Paul Schneck
Guest Editor

DEVELOPMENT AND APPLICATION OF NASA's FIRST STANDARD SPACECRAFT COMPUTER

To provide the autonomy needed by low, earth-orbiting satellites, NASA's first standard on-board processor requires changing only interfacing hardware from mission to mission.

CHARLES E. TREVATHAN, THOMAS D. TAYLOR, RAYMOND G. HARTENSTEIN, ANN C. MERWARTH, and WILLIAM N. STEWART

As a means of providing the autonomy needed for many of NASA's low, earth-orbiting satellites. NASA's first Standard Spacecraft Computer (NSSC-1) has evolved from the on-board processor that was flown as an experiment on the Orbiting Astronomical Observatory C (OAO-C) in 1972. As the computer was applied to other missions, implementation improvements were made. Even before the launch of OAO-C, a program was initiated to improve reliability and reduce size and weight. The result of the program was that the advanced on-board processor (AOP) was flown on the Landsat B and C missions where it was used to augment the stored command capability and perform on-board telemetry limit checking. On the International Ultraviolet Explorer (IUE), the AOP performed attitude-control computations; on OSS-1, it provided the stored commands necessary for efficient instrument control. In 1974, the computer became a NASA standard spacecraft component. The NSSC-1 has flown on two missions: the Solar Maximum Mission and Landsat D. Its functions include stored command processing, attitude control, limit checking and corrective action, and backup telemetry. The NSSC-1 is scheduled to fly on four missions that will be launched in the 1980s. One of the major benefits of the central on-board computer is that significantly different mission requirements can be met through changes to only flight software.

BACKGROUND

From the National Aeronautics and Space Administration's (NASA's) infancy in the late 1950s, unmanned low earth-orbit spaceflight missions have rapidly increased in sophistication to meet the needs of an eager scientific community. As early as the mid 1960s, the growth in science objectives required that space observatories accommodate elaborate instruments and include stored command processors to provide efficient and complex on-orbit operations. This level of autonomy proved so successful that mission and operations managers desired almost immediately to use on-board computers as a means of improving observatory performance even more. Increases in spacecraft autonomy were motivated by the need to

- extend stored command capability from 2 or 3 hours to up to 24 hours,
- automatically modify on-board observations based on detected phenomena,
- provide monitoring of instruments and equipment when not in view of the ground and either generate summary messages for transmission to ground during station contact or command the observatory to a safe mode in the event of anomalous conditions, and
- provide the flexibility in orbit to work around potential failures of mission-critical functions.

In order to meet these needs, the Goddard Space Flight Center (GSFC) developed a medium-scale, general-purpose computer that was flown as an "experi-

---

This article was written in 1983. Since then, two prospective missions mentioned in the article have been accomplished. Landsat-D prime was launched, and the Solar Maximum Observatory was successfully repaired.

ment" on the Orbiting Astronomical Observatory C (OAO-C) in 1972. This first computer was referred to as the on-board processor (OBP). As the computer was applied to other missions, implementation improvements were made and its functional assignments on user observatories were expanded. The first notable example of this was the processor's additional performance of attitude-control computations on the International Ultraviolet Explorer (IUE). A later example was a more general-purpose application on the Multimission Modular Spacecraft (MMS). In this case, the computer provided the autonomy and flexibility necessary for the MMS to support a variety of solar, stellar, and earth-referenced missions without requiring redesign of spacecraft hardware. With a history of flight experience and multimission application, the machine, now known as the NSSC-1, was selected as NASA's first Standard Spacecraft Computer in 1974.

During early definition of the OBP, several hardware and software requirements were defined to provide for flexibility through low-power, radiation-resistant hardware, an expandable, nonvolatile memory, and an "easy to use" support software system. As other requirements evolved, the method of interfacing the computer with the user observatory proved to be the most important factor in achieving the desired flexibility. By interconnecting the computer with the spacecraft command and telemetry system as shown in Table I, an extension of the Operations Control Center was essentially placed on board.

Required command reliability was ensured by the use of self-test software that had to be successfully executed before distribution of commands. All computer-user missions launched to date have made use of this relatively simple interface. The IUE and MMS applications required an additional input and output to allow the computer to directly address any telemetry channel for acquisition of data at a higher rate needed for attitude control and, if desired, serve to control the telemetry format.

Acronyms and Special Terminology

AOP: Advanced on-board processor
ATCP: Absolute time command processor
DMA: Direct-memory access
DTL: Diode-transistor logic
IUE: International Ultraviolet Explorer
MMS: Multimission Modular Spacecraft
MSI: Medium-scale integrated (chips)
NSSC-1: NASA's first Standard Spacecraft Computer
OAO-C: Orbiting Astronomical Observatory C
OBP: On-board processor
OSS-1: First Office of Space Science
RTCP: Relative time command processor
SCP: Stored command processor
SDVF: Software Development and Validation Facility
SEX: System exerciser
SMM: Solar Maximum Mission
ST: Space Telescope
STS-3: Third Shuttle Flight
TTL: Transistor-transistor logic

TABLE I. Generalized On-Board Computer Interfaces

Input/Output Functions	Description/Use
Input from command receiver	Program code, tables, and delayed command loads
Input from telemetry equipment	All engineering and science data for some missions
Output to command distribution equipment	Commands for execution by observatory subsystems and instruments
Output to telemetry multiplexer	Computer-generated messages to be transmitted with observatory format
Output to transmitter	Memory content as a dedicated telemetry format that allows readout of memory independent of program integrity

COMPUTER HARDWARE DESIGN

The computer hardware design was based on rather general requirements, although they were derived with the expectation that OAO-C would be the first flight mission. The task was to develop a machine suitable for use as a centralized spacecraft computer with only the interfacing hardware requiring change from mission to mission.

To meet the objective of multimission capability, the OBP was required to

- possess a comprehensive instruction set;
- be reprogrammable in orbit;
- be composed of low-power, radiation-resistant circuits;
- be capable of memory expansion without paying a power penalty;
- have buffered input and output channels and a priority interrupt structure;
- be capable of interconnecting redundant components as a mission option.

