TRIAD PROGRAMMABLE COMPUTER

J. A. Perschy, B. M. Elder, and H. K. Utterback

The computer reviewed in this article is the third generation digital data handling system for the Navy Navigation Satellites. It is also the first APL general-purpose computer used as an integral part of the data-handling system on board an unmanned satellite. First covered is the computer's relation to other satellite systems, followed by the computer's organization. Special attention is given to the priority interrupt structure and energy conservation. The remaining sections summarize the design of the computer software and the in-orbit performance.

Introduction

THE TRIAD SATELLITE LAUNCHED IN LATE 1972 provided APL with its first opportunity to fly a general-purpose computer. The computer served as the satellite digital controller executing programs in what is called a priority interrupt time share mode. The utilization of this computer represents a significant advance in the implementation of logic functions on board satellites. TRIAD exemplifies the value of a small, programmable computer in managing and supporting complex satellite missions. The computer was required to provide the following services:

1. Support attitude stabilization by initiating delayed commands.
2. Provide data storage and processing in conjunction with the attitude, disturbance compensation, environmental survey, power, and thermal systems.
3. Broadcast a valid navigation message resulting in successful ground navigation.
4. Operate the satellite Greenwich Time Clock.
5. Exercise the Pseudo-Random Noise mode on the radio frequency downlink.

The use of a general-purpose programmable computer to perform these services was contingent upon obtaining high program storage efficiency and low operating duty cycle. The attainment of both established the advantage of using a general-purpose programmable computer instead of a special-purpose nonprogrammable computer. To satisfy the navigation satellite message storage and formatting requirements, a special-purpose nonprogrammable computer of nearly equal complexity would have been required and would not have provided for programmable stored telemetry and delayed commands.

Computer Relation to Other Satellite Systems

Figure 1 shows the computer relation to other satellite systems. The data link from the earth-bound station to the computer is via the redundant receivers. The output of the receivers is detected as either a 1000-bit-per-second serial data stream or a 10-bit-per-second serial data stream, depending on the type of modulation the earth-bound station places on the radio frequency carrier. Re-

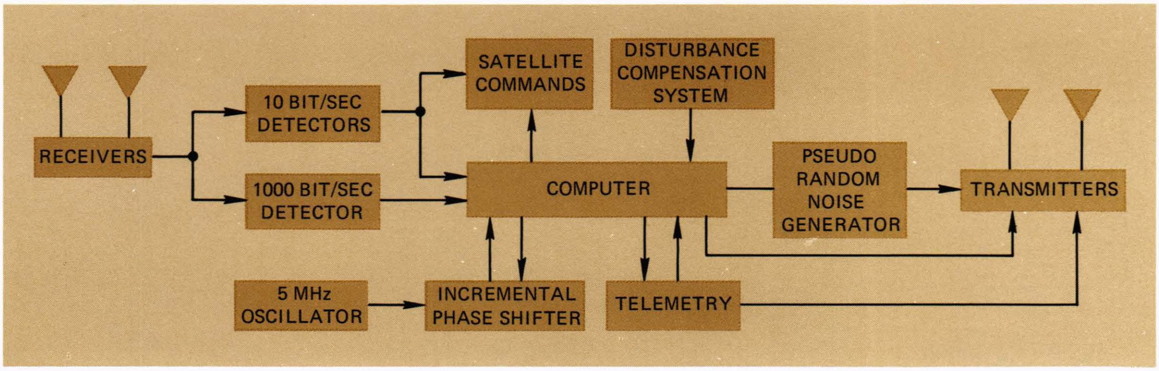

Fig. 1—Computer relation to other satellite systems.

dundant 10-bit-per-second detectors may either send data to the computer or command the satellite into one of its many operating modes. The 1000-bit-per-second detector is used exclusively for loading the computer memory. Long navigation messages and programs are loaded via the 1000-bit-per-second detector.

