TECHNICAL ADVISEMENT MEMORANDUM NO. 171-5

ONBOARD CHECKOUT AND DATA MANAGEMENT SYSTEM (OCDMS) COMPUTER SELECTION CRITERIA

25 August 1967

PRC D-1180

Prepared by

D. H. Stomberg

PLANNING RESEARCH CORPORATION
LOS ANGELES, CALIF. WASHINGTON, D.C.

PRC D-1180

This document was prepared by Planning Research Corporation under Contract Number NAS8-20367, "An Airborne Evaluating Equipment Study," for the George C. Marshall Space Flight Center of the National Aeronautics and Space Administration. The work was administered under the technical direction of Qualtiy and Reliability Assurance Laboratory, Marshall Space Flight Center, with Walter T. Mitchell acting as project manager.

TABLE OF CONTENTS

LIST OF REFERENCES

Periodicals

1. Flynn, Michael J., "Very High-Speed Computing Systems," Proceedings of the IEEE, December, 1966, Vol. 54, No. 12, pp. 1901-1909.

2. Slotnik, D. L., Borch, W. C., and McReynolds, R. C., "The Solomon Computer - A Preliminary Report," Proc. 1962 Workshop on Computer Organization, Washington, D. C., Spartan 1963, pp. 66-92.

3. Senzig, D. N., and Smith, R. V., "Computer Organization for Array Processing," Proc. AFIPS 1965 FJCC, pp. 82-91.

4. Comfort, W., "Highly Parallel Machines," Proc. 1962 Workshop on Computer Organization, Washington, D. C., Spartan, 1963, pp. 126-155.

5. Holland, J. H., "A Universal Computer Capable of Executing an Arbitrary Number of Subprograms Simultaneously," Proc. EJCC 1959, pp. 108-113.

6. deRegt, Maurits, P., "Negative Radix Arithmetic," Computer Design, May 1967, Vol. 6, No. 5, pp. 52-63.

Technical Reports

7. PRC D-1177, "Support System for Onboard Checkout and Data Management System (OCDMS), Saturn Apollo Applications Program," 23 August 1967. Revision A.

8. PRC D-1178, "Supervisory System for Onboard Checkout and Data Management System (OCDMS), Saturn Apollo Applications Program," 23 August 1967. Revision A.

9. PRC D-1237, Technical Advisement Memorandum No. 171-1, "A Survey of Spaceborne Computers," 27 June 1966.

10. PRC D-1336, "Onboard Checkout and Data Management System: Hardware Definition," 11 November 1966.

11. PRC D-1398, "Performance and Design Characteristics for The Onboard Checkout and Data Management System (OCDMS)/ Apollo Applications Program," 30 June 1967.

LIST OF REFERENCES (Continued)

12. PRC D-1417, Technical Advisement Memorandum No. 171-4, "Onboard Checkout and Data Management System, Control and Display Unit," 19 July 1967, Revision A.

13. PRC D-2001, Technical Advisement Memorandum No. 171-7, "Preferred Computer Manufacturer-Supplied Software," 28 August 1967.

14. PRC R-850, "Onboard Checkout and Data Management System: Software Design," 17 November 1966.

Papers

15. Lewis, D. R. and Mellen, G. E., "Stretching LARC's Capability by 100-A New Multiprocessor System," (presented at the 1964 Symposium on Microelectronics and Large Systems, Washington, D. C.).

16. Williman, A., Koezela, L., and Burnett, G., "Multiprocessor Organization for Manned Mars Mission" (presented at the Spaceborne Multiprocessing Seminar, Boston, Massachusetts, 31 October 1966).

17. Alonso, R., Hopkins, A. L., and Thaler, H. A., "Design Criteria for a Spacecraft Computer," (Ibid).

18. Avizienis, A., "System Organization of the JPL Self-Testing and Repairing Computer and Its Extension to a Multiprocessor Configuration (Ibid).

I. INTRODUCTION

The definition phase of the Onboard Checkout and Data Management System (OCDMS) has established software as an intergral part of the system. That is, the General Specification for Performance and Design, Reference 11., identified three major OCDMS functional areas which includes the Computer Subsystem, Communication Subsystem, and Software Subsystem. In many respects, this design approach differs from the usual system definition which results from the study preceding the acquisition of computer-based systems. Computer problem-solving application always require defined software specifications, but not normally as integral to the definition of the system design. Thus, parts of the software such as the interrupt-handling procedure, the core-allocation scheme, and the task-scheduling algorithm will have substantial effect upon the overall performance of the system.

Moreover, the software design criteria because of its system significance and tendency to standardize application approaches can help select certain operational features of the other main subsystems. Therefore, the purpose of this memorandum is to present the results of analysis directed to such an objective. Software motivated criteria to aid in selection of the OCDMS computer are developed. Compatibility with the OCDMS operating concepts, manufacturer-supplied software availability, and cost implications of supplementing that software are major areas of selection concern.

However, the decision factors developed in this text must not be considered in conflict with the computer performance/design requirements detailed by the OCDMS General Specification, reference 11. This criteria is intended to aid in selection of a single computer from several which may satisfy the specifications, or which present possible hardware trade-offs in favor of software development. The requirements identified by the General Specifications are nevertheless proper prerequisites for the OCDMS computer and are not explicitly repeated in this text. No less significance should be attributed to reliability, low power demands, and space conservation because of the emphasis given here to computer programming aspects.

II. MISSION BASELINE

The general concensus of opinion expressed by Williman, Alonso, and Avizienis in related recent engineering literature (see references 16, 17, and 18) is that future mission complexity, by itself, justifies highly sophisticated spaceborne computers which can be applied to the reduction of operational complications. Redundant, highly modular, parallel, and reconfigurable systems are considered the basic organizational structure capable of providing successful attainment of mission objectives.