Early in the development phase, Westinghouse Electric Corporation was contracted to assist an in-house GSFC engineering group with definition and design of the system. As the design progressed to the breadboard stage and the decision was made to fly the computer on OAO, Westinghouse was assigned the job of developing the central processor unit (CPU). The input/output (I/O) unit, which in this case had to be tailored to fit into an existing spacecraft, was designed and implemented by GSFC engineers who were familiar with the OAO design and had direct contact with Project system engineers and their contractor, Grumman Aerospace Engineering Corporation. Two important ground rules were that interconnecting the computer would not require modification of existing spacecraft boxes and that the OBP could in no way degrade the reliability or original operations concepts if it failed or was turned off for any reason. The command output logic, for instance, had to emulate a command receiver output and be designed to automatically trip off if a ground command signal was received. Interestingly, this automatic trip-off proved to be a nuisance later on in orbit when frequent false command carrier signals were received over certain parts of the world.

OBP Technology Selection

GSFC and Westinghouse jointly decided to use diode-transistor logic (DTL) circuits manufactured by Fairchild Semiconductor Corporation for both the CPU and I/O modules. These circuits were selected on the basis of being the lowest power bipolar logic devices available on the GSFC Preferred Parts List. Care was taken in screening the circuitry as well as the packaging methods since size and weight restrictions precluded any redundancy in the hardware. The hardware for OAO-C included four memory modules of 4K words each so that some redundancy in this area was possible. Memory development had been progressing as a parallel effort to the CPU and I/O with both plated wire and magnetic core technologies being considered. A low-power core design was selected for the OAO-C mission.

Design

The CPU instruction repertoire was selected to be a simple fixed-point set with full arithmetic, logic, and program transfer capability. A memory protection feature was implemented to preclude undesirable write cycles in areas of memory containing instructions and constants and provide for noninterference of application programs. Upper and lower storage limits were controlled by a storage limit register that could be loaded only by processing an interrupt. Other OBP features were two I/O channels with four device codes each, 16 commandable priority interrupts, and dual-ported memory buses that provided for full memory redundancy in later systems. Memory bus contention logic was included to arbitrate between the CPU and the two direct-memory access (DMA) channels. This logic was designed so that the CPU would be guaranteed a memory access cycle between each two DMA cycles to avoid lockout that could otherwise be caused by faulty I/O channel hardware. Initiation of an interrupt was permitted, when present, between execution of instructions, and the eight memory cycles required for interrupt, save, and load operations were guaranteed to be contiguous once initiated.

An 18-bit word length was chosen to provide a reasonable address field (12 bits), 31 major operation codes (5 bits), and 1 bit to select an index register to be used to modify operand addresses. Of the 55 instructions implemented, 24 were minor operation codes not requiring memory access and were specified by using the low order bits of the memory address field.

OBP Flight Hardware

Following the fabrication and checkout of a breadboard OBP, which was used to certify the design and later to support software development, implementation of flight quality hardware was started.

The packaging technique chosen for housing the 1700 integrated circuits necessary for implementing the CPU and I/O modules was a vertical stack of 6 × 6-inch cards with up to 125 flat packs each. The flat-pack leads were stitch-welded to pins on one side of the boards, and the interconnect wiring was welded to the same nailhead pins on the opposite side of the board.

The first flight memory modules were originally planned to be plated wire units, primarily because of low power consumption and attractive projections of low recurring cost. Because a number of development problems caused a delay in this plan, a more mature core technology was used. Electronic Memories and Magnetics (EM&M) Corporation was contracted to produce a low-power version of their SEMS-5 product line. By power switching memory modules of 4096 words by 18 bits each, EM&M developed a product line called SEMS-5L. Since the power switching was done on a cycle-by-cycle basis, several memories could be used on the same bus to increase capacity and still dissipate only the maximum continuous power of a single memory plus a small standby level for the remaining inactive memories.

Advanced On-Board Processor Hardware

Before launch of OAO-C, a program was initiated to improve reliability and reduce size and weight by converting the 1,700 small-scale integrated circuits of the OBP into medium-scale integrated (MSI) chips. The technology selected was a low-power, transistor-transistor logic (TTL), customized metallization multigate array with approximately 130 gates per chip. These devices had been developed for the Jet Propulsion Laboratory (JPL) by Harris Corporation for implementing the Self Test and Repair computer planned at the time for use on the Grand Tour Mission. The circuits were rad hard and proved very reliable. A life test was run by JPL, which lasted for 10,000 hours at temperatures between 125°C and 150°C, and the only failures were in the ovens and test equipment. Experience in orbit has been as good as the life test with no failures after millions of device-hours on four flight missions.

A design goal for the advanced on-board processor (AOP) was to package the CPU logic and most of the I/O logic (general-purpose portion) on a 5 × 7-inch card. Each MSI device required approximately a square inch of board space, which meant that even if both sides were used, the number of chips was limited to 70. Westinghouse engineers met the goal by partitioning the logic into 69 chips, using 27 different chip types. The design included an increase of I/O channels from 4 to 16 by using memory instead of hardware for channel control information. The same MSI gate arrays, with 6 additional chip types required, were used to implement the circuits needed to interface the AOP with the user spacecraft. AOP size, weight, and power were further reduced by replacing the core memory modules with plated wire units developed by Motorola, Inc.

NSSC-1 Hardware

In 1974, the computer was selected as a NASA standard spaceflight component. In compliance with the policy requiring the establishment of industrial sources for all standard equipment, IBM Corporation was awarded a contract to produce the NASA Standard Spacecraft Computer (NSSC-1). This latest version of the computer uses 8K word power-strobbed core memory modules previously developed by IBM and MSI circuits produced by TRW. Before the IBM contract award, Harris Corporation had terminated production of the gate array circuits used in the AOP, and TRW had been contracted to produce functionally equivalent MSI chips. The TRW design fortunately provided a flexible gate interconnect capability that permitted a one-to-one transfer of the MSI layouts including identical flat-pack pinouts. Although the TRW circuits consumed slightly more power than the Harris devices, they were 50 percent faster, operated over a wider temperature range, and, to date, have proved to be just as reliable. With use of the faster TRW circuits, the computer's computational rate was increased by 25 percent. The NSSC-1 has flown on two missions to date and is planned for at least four others. All NSSC-1 applications are fully redundant with two processor modules and six to eight core memory modules (48K to 64K words).

