MASSACHUSETTS INSTITUTE OF TECHNOLOGY

APOLLO

GUIDANCE, NAVIGATION AND CONTROL

R-700

MIT's ROLE IN PROJECT APOLLO

FINAL REPORT ON CONTRACTS
NAS 9-153 AND NAS 9-4065

VOLUME III

COMPUTER SUBSYSTEM

by

ELDON C. HALL

AUGUST 1972

MIT CHARLES STARK DRAPER LABORATORY

CAMBRIDGE MASSACHUSETTS 02139

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

APOLLO

GUIDANCE, NAVIGATION AND CONTROL

Approved _____ Date 25 Aug 72
D. G. HOAG, DIRECTOR
APOLLO GUIDANCE AND NAVIGATION PROGRAM

Approved _____ Date 25 Aug 72
R. R. RAGAN, DEPUTY DIRECTOR
THE CHARLES STARK DRAPER LABORATORY

R-700

MIT's ROLE IN PROJECT APOLLO

FINAL REPORT ON CONTRACTS
NAS 9-153 AND NAS 9-4065

VOLUME III

COMPUTER SUBSYSTEM

by

ELDON C. HALL

AUGUST 1972

MIT CHARLES STARK DRAPER LABORATORY

CAMBRIDGE, MASSACHUSETTS, 02139

ACKNOWLEDGMENTS

This report was prepared under DSR Project 55-23890, sponsored by the Manned Spacecraft Center of the National Aeronautics and Space Administration through Contract NAS9-4065.

The author expresses appreciation to many individuals within Draper Laboratory who contributed to the following sections of the report. In particular:

A. Hopkins	Section 1, 2
A. Green, R. Filene	Section 4
A. Harano	Section 5
D. Bowler	Section 6
D. Hanley, J. Partridge	Section 7.

The publication of this document does not constitute approval by the National Aeronautics and Space Administration of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

Published by the Charles Stark Draper Laboratory of the Massachusetts Institute of Technology
Printed in Cambridge, Massachusetts, U.S.A., 1972

FOREWORD

The title of these volumes, "MIT's Role in Project Apollo", provides but a modest hint of the enormous range of accomplishments by the staff of this Laboratory on behalf of the Apollo program. Man's rush into spaceflight during the 1960s demanded fertile imagination, bold pragmatism, and creative extensions of existing technologies in a myriad of fields. The achievements in guidance and control for space navigation, however, are second to none for their critical importance in the success of this nation's manned lunar-landing program, for while powerful space vehicles and rockets provide the environment and thrust necessary for space flight, they are intrinsically incapable of controlling or guiding themselves on a mission as complicated and sophisticated as Apollo. The great achievement of this Laboratory was to supply the design for the primary hardware and software necessary to solve the Apollo guidance, navigation and control problem. It is to the credit of the entire team that this hardware and software have performed so dependably throughout the Apollo program.

The quantum leap in technology nurtured by the Apollo program has been and should continue to be of immensely significant benefit to this country—socially, economically and in terms of its national esteem. It is the responsibility of all those who contributed to the proud achievements of Apollo to convince their countrymen of the directions this nation ought to follow in implementing these newly gained—and hard fought for—advances.

C. Stark Draper, President
Charles Stark Draper Laboratory

R-700

MIT'S ROLE IN PROJECT APOLLO

Final Report on Contracts
NAS9-153 and NAS9-4065

VOLUME III

COMPUTER SUBSYSTEM

ABSTRACT

The Apollo guidance computer (AGC) is a real-time digital-control computer whose conception and development took place in the early part of 1960. The computer may be classified as a parallel, general-purpose or whole number binary computer. This class of computer is representative of most of the ground-based digital computers in existence in the late 1950s, when the precursers of the AGC were being designed. However, at that time few computers of this class had been designed for the aerospace environment, and those few embodied substantial compromises in performance for the sake of conserving space, weight, and power. The design of the AGC capitalized on advancements in digital computer technology in order to provide significant improvements in computational performance and still conserve space, weight and power.

The AGC is the control and processing center of the Apollo Guidance, Navigation and Control system. It processes data and issues discrete output and control pulses to the guidance system and other spacecraft systems. An operational Apollo spacecraft contains two guidance computers and three DSKYs (keyboard and display unit for operator interface), with one computer and two DSKYs in the command module, and one of each in the lunar module. The computers are electrically identical, but differ in the use of computer software and interface control functions. As a control computer, some of the major functions are: alignment of the inertial measurement unit, processing of radar data, management of astronaut display and

controls and generation of commands for spacecraft engine control. As a general purpose computer, the AGC solves the guidance and navigation equations required for the lunar mission.

The operational experience* in the Apollo guidance systems includes 20 computers which flew missions and another 25 flight type computers which are still in various phases of prelaunch activity including storage, system checkout, prelaunch spacecraft checkout, etc. These computers were manufactured and maintained under very strict quality control procedures with requirements for reporting and analyzing all indications of failure. Probably no other computer or electronic equipment with equivalent complexity has been as well monitored and documented. Since it has demonstrated a unique reliability history, it is important to record the technique and methods which have contributed to this history.

by Eldon C. Hall
August 1972

* The operational experience includes missions through Apollo 15 which flew in August of 1971. Three Bl I and one Bl II computers flew in unmanned missions. Sixteen Bl II computers flew manned missions.

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

LIST OF ILLUSTRATIONS

Figure

LIST OF TABLES

PREFACE

"I believe this nation should commit itself to achieving the goal before this decade is out of landing a man on the moon and returning him safely to earth." With these words, spoken on 25 May 1961, President John Fitzgerald Kennedy stated for all Americans the challenge of the APOLLO project. The Massachusetts Institute of Technology Instrumentation Laboratory* was selected to design and develop the hardware and software of the APOLLO Guidance, Navigation and Control system for safe and self-sufficient translunar flight, lunar landing, and return. This Final Report describes that work — a most demanding, innovative, and rewarding task.

This report presents the Draper Laboratory's efforts in Project APOLLO from the original contract award in mid-1961 through July 1969. The report is organized in five volumes:

VOLUME I: PROJECT MANAGEMENT AND SYSTEMS DEVELOPMENT

VOLUME II: OPTICAL, RADAR, AND CANDIDATE SUBSYSTEMS

VOLUME III: COMPUTER SUBSYSTEM

VOLUME IV: INERTIAL SUBSYSTEM

VOLUME V: THE SOFTWARE EFFORT

Volume I emphasizes what was done in terms of resource allocation and systems development and contains Appendices A and B; Volumes II through IV describe the hardware subsystems in detail, with emphasis on the final design configurations; Volume V fully treats the Laboratory's software effort. Appendix A presents abstracts of significant research and engineering reports and theses written under Contracts NAS 9-153 and NAS 9-4065. Appendix B is a bibliography of all such reports and theses prepared through June 1969. This date is also the cutoff for all discussions within this report, except for APOLLO 11 — the first manned lunar landing and return.

*The Laboratory was renamed the Charles Stark Draper Laboratory in January 1970.

SECTION 1.0
SUBSYSTEM DEVELOPMENT

1.1 EVOLUTION OF THE COMPUTER SUBSYSTEM

The APOLLO guidance computer (AGC) is a real time digital control computer whose conception and development took place in the early part of 1960. The computer may be classifed as a parallel, general-purpose or whole number binary computer. This class of computer is representative of most of the ground based digital computers, in existence in the late 1950s, when the precursors of the AGC were being designed. Few computers of this class had been designed by that time for the aerospace environment, and those few embodied substantial compromises in performance for the sake of conserving space, weight, and power.

The APOLLO guidance computer was a descendant of a series of MIT/IL designs. In 1959 and 1960, MIT/IL designed a unique machine with a self-contained guidance and navigation capability intended for use in a proposed space vehicle to photograph Mars and return. This machine, although never built, was designed to use magnetic core-transistor logic. This type of logic circuitry is desirable because of its low power consumption, but it tends to be slow and bulky. The Mars computer had very limited capabilities due to the constraint on physical size but, since computation speed requirements were not severe, an adequate machine was possible within the state-of-the-art at that time. The instruction repertoire, word length, and number of erasable memory cells were also very limited. Provision was made, however, for a moderately large amount of fixed memory for instructions and constants. A high density memory of the read-only type, called a rope memory, was developed especially for this purpose and was carried over into the design of the APOLLO computer.

The rope memory, being a transformer type, depends for its information storage on the patterns with which its sensing wires are woven at the time of manufacture. Once a rope memory is built, its information content is fixed and is unalterable by electrical excitation; it can be altered only by re-manufacture, repair, or destruction. An erasable memory, on the other hand, is characterized by its ability to be loaded and altered electrically at any time. Rope memories were used in the APOLLO guidance computer because of their high density and information retention advantages. However, ropes placed a burden on software delivery schedules because of the

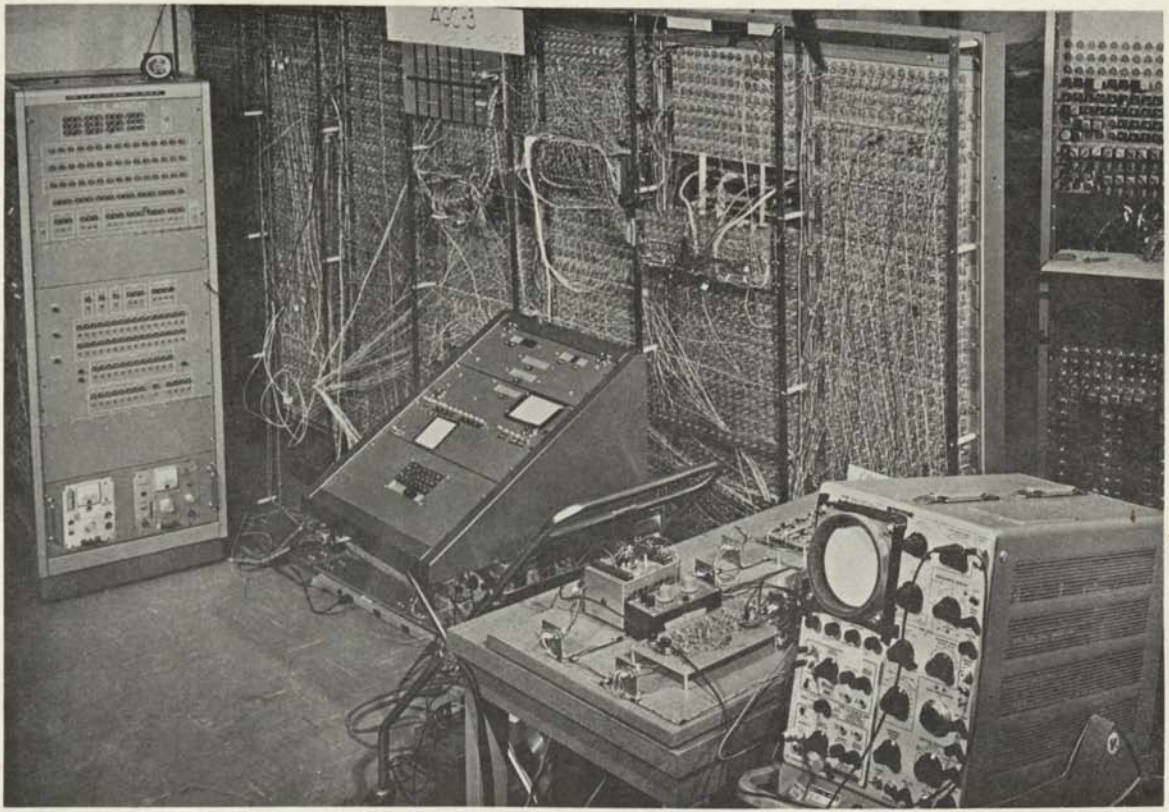


Fig. 1-1 AGC-3 Core-Transistor Logic

additional time required for manufacturing the rope memories. Also system integration and testing was less flexible than was desirable, but these disadvantages were considered acceptable when compared with the advantages of high density and reliability.