No serious argument appears to exist that the computer selection criteria should not depend on the definition of information processing requirements, and as such will be highly mission dependent. Thus, operational demands with impact on computer applications will be most conveniently identified in terms of:

- Tracing the information flow of all command/control messages;
- Identification and location of all potential information transformation points;
- Grouping operational steps between control points;
- Developing flow charts which represent control points, the flow of information and/or time, and the allocations of resources.

The above must be examined on the basis of the particular projects with which a given OCDMS configuration is used on (e.g., launch vehicle, spacecraft, launch support, integration test and checkout). Variables must be identified that relate computer applications with:

- crew and support personnel operations
- mechanical and structural systems
- electrical and network systems
- guidance and navigation systems
- propulsion and attitude control systems
- life support and environment control system
- data and communication systems
- GSE and facility systems
- scientific/engineering experimentation systems
- other ground/vehicle systems as established by the product configuration baseline.

Present analysis indicates that these kinds of variables will generally involve operations which permit simultaneous independent operations on each individual variable within the set. This is important not only from the standpoint of establishing potential applications of OCDMS, but also is a governing factor which heavily influences the development of this computer selection criteria, as will be shown.

III. OPERATIONAL REQUIREMENTS

A probable outcome of multiple mission operational requirements will be a family of OCDMS spaceborne computers. It is reasonable to expect that no single computer configuration will satisfy the set of anticipated missions which are likely to include earth-orbit space stations, lunar exploration, Mars and Venus orbital missions, Mars manned landing, and other manned or unmanned planetary/solar system probes. Even superficial analysis of operational aspects of such missions indicates a wide range of performance criteria, indices, object functions, and penalty functions which will be involved.

The OCDMS development has to-date and will continue to follow the guidelines of the Apollo Configuration Management Manual, NPC 500-1. This means that the operation requirements that will influence the particular OCDMS computer configuration will be expressed in terms of:

- Performance Characteristics
- System Definitions
- Operability
  - Reliability
  - Maintainability
  - Useful Life
  - Human Engineering
  - Safety
  - Environment (Operating and Induced)

Other OCDMS documentation has described and specified these quantities in considerable detail. (References 7-14) The reason for referring to these documents is to emphasize that they represent principle elements of the data base, or in some way relate to parameters in the OCDMS computing process.

A sequence of instructions will perform on these parameters or data sets in the OCDMS computing process just as in all other computer applications.

A computer program always consists of an ordered set of instructions. Each instruction performs a combinatorial manipulation on one or two elements of the data set. If the element is a single bit and only one such bit can be manipulated at any one computational period, then the process is strictly serially sequential.

However, the usual case is to introduce data whose elements more closely correspond to the natural information quantum established by the operational requirements (characters, integers, floating-points numbers, etc.). Since the size of the datum has increased, so too has the number of combinational manipulations that can be performed (manipulation on two n-bit arguments have two 2^n possible outcome). Of course, attention is normally restricted to those operations which have arithmetic or logical significance.

All computer operations from the standpoint of software will ultimately resolve into a set of instructions of the following types:

- Control and Status Switching
- Branch/Jump/Skip
- Load and Store
- Arithmetic
- Logical
- Shift/Convert
- Input/Output

Tables 1 and 2 show the particular instruction repertoire of the two current contending leaders of the computer to be selected for OCDMS. We shall be mainly concerned with techniques to extend computation performance in a context of optimizing the interactions between given sequences of these kinds of instructions and the data stream which can be traced to sources established by mission requirements.

IBM 4π-EP INSTRUCTION LIST

A. Fixed Point Instructions	Execution Time (usec.)
Load	1.9
Load	5.0
Load Halfword	5.0
Load and Test	1.9
Load Complement	2.1
Load Positive	2.1
Load Negative	2.1
Add	2.1
Add	5.0
Add Halfword	5.4
Add Logical	2.1
Add Logical	5.0
Subtract	2.1
Subtract	5.0
Subtract Halfword	5.4
Subtract Logical	2.1
Subtract Logical	5.0
Compare	2.1
Compare	5.0
Compare Halfword	6.0
Multiply	9.2
Multiply	10.4
Multiply Halfword	11.6
Divide	20.0
Divide	20.8
Store	5.0
Store Halfword	5.0
Shift Left Single	Variable
Shift Right Single	Variable
Shift Left Double	Variable
Shift Right Double	Variable
B. Logical Instructions	
Compare Logical	1.9
Compare Logical	5.0
Compare Logical	5.4
And	3.4
And	5.0
Or	3.4
Or	5.0
Exclusive Or	3.4
Exclusive Or	5.0
Test Parity	5.0

TABLE 1

	Logical Instructions	Execution Time (usec.)
	Test Under Mask	5.0
	Sumcheck	Variable
	Insert Character	5.0
	Store Character	5.0
	Load Address	2.9
	Shift Left Single Logical	Variable
	Shift Right Single Logical	Variable
	Shift Left Double Logical	Variable
	Shift Right Double Logical	Variable
C.	Branching Instructions	
	Branch on Condition	4.2
	Branch on Condition	4.4
	Branch and Link	4.0
	Branch and Link	4.1
	Branch on Count	4.2
	Branch on Count	4.4
D.	Status Switching Instructions	
	Load Program Status Word (PSW)	7.5
	Load PSW Special	9.0
	Set Program Mask	2.1
	Set System Mask	5.0
	Change Priority Mask	5.4
	Supervisor Call	15.0
	Set Storage Key	4.5
	Insert Storage Key	5.0
E.	Input/Output (I/O) Instructions	
	Start I/O	Variable
	Test I/O	Variable
	Halt I/O	Variable
	Test Channel	Variable
	Read Direct	Variable
	Write Direct	Variable

TABLE 1 (Continued)

