X-562-69-65
PREPRINT
NASA TM X-63492

ONBOARD PROCESSOR: PHILOSOPHY, DESIGN, AND OPERATION

FEBRUARY 1969

GSFC
GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

X-562-69-65

ONBOARD PROCESSOR: PHILOSOPHY, DESIGN, AND OPERATION

Ronald M. Muller
Space Electronics Branch
Information Processing Division

(Presented December 1968 in Paris, France,
at the CNES International Conference on
Aerospace Computers in Rockets and Spacecraft)

February 1969

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

ONBOARD PROCESSOR: PHILOSOPHY, DESIGN, AND OPERATION

Ronald M. Muller
Space Electronics Branch
Information Processing Division

ABSTRACT

The onboard processor (OBP) is a general-purpose stored-program spacecraft computer designed to play a central role in a spacecraft's operation, control, and data-handling. Standardized electronics allow it to take on the many configurations needed for a variety of missions without the expense of developing specialized electronics. Its programs use an English-like assembly language to facilitate program writing. Operation of the OBP in orbit presents new problems to the ground system; the design of the flight electronics and programs attempts to minimize this impact.

CONTENTS

ONBOARD PROCESSOR: PHILOSOPHY, DESIGN, AND OPERATION

INTRODUCTION

The onboard processor is a computer designed to fill a need for spacecraft electronics that can reliably perform complex data manipulations and at the same time be general-purpose.

In the spring of 1966, development began on a device with high computational capability and a family of standardized input/output (I/O) devices. This type of device permits establishment of onboard processing systems to fit many different missions simply by writing the appropriate programs and selecting the I/O needed. The system can be further tailored to a given mission by varying the size of the random-access memory used. This approach allows development and thorough testing of hardware long before specific requirements are known. In fact, the user can change the processing after launch by transmitting a new program to the computer; this has the obvious economic advantage of not having to redesign and requalify hardware for each different mission. It also allows the development of the spacecraft to proceed by using only very general requirements to pick the size of the memory and the characteristics of the I/O.

A laboratory computer was built and has operated in excess of 1000 hours. The first flight of this machine is now scheduled for the fall of 1970. One result expected from the application of this concept is an actual decrease of the total complexity (parts count) on a spacecraft. For instance, an experimenter need only process his data to the point of obtaining voltage levels suitable for the I/O interface; all other processing can be done by software. The processor occupies less than 1000 cubic inches for one central processing unit (CPU), input/output, and 16384 words of memory; it weighs less than 40 pounds, and consumes a peak power of 36 watts. Average power consumption is proportional to memory usage: if the ORP is busy only half the time, average power is 22 watts. Standby power consumption is 8 watts.

DESIGN PHILOSOPHY

Major design requirements, in decreasing order of importance, were:

- High reliability
- Ease of programming and reprogramming

- Low power consumption
- Low weight
- Small size

These requirements were satisfied, even though certain compromises were necessary because of conflicting requirements. The resulting product has the following characteristics:

- Eighteen-bit word
- Two's complement number representation
- 2.5-microsecond memory-cycle time
- 6.25-microsecond add time
- 42.5-microsecond average multiply time
- Fifty instructions
- Up to sixteen priority interrupts

The onboard computer has as its base a dual bussing system, more reliable than a single-bus technique, allowing a variable memory size up to a maximum of 65K words in 4096-word increments. It also facilitates incorporation of standby redundant CPU's and I/O devices as needed for enhanced reliability during a long mission. The processor includes a dynamic memory-protection feature to ensure that writing occurs only in desired memory areas; it also contains program control over interrupt priority. The program permits specifying the location of the binary point, and scaling is automatic, which provides some of the features of floating-point without the accompanying hardware and memory penalties.

The machine language chosen for this machine is a restricted English. Programs written in this language are self-documenting and enable understanding of each program by all parties familiar with English. A user can specify a set of mnemonics or other words for each English word. These words could also be taken from another language; this is illustrated further on in this report by a program written in French. Other natural languages may be used also. An additional benefit of this is that a French program can be machine-translated into English (and vice versa) so that, on some future international effort, language would be no barrier to understanding of each other's programs.