The Mars computer incorporated a method of accommodating real time inputs and outputs that was unusual at the time but is widely used now. This was the program interrupt method which had been introduced in large scale computers a few years previously. A novel use of the interrupt was for single accesses to memory for trivial processes such as incrementing or shifting. This process has become known as memory cycle stealing and is widely used in linking computers to such peripheral units as magnetic tape units and card readers. The use of the interrupt in its various forms was an important addition to the aerospace computer field, and was a major architectural attribute of the AGC.

Another important aspect of the Mars computer which had evolutionary significance for the APOLLO computer was the use of an interpretive program; that is, a large library of subroutines which perform various higher level arithmetic and language operations. Without the interpretive instructions, the limited instruction repertoire of the computer could have been costly in the amount of memory needed to contain the mission program. The interpretive program is a means of trading off execution speed against instruction repertoire. The Mars computer appeared to have a commodious set of instructions, that allowed substantial economies in program writing, where the more powerful "instructions" were actually executed by subroutines written in the more restrictive basic instruction repertoire.

The Mars computer design was followed by a number of subsequent revisions and additions embodied in several models designated Mod 1A, 1B, 3C, and AGC3 each of which incorporated the rope memory, core-transistor logic, and the same general architecture relating to interrupts, word length, and instruction repertoire. Word length varied from 12 to 24 bits and instruction repertoire from 4 to 32 basic instructions with additional involuntary and interpretive instructions.

Mod 3C was notable in this series for the introduction of a 512-word 16-bit coincident-current magnetic core erasable memory. This followed the commercial development of lithium ferrite cores with a low enough temperature coefficient to be practicable in a broad temperature range application. The Mod 3C hardware was eventually converted to the AGC3. See Figure 1-1. This was the last in the series of core-transistor logic computers and the first prototype computer developed for the APOLLO guidance system.

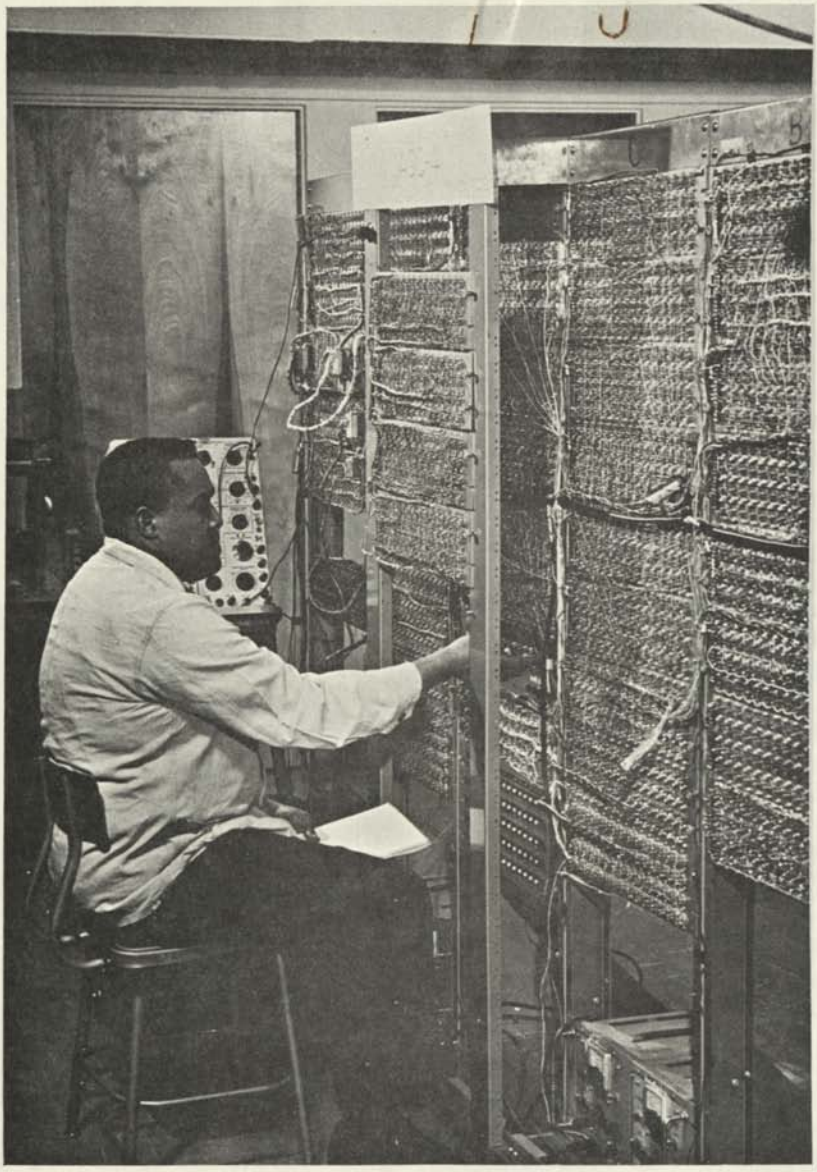


Fig. 1-2 AGC-4 Engineering Model

In AGC3 the erasable memory had 1024 words, also 16 bits long. Its rope memory held 12,288 words, and it had 11 instructions in its repertoire. Normal instruction execution time was 40 microseconds, of which approximately 20 microseconds were consumed by the two memory access cycles required to fetch (or store) instructions and data. The remaining time was occupied in manipulation and transfer of numbers among the central and special registers, including the adder and various buffering and editing registers.

The editing and input/output operations, which are commonly handled by special instructions, were processed by special memory cells instead. For example, to shift a word in this family of machines, the word is stored in a special register designed for this purpose and read out again. The shifting is accomplished between the store and read registers. This technique sacrifices memory, but is essential in cases where the instruction repertoire is severely limited.

During the evolutionary period of the Mars computer and its descendants (1958 to 1962), the hardware technology available to the aerospace computer designer was developing rapidly. Mention has already been made of the adaptation of the three-dimensional magnetic core array to the temperature environment. Another important line of evolution was in semiconductor technology, where silicon transistors progressed to planar forms, then epitaxial form. Ultimately monolithic integrated circuits were developed. Still another area of development was in packaging, where the introduction of welding, matrix interconnection techniques, and machine wirewrapping allowed significant reductions in volume and weight while coincidently enhancing reliability. These packaging techniques were reduced to practice and used by MIT/IL in the development of the POLARIS guidance computer. The experience gained and the techniques developed during the POLARIS program were used more or less directly in the APOLLO computer design.

Integrated circuits were in development by the semiconductor industry during the late 1950s under Air Force sponsorship. In late 1961, MIT/IL evaluated a number of integrated circuits for the APOLLO guidance computer. An integrated circuit equivalent of the Mod 3C computer was constructed and tested in mid-1962 to discover any problems the circuits might exhibit when used in large numbers. This became the engineering model of AGC4. (See Figure 1-2.) Reliability, power consumption, noise generation, and noise susceptibility were the primary subjects of concern in the use of integrated circuits in the AGC. The performance of the units under evaluation was sufficient to justify their exclusive use in place of core-transistor logic with the exception of a portion of the erasable memory addressing circuitry, where metal tape cores were retained as a medium for current switch selection.

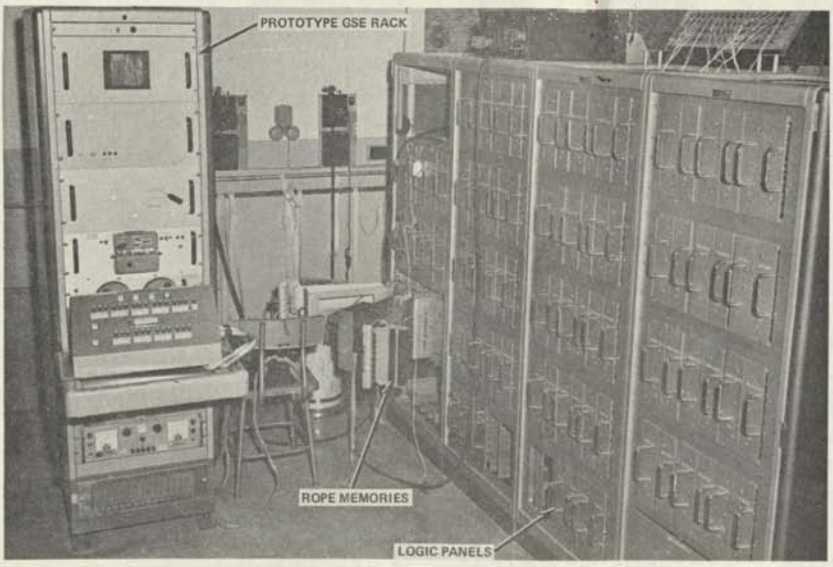


Fig. 1-3 AGC-4 Front View

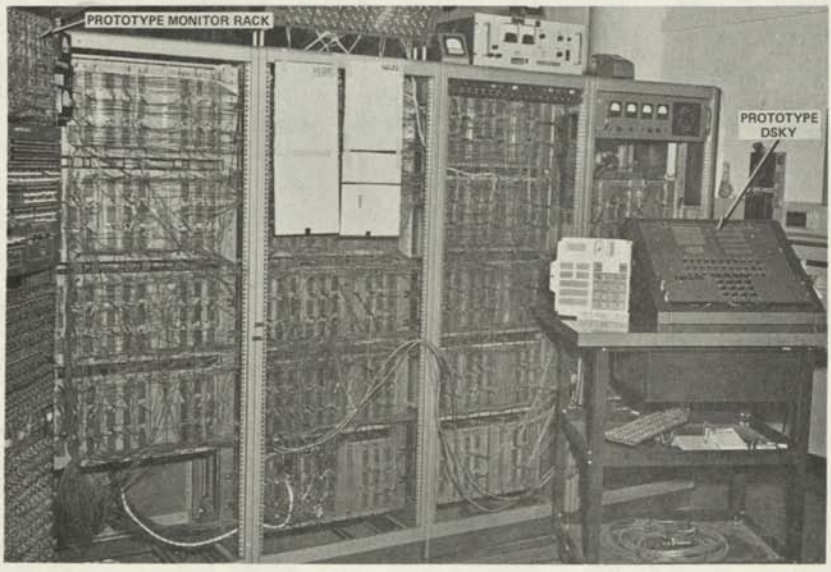


Fig. 1-4 AGC-4 Back Panel View

Accordingly, the APOLLO Block I computer was designed to use integrated circuit logic along the functional lines of, and as a replacement for, the core transistor logic of the AGC3 computer.

The first rack-mounted guidance computer emerged in early 1963 as AGC4 (see Figures 1-3 and 1-4) with integrated circuit logic, fixed rope memory, coincident-current-erasable memory, and discrete-component circuits for the oscillator, power supplies, certain alarms, interface, and memory electronics. The rope memory contained 12,288 words, but this figure was shortly revised upward to 24,576, made possible by designing the rope modules with the eventual expansion in mind. No particular mission-related need for this expansion had been identified other than an uneasiness about the possible insufficiency of the 12K memory. Within a year, when the first mission program requirements had been conceived, documented, and collected, there was increasing concern about the possible insufficiency of the 24K memory, which prompted further expansion of the computer memory by mid-1964 to 36K. Figures 1-3 and 1-4 also show some of the early engineering test equipment, including the early ground support equipment (GSE), the monitor equipment, and display and keyboard (DSKY). The circuit and functional characteristics designed into the AGC-4 computer, together with the DSKY and the GSE, became the basis for the first (or Block I) production version of the APOLLO computer. The term, Block I, was used to distinguish this equipment from that which was developed later, and that was classified as Block II.