LITTON L-304 INSTRUCTION LIST

INSTRUCTION	EXECUTION TIME (μSEC)†
A. DATA TRANSMISSION INSTRUCTIONS	
LOAD RH	6.6
STORE RH	6.6
LOAD DOUBLE	6.4
LOAD TWO'S COMPLEMENT	7.2
LOAD ABSOLUTE	7.2
STORE DOUBLE	6.4
MOVE AND INSERT	7.4 + 0.8n*
MOVE AND ZERO	6.6 + 0.8n*
EXCHANGE	8.8
EXCHANGE DOUBLE	9.4
B. ARITHMETIC INSTRUCTIONS	
ADD	7.2
SUBTRACT	7.2
REPLACE-ADD	7.4
REPLACE SUBTRACT	7.4
ADD ABOSLUTE	7.2
SUBTRACT ABSOLUTE	7.2
MULTIPLY	18.8 + 0.8n**
DIVIDE	33.2
C. LOGIC INSTRUCTIONS	
EXCLUSIVE OR	7.2
INCLUSIVE OR	7.2
LOGICAL AND	7.2
REPLACE EXCLUISVE OR	7.4
REPLACE INCLUSIVE OR	7.4
REPLACE LOGICAL AND	7.4
D. SHIFT INSTRUCTIONS	
SHIFT LONG LEFT	7.6 + 0.8***
LONG SHIFT RIGHT, ALGEBRAIC	7.6 + 0.8***
NORMALIZE LONG LEFT	10.6 + 0.8***
SHIFT AND COUNT	8.4 + 0.8***
REFLECT	7.6 + 0.8***

* n = NUMBER OF SHIFTW REQUIRED TO ALIGN THE FIELD

** n = NUMBER OF "1" BITS IN MULTIPLIER

*** n = NUMBER OF PLACES SHIFTED

TABLE 2

INSTRUCTION	EXECUTION TIME (μSEC)†
E. <u>TRANSFER INSTRUCTIONS</u>	
MODIFY RH BY TWO; TRANSFER IF RH ≠ 0	7.2
MODIFY RH BY ONE; TRANSFER IF RH ≠ 0	7.2
TRANSFER UNCONDITIONAL	4.4
TRANSFER UNCOND. AND STORE LINK	6.4
TRANSFER ON CONSOLE X SWITCH	4.4
TRANSFER IF RH = 0	6.4
TRANSFER IF RH ≠ 0	6.4
TRANSFER IF RH IS NEGATIVE	6.4
TRANSFER IF RH IS POSITIVE	6.4
F. <u>JUMP INSTRUCTIONS</u>	
JUMP THREE WAY	7.4
COMPARE, JUMP IF LESS	7.2
COMPARE, JUMP IF EQUAL	7.2
COMPARE, JUMP IF UNEQUAL	7.2
COMPARE, JUMP IF GREATER	7.2
GATED COMPARISON, JUMP IF INSIDE	9.4
GATED COMPARISON, JUMP IF OUTSIDE	9.4
TEST LOWER BIT, JUMP IF 0	5.2
TEST UPPER BIT, JUMP IF 0	5.2
TEST LOWER BIT, JUMP IF 1	5.2
TEST UPPER BIT, JUMP IF 1	5.2
G. <u>MISCELLANEOUS INSTRUCTIONS</u>	
NO OPERATION	2.2
EXECUTE	2.2
SET LOWER BIT	5.2
SET UPPER BIT	5.2
RESET LOWER BIT	5.2
RESET UPPER BIT	5.2
STORE ALL ZEROS	5.2
HALT	2.2
H. <u>INPUT/OUTPUT INSTRUCTIONS</u>	
MEMORY BANK ASSIGNMENT	4.4
INPUT TO REGISTER	7.2
OUTPUT FROM REGISTER	7.2
EXTERNAL DEVICE COMMAND	7.6
MEMORY BANK DESIGNATOR	4.4

† ASSUMING NO MEMORY OVERLAP. IF MEMORY OVERLAP OCCURS, SUBTRACT 0.6 MICROSECOND FROM THESE TIMES.

TABLE 2 (Continued)

IV. COMPUTER CHARACTERISTICS

A. Organization

We shall be concerned with three different kinds of computer characteristics, in our selection critera, namely

- organizational features
- central processor unit (CPU) features
- input/output (I/O) features.

Of the three, organization has the most profound influence on overall aspects of the software system. CPU Characteristics such as the number system, processor storage, timing and machine programming greatly effect coding practices and performance at execution time. However, the computer organization has equivalent impact in these areas, and additionally effects most significantly the computer program structure, the extent of automation processes, and therefore the amount of software technology involved.

In order to make this discussion apropos to the development of computer selection standards, the following definitions are adopted from Flynn (reference 1).

- Instruction Stream is the sequence of instructions performed by a machine.
- Data Stream is the sequence of data called for by this instruction stream (including input and partial or temporary results).
- Bandwidth is an expression of time-rate of occurrence. In particular, computational or execution bandwidth is the number of instructions processed per second and storage bandwidth is the retrieval rate of operand and operational storage words (words/second).

- Latency or latent period in the total time associated with the processing (from stimuli to response) of a particular data unit at a phase in the computing process.

The concepts involved with these definitions are quite useful in categorizing basic computer organizations and other features in an attempt to avoid the ambiguous term "parallelism." Organizations are conveniently characterized by the multiplicity of the hardware provided to service the Instruction and Data Streams. The multiplicity is taken as the maximum possible number of simultaneous operations (instructions) or operands (data) being in the same phase of execution at the most constrained component of the organization. The ratio of the number of simultaneous instructions processed to the constrained multiplicity is called the confluence (or concurrence) of the system.

Four organizational classes can be distinguished as a result of the above definitions:

1) Single Instruction Stream-Single Data Stream (SISD).
2) Single Instructions Stream-Multiple Data Stream (SIMD).
3) Multiple Instruction Stream-Single Data Stream (MISD).
4) Multiple Instruction Stream-Multiple Data Stream (MIMD).