SOFTWARE DEVELOPMENT

Ground Support Software

Definition and implementation of the OBP system included the development of the support software necessary for making the computer "easy to use" both before and after launch. Elements of this software were an assembler/loader/simulator and a number of other support programs needed for ground-based computers that controlled observatory integration and test and in-orbit operations. As a task under the computer system development contract, Westinghouse designed the assembler, loader, and simulator for the OBP to be hosted on the XDS 920 computer. The GSFC engineering group developed the remaining support software elements that were somewhat mission unique.

The assembler was to be free form, using verbs for instruction mnemonics so that the sentence "LET X PLUS Y YIELD Z" was assembled into three instructions: Load the accumulator with X, add Y to the accumulator, and store the contents of the accumulator in Z. A paper on the OBP was given in Paris in which French verbs were used, and a program was presented in both English and French. Several factors led to dropping the English-like assembler, and a traditional assembler was developed for the OBP prior to its use on OAO-C.

The loader "linked" undefined and externally defined variables and assigned memory space to program segments and to data segments. To take advantage of a register that assigned areas of memory into which data could be written, the loader had two location counters for use in assigning memory space. Variable data used one counter and were assigned space in memory where writes are permitted, whereas code and literals would use the other location counter and were assigned space in a functional read-only memory. (The physical memory was read/write memory. A hardware register was used by logic to prevent writing in protected areas.)

The simulator would execute OBP programs and could provide an instruction-by-instruction trace of register contents, a running execution time indicator, and a formatted dump of memory on request. Although the simulator was useful for module debug, its slow execution time (approximately 1000:1 slowdown) limited its use for large program checkout.

Interface equipment was developed to handle the transfer of control and data signals between the XDS 920 and a breadboard model of the OBP. A system exerciser (SEX) program was written for the XDS 920 that could load OBP memory from an image created by the loader, dump and display OBP memory, start OBP program execution, and patch OBP memory from operator commands. A program was written for the OBP to transfer the contents of all registers to the 920 after executing each instruction so that an operator could trace the execution of selected program segments while in this mode. The 920/breadboard system was useful in OBP executive program checkout since the breadboard was electrically identical to a flight OBP and the checkout system could be run at a real-time rate to uncover timing problems. The SEX program was installed on later models of host machines and was used to check out flight computers developed for the OAO, IUE, and Landsat Projects.

The OAO-C Satellite Control Center had three XDS 930 computers. When the OBP was selected for flight on the OAO-C satellite, the assembler/loader/simulator system was run on one of the control center computers for program development. A significant effort went into providing software for the control center to handle the needs of a flight computer. Programs were developed to selectively load the OBP from a memory image produced by the OBP loader. The OBP could be selectively dumped, and resulting images could be compared with load images with formatted listings of noncompare locations. Several other ground-support functions were developed.

In the OAO control center, a Honeywell 516 computer with interfacing hardware provided an exact electrical simulation of the OAO command and data-handling system. A breadboard model of the OBP was connected to the Honeywell minicomputer in the same manner as the flight OBP was connected to the satellite's data system. The OBP/516 subsystem could operate in one of three modes. In one mode, the 516 could receive commands from and send telemetry to one of the 930 computers that provided the normal control center functions of command generation and telemetry display. In this mode, the OBP could be loaded and dumped, and all timing functions could be checked by running the OBP at a real-time rate with its operation system controlling various combinations of applications programs. The OBP/516 could also operate in a standalone mode. In this mode, the OBP could be run at a real-time rate or be single-stepped. This mode was useful in tracing the execution of a program through the inspection of a panel that displayed the contents of the OBP registers. In the third mode, one 930 was used as a ground station telemetry and command computer, and another 930 simulated the subsystems of the spacecraft. The spacecraft simulator was attached to the OBP/516 hardware. All application programs could be checked in this mode since the external environment was simulated for all OBP functions that required feedback. A task was given to a group not involved in OBP program development to write test programs for this last mode to verify that all OBP program paths were correctly executed. The similarity of the test environment and the actual flight environment proved of such value that a test system philosophy was established on OAO that has followed flight computer development at GSFC to the present time.

After the OBP, the next significant change in the support software for the computer (then called the AOP) was for the IUE Project. The assembler/loader was rehosted from the 24-bit XDS 930 to the 32-bit XDS Sigma 5 used in the IUE spacecraft integration system. Since the primary use of the on-board computer for IUE was attitude-control computations, the ground test system emphasized attitude-control simulation. A rigid-body simulation of the IUE spacecraft was developed by the Guidance and Control Branch at GSFC that could execute at a real-time rate on the Sigma 5 computer. After being initialized with starting spacecraft rates and starting orientation relative to a guide star, the simulator received reaction wheel and gas jet commands and produced a count of gyro rate pulses and star tracker error counts each sampling interval. This simulator program was integrated into the Sigma 5 computer that also provided the real-time telemetry, command, and display functions to support spacecraft test during hardware assembly. The spacecraft test system was developed by GSFC's Electrical Test Branch and could operate in one of two modes: In one mode, the Sigma 5 was cabled directly to the IUE Observatory for assembly testing. In the other mode, the Sigma 5 was connected to electrical equivalents of spacecraft equipment, spare flight models of the command subsystem and the data subsystem. A breadboard OBC was connected through interfacing hardware to the spacecraft dynamics simulator in the Sigma 5 and also to the command and data subsystems. This assemblage of equipment served two purposes: It provided a complete environment for AOP software development and test. The system also supported electrical checkout of flight equipment that could connect to or replace the command, data, and AOP boxes.

In addition to the load, dump, compare, patch, and display software that had been written for ground computers to support an AOP, "data block" generation and load software was written for IUE. The database for the IUE computer was partitioned into 32-word data blocks with parity to provide a convenient means of controlling the flight program from the ground and ensuring error-free transmission of the data to the satellite. The IUE control center developed software to support OBC operations, and the assembler/loader package was transported to the XDS Sigma computers in the control center.