OPERATION

Hardware

Figure 1, a simplified block diagram of the onboard processor (OBP), shows the basic dual bus interconnection used to achieve both flexibility and high reliability. The two sets of buses are electrically independent, so that no single failure can cause the loss of more than half the memory. Only a few kinds of failures can cause the permanent loss of this much memory: for instance, a bus wire shorted to the structure. If the failure is internal to any of the boxes, the user can command that box OFF and turn a spare ON. The OFF command disconnects both power and ground, so that all nonoperating boxes are electrically "floating." This approach permits operation in spite of failures and allows a "graceful" rather than a catastrophic loss of capability. These features — the dual bus, the standby redundancy, and the ability to reprogram by command — all enhance the reliability of the system.

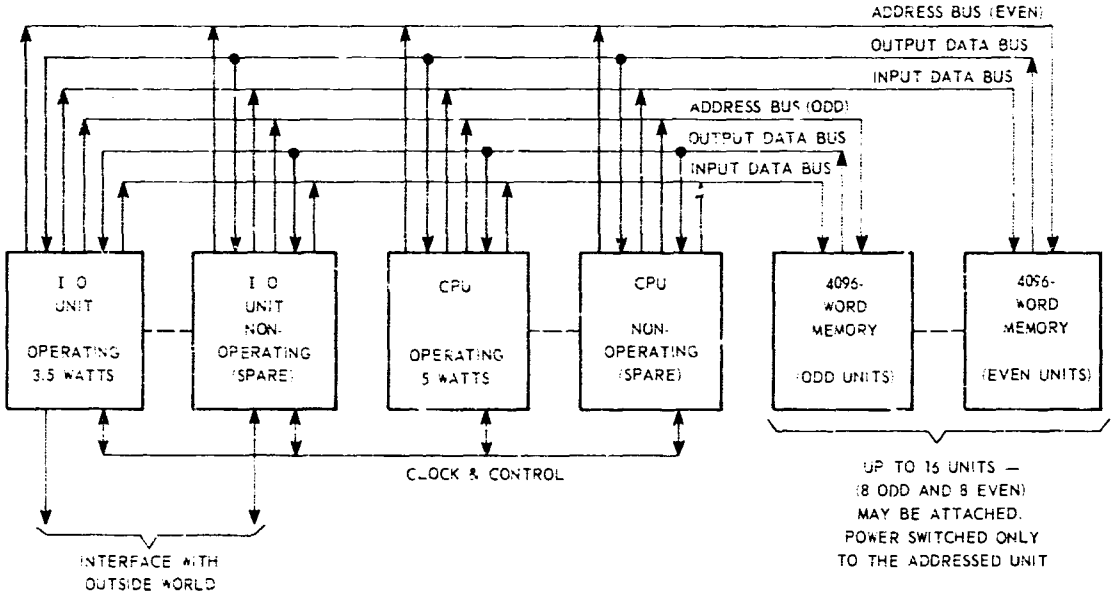

Figure 1. Block Diagram, Onboard Processor

The bussing interfaces will accommodate up to three CPU's (two as spares), up to three I/O channels, and up to sixteen 4096-word memory units. The I/O units and memory units may be turned on in any combination.

Characteristics of each block are:

	Size (in^3)	Weight (lb)	Power (w)
CPU	200	6	5
I/O	160	5	3
Memory	135	7	28

Memory power is a function of memory usage. The power figure shown represents 80-percent use of available memory cycles at 2.5 microseconds per cycle. Standby memory power is less than 100 milliwatts power unit of 4096 words. Only one memory can be used on a given cycle; therefore, memory power consumption is not a function of the number of memory units flown.