SISD

We may better understand the listed organizational classes in an established reference by giving examples of the different machine configurations. The RCA-110A, the IBM 704-709, 7090 are examples of SISD. Confluent SISD processor organizations achieve increased bandwidth by various techniques of overlapping the various sequential decision processes. Examples of configurations which may be described in this manner include the

CDC 6600 series, and the IBM System 360 and 4π series. Exhibit 1 illustrates the SISD organization.

SIMD

Slotnik (reference 2) and others describe the classic SIMD-type structure as the Solomon computer which recently has been superseded and modified to some extent by ILLIAC IV. The Litton L-304 computers may be considered a member of this class. Exhibit 2 represents the basic SIMD Organization.

MISD

MISD-type structures have received much less attention, but have been proposed by Senzig and Smith (reference 3), and others. An example of such a structure is shown in Exhibit 3. A wide bandwidth execution unit is shared by a number of virtual machines operating on program sequences independent of one another. Each virtual machine has access to the execution hardware once per cycle. Each machine has its own instruction memory and interactions between instruction streams occur only via the common data memory.

MIMD

If the organization shown by Exhibit 3 is reconstructed so that the data and instruction streams are maintained together in private storage, we have an example of MIMD. There is no interaction or minimum interaction allowed between these virtual machines. So long as latent time for execution of any operation is less than a memory cycle, no problem arises due to branching or looping. General MIMD structures have widely described (references 4, 5, and 15) with large-scale multiplicity proposed by Holland, and more restricted implementation being undertaken by Burroughs and Univac (reference 15).