The erasable memory contained 1024 words, of which the first 16 locations were reserved for the central and special registers external to the core memory unit. These registers were used as flip-flops in the logic of the computer.

Both the fixed and erasable memories were operated at approximately a 12 microsecond memory cycle time (MCT), which is the time interval between erasable memory accesses. This was quite leisurely for the erasable memory, and allowed for content modification between reading and writing such as incrementing and shifting functions that were needed for memory cycle stealing operations. The rope memory is inherently slow, and was operating a good deal less leisurely. There was no hierarchical distinction between the fixed and erasable memories; both were accessible by any instruction. The instructions writing into memory would of course fail to alter the contents of fixed memory.

The integrated circuit logic section for the Block I computer was designed using a three-input NOR gate as a building block, with one gate in a TO-47 type package. These gates were relatively simple in form, consisting of the equivalent

of three NPN bipolar transistors and four resistors connected as a modified direct-coupled transistor logic or DCTL NOR gate. The arrangement is also referred to, less precisely, as resistor-transistor logic or an RTL NOR gate. The gates were interconnected using a welded nickel ribbon matrix and packaged in modules containing 120 gates and a 140-pin connector. With this method of packaging, thirty-six logic modules were required for the logic functions of the Block I computer. The computer would have required fewer integrated circuit packages if a variety of logic building blocks had been used including, for example, a gated flip-flop. It was estimated, however, that the problem of producing and qualifying even a second circuit type would have outweighed the advantages of a variety of logic types. In retrospect, this estimation was correct. The single logic type simplified packaging, manufacturing, and testing, and gave higher confidence to the reliability predictions.

One other integrated circuit was used in the Block I computer, a differential amplifier for sensing memory outputs. This device was developed especially for the guidance computer, since a differential amplifier capable of memory sensing was not available at that time. The sense amplifier contained the equivalent of six NPN bipolar transistors and eight resistors. Such units were preferred over discrete-component sense amplifiers not only for their small size, but also for the close match of characteristics and tracking desired between components of the differential stage.

As an adjunct to the APOLLO guidance computer, a display and keyboard unit was required as an information interface with the crew. The original design was made during the latter stages of development of the Mod 3C computer, at which time neon numeric indicator tubes of the "Nixie" variety were used to generate three 4-digit displays for information, plus three 2-digit displays for identification. These were the minimum considered necessary, and they provided the capability of displaying three-space vectors with sufficient precision for crew operations. The 2-digit indicators were used to display numeric codes for verbs, nouns, and program numbers. The verb-noun format permitted communication in language with syntax similar to that of spoken language. Examples of verbs were "display", "monitor", "load", and "proceed," and examples of nouns were "time", "gimbal angles", "error indications", and "star identification number." A keyboard was incorporated along with the display to allow the entering of numbers and codes for identifying them. The unit acquired the designation DSKY, for display and keyboard.

The DSKY design incorporated segmented electroluminescent numeric displays instead of neon, and a 5-digit instead of 4-digit display to accommodate a base eight (octal) display of a 15-bit computer word. In the Block I computer, two different physical outlines were generated for the DSKY, one for the navigation station in the

vicinity of the GN&C system (lower equipment bay), and one for the main control panel located above the crew couches. Figure 1-4 shows a main panel DSKY beside an engineering version of the design.

In 1964, when the APOLLO GN&C system underwent redesign for both the command module (CM) and lunar module (LM) spacecraft, the requirements for the current operational version of the APOLLO computer and DSKY were conceived. This version of the computer became known by the various designations AGC, Block II AGC, LGC (lunar module guidance computer), and CMC (command module computer).

As was stated previously, the need was evident for an increased capability in both the fixed and erasable memories over the Block I computer. There were two major reasons for this: one was the experience gained with mission-related programs for Block I, and the other was the identification of new functions for the BLOCK II system, including the ambitious digital autopilot function. Both memory expansions were accommodated with a moderate impact on existing designs. The braid memory, a new form of fixed memory with some similarities to the rope memory but with several potential advantages, was under development for possible inclusion into the Block II computer. But, because braid development was not sufficiently fast for the Block II schedule, the rope memory was retained, with an increase in capacity from 24,596 words to a total of 36,864, a factor of 1.5 over Block I. This was made possible by increasing the number of sense lines in each module. The mechanical design of the rope modules and computer was changed to allow rope removal and insertion without removing the computer from the spacecraft or breaking any connections other than those of the rope modules themselves.

The erasable memory capacity was doubled to 2048 words. This increase was made with a small increase in driving circuitry, twice the number of cores, and an overall volume reduction owing to more efficient space usage and the use of smaller driving transistors.

In the logic design area, the number of input/output operations was greater than in Block I, and the overall speed requirement was similarly greater, both largely due to the incorporation of the autopilot function into the computer. The input/output requirement was met by adding a number of special circuits for such interfaces as the radar and hand controller, and increasing the number of interface circuits used in Block I, such as counter (memory cycle stealing) inputs, and discrete inputs and outputs. The speed requirement was met by reducing the number of memory cycles required to perform some of the instructions and expanding the instruction repertoire. In order to conserve power, the logic circuitry was made slightly slower in the

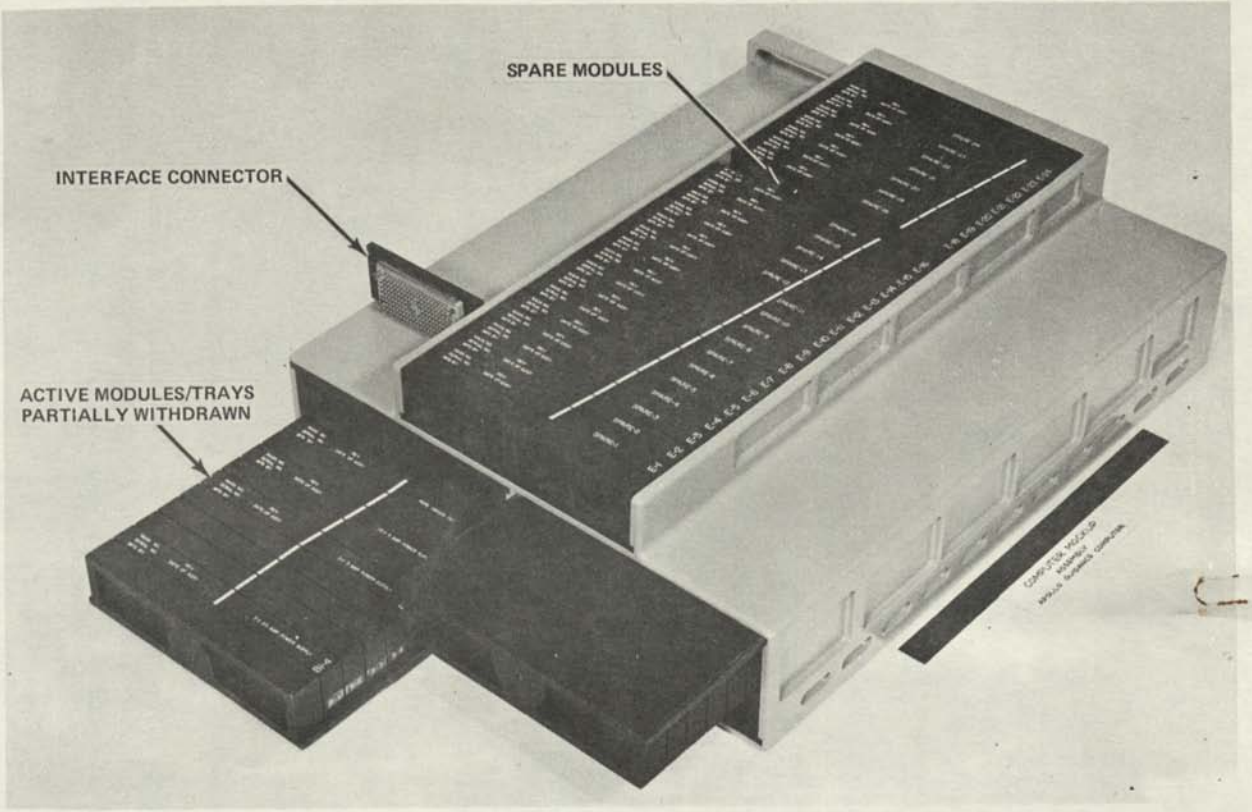


Fig. 1-5 Computer Modular Construction

Block II design because of a change to low power integrated circuit logic. The number of instructions was increased from 11 to 34 to include more flexible branching and data handling, some double-precision capability, and special input/output instructions. Some instructions, including multiply and divide, were made faster through the use of extra logic, particularly in the adder circuitry, where the time to propagate carries was reduced to about one-third its duration in Block I. As a result of these changes, the number of gates rose about 1400, from about 4300 to approximately 5700.

Improvements in integrated circuit technology led to the adoption of the new low power NOR gate for the Block II computer. Though still a modified DCTL three-input gate, this circuit dissipated less than half the power of the Block I gate. Additionally, two of these gates were made on a single silicon chip and mounted in a 10-lead "flatpack" container. This resulted in doubling the packaging density relative to the Block I approach. The power reduction was obtained in these logic units by increasing the output impedance, resulting in an increased rise-time sensitivity to stray wiring capacitances. Two steps were taken to permit the use of the new device and effect a saving in power: one, multilayer etched boards were adopted as the means of interconnection within modules in place of the welded ribbon matrices, the change decreased the stray capacitance, and two, the clock timing circuit was improved to accommodate greater uncertainties in signal propagation delays.

The Block II computer design resulting from the change in technology roughly doubled the speed, raised between 1.5 and 2 times the memory capacity, increased input/output capability, decreased size, and decreased power consumption. The Block II DSKY was similarly redesigned. The functional characteristics of the DSKY were essentially unchanged. The new design that resulted was mainly a smaller mechanical envelope, that was the same for the three locations: two in the command module and one in the lunar module. Additionally, the Block II computer and DSKY were constrained by new mechanical requirements such as the enviromental seal on all connectors or modules that could be subjected to, and damaged by, the high moisture content of the spacecraft.

The mechanical design concepts, derived from experience with welded cordwood construction and other construction techniques, were applied very successfully to the packaging of the POLARIS guidance system and guidance computer. The major APOLLO mission requirements that had significant impact on the early configurations were: the requirements for inflight repair, for mounting on the spacecraft cold plate structure, and for the spacecraft cabling interface. This resulted in a configuration having modular construction and removable trays. Refer to Figure 1-5. The housing also contained a tray with spare modules.