The computer may send delayed satellite commands to the satellite command system. The commands are loaded into the computer when the satellite is in view of the earth-bound station, along with the desired time at which the command is to be executed. The commands may be executed at any position in the orbit of the satellite. Six signal lines from the Disturbance Compensation System give the computer six microthruster valve on-off waveforms for partial data reduction, formatting, and storage. The computer provides timing and direct on-off commands to the Pseudo Random Noise Generator. For navigation the computer outputs the Navy navigation message directly to the transmitter. This message is very precisely timed by the computer real-time clock, and changes format every two minutes under computer program control. Via the telemetry system, the computer may store data on the satellite's behavior over long intervals of time and transmit these data when the satellite is in view of the earth-bound station.

The 5-MHz oscillator is used to generate the precise computer time signals. These time signals are made more accurate by the Incremental Phase Shifter which is discussed in this issue of the *APL Technical Digest*. The computer provides programmed counts, similar to delayed commands, to the Incremental Phase Shifter to regulate the phase-shift rate of the 5-MHz signal for frequency drift correction.

Modes of Operation

Viewed from a station on earth the computer has two uplink modes: operational and restart.

In the operational mode a program contained in Read-Write Memory controls the computer. This program makes the computer accessible to other satellite subsystems. In this mode, commands sent from the station on earth, when properly introduced and formatted, are gated by the computer to the command system.

Restart is a recovery mode. In this mode commands are not monitored by the computer but are routed directly to the command system. In addition, the restart mode provides the earth-bound station access to a basic loader program stored in a nonerasable memory called the Read-Only Memory in the computer. The restart mode can be activated by command or by fault indicators within the computer. The fault indicators consist of (a) a short-term counter whose overflow indicates the absence of computer operation in excess of 2.5 minutes, (b) a long-term counter whose overflow signals a failure to command or perform computer injections for a selected period of time (hours or day), and (c) a delayed command bit rate that exceeds specified limits.

Computer Organization

Figure 2 is a simplified diagram showing the computer organization. The data paths represented as solid lines actually consist of sets of 16 parallel lines. Other data paths exist but for clarity are not shown. The interface with the other satellite systems is through the Input-Output Logic. The Read-Only Memory contains the nonvolatile program used for loading the Read-Write Memory via the ground station RF link. The Read-Write

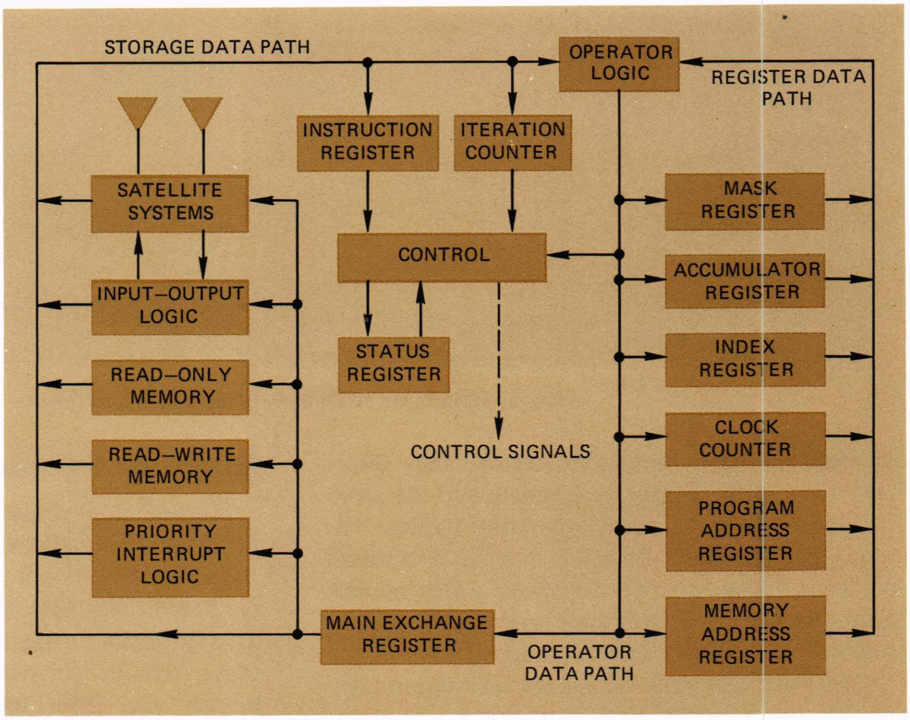

Fig. 2—Computer organization.