A Software Development and Validation Facility (SDVF) was established by the Multimission Modular Spacecraft (MMS) Project to develop standard executive and stored command handling programs for the computer (NSSC-1) in the MMS spacecraft. The Solar Maximum Mission (SMM), the first Project to use the MMS spacecraft, contributed to the development of the SDVF and added a flight dynamics simulator for attitude control program checkout. The SDVF was similar to the IUE test system in that there were hardware equivalents of the computer and the spacecraft command and data-handling subsystem connected to a telemetry, command, and display ground computer. The dynamic simulator of the observatory developed by SMM ran in a DEC PDP 11/70 and was connected to a Sigma 5 computer that provided telemetry, command, and display functions. The capability to load computer data blocks from the ground was changed to a table-load capability on SMM that permitted the loading and dumping of large tables. The assembler/loader system was used for program development on SMM.

A significant ground-support software upgrade was made by General Electric (GE) Company, the mission contractor for Landsat D, which was the second user mission of the MMS. GE wrote a new assembler, loader, and simulator program for the NSSC-1 and hosted the system on a VAX 11/780 computer. The package has more features than the original system. A test system for the computer on Landsat D was developed, which was similar in concept to the test systems developed for IUE and SMM. This system has a breadboard flight data system connected to a breadboard version of the flight machine. The data system sends commands to and receives responses from a dynamic simulation of the observatory, and a ground computer provides telemetry, command, display, and computer load/dump/compare functions.

A HAL-S compiler for the NSSC-1 was developed by Intermetrics, Inc. The cross-compiler was hosted on an IBM 360. The compiler has not been used to date because of execution-time inefficiency. The major task for the NSSC-1 has been attitude-control computations, which are lengthy and time critical. There has not been enough time margin to take advantage of a high-level language for the NSSC-1.

Flight Executive Software

Standard software developed for the NSSC-1 consists of an executive program with a status buffer and a stored command processor. The executive program, or exec, performs several functions: It handles all input and output that includes issuing commands to and receiving and formatting sampled data from instruments and spacecraft subsystems. It processes all interrupts and schedules the execution of application programs or tasks. It communicates with the ground by handling requests for various services and by maintaining a status buffer that is a time-tagged log of significant events. The computer also forms messages that are sampled by the flight data system and are included in the data telemetered to the ground.

The allocation of functions to the exec has not changed from the original to the current version. Spacecraft and instrument data have been brought into the computer synchronously, using a DMA channel on all versions of the exec. In later versions of the exec, the input data items have been individually named so that an application program may refer to the data without regard to its relative position in an input buffer. The issuance of commands has grown in sophistication from one first-in, first-out list of single commands to three prioritized queues of command requests where each request is issued on 16-millisecond centers and represents from 1 to 16 commands.

The scheduling of application programs has always been based on a time-sliced system that was synchronous with the receipt of data. On OAO, a frame of telemetry data had 65 words. These data were telemetered to the ground and were also inputted to the OBP. The spacing between words was 25 milliseconds, and the entire frame took 1.6 seconds to input to the computer. An interrupt was generated as each word was received and used as the clock for the time-sliced task scheduler. A decision was made each 25 milliseconds as to which application program would receive control of the machine. Each application program could start on any of the 65 words per frame, it could keep the computer for a selectable number of words, and it had a relative priority in case there was contention among application programs for the computer. Application programs could be individually initiated or terminated by ground command.

This structure was modified slightly on IUE. The clock that was synchronous with the receipt of telemetry data had a period of 50 milliseconds, and the same 50-millisecond interval was used as a time-slice for the task scheduler. Each task had a relative priority, a selectable maximum number of time-slices per execution, and a selectable number of time-slices between executions.

More control was given to application programs on IUE in that one program could request that another program be either initiated or terminated. There were also two input data channels for IUE: One channel received the data that were telemetered to the ground, and the other channel was used to input data only for the computer. The second channel was necessary for a higher sampling of attitude sensor data for attitude-control computations.

The exec developed for MMS was used on SMM and on Landsat D with minor modifications. The task scheduler for the MMS exec has 64 time-slices with a duration of 16 milliseconds each. Application programs may be executed in any order through entries in a 64-word table corresponding to the time-slices. Again, tasks are assigned a maximum number of slices for protection against infinite loops, and each task has a relative priority for resolution of contention for execution by more than one task. A separate set of low frequency tasks may be executed on individual multiples of 1.024 seconds, which is the time required to pass through all time-slices for the tasks having a higher frequency of execution.

A data-gathering convenience was added to the MMS exec. Data to be telemetered to the ground are continuously received by the computer using a DMA channel. If an application program needs data sampled at a higher frequency than used for telemetry, then a set of data may be separately issued to the computer on a different DMA channel coincident with selectable time-slices in the 1.024-second scheduling period.

The ground can issue commands to the exec by sending a command to the spacecraft with proper address and coding for the flight computer. The class of ground requests for the exec on OAO, IUE, SMM, and Landsat D has not changed markedly. The major functions are program-initiated loading or dumping of memory, either initiating or terminating the execution of tasks, and dumping or resetting a status buffer.

The status buffer is an area of memory into which the exec and application programs can enter messages tagged with the time of entry. The status buffer has been of value for low earth-orbiting missions by providing a trace of "back orbit" events. On Landsat D, an exec modification was made that allows a continuous readout of the status buffer through normal telemetry without resorting to a dump of memory that would interrupt a data channel to the ground. Another type of ground command to the exec controls a stored command processor, which is discussed in the following section.

Stored Command Processor

Continual issuance of on-board commands is needed for controlling spacecraft subsystems and instruments. It is not possible to issue these commands from the ground because of limited contact time between ground stations and the spacecraft. For this reason, low earth-orbiting spacecraft implement some form of stored command processor (SCP) in which a set of commands with time tags is sent to the spacecraft during a contact. Then, the SCP periodically compares the time tags with a spacecraft clock and issues those commands whose times agree with the clock. Prior to the use of on-board computers, SCPs were implemented in hard-wired logic. The capability of SCPs has increased with the use of flight computers.