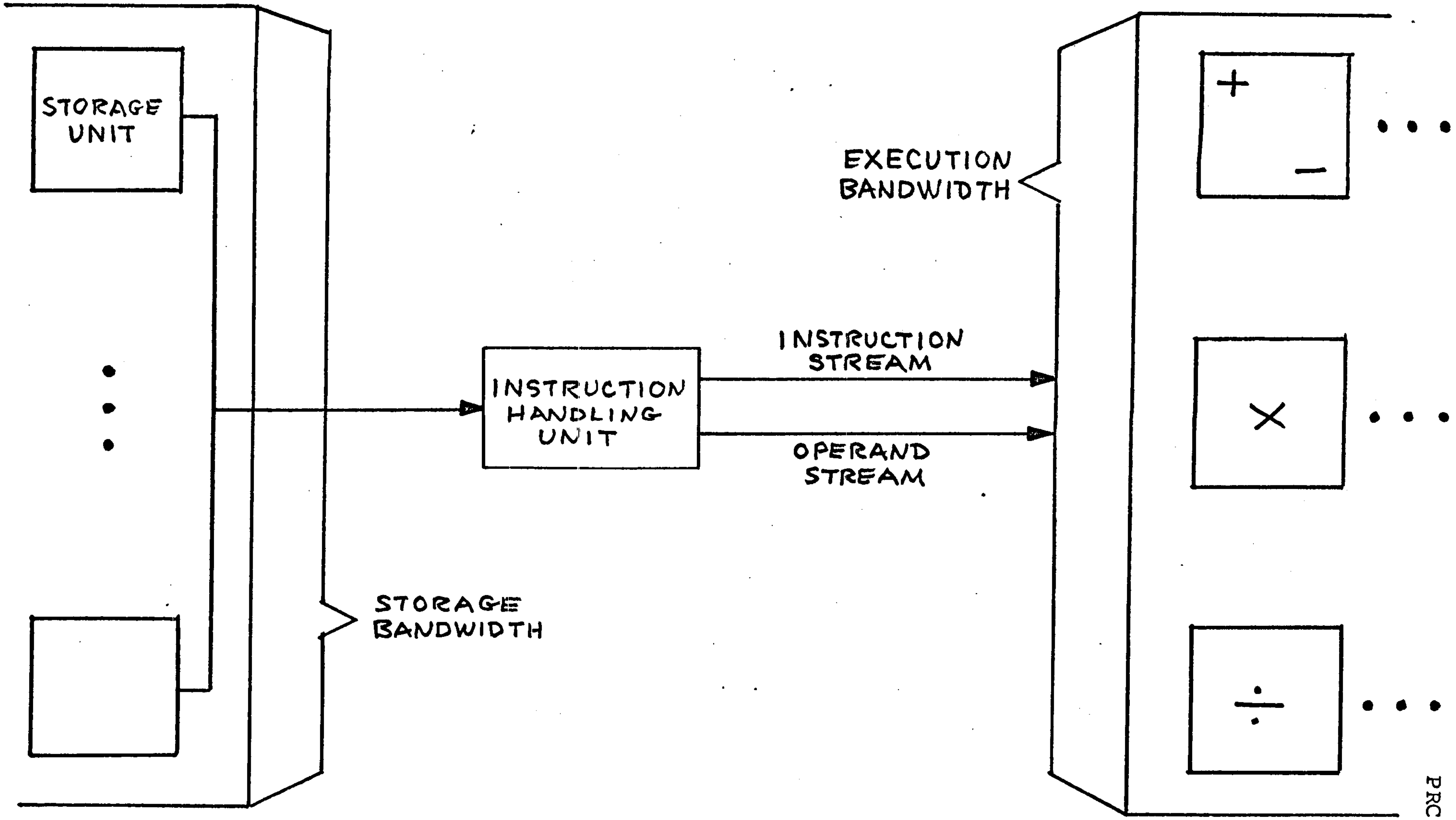

EXHIBIT 1 SISD ORGANIZATION

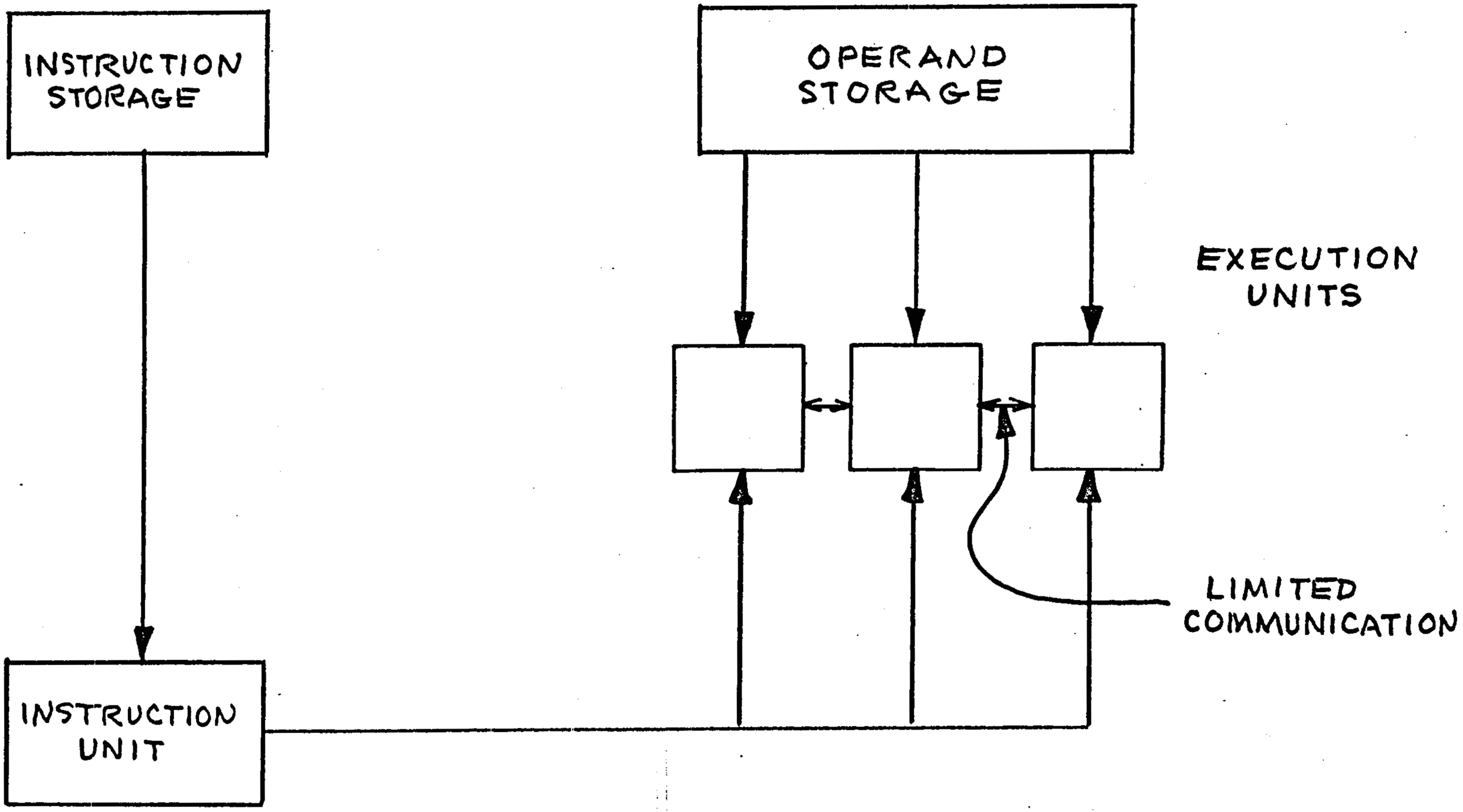

EXHIBIT 2 SIMD ORGANIZATION

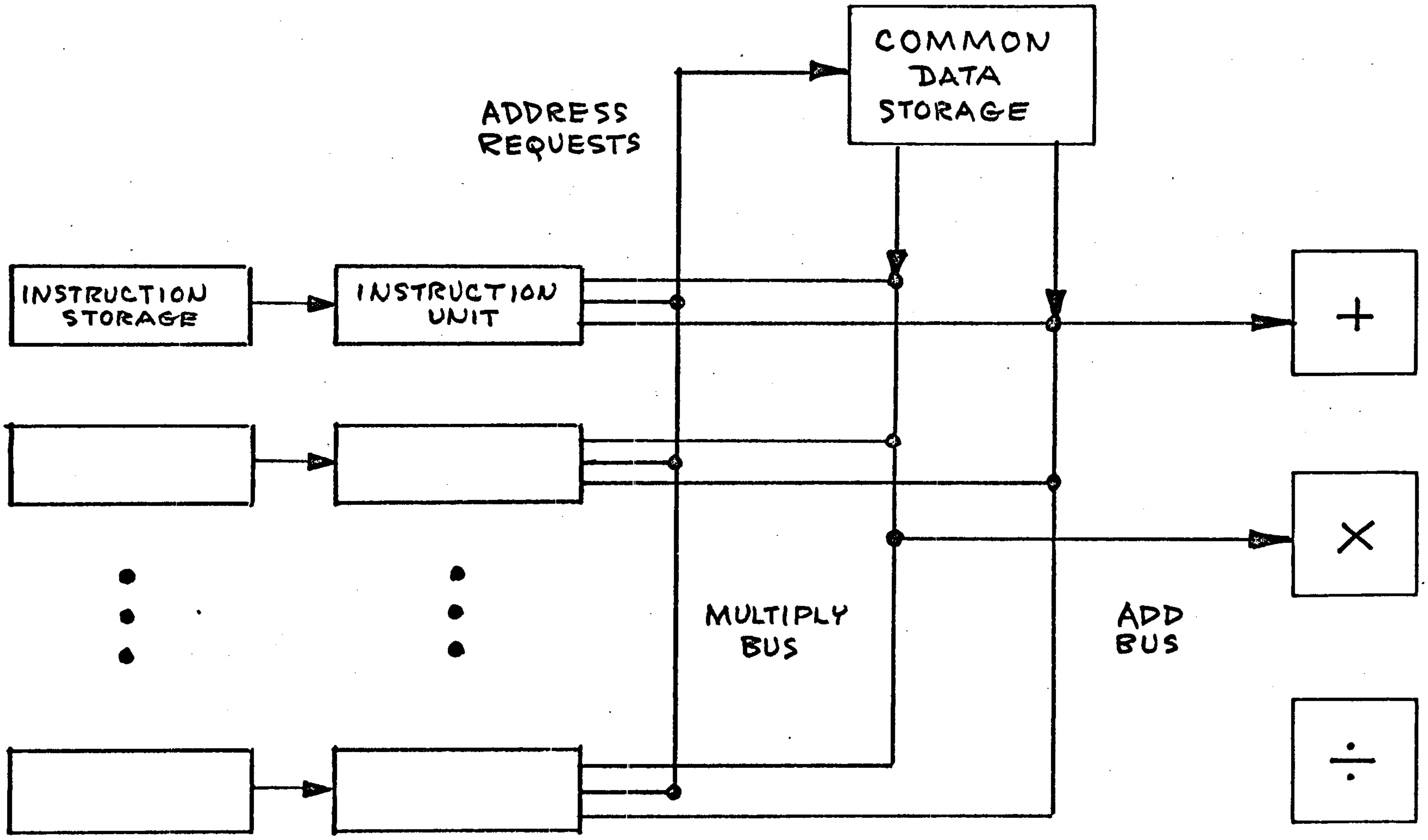

EXHIBIT 3 MISD ORGANIZATION

B. Central Processor Unit (CPU) Characteristics

The CPU features which relate to software performance can conveniently be discussed in terms of four major items:

- Number Systems
- Storage Characteristics
- Timing Characteristics
- Machine Programming Characteristics

1) Number Systems--The representation of numbers is, of course, vital to any discussion of computers and computer applications. Modern digital computer design and programming are fundamentally constrained by the implemented concepts of numerical magnitude and polarity.

It is not within the scope of this document to be overly tutorial about various radix (or base) representations, or the associated conventions for handling positive and negative quantities.

However, one of the most fundamental ways of characterizing a computer is by its number-base and word-size, i.e., the kind and number of digits that can be retrieved from or stored in the computer memory in a fixed time interval. In many cases, the memory cycle time and word size alone constitute data upon which a fair guess of performance can be based. That is because performance is primarily memory access limited (if it is not I/O limited), and therefore depends upon the rapidity of memory-access and upon the value which accrues to memory-access.

The word-size of the computer establishes the various quantization properties of the variables processed by the CPU. Gross quantization is important from our standpoint in several situations:

- Where weight is at a premium, as it will be in the OCDMS computer, gross quantization reduces the required memory capacity of the computer.

- Noise in the communication system places a lower practical limit on analog-to-digital conversions so that input is grossly quantized.
- In the mixed system of analog and digital elements, it becomes pointless to quantize to a degree of precision that is inconsistent with the accuracy of the analog components.
- For a fixed program the bandpass of a digital computer decreases directly with word length. Therefore, in the interest of maintaining adequate bandpass, gross quantization is desirable, i.e., use short word lengths.

The larger the word size of the computer, the larger the address, operation code, instruction, or combination of these data that can be accessed with a single memory reference. Increasing the number of each of these program-related units of information accessed with a single memory-reference has some benefits; however, a substantial amount of register storage (to hold waiting instructions and temporarily unused addresses) and other logic circuitry is required to obtain improved performance in this way. But such benefits are confined to program access: no benefits are obtained in the accessing of addresses and instructions themselves