Memory is used for data storage and other programs. The Priority Interrupt Logic is described in more detail in the following section. The Operator Logic is capable of a wide variety of operations including addition, subtraction, shifting, inverting, gating, forming the logical AND, OR, and Exclusive OR, and combinations of the above operations. The output of the Operator Logic, caled the Operator Data Path, is the input to the arithmetic registers. The Mask Register contains the carry, overflow, and input-output flags, the power mode control bits, and eight interrupt masks. The Accumulator Register is used to store intermediate computation results. The Index Register is used primarily as a memory-addressing aid. The Clock Counter Register consists of a 16-bit binary counter and a 16-bit synchronizing register. Negative overflow of the counter causes real-time clock interrupt. Operations that are performed with the Accumulator Register can also be performed with the Clock Counter. The Program Address Register holds the memory address of the next instruction. The Memory Address Register holds the current memory address. The Main Exchange Register is used primarily as temporary storage for data to be loaded in Read-Write Memory.

The control portion of the computer consists of a partially populated, or associative, Read-Only Memory addressed by the State Register, Iteration Counter, Instruction Register, and the Operator Data Path. The control signals sequence the data flow paths. The control unit generates 56 control signals, called microinstructions. These microinstructions in various sequences and combinations activate the computer to execute 201 types of instruction. Thus, for example, the memory reference instruction "transfer the content of a certain storage location from the memory to the Accumulator Register" would be executed by means of a sequence of microinstructions, such as: place the content of the memory on the Storage Data Path, transfer the content of the Storage Data Path through the Operator Logic to the Operator Data Path, and finally transfer the content of the Operator Data Path into the Accumulator Register.

Interrupts—The computer normally communicates with external devices, and also the Input-Output Logic in its own interface, by way of the computer priority interrupt logic, in Fig. 2. Interrupts occur independently of and nonsynchronously to the computer program in execution and cause the content of the Program Address Register to be automatically modified, changing the sequence of instruction execution. Interrupts have unique memory locations dedicated to them. The action of an interrupt causes the program to jump to the location whose address is stored in the dedicated location. The resulting location is the start of a short program (subroutine) provided to service the interrupt. Before starting the service subroutine, the computer stores the address of the location from which the jump occurred to provide a return link to the original program that was being executed prior to the interrupt.

Device interrupts are accepted by the computer in a priority sequence if two or more functions are trying simultaneously to gain access to memory. Once the external subsystems are interconnected to the computer the priority of a particular subsystem is fixed. However, some flexibility is afforded the program that allows selected interrupts to be inhibited by controlling an Interrupt Mask Register. Table 1 lists the thirty-two interrupts in order of priority.

TABLE 1

INTERRUPT ASSIGNMENTS, LISTED IN PRIORITY ORDER

Priority	*Assignment*
1	Highest Priority Interrupt (Unassigned)
2	Restart Mode Data Load
3	10-Bit-Per-Second Data Load
4	1000-Bit-Per-Second Data Load
5	Data Load Register Full
6	Telemetry Frame Synchronization
7	Navigation Message and Timing Interval
8	Telemetry Data Available
9	Telemetry Requests Computer Status
10	Telemetry Requests Stored Telemetry Data
11	Unassigned
12	Unassigned
13–24	Six Thruster On and Off Interrupts
25–32	Program Initiated Interrupts

Memories—The TRIAD computer Read-Write Memory contains 4,096 words. The word length is 16 bits. A data word is stored and read out from the memory in parallel (i.e., 16 bits at a time). The cycle time of the memory is 2.5 microseconds. A cycle consists of either a read operation followed by a restoration of the data readout, or a read operation followed by a write operation in which new data are stored in the addressed word location. The Read-Write Memory system uses a purchased high reliability core memory stack with decoding diodes attached. Sixty-four hybrid microcircuits are used in the drive and sense circuitry of the memory system. The complete system uses integrated circuits, hybrid circuits, and discrete components. The memory is designed for complete power shutdown when not being addressed by the computer.