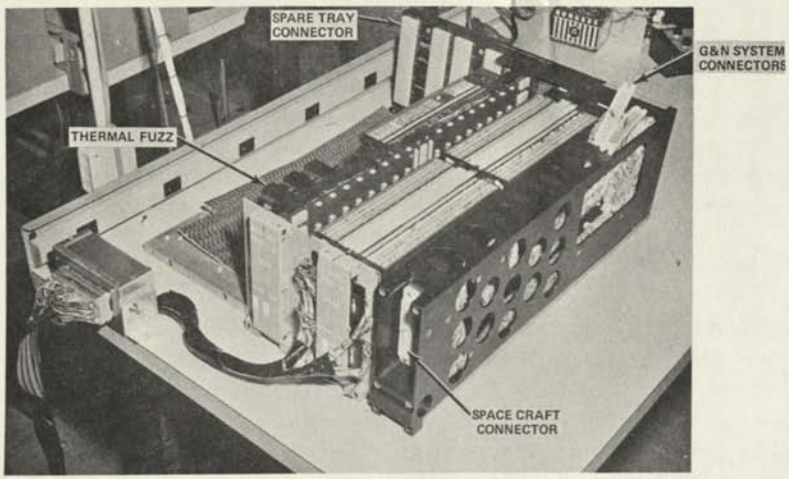


Fig. 1-6 Block 0 Computer

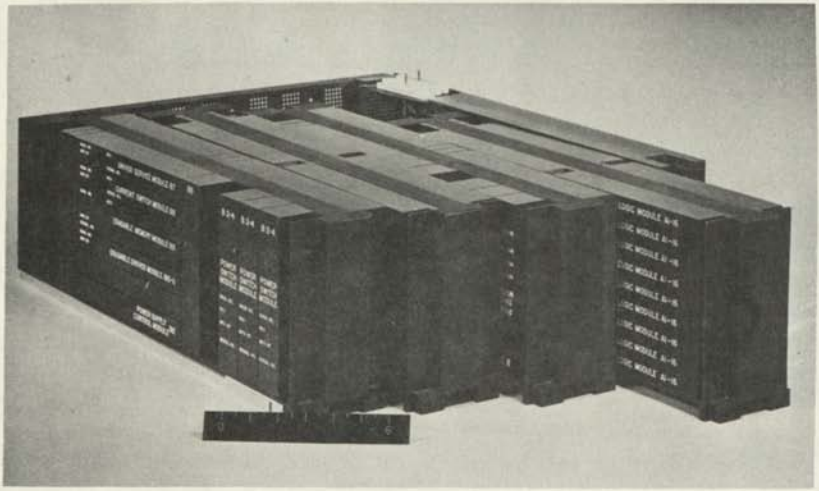


Fig. 1-7 Dual Computer Model

Various mechanical and thermal interface problems dictated a change in configuration to what became the Block 0 computer shown in Figure 1-6. The early production computers were of this configuration. The mechanical design was not stabilized, however, until the requirement for inflight repair yielded to the more important requirement for moisture proofing. This in turn led to a significant change in mechanical configuration using the same module designs that were used in Block 0. Figure 1-7 is a photograph of the design resulting from moisture proofing. This was the final Block I mechanical configuration for the computer and DSKYs.

Before the inflight repair requirement was deleted, there were a series of mechanical designs, coupled with studies, to determine the feasibility of fault isolation in flight or the feasibility of dual computer operation with manual switchover. Figure 1-8 is a mockup of one version that provided one complete set of spare trays. If the problems of switchover could be solved, the spare trays could be interconnected in a dual operating mode. The configuration, shown in Figure 1-6, is the only version of this general configuration that was built with space for two spare trays mounted beside the two active trays. Other constraints on this design included: the right-hand tray containing cabling to interconnect the computer with the rest of the GN&C system through the top and with the spacecraft cabling through the front connectors, and the "thermal fuzz" material to provide heat transfer to the spacecraft cold plate. The change to the final Block I design (Figure 1-7) was accomplished after the requirement for spares and "thermal fuzz" were eliminated.

The increased functional requirements resulting in the Block II design, discussed earlier, also resulted in a new mechanical design. The resulting design of computer and DSKY is illustrated in Figure 1-9. This was a complete mechanical redesign: the end product was not only smaller and lighter than Block I, but also provided better environmental sealing, easy access to fixed memory for replacement in the spacecraft, and commonality between the lunar and command module mounting. Internally, the same type of modular construction was used as in the Block I version.

1.2 INITIAL DECISIONS

Many of the initial decisions surrounding the computer development could be easily categorized by the single statement that the task was seriously underestimated. The realization that reliability, weight, volume, and power consumption would be critical factors governing the success of the missions made for a limited view of the computational requirements. The result was excessive emphasis on minimizing the functional capacity of the computer. In addition, program costs, scheduling requirements, and the state-of-the-art of components constrained the initial

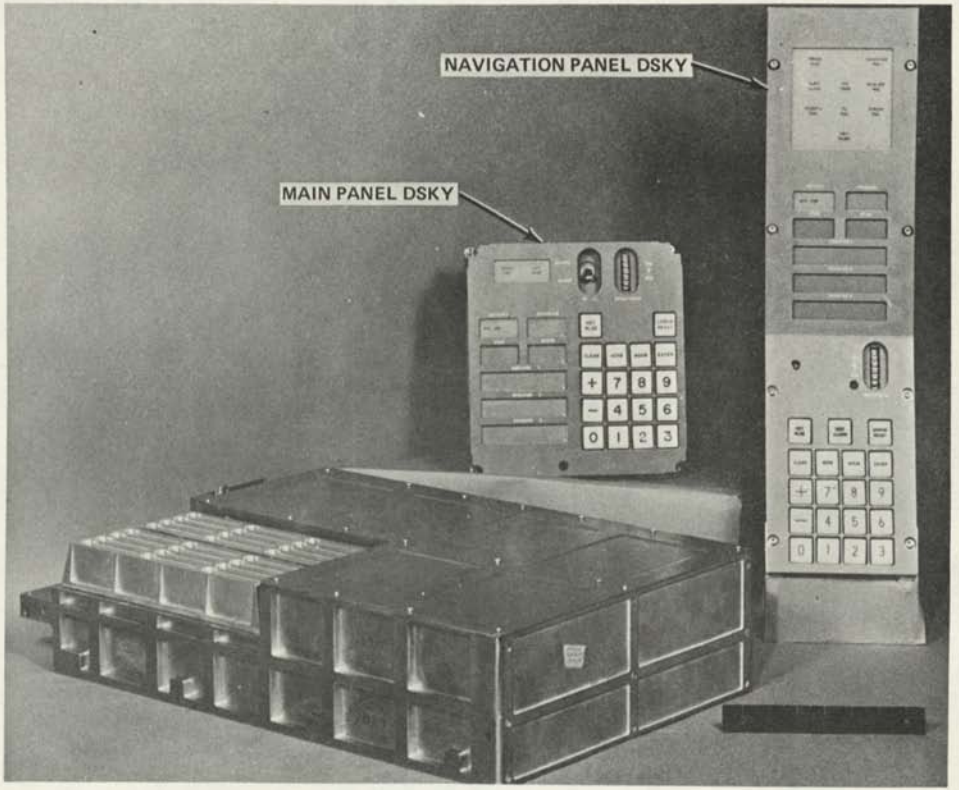


Fig. 1-8 Block I Computer and DSKYs

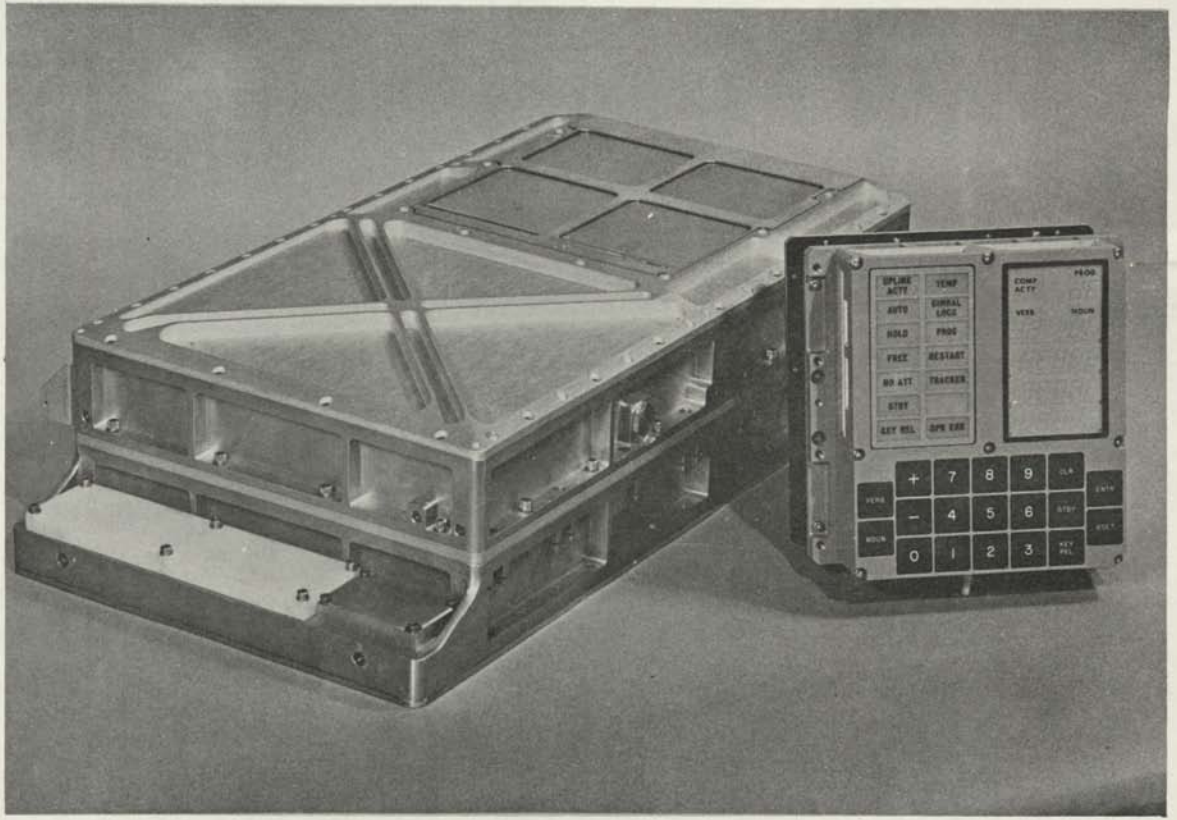


Fig. 1-9 Block II Computer and DSKY

considerations for adequate functional capacity. Also, in the light of the early limited knowledge of the mission requirements, it would have been impossible to justify a computer with the capacity and features of the Block II configuration.

Power, weight, and volume factors were prime considerations in determining word length, memory size, instructions, and interfaces. A word length of 15 bits, for example, was considered sufficient for most arithmetic and control operations. Double-precision computation (28 bits) was sufficient to provide the accuracy for navigation computations. Memory size necessitated consideration of memory type. For scratch pad memories, the only suitable candidate was the coincident-current ferrite-core type, and since these were costly in terms of power consumption and volume, it was important to minimize both the number and length of words. The problem of size and power consumption, as well as the reliability of data storage, was strong justification for using a fixed memory and resulted in the proposal for a wired rope type memory. An erasable capacity of 512 words was considered very marginal; therefore, the capacity was increased to 1024 words very early in the design process. The design philosophy of limited capacity was followed by the use of a minimum number of simple instructions, a limited interface capacity, as well as a limited memory capacity. This philosophy was a logical balance between computing capacity, the computer's mechanical limitations and component state-of-the-art.

In consideration of the control and display capability, there was a great deal of interest in an oscilloscope-type display subsystem, but the complexity of the electronics and the complexity of the computer required to format the data were sufficient to eliminate this concept. An alternative approach resulted in the DSKY design that was considered an absolute minimum for the lunar mission. Other spacecraft display and control interfaces with the computer were ruled out in Block I design for various reasons. Typically, they involved the whole question of reliable operation of a digital computer. The philosophy of limited control and display through the computer restricted the flexibility of the GN&C system. While the Block II design provided expansion in the control and display area, even this later configuration was limited by these earlier decisions.

One of the major early decisions, that greatly influenced design and computer operation, was the provision for inflight repair. To some extent, this decision affected the design of both the display hardware and software, since inflight repair implies the ability to detect and diagnose a failure. Inflight repair was later eliminated as a requirement, but the failure detection and software restart capability remained.