as the operands to be processed. Increasing the size of each of these program-related units of information on the other hand, has a substantial, although conditional, effect on performance. Increasing the address size is of great value if the storage to be addressed may be increased. Increasing the instruction size permits a large variety of instructions, and thus allows more special-purpose features. These benefits may or may not be significant, depending upon the job to be performed.

Just as the word-size depends on the scope of application so too is the choice of number representation. The choice of number system and polarity is for the most part independent of the physical devices being used. In current usage most computers are either binary or binary coded decimal systems. Negative numbers are normally represented by sign-and-magnitude, radix-complement, or diminished radix-complement. Recently introduced, however, is negative-radix arithmetic for which the proponents such as deRegt promise revolutionary changes in digital computer organization and design philosophy (see reference 6). Advantages projected include:

- significant reduction of system-level logic and inter-connections;
- variable word-size operations, modular construction, program control of register length, and program control of computer configuration through negative-radix mechanics will promote easy hardware redundancy and graceful system degradation;
- increased modularity of complex units which will in turn be made economically possible by the negligible cost difference between simple and complex integrated circuits.

2) Storage Systems--The high-speed, random access memory characteristics are always a most significant aspect to any computer selection criteria. The most important selection characteristics of internal memory are organization, capacity and timing. The ferrite core memories available in most computers of today are classified in logical organizations as follows:

- 3D, or coincident-current;
- 2½D, 3-wire-in smaller and faster main memories (10 ^{6}bits);

- 2½D, 2-wire-in the larger, slower bulk core memories, (5 x 10^6 bits and greater);
- 2D, or linear select organization.

Thin film memories which are now becoming available fall into two general groups: the planar film memories and the cylindrical film (plated wire) memories. Both types are usually organized as 2D (linear select) configurations, but 2½D plated wire memories have also been developed. The thin film memories have cycle times ranging for 0.5 microsecond to 100 microseconds and will undoubtedly become much more important for applications in which fast cycle time is a requirement.

The organization of computer memory has a direct effect on system performance and cost. The cost, performance, and reliability of the system are a function of the organization which includes not only the logical structure indicated above (2D, 3D, etc.), but also the size and logical dimensions of the memory modules.