The TRIAD computer Read-Only Memory contains 64 words. The Read-Only Memory does not have a write cycle since information is permanently written into it. The word length of 64 is sufficient to store a simple program for loading the core memory over the radio frequency data link. This program accepts the starting address and number of words to be loaded, performs an odd parity check on each word, called row parity, an odd parity check on every bit position of all words, called column parity, and if all row and column parities are odd, transfers program control to the last word loaded in core memory.

Packaging—Figures 3 and 4 show two views of the prototype flight computer logic. The right-hand card on the open view, Fig. 3, contains the Instruction Register, Iteration Counter, Status Register, and Control. The connector in the upper right corner is used for preliminary test monitor and display. The next card contains the Operator Logic and arithmetic registers. These two cards combine to form what is called a central processing unit. All of the first card containing the control is fabricated with medium power microcircuits. This card is pulse powered and shuts down completely when not in use. The registers on the operator logic card are fabricated with low power microcircuits and are powered when low priority computing is in process, or shut down when computing is not in process. The remaining two cards contain peripheral logic consisting of the uplink data input logic and control circuits, delayed command format error detection circuits, transmitter modulation encoder, telemetry Input-Output Logic, and Priority Interrupt Logic. The Read-Only Memory, Storage Data Path, and connectors for

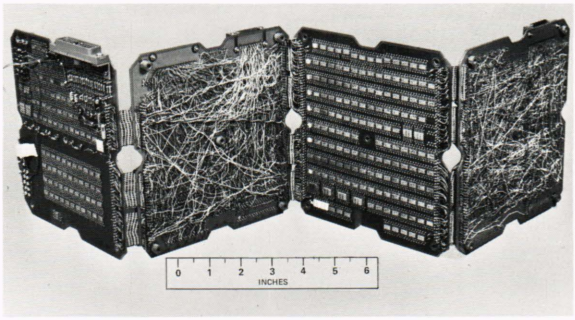

Fig. 3—Logic boards, open view.

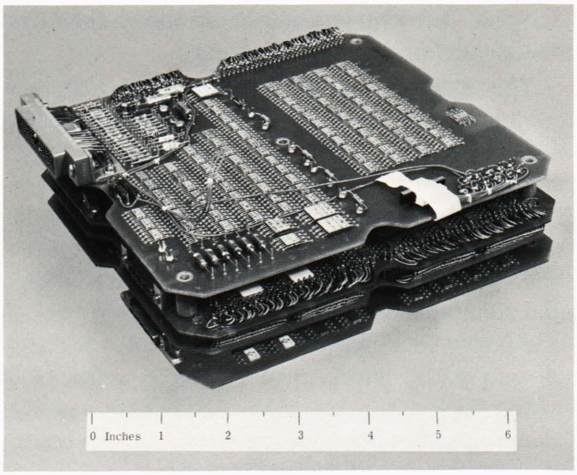

Fig. 4—Logic boards, folded view.

the core memory and satellite systems are also in the peripheral cards.

The construction method consists of point-to-point welded wire on feedthrough pins for signal lines and laminated planes with access holes for power and ground connections. Microcircuits are parallel gap-welded to the top of the board as a final step in board fabrication. The boards, bolted into a magnesium casting, weigh 2 pounds. The core memory, which consists of two boards and the core stack bolted into another magnesium casting, weighs an additional 2½ pounds. The two castings clam-shelled together in the satellite occupy 200 cubic inches and weigh a total of 4½ pounds. The power processor, Fig. 5, which consists of current limiters, energy storage capacitors, and gated voltage regulators, weighs 2 pounds and occupies 50 cubic inches.

