964.