In addition to memory organization, the characteristics which are most likely to effect the performance in OCDMS applications include:

- Communication and control lines to and from storage-special indicators may be needed such as:
  - Address Register CLEAR
  - Information Register CLEAR
  - All Registers CLEAR
  - END-OF-CYCLE Indication
  - Read Access Complete
- General handling of the set and clear for registers, whether to clear all registers automatically and control flip-flops at the end of the cycle, or to set the registers directly to the new state.

- Incrementing the address register from a single control line by making the register into a counter.
- Mode of data transfer--the data and control can be transferred either single-ended or double-ended by transmitting only the "true" of the data, or both the "true" and "complement" of the data.
- Parity generation and checking (considered mandatory for the OCDMS application).
- Memory retention on power down (essential for the OCDMS computer).

3) Timing--Timing characteristics refer to the instruction execution times such as given by Tables 1 and 2, and the several aspects of memory-timing such as cycle time, read access time, and various latency periods. The system concern is to be assured that computer through-put is sufficient for the application. In the OCDMS application, concurrent activity requirements result directly in a situation that computer I/O, data acquisition, and data processing rates may only be known in a statistical sense. That is, to satisfy the functional requirements, a combination of operation conrol program must be allowed to compete timewise for computer and I/O actions. Therefore, the actual rates will be variable and can be described only by the statistical distribution of the I/O demand, and of the associated device service times. CPU performance characteristics or design features that are compatible or facilitate programming in this environment will be of significant advantage.

4) Machine-Programming--The remaining kinds of CPU characteristics which relate to software performance may be listed in a machine-programming class. Items covered include the following:

- Number of instructions
- Addresses/instruction
- Number of index registers
- Indirect addressing
- Floating point hardware
- Interrupt control
- Special features:
  - Micro-programming
  - Hardware queuing
  - Overflow/underflow protection
  - Double precision arithmetic
  - Extended operands
  - Segmented-word operations

These kinds of characteristics are generally explained in extensive detail for any computer under consideration. Thus, no further consideration is given to them here.

C. Input/Output (I/O) Characteristics

The third main category of computer characteristic covered by our OCDMS selection criteria involves I/O communication controls and peripheral devices associated with the OCDMS computer. The communication controls apply to:

- Number of channels
- Devices/channel
- Minimum data rates
- Maximum data rates
- Number of interrupts/levels
- Data representations
- Buffer sizes
- Multiplexing and de-multiplexing
- Asynchronous operations

The consideration of most peripheral devices, of course, is limited to their possible usage in conjunction with the Support System. The Supervisor System by current definition has an interface limited to magnetic tape units, remote operator's console, and to the communication subsystem which is composed of: (see reference 11)

- Computer Interface Unit
- Signal Adapters
- Control and Display Unit

Performance characteristics and design attributes of interest for the peripheral devices include:

- Magnetic tape units
- Punched card units
- Paper tape units
- Line printers
- Disc storage
- Magnetic cards
- Inter-active displays
- Console typewriters
- Control panels
- Displays

V. COMPARATIVE ANALYSIS

The concluding remarks in Section II indicates that S/AA experiments and the application of OCDMS will generally lead to automated procedures operating on sets of variables for which simultaneous independent operations can usually be performed on the individual variables. This suggests that the technical areas involving software will have a common denominator involving calculations and operations that may be expressed by matrix solutions or a mesh of numerical values. The calculations are typified by mathematical techniques used in the solution of linear systems, the calculation of inverses and eigenvalues of matrices, correlation and autocorrelation, and numerical solution of systems of ordinary and partial differential equations. Such calculations are encountered throughout the entire spectrum of applications in space vehicle checkout, communication guidance and control, orbit calculations, character recognition hydrodynamics, heat flow, diffusion, radar, data processing, numerical weather forecasting, and data management.

This being the case, the Litton L-304 Computer has some basic advantages over the IBM 4π-EP in terms of its overall organization. As indicated in Section IV, the Litton L-304 may be classified as a SIMD-type structure. There are essentially 64 universal execution units each with its own access to operand storage. The single instruction stream can act on 64 operands without using the confluence techniques such as employed in the 4π-EP. Communication between units is restricted to a predetermined neighborhood pattern and must also proceed in a universal uniform fashion established by automatic program queuing and triggering mechanism, and the procedure for program level activation.

The IBM 4π-EP is an example of the confluent (concurrent) SISD-type structure which employs techniques to overlap the various sequential decision processes which make up the execution of the instruction. These techniques achieve a significant increase in the memory bandwidth and the execution bandwidth in comparison to the basic SISD organization. However, one particular bottleneck can not be avoided. This bottleneck is the need to decode one instruction in a given unit of time. Thus, no more than one instruction can be retired in the same time quantum, on the average. If the organization is extended by taking 2, 3 or n different instructions in the same decode cycle, and no limitation placed on instruction interdependence, the number of instruction types to be classified would be increased by the combinational amount (M different instructions taken n at a time represents M^n different outcomes), and the decoding mechanism becomes correspondingly more complex. Restrictions must be placed on the occurrence of either specific types of instructions or instruction dependencies. This, in turn, demands restrictive programming practices, and/or constrains the possible extent of the OCDMS application.

Although the L-304 has distinct advantages with regard to the above considerations, nevertheless, it represents certain difficulties too. These difficulties include:

1) Latency in the instruction stream for SISD-type branches is replaced by latency in the data stream caused by operand communication (forwarding problems).
2) SIMD organization is inconsistent with much current software technology and techniques such as compiler algorithms and operating system procedures.
3) The universality of the execution units deprive them of maximum efficiency.