Energy Conservation—Most of the computer is pulse powered and shuts down when not in use. When an interrupt is recognized, the computer begins to execute the interrupt task at its maximum computing rate. At its maximum computing rate, the computer, using core memory for both program and data storage, executes 250,000 instructions per second at a power level of 62 watts. Sufficient energy is stored on capacitors for 2 milliseconds of operation. This fast computing rate is necessary to accomplish certain tasks related to high priority interrupts quickly and efficiently. Lower priority tasks are accomplished at a power level the satellite power system is capable of continuously supplying. The computer has four lower priority computing rates, which may be selected under program control by two bit positions in the Mask Register. These computing rates are as follows:

Instructions/second	*Power (watts)*
2,000	1.5
4,000	2.0
8,000	3.0
16,000	5.0

The TRIAD computer, using its energy-hungry core memory for both program and data storage, requires 200 microjoules per instruction from the computer power processor. Giving this a more useful interpretation for a small satellite, the computer can process at a 1 watt average power level, an average of 50 interrupts per second with an average of 100 instructions per interrupt.

Instruction Types—The computer functions are accomplished through software subroutines comprised of instructions. The computer examines one instruction at a time from memory. Each instruction may have one memory address associated with it. The computer has a repertoire of 201 16-bit instructions, which are divided among the following six types: memory reference, immediate, conditional branch, register reference, input/output, and shift.

A memory reference instruction is one that involves either the Read-Only Memory or the Read-Write Memory. Typically information is either read out of storage (load) or read into storage (store). In a load memory reference instruction the content of the storage location addressed in the Memory Address Register, Fig. 2, would appear on the Storage Data Path. In a store memory reference instruction, the content of a register would be transferred to the Main Exchange Register and then into a core memory storage location. By means of a memory reference instruction,

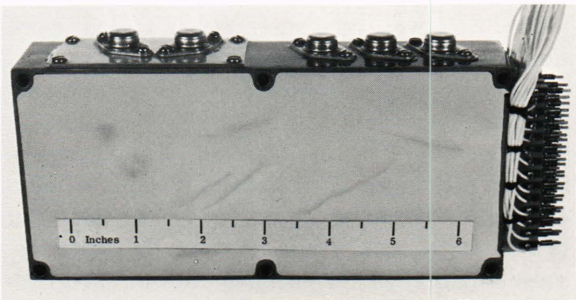

Fig. 5—Power processor.

the computer registers may be loaded with data stored in the memory, or the data in these registers may be stored in the memory. Most arithmetic and logic instructions performed are memory reference instructions. A class of memory reference instruction, called index instructions, may be executed to cause the content of a storage location to be incremented and the next instruction to be skipped or executed accordingly as the result of the comparison is negative, zero, or positive. Index instructions are very powerful when used to modify program flow paths. Another type memory reference instruction may be executed to cause the content of the storage location specified in the instruction address to be executed as the next instruction. This is called a jump instruction. Also, a subroutine jump may be made, such that after execution of the subroutine the original program will be resumed at the point at which it was interrupted.

In contrast to a memory reference instruction, which requires an extra memory cycle to obtain the operand, an immediate instruction is one in which one-half of the 16 bit instruction is the operand. An immediate instruction may be used to perform many of the same operations of a memory reference instruction, but faster and with less energy.

A register reference instruction is one in which operations are conducted on the contents of two registers. Approximately one-half of the 201 instructions are register reference instructions.

A conditional branch instruction is one that causes a program jump provided that a test is or is not satisfied. The test may be to check whether the content of the accumulator register is zero, positive, or even, if the content of the index register is positive or even, if the carry or overflow flags in the Mask Register are zero, or if external input-output conditions have been met.

A shift instruction causes the content of a register to be shifted to the right or left or to be rotated either end around. Either the accumulator or index register may be shifted or rotated up to 16 positions, or these two registers may be linked and shifted or rotated together up to 32 positions. The type of shift may be either arithmetic for scaling signed numbers, or logical for independent bit manipulation.

An input-output instruction is one that causes data to be read into or out of the computer to some external device, such as the telemetry or bit detector.