Additional decisions, considered to be major contributions to the success of the program, were as follows:

1. Hardware provisions for an interrupt structure, with software containing an executive, wait list, and an interpreter to aid real-time computation for guidance and control.
2. Hardware provisions for a limited but powerful instruction set, 1's complement arithmetic, and the ability to perform multiple precision arithmetic simply.
3. Hardware provisions for uplink, downlink, and compatible onboard displays for relatively simple man-machine communications, both onboard and on the ground.
4. Emphasis upon "make it right" to obtain reliability, rather than upon redundancy at the component level.
5. Use of mechanical packaging techniques of proven manufacturability and reliability.

1.3 EXPANDING REQUIREMENTS

The following is a partial listing of mission type requirements that had an impact on computer designs. The time period during which the requirement was in a state of flux is indicated.

1.3.1 Memory Size (1962 to 1969)

During the entire program there was concern over the limited memory size. The debate included the ratio between fixed and erasable memory. At various points in time during the software development there were requests for expansion of the fixed memory. At other times the request was to expand the erasable memory. There was continuous evaluation of the hardware and supporting hardware developments designed to expand either fixed or erasable. Partially due to the lack of a firm requirement for expansion in memory but also due to the available memory technology, there were very few attempts to expand the memory beyond the Block II capacity. The latest design exercises occurred in early 1969 with a proposed expansion of the erasable memory from 2K to 16K words. The proposal was rejected because of insufficient justification. This was the only proposed increase in memory since the changes introduced in the Block II design, but it was fairly clear that the requirement for the additional memory was not firm enough to justify the change. The other changes in memory size came early enough in the development to make the justification somewhat simpler. In general, the memory sizing was determined

more by what the hardware technology could provide rather than a requirement established for the mission.

1.3.2 Implementation Meetings - 1964

This was a series of meetings between MSC, MIT, and the two spacecraft contractors which were to determine the system requirements. The output of these meetings resulted in the determination of many new computer requirements that lead to the change from Block I to II. The major changes affecting the computer were expanded functional requirements such as the digital autopilot, increased mode control, definition of radar interface, hand controller interfaces, hermetic packages for moisture-proofing, compatible lighting between spacecraft and DSKY, and other human factor requirements.

These new requirements justified the increase in memory size, increase in interface capacity, increase in overall functional capability, and the introduction of the constraints on the DSKY that lead to the Block II universal design.

Another result of the implementation meetings was a definition of the philosophy for guidance system backup and redundancy. This philosophy provided for operating modes with independent means for guidance and control if the guidance system failed. In the case of the command module, the ground control would provide the capability of safe return to earth and, in the case of the lunar module, there was a separate guidance system to provide abort capability. With these guidance backup modes it was no longer necessary to have inflight repair or a redundant computer in order to meet the crew safety requirements of guidance. The relaxation of the requirement to provide backup to the computer within the guidance system, and the introduction of moisture-proofing provided a definite guide to mechanical configuration of the Block II.

1.3.3 Weight Savings (1962 to 1965)

A concerted effort to save weight was instrumental in initiating many design studies that resulted in changing the Block II computer frame design from aluminum to magnesium. The design change saved about 12 pounds. In parallel with, and partially resulting from the change to magnesium, was an increased concern about materials in general. The magnesium required a corrosive-resistant finish that would meet the contamination and corrosion-resistance requirements of the spacecraft.

1.3.4 Electromagnetic Interference Specifications - 1963

The contract requirement for the system to meet EMI specifications was added late in 1963. The Block I design at that point was well along and had considered some of the requirements. It was quite well known that a strict interpretation of the requirements could not be met by the computer. After the specification was added as a contractual requirement, the design impact was evaluated, and Block II was designed to meet or exceed the requirements. Since digital computers are quite susceptible to errors induced by external electromagnetic interference, both designs had features which would reduce the sensitivity to interference.

1.3.5 Fire Proofing - 1967

The requirement to investigate and fire-proof the system resulted from the APOLLO 4 accident. This added a new dimension to the materials problem but had no impact on the computer design. The DSKY design was changed to provide a cover for the alarm panel and fire proof caps on the keys, since the material in the panel and key caps were flammable.

SECTION 2.0
COMPUTER SYSTEM DESCRIPTION

2.1 INTRODUCTION

The computer subsystem is the control and processing center of the guidance, navigation and control system. It processes data and issues discrete output and control pulses to the guidance system and other spacecraft systems. Functionally, the computer subsystem consists of the APOLLO guidance computer (AGC) and the display and keyboard unit (DSKY). The computer is a parallel digital real time control computer with many features of a general purpose computer. As a control computer, some of the major control functions are: alignment of the inertial measurement unit, processing of radar data, management of astronaut display and controls and generation of commands for spacecraft engine control. As a general purpose computer, the AGC solves the guidance and navigation equations required for the lunar mission.

An operational APOLLO spacecraft contains two guidance computers and three DSKYs, with one computer in the command module, and one in the lunar module. The computers are electrically identical, but differ in the use of computer software. There are two DSKY units in the command module, a DSKY mounted on the main display and control panel and a DSKY in the lower equipment bay at the navigation station. (See Figures 2-1 and 2-2). The lunar module has a single DSKY unit. (See Figure 2-3). In this chapter, because it is concerned primarily with the hardware aspects of the APOLLO guidance computer, no distinction is made between the command module and lunar module computers, except where significant operational differences exist. References to the "APOLLO guidance computer", the "computer", and the "AGC" are equivalent in meaning. Other project APOLLO documentation may refer to the command module computer (CMC), the lunar module computer (LGC). These designations are not used in this report unless specifically defined.

Section 1.0 described the evolution of the Block II AGC configuration through a series of technological developments, including the Block I prototype design, and also examination of the requirements. Since the Block II design was the only configuration used for manned operational flights, the following functional description concentrates on the Block II configuration, but identifies the more significant functional differences in the Block I design. The Block II DSKY was functionally equivalent but mechanically different from Block I. Table 2-I compares the functional characteristics of the two computers.