Central Processing Unit (CPU)

After a study of desirable computer characteristics, the CPU for the onboard processor was designed as a parallel two's complement machine using low-power diode transistor-logic circuits. Eighteen-bit data and instruction words provide a convenient instruction format and serve to handle most quantities of data. The available instructions total fifty; thirty of these access the memory, and are indexable. The addressable registers include an 18-bit accumulator; an 18-bit extended accumulator for multiply, divide, and double-length operation; an 18-bit subscript (or index) register; a 4-bit page (or block address) register to provide maximum access to as much as 65K bits of memory; and a 6-bit scale register for noting the locations of the binary point in the fixed-point data words.

Typical instruction/execution times are:

Add	6.25 microseconds
Multiply (av.)	45 microseconds
Divide (av.)	90 microseconds
Shift	6.25 + 1.25 (number of bits shifted) microseconds
Conditional transfer	5 microseconds
AND/OR	6.25 microseconds

One special feature of the CPU is the scale register used by the programmer to control the location of the binary point associated with numerical quantities by automatically shifting before the divide and after the multiply operations. The scale register allows him to locate the binary point wherever it will best suit his data. Standard fractional arithmetic is available by setting the scale register to zero; standard integral arithmetic is available by making the scale register equal to 17. The scale-register concept appeared to be the best compromise between fixed- and floating-point operations; the relocatable binary point is not as flexible as floating-point, but it does require less hardware to implement while at the same time it retains some of the desirable features of floating-point. In brief, the value of the characteristic is carried in this register, therefore memory words are shorter for the same degree of accuracy.

Another special feature of the CPU is its ability to handle conditional program transfers. A conditional transfer occurs when a 1-bit register known as the D (decision) register is set (this may be done by any of the several test instructions included in the CPU). As it may be desirable to combine several tests before making a transfer, the OR/AND (O/A) flip-flop controls the way in which each condition will affect D, and thus the conditional transfer. This flip-flop, which is under program control, can be set so that the upcoming test results will be either ORed or ANDed with the previous value in the D register. The number of test instructions which can be combined in this manner is unlimited.

To prevent one user from destroying another user's program or data, the storage-limit register reserves a block of memory in which the operating program may write. These blocks are in increments of 128 words. To provide this feature, the 18-bit register is divided into two 9-bit fields: the first specifies the most significant 9 bits of the first address that may be written, the second defines the most significant 9 bits of the last address that may be written. Only the Executive program can alter the contents of the limit register, and these instructions are forbidden to all user programs.

The processor includes a flexible interrupt system which can include up to sixteen levels of priority. Such priority systems present a potentially serious problem, in that a high-priority interrupt could fail and continue to demand CPU time unnecessarily. If this should occur, the contents of a mask register can be changed by ground command in order to honor or to lock out a given interrupt level. This mask register is dynamically controlled by the Executive program which uses information supplied by ground command.

Memory

The random-access memory consists of one or more memory units of 4096 words of 18 bits each. This memory uses temperature-compensated 20-mil

cores with a standard 3D addressing technique. Access time is 850 nanoseconds, and full cycle time is 2.5 microseconds. An unusual feature of the memory is cycle-by-cycle power switching that makes memory power consumption proportional to the frequency of memory access, and permits adding 4K memory units without paying a power penalty because per-unit standby power consumption remains less than 0.1 watt.

Input/Output (I/O) Unit

Successful use of the OBP depends upon its ability to be reconfigured with relative ease from application to application. This ability is achieved by keeping the OBP unchanged and configuring the I/O unit for each mission. The standard I/O has been designed so that a major part of it will remain unchanged for each use. For most applications, special I/O hardware will be required to supplement the standard I/O. A typical I/O will contain about 75 percent as much circuitry as the CPU.

Standard I/O — The standard I/O unit considered adequate for most spacecraft applications contains six input digital-data channels, three output data channels, two control channels (for cycle-steal operation), 16 levels of interrupts (one interrupt channel per level), and appropriate input/output control. Included as a part of the I/O are the system clock (800 kHz) and the memory-bus controller.