On OAO-C, the SCP function was performed in an IBM logic unit called a primary processor that could store 256 commands. The OBP was used to extend the stored command capacity by holding an additional 1024 commands as eight pages of 128 commands each. As the primary processor worked through one-half of its

256 commands, it would begin executing the other half and the OBP would transfer a page of commands to overlay the space just used. There was a parity word associated with each page loaded into the OBP to verify correct loading from the ground so that the flight memory did not have to be dumped each time it was loaded with commands. In addition to the auxiliary command storage, the OBP implemented stored command logic with time compares for 16 commands. The ground could use the small SCP to handle commands not contained in the primary processor load.

On Landsats B and C, the AOP was used both as an SCP and as a generalized device to check status data for out-of-limits and possibly take corrective action. The time granularity of stored commands was one second on these two missions. Each second, the AOP would search the time tag of all commands and issue those commands whose time equaled the spacecraft clock. Commands were not necessarily stored in time sequential fashion.

Since IUE is in geosynchronous orbit, an SCP is not required. However, a command processor was implemented to accurately control a sequence of commands used for firing a gas jet. The jet was periodically fired to maintain IUE's geosynchronous orbit. The firing of smaller jets was also controlled by the AOP to prevent the saturation of reaction wheels.

The SCP developed by MMS has two main parts: One part executes commands stored in time sequential order for comparing command time tags with a spacecraft clock. This is called the absolute time command processor (ATCP). The time resolution of the ATCP is one second. The other part contains a set of command sequences with time delays between each command in a sequence. Any sequence can begin execution at any time, and several sequences can execute simultaneously. The program that handles the execution of the command sequences is called the relative time command processor (RTCP). The delays between commands in a relative time sequence are specified as multiples of one second. Commands with zero delays are executed one after another on one-millisecond centers. A relative time sequence can be started by a ground command, by a command in the ATCP, or by an application program. These RTCPs have the advantage of providing command sequences that can be executed on a recurring basis throughout the life of the mission without requiring daily command reload that is necessary for ATCP operation. The RTCPs can also be used for functions such as controlling the synchronous operation of several instruments as a means of achieving coordinated science.

Instrument Control

Many of the early instruments in space required little control, just power on and off commands in some cases. As instruments became more sophisticated, they became more flexible, thus requiring more complicated instrument control. Instrument control functions include configuring the instrument by commanding mechanisms such as filter wheels and scanning mirrors, controlling data collection, monitoring the health and safety of the instrument, and processing data. Some instruments have autonomous control functions that include automatic gain adjustment and data dependent alteration of operational sequence. The clear trend in approaches to instrument development is to use imbedded microprocessors for instrument control. Another approach is to use a central machine for controlling several instruments. However, the latter introduces difficult management and technical interfaces unless the instrument group and the central computer group are in the same organization.

Since the IUE instrument can be continuously commanded from the ground, the instrument was designed for ground control. It was found, however, that the control function could be more efficiently carried out by the AOP on IUE where, essentially, the operating command procedure was sent as a block of commands to the AOP, which issued the commands in the correct-timed sequence to the instrument. This provided better command timing and also relieved the ground computer of a time-consuming task.

Several instruments on SMM were controlled by imbedded microprocessors. One instrument that did not have an imbedded microprocessor was controlled by the NSSC-1. This control was permitted by the general telemetry and command interface that existed between the NSSC-1 and all instruments and subsystems.

Most instruments on Space Telescope (ST) have microprocessors. However, the NSSC-1 on ST uses MMS-type interfacing circuits so that it has a general data/command connection with the instruments. In addition to an SCP function, general limit checking, telemetry format control, and other functions, there is memory space allocated to each of the five instruments in the NSSC-1. Typical uses of this program space are health and safety monitoring, the issuance of macrocommands to the imbedded microprocessor, backup of the microprocessor, and the computation of small slew commands to control telescope pointing. The NSSC-1 performs higher bandwidth control functions for one ST instrument that has no microprocessor.

Attitude Control

Attitude-control computations involve the processing of attitude sensor data to estimate the attitude of the spacecraft with respect to some coordinate system. The desired attitude is then determined. The difference or error signal is then passed through a control law, which results in a control voltage to torquing mechanisms.

On IUE, the AOP is used for fine attitude control, slew control, and coarse attitude control during an orbital adjust maneuver. The attitude sensors used by the AOP are a set of six gyros and two star trackers that could provide two axis error signals with respect to a selected guide star. Other attitude sensors are not used by the computer but are used by analog circuitry for coarse attitude control. Spacecraft torquers on IUE are an orthogonal set of momentum wheels and orthogonal pairs of low thrust jets. Coarse attitude can be computed on the ground from sun sensor and earth sensor data. The ground determines the celestial attitude of IUE more precisely by examining star fields taken by the on-board star tracker. The star tracker can also track a sufficiently bright star within the field of view of the tracker.

The nominal operational sequence is to control the attitude through the AOP during an observation by using gyro and star tracker data in a Kalman filter to control the momentum wheels. The star tracker signal used is the offset between a guide star and a target star in the field of view of the tracker. This error can either be filtered and used to periodically adjust a drift term for the gyros or be used directly in the computation of each control voltage. For situations in which there is no guide star near an object to be viewed, fine control is maintained with a gyro-to-wheel loop. Satellite repositioning to view the next object is accomplished by a series of single-axis slews under computer control using a gyro-to-wheel loop. The computer fires small thrusters if an attitude error exceeds a threshold value during the burning of a large thruster and during an orbit adjust maneuver. Fine pointing and slew maneuver computations occur each 200 milliseconds on IUE, and attitude control during orbit adjust occurs each 50 milliseconds.

SMM and Landsat D attitude-control computations are progressively more complex than those on IUE. For SMM, the observatory points toward the sun, and a high-resolution sun sensor is used as the attitude reference sensor to update bias terms for the gyros. Momentum wheel saturation is prevented on SMM by controlling current loops around three orthogonal magnets to interact with the earth's magnetic field. The resultant force tends to move the spacecraft in such a direction that the torque is removed by decreasing the speed of the momentum wheels, thus preventing wheel speed saturation. The SMM also performs an expansion on 16-minute centers of a Fourier power series that represents spacecraft position as a function of time. An interpolation is performed between those points each 32 seconds, and the satellite position data thus obtained are included in telemetry so that they do not have to be separately merged for science data processing on the ground. The coefficients of the Fourier power series are generated on the ground from an extrapolation of satellite tracking data and then sent to the spacecraft for inclusion in telemetry to have a complete science data set. Attitude-control computations occur each 512 milliseconds on SMM.