States—Execution of an instruction often requires that a component of the computer such as the Operator Logic be used more than once. Therefore, for considerations of timing, the control signals in Fig. 2, called microinstructions, are executed in groups and a common time interval is used for executing the microinstruction groups. The microinstructions grouped together for execution constitute a certain state of the computer. The Status Register in Fig. 2 defines the 16 states used in executing the 201 instructions. Some simple instructions require only a few states for execution, whereas other complicated instructions require the use of a large number of states. Sequences in which the computer can execute the 16 states are shown in the simplified state sequence diagram, Fig. 6. The detailed state sequence diagram would look very complex to a casual observer. For simplicity in Fig. 6, one or more states are grouped according to purpose.

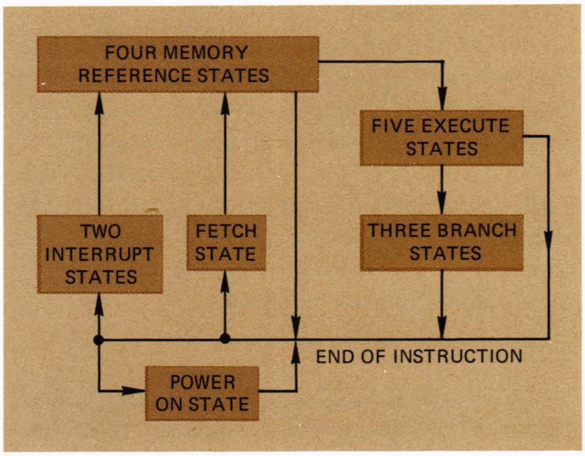

Fig. 6—State sequence diagram.

Upon entering the Fetch state, the memory (Fig. 2) is started to read the next instruction. During the Fetch state the Instruction Address Register is incremented by one count. Toward the end of the Fetch state the instruction is available on the Storage Data Path. The instruction is decoded by the control to determine whether it is a memory reference type instruction. If the instruction is a memory reference type, the next states entered are the memory reference states. If the instruction is not a memory reference type, the next states entered are the execute states.

The memory reference states generate the address of the operand of a memory reference instruction. This address is placed in the Memory Address Register as the memory reference state group is exited.

As the execute state group is entered from the memory reference state group, a memory cycle is begun which either reads data from the memory or writes data into the memory. For an interchange type instruction both memory read and write occur during the execute states. All arithmetic and logic computations are performed in the execute states.

The branch states are used to change the program flow by modifying the contents of the Program Address Register.

The interrupt initiation states are used to break the normal flow of a program between instructions and force a subroutine jump to a program determined by the Priority Interrupt Logic. If an interrupt is present at the end of an instruction, the interrupt states are entered instead of the Fetch state.

The Power On state is forced into the Status Register while power is being turned on. If the arithmetic registers had been left on during power down, the content of the Program Address Register is transferred to the Memory Address Register. If the arithmetic registers had been turned off during power down, "1's" are placed in these registers. The Power On state is then exited as if an instruction were completed.

Software

The preceding discussion of the flight computer hardware alludes to the computer capabilities; however, without the software or programs, the general purpose digital computer accomplishes nothing.

The computer was built to perform a repertoire of specific real-time functions that complement the operation of the other satellite systems. In short, mission requirements called for a continuously operating system of software to manage various combinations of diverse functions in real time, interact with independently running satellite systems, and maintain a precise clock. The major software projects included the formulation of:

1. A programming language and an associated assembler;
2. Special-purpose diagnostic and utility programs;
3. A real-time simulation of the expected satellite hardware environment;
4. An in-flight operating system capable of coping with various real-time operational demands.

The need for a convenient way to write code and prepare data for input to the computer resulted in an assembly program. This program is written in PL/I on the IBM 360/91. Special diagnostic programs aided in testing the computer before satellite installation. The need for a realistic satellite environment to debug the real-time software for the flight computer was satisfied by a comprehensive simulation program. This program was written on the Xerox Data Systems Sigma 3 computer. A special-purpose interface was built to handle communication between the Sigma 3 and the TRIAD computer. A monitor panel was also built to display various logic states and all computer-telemetry status data from the computer.