The OBP instruction set includes an input/output instruction which allows a data-output/data-input or function input, or causes activation of one of the two control channels. Functionally, the six input and three output data channels can throughput data to or from memory by means of the CPU accumulator, under program control. Also, four of these same channels can obtain direct access to memory addressing and read/write control. This direct I/O operation is called "cycle-steal" because the I/O steals memory cycles from the CPU to produce an interleaving of memory activity without actually interrupting CPU program execution. The system, which includes a program-controlled priority-interrupt feature, can control transfers of data at rates up to 25,000 words per second using the CPU and 400,000 words per second using cycle-steal.

The two control channels provide addresses and read/write control to the memory bus for any data channel assigned to operate in the cycle-steal mode. This mode can be applied simultaneously to any two of the four applicable data channels. Operation of both channels can be initiated, terminated, or reassigned by program execution or by the external command system.

Special I/O — Every application will require some special I/O hardware to supplement the usual standard I/O equipment needed; the amount and type required

for any application will depend upon the specific tasks to be performed on board the spacecraft. Examples of special I/O hardware are multiplexers, analog-to-digital and digital-to-analog converters, buffer registers, serial-to-parallel and parallel-to-serial converters, level converters, impedance-matching circuits, etc. These devices, available as standard modules, are considered special-purpose because they are tailored for a given application.

OPERATION

System Programming

Ease of programming and reprogramming the computer was a major design requirement second only to reliability. Much of the circuitry already described was included as the direct result of a programmer's suggestion.

The flight computer program actually consists of an overall control program (called the Executive) and a number of user programs. The Executive determines which user program will run on the machine at any given time; it also handles interrupts, and has exclusive control of the storage-limit register. When the machine is idle, the Executive may run a machine self-test program or a diagnostic program.

Language

The syntactic structure and rules of grammar for the OBP assembly's language are very closely allied with those of English. A program in the OBP language consists of a group of sentences composed to accomplish a given task. The nouns or noun phrases of the language represent the data being operated upon or computed; these nouns are selectable by the user to be descriptive of the data quantity. The verbs of the language are English verbs denoting computer instructions.

The fact that the OBP language resembles English provides the assembly-language program with the desirable feature of self-documentation: in other words, the program is an English-language explanation of itself, and this makes program maintenance and modification efficient. Because a given program may require modification at any time from the time it is written until long after launch, complete and up-to-date documentation is essential. The English-like language also facilitates understanding of the program by persons other than professional programmers; thus, the scientist engages in a spacecraft experiment can determine if his programmer has faithfully carried out his duties.

Procedures

The user describes his desired processing in restricted grammatical English. This description is then punched on cards and processed by a portion of the support program called the assembler. The programming language contains a number of features incorporated with the aim of making it similar to English:

- The language is "free form," relieving the user from the requirement to assign specific fields of the language to specific card columns. Text may be freely continued from one card to the next. Also, the user may create paragraphs, indent text for emphasis, etc., just as he would in English.

- Free use of standard English punctuation is permitted. Periods, commas, exclamation points, etc., are allowed. The double use of the symbol (-) as a hyphen or as a mean of continuing a word across a line has been preserved.

- Additional expository text not essential to the main thought of a sentence is written as a parenthetical expression; these comments are not processed by the assembler.

As an aid to a programmer who prefers to use words or letter combinations of his own choosing, the assembler contains a translation feature which converts these to their English equivalents and then assembles normally. There is no restriction on these letter combinations, which could be words from another natural language.

To illustrate the power of this feature, a simple program to solve the equation

$$\text{NORM} = \sqrt{X^2 + Y^2}$$

is written below, first in English and then in French. Either program could be processed by the assembler.

English

> Let X times X yield X squared. Let Y times Y plus X squared, transformed by square root, yield norm. If it is less than 1, then go to compute inside the unit circle; otherwise, . . .