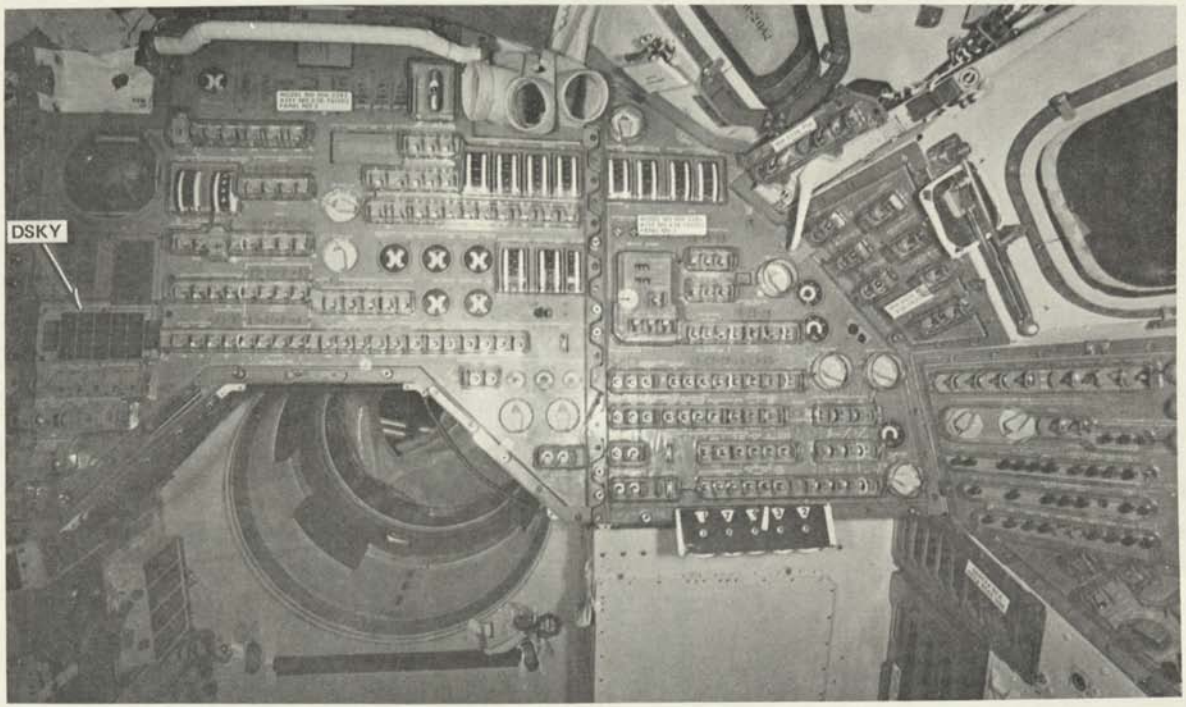


Fig. 2-1 CM Main Display and Control Panel

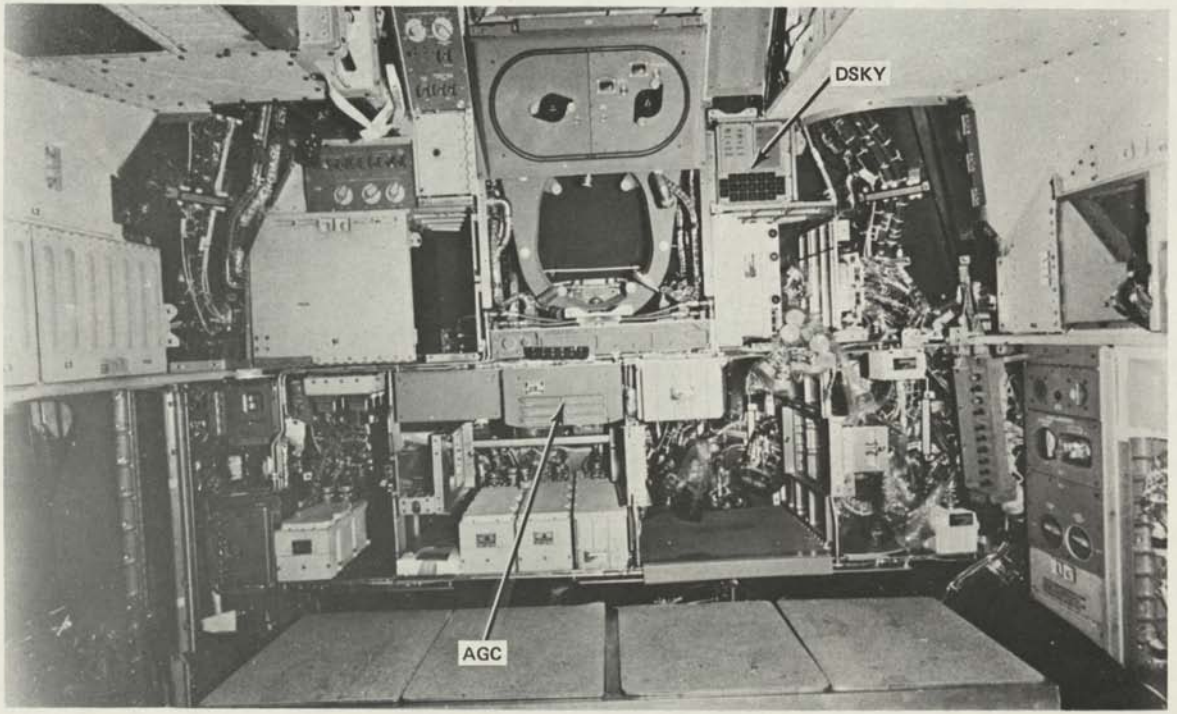


Fig. 2-2 CM Lower Equipment Bay

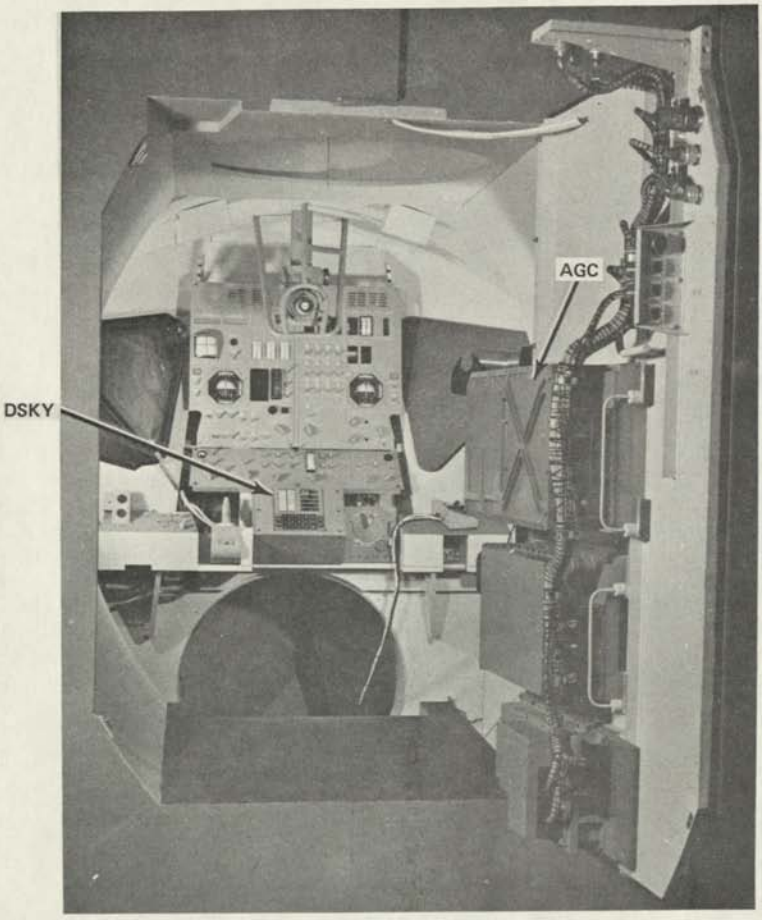


Fig. 2-3 LM Display and Control Panel

TABLE 2-I

COMPUTER CHARACTERISTICS

PERFORMANCE CHARACTERISTICS	BLOCK I	BLOCK II
Word Length	15 Bits + Parity	15 Bits + Parity
Number System	One's Complement	One's Complement
Fixed Memory Registers	24,576 Words	36,864 Words
Erasable Memory Registers	1,024 Words	2,048 Words
Number of Normal Instructions	11	34
Number of Involuntary Instructions (Interrupt, Increment, etc.)	8	10
Number of Interrupt Options	5	10
Number of Counters	20	29
Number of Interface Circuits	143	227
Memory Cycle Time	11.7μsec	11.7μsec
Counter Increment Time	11.7μsec	11.7μsec
Addition Time	23.4μsec	23.4μsec
Multiplication Time	117 μsec	46.8μsec
Double Precision Addition Time	Subroutine (1.65 millisec)	35.1μsec
Number of Logic Gates (Microcircuits)	4,100	5,600 (2,800 pkgs)
Volume	1.21 cubic ft	0.97 cubic ft
Weight	87 pounds	70 pounds
Power Consumption	85 watts	55 watts

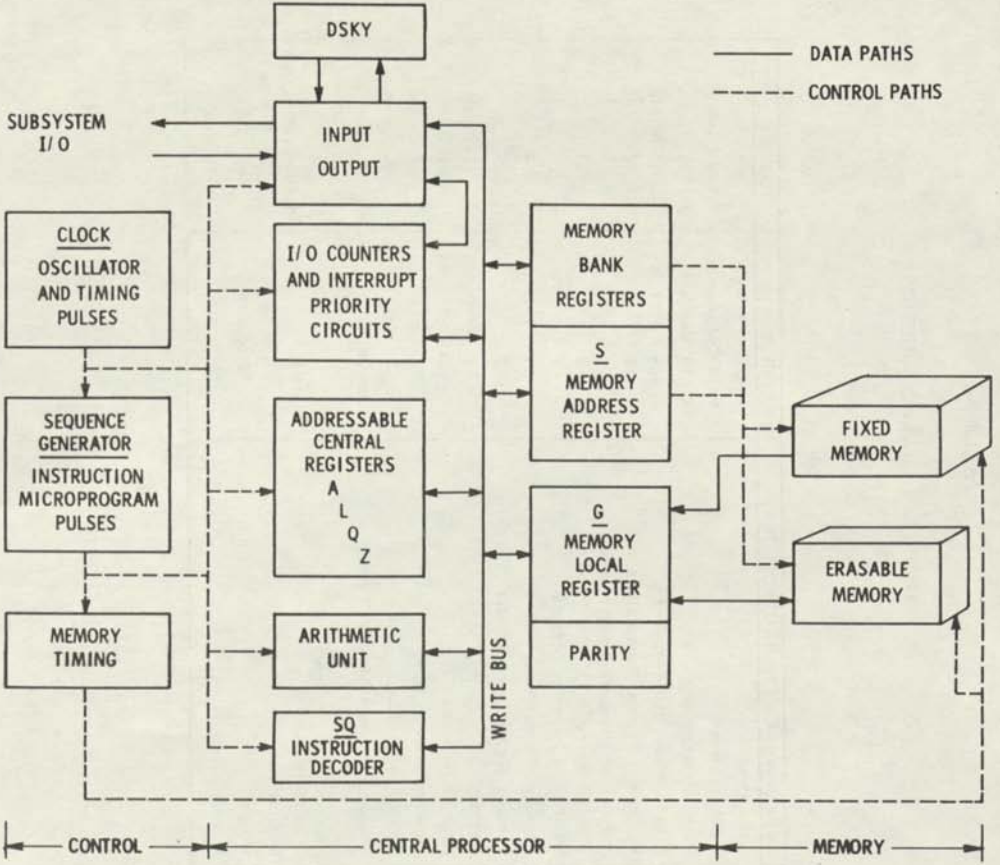


Fig. 2-4 Computer Simplified Block Diagram

2.2 GENERAL FUNCTIONAL DESCRIPTION

2.2.1 Computer

The computer has three principal sections, a memory, a central processor, and a control section. Refer to Figure 2-4. The memory section is divided into two portions, fixed and erasable. The fixed, or read-only, portion has 36,864 words and erasable portion has 2048 words. The central processor section includes an adder, an instruction decoder, interrupt control, a memory address decoder, a memory local buffer register, and a number of addressable registers with either special features or special use. The backbone of the computer is the set of 16 write buses that are the means for transferring information between the various registers of the central section shown in Figure 2-4. The arrow-heads indicate possible direction of information flow between various registers of the central processor. The S register stores the memory address. The G register provides a buffer for the memory data. The SQ register bears the same relation to instruction as the S register bears to memory addresses. The central and special registers (A, L, Q, Z, etc) and a set of input/output channel registers are addressable explicitly. Their properties are shown in Table 2-II. The arithmetic unit is an adder with shifting gates, temporary storage, and control logic. The priority circuits control the incrementing or shifting of a set of erasable memory registers and control the program interruption for a number of involuntary sequences. The control section includes the timing and sequencing logic. The clock provides the primary time reference for all spacecraft systems and through a countdown chain, timing pulses at various frequencies for computer internal and external usage. The sequence generator provides the basic memory timing and the sequences of control pulses (microprograms) which constitute the instruction that the central processor can perform.

2.2.2 Display and Keyboard

Essentially the Display and Keyboard (DSKY) was designed to permit exchange of digital information between the subsystem operator and the computer. Most basic functional capabilities were determined at the time the DSKY was designed. These functions relate the method or language of communication between the computer and operator.

The DSKY (see Figure 2-5) contains a small keyboard and a set of display elements. The keyboard provides an interface with the computer. It causes program interrupt and provides a five bit code each time a key is depressed. The computer program

TABLE 2-II

ADDRESSABLE CENTRAL AND SPECIAL REGISTERS

REGISTER	OCTAL ADDRESS	PURPOSE
A	0000	Central accumulator. Most instructions refer to A.
L	0001	Lower accumulator. Used in multiply, divide, and all double-precision operations.
Q	0002	Return address register. If a transfer control (TC) operation occurred at line L, (Q) = L = 1.
EB	0003	Erasable bank register, bits 9, 10, 11.
FB	0004	Fixed bank register, bits 11, 12, 13, 14, 15.
Z	0005	Program counter. Contains L + 1, where L is the address of the instruction presently being executed.
BB	0006	Both bank registers: Erasable, bits 1, 2, 3. Fixed, bits 11, 12, 13, 14, 15.
--	0007	Contains zero.

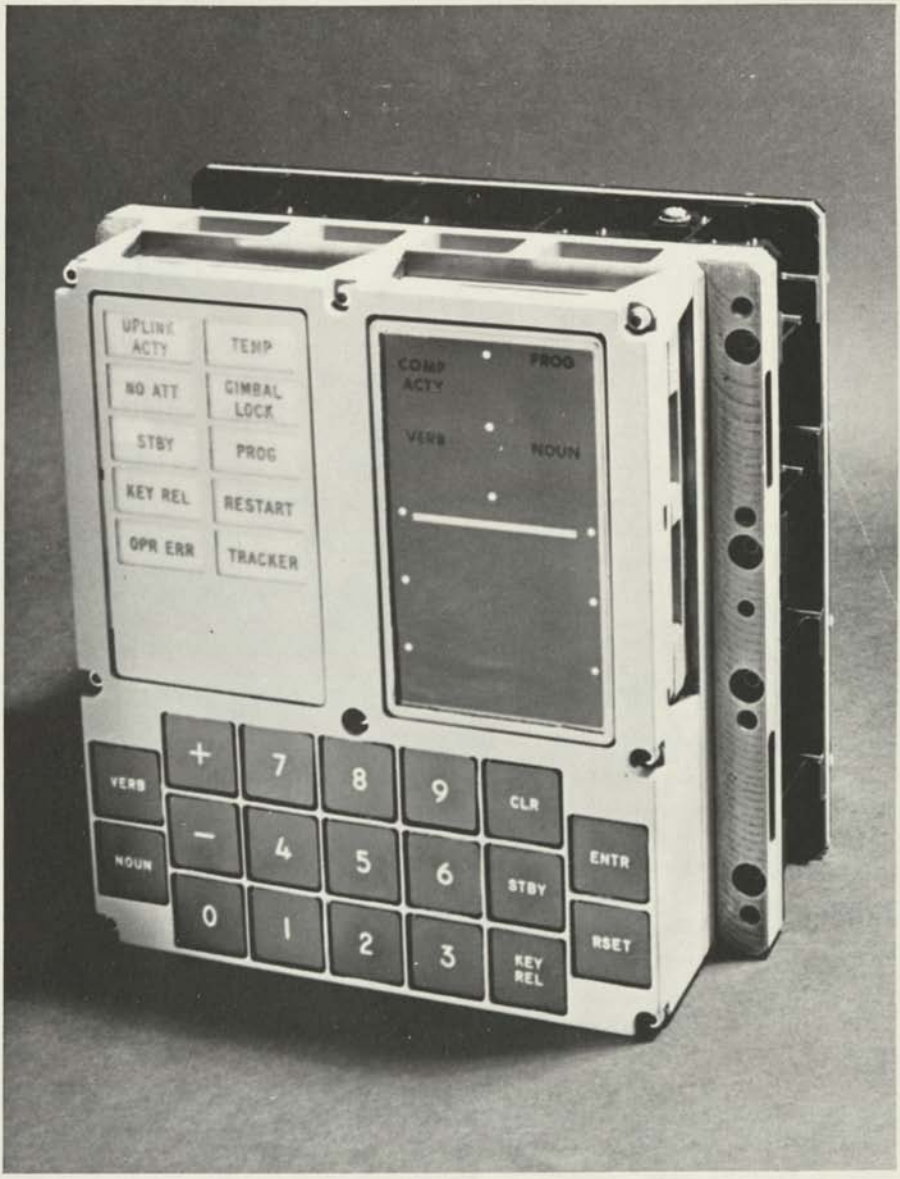


Fig. 2-5 DSKY Front Panel

processes the key code according to a predetermined set of routines and displays the results in the registers of the display panel. The display contains a set of discrete status or caution elements and a set of numeric registers for data. The display elements are switched on or off according to coded commands from the AGC. The AGC code is decoded, and the corresponding display elements are turned on or off using a matrix of miniature latching relays contained in the DSKY.