The two instruments on Landsat D take images of the earth in several spectral bands. The attitude-control system positions the observatory so that the instruments are nadir viewing. The computations for the control system are made by the NSSC-1. The basic approach is to establish an inertial attitude from star tracker and gyro data and to determine a desired pointing based on predicted satellite position data. The on-board attitude estimation process uses a Kalman filter. Attitude estimate and control signals are computed twice a second.

Limit Checking/Corrective Action

This process compares values contained in telemetry against upper and lower limits. If a value exceeds the limits, an alarm message is included in telemetry and possibly a corrective command or set of commands is issued by the flight computer. Programs were written to handle specific cases on OAO and SMM. For Landsats B, C, and D, the approach is to use a table-driven program in which telemetry items and their associated limits and corrective commands, if any, can be easily modified by changing a table. The capability also exists to make a compound expression of several comparisons connected by logical operators. The Upper Atmosphere Research Satellite (UARS) and the instrument module for ST will use this generalized approach.

Summary Message Generation

The idea of a summary message collection and transmittal to the ground originated on OAO-C. The mechanism was termed a status buffer, meaning a table in memory into which messages are stored. The messages are tagged with spacecraft time upon entry to the status buffer to give a time history of events. Significant spacecraft, instrument, and flight software events are stored in the status buffer upon occurrence. Normally, the status buffer is sent to the ground by dumping the area of memory in which it resides. Landsat D added the convenience of routinely reading the status buffer on a cyclical basis and including these data in the normal telemetry slots assigned to the computer. This approach is planned for UARS.

Initiate Safe Hold

Safe hold means an analog control mode of a satellite that keeps the spacecraft and instruments "safe" with adequate power and within temperature limits. The control mode is usually either to maintain coarse earth pointing for earth-pointing missions or to establish and maintain an attitude that provides sunlight normal to the solar array. This mode may be entered automatically if the computer does not periodically issue pulses to the control system signaling computer health. The other way of entering safe hold is for the computer to diagnose a control system error and then to command a safe hold condition.

Time Code Generation

This function entails the generation of a time code with which to annotate data to simplify processing by the ground. On SMM, spacecraft time was converted to Greenwich mean time for annotating spacecraft position data included in the telemetry slots assigned to the computer. On UARS, universal time will be computed by adding a time increment to the time on regular intervals. The ground must be aware of time delays in the end-to-end data system to make periodic (once a day or so) corrections to the incremental value to be added each period. This will have the effect of making slope changes, not resets, so that the time value will be monotonic.

Backup Telemetry Generation

On IUE and on all satellites in which the NSSC-1 is connected to MMS data system components, there are two ways of generating the telemetry format: One way is for a read-only memory (ROM) unit to generate the addresses of telemetry items to cause their time-ordered sequence of sampling. This occurs independently of the flight computer. Another way to generate the sequence of addresses of telemetry items to be sampled is to switch from the normally used format ROM to the flight computer where the addresses can be issued from the computer's memory using a DMA channel. This second way has the advantage of being completely flexible since the area of computer memory holding the telemetry addresses can always be changed from the ground. The use of the computer to generate telemetry formats has been very useful in early observatory integration work before determining the final format that needs to be written into a ROM. To date, the computer-generated telemetry formatting has not been needed in orbit because there have been no operational or data processing requirements for adjustment of the format defined at launch.

COMPUTER UTILIZATION

User Mission and Functions

To date, seven space missions have been launched that use either the OBP, AOP, or the NSSC-1 versions of the computer. Four additional missions are planned for launch during the 1980s. These 11 users are listed in Table II, which presents an overview of computer utilization on the different missions. Functions included in the table are those described in the preceding discussion of flight software. Some functions, such as the SCP, are not identical for all missions because of the uniqueness of the application or the continued upgrade of the flight software. Also indicated in the table are the computer version and memory size and technology used for each mission.

With increases in application experience, the computer has been used more and more as an integral part of the observatory designs to perform a number of planned mission-critical functions during orbital operations. In addition to these planned functions, the computer has also been used to perform a number of tasks conceived after launch to deal with unforecasted events and conditions on board. Four of the seven missions launched to date have taken advantage of the computer's telemetry and command interface and its ability to be reprogrammed in orbit to either extend mission life or avoid degradation of performance.

OAO-C

At launch, the OBP was used on OAO-C to extend the stored command capability, generate summary messages relating to selected back orbit data, provide emergency commanding for critical out-of-limit conditions, and provide analysis of power system performance. During the mission lifetime of almost nine years, the computer was reprogrammed on several occasions to work around equipment degradation or failure and to provide improved performance of the observatory and instruments. Discussions of some of the more significant reprogramming activities follow:

1. About three months after launch, minor degradation of the power bus voltage level required that the hardware undervoltage detector be disabled to avoid faulty operation. In this case, the OBP was used to provide the undervoltage protection feature by commanding off nonessential loads if an unacceptably low voltage level was detected. Although enabled, a low-voltage condition was never

TABLE II. On-Board Computer Applications

Mission	Launch Year	Computer Version	Memory Size and Technology	Functions*							
				Stored Commands	Instrument Control	Attitude Control	Limit Checking/ Correlative Action	Summary Message Gen.	Initiate Safe Hold	Time Code Gen.	Backup Telemetry Gen.
OAO-C	1972	OBP	16K Core	✓	✓		✓	✓	✓		
Landsat B	1975	AOP	4K PW	✓			✓				
IUE	1977	AOP	12K PW		✓	✓			✓		✓
Landsat C	1977	AOP	4K PW	✓			✓				
SMM/MMS	1980	NSSC-1	48K Core	✓	✓	✓	✓	✓	✓	✓	✓
OSS-1	1982	AOP	8K PW	✓							
Landsat D/MMS	1982	NSSC-1	64K Core	✓		✓	✓	✓	✓		✓
Landsat D′/MMS		NSSC-1	64K Core	✓		✓	✓	✓	✓		✓
Inst. Module for ST		NSSC-1	64K Core	✓	✓		✓	✓			✓
Gamma Ray Obs.		NSSC-1	64K Core	✓				✓			
UARS/MMS		NSSC-1	64K Core	✓		✓	✓	✓	✓	✓	✓

* Functions are arranged in the order in which they were described in the text.

experienced, and this function was not executed during the life of the mission.