French

> Soitque X multiplier par X rendre X au carre. Soitque Y multiplier par Y plus X aucarre transforme par racine carre rendre norme. Si cela est inferieur a 1, alors allez a calculer dans la surface du cercle; sinon, ...

Other natural languages having sentence structure similar to English could also be used. A natural byproduct of this is machine translation of programs; thus, on some future international effort, language need not be a barrier to understanding each other's program.

Subroutines

One special class of programs consists of subroutines, which are programs that can be shared by all user programs. Circuits are designed so that one user program can be interrupted at any point in his program including a subroutine; the interrupting program may then use the same subroutine. When the machine is given back to the interrupted program, it will pick up just where it left off.

Other Support Programs

Other support programs in addition to the assembler are now available. A relocatable loader assigns a specific memory location to each program instruction; this is necessary because all programs are written with relative (not absolute) addresses assigned. Also available is a programmed simulation of the spacecraft computer; these programs are operational on the Univac 1108, the GE 625, and the SDS 920 computers.

OPERATION IN ORBIT

A major concern during the design phase has been the operation of this computer in orbit. The electrical interface is deceptively simple: one radio link from ground to spacecraft (command), and one radio link from spacecraft to ground (telemetry). This simplicity tends to mask the complexity of the numerous steps required to convert human thoughts into commands, or to convert the received telemetry message into a form understandable by the human user. This complete data system must be carefully considered, and the onboard circuitry and programs designed, to complement the ground capabilities. This is true of any spacecraft telemetry and command system, but especially of an onboard computer.

Reprogrammability

Because the OBP can be reprogrammed while in orbit, it must include some means of checking out new programs and verifying their accuracy. For the reprogramming feature to be of most value, the rewriting, checking, and transmission to the orbiting computer memory of a new program should take only a short time (a few days to a week).

Command

The computer may be allowed to generate commands, but not without some risk: for instance, it might send a command that would damage the spacecraft. Such commands are few in number and should be preventable; for instance, it could be mandatory that a pair of commands be sent in sequence before the execution of a command would be permitted. However, the benefits of permitting onboard command control usually outweigh the disadvantages. Such control allows long-term spacecraft operation without ground-station attendance except for extraction of accumulated data; it can also literally save a mission by recognizing a dangerous condition on board and commanding the spacecraft into a safe condition. If the OBP is allowed to generate commands on its own, these actions must be relayed to the ground at the earliest opportunity.

Data Compression

One particular reason for using an OBP is to compress data into more compact form so that they will fit into the available telemetry bandwidth. This compression significantly affects the ground data system because the process usually removes most of the redundancy. This consequently makes it more difficult to determine whether the onboard processor and the ground system are performing properly. Inclusion of a few samples of raw data in the compressed data should enable users to check the onboard processing; also, the communications link should include some sort of error-detecting coding, and the transmitter should be sufficiently powerful to ensure data that are virtually error-free. The benefits of compressing by 100 to 1 or 20 db should compensate for putting back a few db to ensure complete intelligibility in the ground processing.

BIBLIOGRAPHY

1. Stabler, E. P., and C. J. Creveling. "Spacecraft Computers for Scientific Information Systems." Proceedings of IEEE, 54:12. December 1966

2. Styles, F. J., T. D. Taylor, H. M. Tharpe, and C. E. Trevathan. A General Purpose On-Board Processor for Scientific Spacecraft. NASA/GSFC X-562-67-202. July 1967

3. Tharpe, H. M., and D. K. Sloper. "A Low Power Computer to Improve Overall Spacecraft Performance." Speech to 19th Congress, International Astronautical Federation. New York, October 13-19, 1968

4 Tharpe, H. M., C. E. Trevathan, J. Novello, and R. Hartenstein. A General Purpose On-Board Processor for Spacecraft Application. NASA/GSFC X-562-68-387. October 1968

5. Merwarth, A., and T. D. Taylor. Software for the Space Electronics Branch On-Board Processor, NASA/GSFC X-562-68-388. November 1968