2.3 SUBSYSTEM CHARACTERISTIC

The computer subsystem has several unique features that have contributed to the computer's success in meeting the general requirements for high performance in terms of mathematical computations and interface control and yet maintain high reliability, low weight, volume, and power consumption.

2.3.1 Word Length

The computer is a "common storage" machine. This means that instructions may be executed from erasable memory as well as from fixed memory, and that data (obviously constants, in the case of fixed memory) may be stored in either memory. The word sizes of both types of memory must be compatible in some sense. The easiest solution was to make equal word lengths of 16 bits. The AGC is somewhat unique in its very short word length, and the reasons for it are of some interest. The principal factors in the choice of word length in the computer were:

A. Minimize power and volume.
B. Provide the precision desired in the representation of navigational variables and data.
C. Provide the word length necessary for the input variables that are entered serially or incrementally.
D. Provide for the instruction word format. Division of instruction words into two fields, one for operation and one for address.

The data words used in the computer may be divided roughly into two classes: one, data words used in elaborate navigational computations, and two, data words used in the control of various appliances in the system. Initial estimates of the precision required by the first class ranged from 27 to 32 bits. The second class of variables could almost always be represented with 15 bits. The fact that navigational variables require about twice the desired 15 bit word length means that there is not much advantage to word sizes between 15 and 28 bits, as far as precision of representation

of variables is concerned, because double-precision numbers must be used in any event. Because of the doubly signed number representation for double-precision words, the equivalent computer word length is 29 bits (including sign), rather than 30, for a basic word length of 15 bits. See Figure 2-6. It was also estimated that a significant portion of the computing had to do with data words required for control, telemetry and display activities, all of which can be handled more economically with short words. A short word allows faster and more efficient use of erasable storage because it reduces fractional word operations, such as packing and editing; it also means a more efficient encoding of small integers.

As a control computer, the AGC must make analog-to-digital conversions, many being those of angles. Two principal forms of conversion are possible: one of which renders a whole number, the other produces a train of pulses that must be counted to yield the desired number. The latter type of conversion is employed by the computer using the counter incrementing feature. When the number of bits of precision required by the conversion exceeds the word length, a double precision word may be formed by programmed scanning of the counter register or by counting the overflows in another counter-register. Whether programmed scanning is feasible depends largely on how frequently this scanning must be done. The cost of using an extra counter register is directly measured in terms of the priority circuit associated with it. In the AGC, the equipment saved by reducing the word length below 15 bits would probably not match the additional expense incurred in double-precision extension of many input variables. The question is academic, however, since a lower bound on the word length is effectively placed by the format of the instruction word.

The instruction word consists of an operation code of 3 bits and a single address of 12 bits. The 8 instruction order codes, provided by the 3-bit field, are augmented by the various special registers to expand the order code and provide operations such as shift right, cycle left, edit, so that a transfer in and out of one of these registers would accomplish actions normally specified by additional order codes. These registers were considered to be more economical than the corresponding instruction decoding and control pulse sequence generation. Hence the 3 bits assigned to the order code were considered adequate, albeit not generous. The address field of 12 bits presented a different problem. At the time of the initial design, it was estimated that 4000 words would satisfy the storage requirements. When the memory was to be expanded, the question then became whether the proposed extension of the address field by a bank register was more economical than the addition of bits to the word length. For reasons of modularity of equipment, adding more bits to the word length would result in adding more bits to all the central registers. This

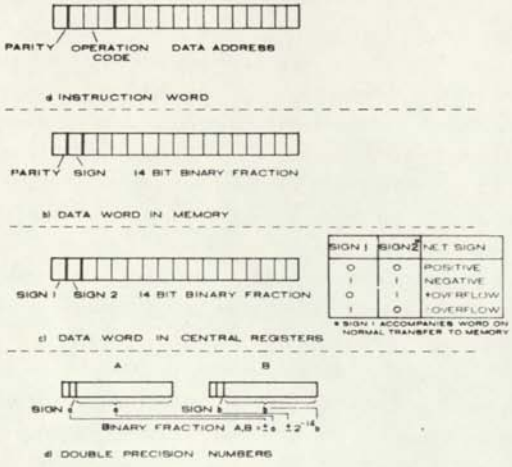


Fig. 2-6 Word Formats

amounts to increasing the size of the non-memory portion of the computer. Hence the address field was expanded by the provision of memory bank register.

In summary, the 15-bit word length seemed practical enough, so that the additional cost of extra bits in terms of size, weight, and reliability did not seem warranted. A 14-bit word length was thought impractical because of the problems with certain input variables, and it would further restrict the already cramped instruction word format. Word lengths of 17 and 18 bits would result in certain conceptual simplicities in the decoding of instructions and addresses, but would not help in the representation of navigational variables. These require 28 bits, so they must be represented in double precision in any event.

2.3.2 Number Representation

In the absence of the need to represent numbers of both signs, the discussion of number representation would not extend beyond the fact that numbers in the computer are expressed to base two. But the accommodation of both positive and negative numbers requires that the logical designer choose among at least 3 possible forms of binary arithmetic. These 3 principal alternatives are: one's complement, two's complement, and sign and magnitude.

In one's complement arithmetic, the sign of a number is reversed by complementing every digit where "end around carry" is required in addition of two numbers. In two's complement arithmetic, sign reversal is effected by complementing each bit and adding a low order one, or some equivalent operation. Sign and magnitude representation is typically used where direct human interrogation of memory is desired, as in "postmortem" memory dumps, for example. The addition of numbers of opposite sign requires either one's or two's complementation or comparison of magnitude, and sometimes both may be used. The one's complement notation has the advantage of having easy sign reversal, this being equivalent to Boolean complementation. Hence a single machine instruction performs both functions. Zero is ambiguously represented by all zero's and by all one's, so that the number of numerical states in an n-bit word is $2^n - 1$. Two's complement arithmetic is advantageous where end around carry is difficult to mechanize, as is particularly true in serial computers. An n-bit word has 2^n states, which is desirable for input conversions from such devices as pattern generators, geared encoders, or binary scalers. Sign reversal is awkward, however, since a full adddition is required in the process. These considerations led to the use of a one's complement number system in the AGC.

		STANDARD					MODIFIED					
		S_1	4	3	2	1	S_1	S_2	4	3	2	1
EXAMPLE 1:	Both operands positive; Sum positive, no overflow. Identical results in both systems.	0	0	0	0	1	0	0	0	0	0	1
		0	0	0	1	1	0	0	0	0	1	1
		0	0	1	0	0	0	0	0	1	0	0
EXAMPLE 2:	Both operands positive; positive overflow. Standard result is negative; Modified result is positive using S_2 as sign of the answer. Positive overflow indicated by $S_1 \cdot S_2$.	0	1	0	0	1	0	0	1	0	0	1
		0	1	0	1	1	0	0	1	0	1	1
		1	0	1	0	0	0	1	0	1	0	0
EXAMPLE 3:	Both operands negative; Sum negative no overflow. End around carry occurs. Identical results in both systems using either S_1 or S_2 as the sign of the answer.	1	1	1	1	0	1	1	1	1	1	0
		1	1	1	0	0	1	1	1	1	0	0
		1	1	0	1	0	1	1	1	0	1	0
						1 carry						1 carry
		1	1	0	1	1	1	1	1	0	1	1
EXAMPLE 4:	Both operands negative; negative overflow. Standard result is positive; modified result is negative using S_2 as the sign of the answer. Negative overflow indicated by $S_1 \cdot S_2$.	1	0	1	1	0	1	1	0	1	1	0
		1	0	1	0	0	1	1	0	1	0	0
		0	1	0	1	0	1	0	1	0	1	0
						1 carry						1 carry
		0	1	0	1	1	1	0	1	0	1	1
EXAMPLE 5:	Operands have opposite sign: Sum positive. Identical results in both systems.	1	1	1	1	0	1	1	1	1	1	0
		0	0	0	1	1	0	0	0	0	0	1
		0	0	0	0	1	0	0	0	0	0	1
						1 carry						1 carry
		0	0	0	1	0	0	0	0	0	1	0
EXAMPLE 6:	Operands have opposite sign; sum negative. Identical results in both systems.	1	1	1	0	0	1	1	1	1	0	0
		0	0	0	0	1	0	0	0	0	0	1
		1	1	1	0	1	1	1	1	1	0	1

Fig. 2-7 Illustrative Examples of Properties of Modified One's Complement

In a standard one's complement adder, overflow is detected by examining carries into and out of the sign position. These overflow indications must be "caught on the fly" and stored separately if they are to be acted upon later. The number system adopted in the computer has the advantage of being a one's complement system with the additional feature of having a static indication of overflow. The implementation of this modified one's complement system depends on the computer not using a parity bit in most central registers. Because of certain modular advantages, 16, rather than 15, columns are available in all of the central registers, including the adder. Where the parity bit is not required, the extra bit position is used as an extra column. The virtue of the 16-bit adder is that the overflow of a 15-bit sum is readily detectable upon examination of the two high order bits of the sum (see Figure 2-7). If both of these bits are the same, there is no overflow. If they are different, overflow has occurred with the sign of the highest order bit.

The interface between the 16-bit adder and the 15-bit memory is arranged, so that the sign bit of a word coming from memory enters both of the two high order adder columns. These are denoted S_2 and S_1, since they both have the significance of sign bits. When a word is transferred to memory, only one of these two signs can be stored. In the AGC the S_1 bit is stored. This is the standard one's complement sign except when overflow occurs during addition, in which case it is the sign of the result of the two operands. This preservation of sign on overflow is an important asset in dealing with carries between component words of multiple-precision numbers.

2.3.3 Multiple-Precision Arithmetic

A short word computer can be effective only if the multiple-precision routines are efficient relative to their share of the computer's work load. In the AGC'S application there is enough use for multiple-precision arithmetic to warrant consideration in the choice of number system and in the organization of the instruction set. A variety of formats for multiple-precision representation are possible. Probably the most common of these is the identical sign representation where the sign bits of all component words agree. The method used in the AGC allows the signs of the components to be different.

Independent signs arise naturally in multiple-precision addition and subraction, and the identical sign representation is costly because sign reconciliation is required after every operation. For example, 64 - 46 = 18, or in double-precision notation, (+6, +4) + (-4, -6) = (+2, -2). Here, since signs are mixed, we reconcile signs by adding 10 to lower order number and subtracting 1 from higher order number to obtain the result, (+1, +8). Since addition and subtraction are the most frequent

operations, it is economical to store the result as it occurs and reconcile signs only when necessary. When overflow occurs in the addition of two components, a one with the sign of the overflow is carried to the addition of the next higher components. The sum that overflowed retains the sign of its operand. This overflow is termed an interflow to distinguish it from an overflow that arises when the maximum multiple-precision number is exceeded.