2. During the second year of operation, a failure occurred in the "fine" cold gas thrusters used by the attitude-control system for automatic momentum unloading of the reaction wheels. The OBP was reprogrammed shortly after the failure to monitor wheel speed and maintain the wheels below saturation by firing "coarse" thrusters that had been placed on board to provide large attitude maneuvers.
3. Shortly after the second year of operation, failure of the engineering data tape recorder caused loss of back orbit data needed for observatory operation. The OBP, already being used to monitor selected engineering data and generate summary messages, was reprogrammed to extend the monitor function. Interestingly, control center personnel preferred using computer-generated messages that provided a quick look at back orbit performance, and loss of the tape recorder did not seriously affect their ability to operate the observatory.
4. Early in the mission, the OBP was used to provide timing signals to the University College of London Experiment as a means of improving resolution by increasing the data-sampling rate. During the third year of operation, the OBP was reprogrammed to send periodic commands to the Experiment Data Handling Equipment to provide improved resolution of the Princeton Experiment Package in a similar manner.
5. In the observatory's fourth year in orbit, a worker was added to the OBP, which enabled the computer to distribute a simulated "star presence" indication to the OAO pointing control system. This signal was produced on the basis of gyro data and was used by Fine Error Sensor guidance logic to provide observation of dimmer stellar targets than had been originally planned.
6. During the fifth year in orbit, the OBP sustained a partial failure of an integrated circuit chip. Reprogramming successfully bypassed this problem for a few months until a more severe intermittent failure in the chip caused additional problems. At this point, only partial operation was possible as a series of program changes was made in an attempt to work around the failure. Finally, in the seventh year, the auxiliary command memory function was disabled; however, other workers essential to the mission continued to operate.

Even with its partial failure in later years, the OBP proved to be a most useful tool on OAO-C. Credit is given to the Grumman operations team and to the OAO Corporation's software development group for innovative use of the computer to maintain successful operation until observatory shutdown in February 1981.

Landsats B and C

Both of these early Landsat missions used the AOP to augment the stored command capability and perform on-board telemetry limit checking. Out-of-limit conditions were reported to the ground as summary messages, and for conditions considered as unsafe, the computer was used to take corrective action that ranged from turning off equipment to initiating observatory safe hold mode. During Landsat B operations, failure of the yaw-axis reaction wheel was effectively worked around by using the computer to control magnetic torquer bars as a function of gyro data. Since the computer for this mission contained only 4K words of memory, the Computer Sciences Corporation's programmer did an outstanding job of implementing the necessary software changes.

Before the launch of Landsat C, the AOP flight program was upgraded to expand the monitor and corrective action functions as a result of Landsat B experience. One software change was made to avoid occasional garbling of command messages caused by conflicts in command activity between the ground and on board. In this case, the AOP was programmed to monitor command receiver status and distribute on-board commands in a timed manner that would allow for interleaving with commands from the ground. A minimum of postlaunch reprogramming for Landsat C has been required.

IUE

The IUE performs ultraviolet astronomy, and the main function of the AOP is to perform attitude-control computations. The AOP is used to control large slews to reposition the spacecraft to view a different object. It also maintains fine pointing control during observations. Other functions performed by the AOP on IUE are instrument control, wheel unloading, orbit adjust, and several anomaly detection functions. Instrument control prepares the camera for an exposure and starts and stops the exposure. The ground computer determines firing durations for selected jets for wheel unloading, and the AOP controls the firings. During orbit adjust, the AOP controls the timed firing of a large jet and maintains attitude control by firing small jets orthogonal to the thrust axis based on gyro data. In addition to these planned functions, the AOP on IUE has successfully been used to work around two potentially mission-critical failures. Since the AOP is used to control the maneuvering and pointing of the spacecraft, no astronomy could occur on IUE without a functional machine. Also, a working set of gyroscopes is required to measure short-term spacecraft motion necessary to maintain accurate pointing for stellar observations. Both of these systems have had partial failures that have been accommodated by the flexibility of having a reprogrammable computer on board.

Since the AOP is mission-critical on IUE, it was configured with a redundant power converter and CPU and an extra 4K words of memory. The IUE has its normal program and data in 8K words of memory and a bare-bones set of program and data in the extra 4K in case there is a problem in the primary 8K. The memory has worked well so far, but the 4K backup has been used several times for control while the 8K prime has been reloaded. The reloading was necessitated several times during early operations when a problem developed in the "ON" computer. (Since the problem has been handled by software, the redundant power converter/CPU has not been activated in fear of difficulty in switching and of finding a similar problem in the redundant units.) The problem is an occasional scrambling of the return vector when an interrupt occurs. The problem seems to be caused by high computer temperature, which resulted in unacceptable switching delays in a critical string of circuits. The fix has been to examine the return vector for reasonableness and then go to an idle loop and wait for the next 50-millisecond interrupt if it is not reasonable. This explanation has made some significant simplifications necessary for brevity.

The IUE has six gyros that are tilted 75 degrees from the roll axis of the spacecraft and that are equally spaced around the roll axis. There is a matrix in the AOP that transforms the outputs of selected gyros into spacecraft coordinates. This was designed deliberately to accommodate possible gyro failures. Three of the six gyros have failed, and each failure has been worked around by sending up a new matrix to the AOP. The philosophy that was used on IUE was to have analog hardware to perform coarse attitude control if the computer or a gyro fails. This gave ground personnel time to think and take action to correct the problem either by switching to a redundant unit or by reprogramming the computer. When a gyro failed, the ground could switch to an analog attitude hold mode, send up the new matrix to the AOP, and resume fine attitude control with the AOP.