These difficulties chiefly represent areas for which new technology must be developed. As the improved techniques become available, then a real potential exists to allow the definition and development of spaceborne computer system with performance and design features to include:

- Self-repair
- Programmed interconnections
- Graceful degradation
- Increased reliability
- Increased simultaneity of operations
- Variable-configuration
- Reduced power consumption

In the particular instance where current OCDMS study efforts may lead to breadboard activities and developmental facilities, the Litton-304 is a better choice if compatibility with OCDMS operating concepts, and technology aspects of software are the major areas of selection concern. The cost implications and availability of manufacturer-supplied software are considered about the same for both Litton and IBM. This is discussed in greater detail in the Preferred Manufacturer-Supplied Software Memo, reference 13. Table 3 summarizes the comparison of computer features between the Litton L-304 and the IBM System 4 Pl-EP.

IBM SYSTEM 4 P1, MODEL EP	LITTON L-304
1. 32-BIT WORD LENGTH CAPABLE OF HALF-WORD OR MULTIPLE-WORD ACCESS.	1. 32-BIT WORD LENGTH CAPABLE OF HALF-WORD ACCESS.
2. 16 REGISTERS EMPLOYABLE AS ACCUMULATORS OR INDEX REGISTERS.	2. 8 REGISTERS PER LEVEL EMPLOYABLE AS ACCUMULATORS OR INDEX REGISTERS.
3. 75-INSTRUCTION REPERTOIRE (NOT INCLUDING FLOATING POINT).	3. 65-INSTRUCTION REPERTOIRE,
4. UP TO 262 INPUT/OUTPUT CHANNELS.	4. UP TO 64 INPUT/OUTPUT CHANNELS.
5. 15.5-HOUR INTERNAL REAL-TIME CLOCK (CAPABLE OF PROGRAM INTERRUPTION)	5. MANY REAL-TIME CLOCKS (EACH CAPABLE OF PROGRAM INTERRUPTION).
6. DIRECT MEMORY ACCESSING TO OVER 4 MILLION WORDS.	6. PROGRAM-CONTROLLED MEMORY, EXPANDABLE TO 131K WORDS.
7. COMMON MEMORY ACCESS BETWEEN PROCESSORS.	7. COMMON MEMORY ACCESS BETWEEN PROCESSORS.
8. IMPROVED INSTRUCTION REPERTOIRE INCLUDING: A. MOVE B. EXECUTE C. BINARY-TO-DECIMAL OR DECIMAL-TO-BINARY CONVERSION	8. EXPANDED INSTRUCTION REPERTOIRE INCLUDING: SUCH IMPROVEMENTS AS: A. MOVE B. EXECUTE C. TEST AND SET BIT D. GATED COMPARISON E. EXCHANGE F. INPUT/OUTPUT TRANSFER DIRECTLY FROM PROCESS REGISTERS G. THOROUGH LITERAL ADDRESSING CAPABILITY TO MEMORY
	9. AUTOMATIC PROGRAM QUEUING AND TRIGGERING.
	10. TOTAL OF 64 PROGRAM LEVELS, EACH WITH OWN PROCESS REGISTERS AND MEMORY BANK SELECTION (SAVED AUTOMATICALLY WITH EACH LEVEL CHANGE).
	11. INPUT/OUTPUT TRANSMISSION VIA 8-BIT CHARACTERS OR 32-BIT WORDS.

TABLE 3 - COMPARISON OF COMPUTER FEATURES

VI. OTHER STANDARDS OF SELECTION

The activity to develop quantitative measures for selection of the OCDMS computer has been a joint MSFC/PRC effort. In the process of doing this, some 31 airborne computers have been evaluated as listed in Table 4 and discussed by reference 9. Computers that satisfy, or which may easily be modified to meet, the requirements of the OCDMS General Specification are still considered candidate systems.

Additional criteria has been established, but not classified, as mandatory. These are listed in four categories below and ordered according to their relative importance.

1) CPU Characteristics

- Index registers: 8 or more
- Hardware queuing
- Double precision arithmetic
- 32-Bit operands
- Hardware overflow protection
- Floating-point hardware
- External interrupts: 32 or more
- Discrete I/O: 32 each
- Micro-programming.

2) Memory

- Memory parity
- Cycle time: 2 μ sec maximum
- Memory modules: independent
- Word-size: 32-bits
- Half-word operations
- Memory modules: independently addressable
- Memory size: expandable to 65,536-bits (8,192 bytes)

Autonetics D26C
Autonetics D26J
CDC 5360
CDC 5400 & -8
CDC 5500
CDC SKY I
General Electric A 224
General Electric A 605
CPI MOD L90-1
Honeywell - Alert
Honeywell H-387
Hughes HCM 205
Hughes HCM 206
IBM 4π-CP
IBM 4π-EP
IBM LVDC
Litton L-304
Litton L-305
Litton L-306
Litton L-2040
Litton L-3050
Litton L-3036
Northrop NDC-1051
RCA VIC-24A
Sperry Mk. XII
Sperry Mk. XIV
Texas Instrument Model 2501
TRW 4418
UNIVAC 1818
UNIVAC 1824-C
UNIVAC 1830-AM

TABLE 4 - AIRBORNE COMPUTERS STUDIED
FOR THE OCDMS APPLICATION

- Memory module size: 8,192-bits (1024 bytes) per module.

3) Input/Output

- I/O Channels: 3-buffered
- CPU and external devices: asynchronous
- Magnetic tape operations: serial data transfer by byte.
- PCM Telemetry operations: serial data transfer by byte.
- Byte rates:

  Magnetic tape - 12 k bytes/sec
  PCM Telemetry - 6 k bytes/sec
  Digital Command System - 50 words/sec
  Timers - 1 k words/sec
  Control Dispaly Unit - 15 k words/sec
  Signal Adapters - 15 k words/sec
- Data transfers: parallel words except as noted above.

4) Physical Features

- Weight: minimum (less than 100 pounds)
- Size: minimum (less than 2 cubic feet)
- Power: minimum (less than 300 watts)
- Reliability: greater than 0.980
- Space qualified: with minimum development
- Cooling: compatible with vehicle
- Maintainability: reasonable preventative maintenance
- Test plugs: without modification
- Availability: within 6 months
- Spare parts: as required.