Diagnostic programs were designed to check the flight computer after installation in the satellite. Each of these programs runs independently of the other. Proper loading and execution are indicated via the RF link from the satellite in the telemetry data stream. All of the diagnostics are real-time programs which interact with other satellite systems that are working independently. The programs are transmitted from the ground station computer disk file to the flight computer via the RF link.

One set of programs checks the basic working condition of the computer itself. The checks include the central processing unit, the interrupt logic, and the core memory. A set of programs tests the ability of the computer to carry on sustained computations. Sustained computations are limited by the charge stored in a capacitor located in the power processor. When the capacitor becomes discharged, a low voltage mode is entered and the computer stops. One program executes an endless loop count. When the computer stops, the count accumulated indicates the total number of instructions that can be executed without interruption. Another program tests the charge recovery rate of the capacitor. A fixed set of instructions is executed with increasing frequency. Eventually more power is consumed in executing the fixed set than can be restored while waiting for the next execution sequence. A total of twelve classes of diagnostic programs had to be prepared to thoroughly check out the computer and its interfaces with the TRIAD satellite subsystems.

Operational Programs

Hardware constraints determined the strategy chosen to deal with the operational programs. The flight computer contains 4,096 16-bit words of core space. This is several times too small to house all mission-related programs and their attendant storage needs. By necessity then, various programs must be jockeyed in and out of the computer without destroying the underlying operational continuity. The computer cannot operate with an average duty cycle exceeding 5 percent. Furthermore, the maximum number of instructions executed continuously in the peak power mode is 500. These restrictions require that extensive computing be done in a background (gated power) mode. This approach becomes even more critical when one allows for the simultaneous occurrence of all possible external interrupts. As a real-time controller for various satellite missions, the flight computer must be capable of initiating and conducting a number of on-board experiments. The maintenance of a clock and an overall job-controlling mechanism is required.

The design of the flight software revolved around the clock counter interrupt and the memory storage capacity restriction. The master control software is resident in memory for all computing modes. This system contains the Time-Register Program, a nonrestart mode loader program (more sophisticated than the Read-Only Memory loader), the Telemetry Frame Synchronization Program, a Computer Memory-to-Telemetry Program, the Telemetry Status Request Program, the Pseudo Random Noise Program, and the Incremental Phase Shifter Program. Table 2 summarizes the master control software. With the master control software serving as the nucleus, the remaining operational programs were able to operate as required, although they were not all resident in the memory at the same time.

The Flight of the TRIAD Computer: Results and Epilogue

In September 1972, TRIAD was launched into a nearly circular polar orbit. The on-board computer was used throughout the ascent phase to store TM data describing the activation of the Radioactive Thermal Generator nuclear power supply.

From this beginning, the computer was regularly used in cycles of gathering around-the-orbit data and then relaying the data to the ground. The large number of on-board experiments created

TABLE 2

MASTER CONTROL SOFTWARE

Function or Program Name	*Size (words)*	*Description*
Interrupt Locations	32	Table of core locations for programs to service internal and external interrupts to the computer.
Common Parameters, Constants, etc.	226	Frequently used parameters, constants, flags, etc., for all programs.
Non-Restart Loader	435	Loads data in Non-Restart mode for Secure or Non-Secure injections at either 1 K bps or 10 bps.
Time-Register Program	110	Keeps a GMT software clock; maintains direct access to high duty cycle programs such as delayed command, navigation message output, Incremental Phase Shifter, and Pseudo-Random Noise programs; reads a real-time event initiation schedule.
Real-Time Event Queue	100	Maintains a queue of real-time events vs. the GMT when events are to be initiated.
Loading Status Feedback Program	77	Provides a real-time feedback to ground station of success or failure of Non-Restart computer injections.
Computer Dump Program	90	Dumps a ground-specified area of flight computer core on the 325 bps telemetry downlink.
TM Frame Sync Program	15	Counts TM frames in parallel with the GMT clock; used mainly in conjunction with TM storage functions.
Status Request Program	15	Fills 48 telltale bits of telemetry data stream with computer status information.
PRN Program	35	Operates the Pseudo Random Noise system in the "continuous" or "chopped" modes.
IPS Program	83	Maintains operations dealing with Incremental Phase Shifter device—i.e., gives epoch adjustment, frequency adjustment, and initialization of Incremental Phase Shifter counters.
Total Space req'd	1,218	

demands to reprogram the computer for different memory map configurations several times a day. Command and computer operations were done solely at APL.