OSS-1

This mission was a pallet-mounted set of space science instruments that flew as an attached payload on the third Shuttle flight (STS-3). The first Office of Space Science (OSS-1) mission used the AOP version of the computer that had been developed as a spare for the IUE and Landsat B and C observatories. For this one-week mission, the computer provided the stored commands necessary for efficient control of instrument operation, generated summary information used by ground controllers, and drove display panels on the shuttle. The computer contained two 4K word memory modules, which were flown primarily for redundancy.

SMM

The SMM contained a complement of solar observing telescopes and instruments and was the first observatory to use the MMS. As such, it was also the first mission to fly the NSSC-1 version of the computer. In addition to performing the planned functions listed in Table II, the SMM computer has been used since the second month in orbit to work around a number of radiation-induced anomalies, and it is now being used to provide for observatory survival. A mission-disabling problem occurred after approximately six months of operation when failures of three undersized fuses in the drive circuits for the on-axis reaction wheels caused a loss in the ability to control attitude. During the three-week period spanning the occurrence of the three failures, the spacecraft was placed into a power positive mode by using ground commands to induce a 1-degree-per-second roll around the solar-pointing axis. Shortly afterward, the NSSC-1 was reprogrammed to provide a more secure closed-loop control. The new control algorithm uses the coarse sun sensor and gyro data as input and commands magnetic torquer bars to develop momentum. The spacecraft, now being held to within a 10-degree cone about the sun, is sufficiently controlled to support survival and a limited level of scientific instrument operation. Significantly, a Solar Max Repair Mission is planned for mid 1984 when astronauts aboard the Space Shuttle will exchange the MMS attitude-control subsystem and restore the SMM to normal operation.

The earlier problem on SMM that resulted in computer reprogramming was caused by cosmic particle radiation. On several occasions, single bit errors were observed in a bipolar memory device located in the MMS communications and data-handling subsystem. The device affected is used in the circuitry that controls spacecraft telemetry format timing, and a number of the errors caused alterations of the spacecraft time code, which created a variety of ground control and data processing problems. The NSSC-1 was reprogrammed to provide a software spacecraft time code for ground use and to monitor the remaining memory device content for radiation-induced upsets. This action has allowed smooth operation, even though memory alterations have continued to be detected.

Landsat D

Landsat D is an earth-referenced observatory containing two imaging instruments and is the second and most recent mission to use the MMS. NSSC-1 applications on Landsat D are indicated in Table II and have been discussed in some detail in the preceding section covering flight software. A point worth noting is that, except for circuit changes needed for correcting the generic problems experienced on SMM, the MMS hardware used to accommodate the two missions is essentially identical. This highlights one of the major benefits of a central on-board computer in that significantly different mission requirements can be met through changes to only the flight software.

CONCLUSIONS AND RECOMMENDATIONS

Some of the experiences, both good and bad, gained in the development and application of the NSSC-1 and its predecessors could well serve as guidelines in the design of new on-board computer systems and in their use as centralized machines on future space missions.

The more important conclusions and recommendations are offered in the following paragraphs:

1. Coding of some of the necessary algorithms with a fixed point design has proved to be somewhat costly in terms of computer time and memory. Future machines should include hardware floating-point arithmetic with sufficient precision.
2. Spacecraft requirements seem to demand continued increases in on-board autonomy and control system performance. This means that new computers should operate at higher speeds and contain more memory.
3. Use of nonvolatile memory technology, such as core or plated wire, has proved almost essential. It can be power-strobbed and therefore expanded with little increase in power. It is immune to radiation and retains its content when power is removed from the computer, either on purpose or by some anomaly. New space computer designs should, as a minimum, use nonvolatile technology for the program portion of memory.
4. Computers used in a centralized role on an observatory should be interfaced in a general-purpose manner within the command and telemetry system. This provides the flexibility needed for closed-loop control operations, any desired level of on-board autonomy, and allows the accommodation of unforecasted observatory needs both before and after launch.
5. In general, science data processing and microcontrol of mission instruments using the central computer should be avoided. Microprocessors within the instruments can best perform these tasks. The reasons for this recommendation are that conflicts for central computer resources are reduced, on-orbit instrument operations are simplified, and most important, instruments can be fully developed and tested before launch with no dependdence on the central machine.
6. All low earth-orbiting observatories designed around the use of a central computer should have a survival or safe hold mode that can be automatically entered if the computer fails to issue signals indicating proper performance. This will allow for safe observatory operation through periods of transient anomalies or failures and will provide sufficient time for ground controllers to make the repair.
7. Although the trend in observatory designs is to load the computer with a number of tasks, there should always be sufficient spare memory and processor time held in reserve to work around problems that will inevitably occur after launch.
8. An extensive software development and test system that has a high-fidelity simulation of the computer's environment is valuable in uncovering timing and logic problems.
9. The flight software should be designed to permit in-orbit reprogramming to enhance operations or work around equipment failures.

CR Categories and Subject Descriptors: C.4 [**Performance of Systems**]—*reliability, availability, and serviceability*; D.2.2 [**Software Engineering**]: Tools and Techniques; D.2.5 [**Software Engineering**]: Testing and Debugging; D.2.9 [**Software Engineering**]: Management—*software quality assurance*; D.4.5 [**Operating Systems**]: Reliability; J.2 [**Physical Science and Engineering**]—*aerospace*; K.6.3 [**Management of Computing and Information Systems**]: Software Management—*software development, software maintenance*

General Terms: Design, Management, Reliability, Verification

Additional Key Words and Phrases: avionics system, PASS

Authors' Present Addresses: Charles E. Trevathan and Thomas D. Taylor, UARS Project, Code 430, NASA/Goddard Space Flight Center, Greenbelt, MD 20771; Raymond G. Hartenstein, Applied Engineering Division, Code 730, NASA/Goddard Space Flight Center, Greenbelt, MD 20771; Ann C. Merwarth, ST Project, Code 400.2, NASA/Goddard Space Flight Center, Greenbelt, MD 20771; and William N. Stewart, MMS Project, Code 408, NASA/Goddard Space Flight Center, Greenbelt, MD 20771.