By the first month of flight, all data-harvesting and experiment-controlling programs had been exercised. The computer successfully provided the following services:

1. Support of attitude stabilization using delayed command function;
2. Data storage and processing in conjunction with the attitude, Disturbance Compensation System, Environmental Survey Package, Radioactive Thermal Generator, and thermal systems;
3. Broadcast of a valid navigation message resulting in a successful ground navigation experiment;
4. Operation of the on-board GMT clock using the two timekeeping systems;
5. Exercise of the special Pseudo Random Noise mode on the 160-MHz downlink.

After a month in orbit, a failure in the telemetry system put an end to 90% of the possible telemetry functions. This drastically curtailed the real-time data available during satellite passes and eliminated the usefulness of most of the data-gathering flight programs.

After the failure, the flight computer became the only means of monitoring the Disturbance Compensation System and its fuel consumption rate. The Disturbance Compensation System thruster program was used extensively in this capacity to support a long-term Disturbance Compensation System experiment. Other major computer activities during this period included broadcasting 50 bits/sec navigation data and managing the on-board clock.

A second failure at the end of the second month—this time in the computer injection logic—prevented further access to the computer. Although all experiments using the computer had been run as planned, the early system failures meant a disappointing data yield.

In judgment of the flight software system, all of the requested combinations of activities were accomplished within the on-board constraints. The realization of continuous computer activity ranged between one and two weeks between disruptions. These problems were due largely to the complex ground support operations needed for TRIAD and the varying operational procedures for different experiments.

Failure notwithstanding, the TRIAD flight computer proved to be an extremely useful and versatile tool in a satellite with such diverse experimental and operational requirements.

Acknowledgment

The authors wish to acknowledge the contribution of C. A. Blackburn in the design of the computer power processor, and the contributions of M. A. Walko and R. S. Yuhasz in the preparation of the operating programs for the computer.

PUBLICATIONS

Compilation of principal recently published books and technical articles written by APL staff members.

I. H. Schroader, C. T. Holliday, and G. B. Bush, "Optical Alignment of a 0.5 Meter Ultra-Violet Spectrometer for Apollo 17," *Proc. Soc. Photo-Optical Instrument. Engineers* **39**, Aug. 1973, 259–269.

D. D. Zimmerman, "Effective Use of Epoxy Adhesives in Microelectronics," *11th Electrical/Electronics Insulation Conf. Record*, Sept. 1973, 330–332.

C. B. Bargeron, "Analysis of Intensity Correlation Spectra of Mixtures of Polystyrene Latex Spheres by Least Squares," *Appl. Phys. Letters* **23**, No. 7, Oct. 1, 1973, 379–381.

J. C. Armstrong and A. J. Zmuda, "Triaxial Magnetic Measurements of Field-Aligned Currents at 800 Kilometers in the Auroral Region: Initial Results," *J. Geophy. Res.* **78**, No. 28, Oct. 1, 1973, 6802–6807.

J. G. Parker and D. N. Ritke, "Collisional Deactivation of Vibrationally Excited Singlet Molecular Oxygen," *J. Chem. Phys.* **59**, No. 7, Oct. 1, 1973, 3713–3722.

F. J. Adrian, E. L. Cochran, and V. A. Bowers, "ESR Spectrum and Structure of the *n*-Propyl Radical," *J. Chem. Phys.* **59**, No. 8, Oct. 15, 1973, 3946–3952.

H. E. Gilreath and C. R. Walton, "Frequency Response of Station-