For triple and higher orders of precision, multiplication and division become excessively complex, unlike addition and subtraction where the complexity is only linear with the order of precision. However, APOLLO programs do not require greater than double- precision multiplication and division. The algorithm for double-precision multiplication is directly applicable to numbers in the independent sign notation. The treatment of the interflow is simplified by a double-precision add instruction. Double-precision division is exceptional in that the independent sign notation may not be used. Both operands must be made positive in identical sign form, and the divisor normalized so that the left-most non-sign bit is one. A few triple-precision quantities are used in the AGC. These are added and subtracted using independent sign notation, with interflow and overflow features the same as those used for double-precision arithmetic.

2.3.4 Instruction Set

The major goals in the computer were efficient use of memory, reasonable speed of computing, potential for elegant programming, efficient multiple-precision arithmetic, efficient processing of input and output, and reasonable simplicity of the sequence generator. The constraints affecting the order code as a whole were the word length, one's complement notation, parallel data transfer, and the characteristics of the editing registers. The following rules governing the design of instructions arose from these goals and constraints: three bits of an instruction word are devoted to operation code, address modification must be convenient and efficient, there should be a multiple instruction yielding a double length product, facility for multiple precision must be available, and a Boolean combinatorial operation should be available. These rules are by no means complete, but give a good indication of what kind instruction set was desired.

The three bits reserved for instruction codes are capable of rendering a selection among eight operations with no further refinement. Two techniques are employed in the computer to expand the number of operations 4-fold. These are called "extension" and "partial codes" respectively. Extension is like using a teletype shift code. When an extend instruction occurs, it signifies that the next instruction

code in sequence is to be interpreted otherwise than normally. By this means, the instruction set could be expanded almost indefinitely at a penalty in speed, for a memory cycle time is required for each extension. In the computer the size of the instruction set is doubled by an Extend operation, that calls forth the less-often used instructions. For example, code 000 selects the Transfer Control instruction unless it is preceded by an Extend, in which case it selects an Input-Output instruction. Table 2-III lists the classes of instruction, and how they are controlled.

Partial codes are instruction codes that encroach upon the address field. This technique capitalizes upon the essential difference between fixed and erasable memory. More specifically, a wider variety of instructions, applicable to register, are not fully applicable for fixed memory. Thus, operation code 101 for addresses 0 through 1777 (octal) selects the Index instruction for the erasable memory, whose address field is also 0 through 1777 (octal). The same operation code for addresses 2000-3777 (octal) selects a Double Exchange instruction for erasable memory, whose addresses are obtained by reducing the address module 2000 (octal). In a similar way, the Transfer to Storage instruction is selected by the same code for addresses 4000-5777 (octal), and the Exchange A instruction for addresses 6000-7777 (octal), both being for erasable memory. Alternatively, the entire fixed memory field may select a different instruction for fixed memory, or else the same instruction may be selected over the entire address field.

Table 2-IV (Parts 1 and 2) lists the normal AGC instructions. These include capability for double-precision data handling and addition. Many of these instructions are similar to one another and share microprogram steps.

Input and output are handled to a large extent by special registers called channels, that are not accessible through the regular address field. In the version of the AGC prior to the present one, this was not true; the input and output registers were addressable for any instruction. Here, the channels are accessible by the input-output instructions alone. A slight extra degree of freedom is provided by making the lower accumulator (L) and return address (Q) registers accessible through channels 1 and 2 as well as through regular addresses 1 and 2. This is primarily to allow the programmer to take advantage of the or and exclusive or input-output instructions.

The remainder of the computer instructions are involuntary or address dependent, and are listed in Table 2-V. The last four (B5 - B8) are not really instructions, but are rather editing operations on all words written into the specified four addresses. They are tabulated as instructions only because such operations have instruction status in most computers.

TABLE 2-III

INSTRUCTION CLASSES

CLASS	TYPE	CONTROL
Regular	Basic	Program
	Extracode	
	Channel	
	Special	
Involuntary	Interrupt	Priority
	Counter	
Ground Support	Keyboard	Operator
	Tape	

TABLE 2-IV

NORMAL INSTRUCTIONS

		CODE	TIME	MEMORY LOCATION
A.	SEQUENCE CHANGING			
1.	TRANSFER CONTROL, SET RETURN ADDRESS	TC	1 MCT	ALL
2.	TRANSFER CONTROL ONLY	TCF	1 MCT	FIXED
3.	FOUR WAY SKIP & DIMINISH BY ONE	CCS	2 MCT	ERASABLE
*4.	BRANCH ON ZERO	BZF	1 or 2	FIXED
*5.	BRANCH ON ZERO OR MINUS	BZMF	1 or 2	FIXED
*6.	FORCE INTERRUPT	EDRUPT	3 MCT	SPECIAL SUBSET
B.	FETCHING AND STORING			
1.	CLEAR & ADD TO ACCUMULATOR, A	CA	2 MCT	ALL
2.	CLEAR & SUBTRACT FROM ACCUMULATOR, A	CS	2 MCT	ALL
*3.	DOUBLE CLEAR & ADD TO A & LOWER ACCUMULATOR, L	DCA	3 MCT	ALL
*4.	DOUBLE CLEAR & SUBTRACT FROM A AND L	DCS	3 MCT	ALL
5.	TRANSFER TO STORAGE	TS	2 MCT	ERASABLE
6.	EXCHANGE A WITH STORAGE	XCH	2 MCT	ERASABLE
7.	DOUBLE EXCHANGE A & L WITH STORAGE	DXCH	3 MCT	ERASABLE
8.	EXCHANGE L WITH STORAGE	LXCH	2 MCT	ERASABLE
*9.	EXCHANGE Q WITH STORAGE	QXCH	2 MCT	ERASABLE

*Each require EXTEND Instruction Proceeding (See Table 2-V)
MCT = Memory Cycle Time

TABLE 2-IV (Cont)

NORMAL INSTRUCTIONS

		CODE	TIME	MEMORY LOCATION
C.	INSTRUCTION MODIFICATION			
1.	INDEX (ADD TO NEXT INSTRUCTION)	NDX	2 MCT	ERASABLE
*2.	INDEX AND EXTEND	NDX	2 MCT	ALL
D.	ARITHMETIC AND LOGIC			
1.	ADD TO A	AD	2 MCT	ALL
*2.	SUBTRACT FROM A	SU	2 MCT	ERASABLE
3.	ADD TO STORAGE & A	ADS	2 MCT	ERASABLE
*4.	MODULAR SUBTRACT FROM A (MIXED NUMBER SYSTEM)	MSU	2 MCT	ERASABLE
5.	ADD 1 to STORAGE (INCREMENT)	INCR	2 MCT	ERASABLE
*6.	INCREASE ABSOLUTE VALUE OF STORAGE BY 1 (AUGMENT)	AUG	2 MCT	ERASABLE
*7.	DECREASE ABSOLUTE VALUE OF STORAGE BY 1 (DIMINISH)	DIM	2 MCT	ERASABLE
8.	DOUBLE ADD A AND L TO STORAGE	DAS	3 MCT	ERASABLE
9.	LOGICAL PRODUCT TO A	MASK	2 MCT	ALL
*10.	MULTIPLY; PRODUCT TO A AND L	MP	3 MCT	ALL
*11.	DIVIDE A AND L BY STORAGE QUOTIENT TO A	DV	6 MCT	ERASABLE
E.	INPUT OUTPUT			
*1.	TRANSFER CHANNEL TO A	READ	2 MCT	CHANNELS
*2.	TRANSFER A TO CHANNEL	WRITE	2 MCT	CHANNELS
*3.	LOGICAL PRODUCT (OF A AND CHANNEL) TO A	RAND	2 MCT	CHANNELS

*Each Requires EXTEND Instruction Proceeding
MCT = Memory Cycle Time

TABLE 2-IV (Cont)

NORMAL INSTRUCTIONS

	CODE	TIME	MEMORY LOCATION
E. INPUT OUTPUT			
*4. LOGICAL PRODUCT TO CHANNEL AND A	WAND	2 MCT	CHANNELS
*5. LOGICAL SUM TO A	ROR	2 MCT	CHANNELS
*6. LOGICAL SUM TO CHANNEL AND A	WOR	2 MCT	CHANNELS
*7. EXCLUSIVE OR TO A	RXOR	2 MCT	CHANNELS

*Each requires EXTEND Instruction Proceeding
MCT = Memory Cycle Time

TABLE 2-V

SPECIAL INSTRUCTIONS

	CODE	TIME	MEMORY LOCATION
A. INVOLUNTARY			
1. TRANSFER TO INTERRUPT PROGRAM, STORE C (Z) AND C (B)	RUPT	3 MCT	SPECIAL SUBSET
2. INCREMENT BY 1	PINC	1 MCT	COUNTERS
3. INCREMENT BY -1.	MINC	1 MCT	COUNTERS
4. DIMINISH ABSOLUTE VALUE BY 1	DINC	1 MCT	COUNTERS
5. SHIFT LEFT LOGICAL INSERT 0	SHINC	1 MCT	COUNTERS
6. SHIFT LEFT LOGICAL INSERT 1	SHANC	1 MCT	COUNTERS
7. INCREMENT BY 1 MOD 2^{15}	PCDU	1 MCT	COUNTERS
8. INCREMENT BY 1 MOD 2^{15}	MCDU	1 MCT	COUNTERS
9. HARDWARE RESTART	GOJ	2 MCT	4000_8
B. ADDRESS DEPENDENT			
1. RESUME INTERRUPTED PROGRAM	NDX	2 MCT	17_8
2. EXTEND	TC	1 MCT	6
3. INHIBIT INTERRUPT (INHINT)	TC	1 MCT	4
4. PERMIT INTERRUPT (RELINT)	TC	1 MCT	3
5. NO OPERATION (NOOP)	TCF	1 MCT	NEXT LOCATION
6. CYCLE RIGHT EACH ACCESS	CYR		20_8
7. SHIFT RIGHT EACH ACCESS	SR		21_8
8. CYCLE LEFT EACH ACCESS	CYL		22_8
9. SHIFT RIGHT SEVEN PLACES EACH ACCESS	EDOP		23_8

MCT = MEMORY CYCLE TIME

TABLE 2-V (Cont)

SPECIAL INSTRUCTIONS

	CODE	TIME	MEMORY LOCATION
C. GROUND SUPPORT EQUIPMENT SEQUENCE			
1. MANUAL DISPLAY MEMORY	FETCH	2 MCT	ALL
2. MANUAL LOAD MEMORY	STORE	2 MCT	ALL
3. MANNED DISPLAY CHANNEL	INOTRD	1 MCT	CHANNELS
4. MANUAL LOAD CHANNEL	INOTLD	1 MCT	CHANNELS
5. MANUAL START	TCSAJ	2 MCT	ALL

2.3.5 Timing and Priority Control

In order to provide characteristics in the computer design facilitating the real time control function of the guidance system, special circuits are provided to maintain a measure of real time and to have the capability of interrupting the normal program steps as the result of timed events or external stimuli. Real time is provided by a countdown chain from the crystal controlled oscillator running at 2.048 MHz. This is counted down to approximately 24 hours. The result is available to the program with a quantization of 312 microseconds. Timing for computational or control jobs is provided by counters that can be set under program control and, after counting down, interrupt the program in operation. The counting is accomplished by incrementing or shifting an erasable memory register. A number of these registers are used for real time control functions of the computer.

Counter incrementing may take place between any two instructions. External requests for incrementing a counter are stored in a counter priority circuit. At the end of every instruction a test is made to see if any incrementing requests exist. If not, the next instruction is executed directly. If a request is present, an incrementing memory cycle is executed. Each "counter" is a specific location in erasable memory. The incrementing cycle consists of reading out the word stored in the counter register, incrementing it (positively or negatively) or shifting it, and storing the results back in the register of origin. All outstanding counter incrementing requests are processed before proceeding to the next instruction. This type of interrupt provides for asynchronous incremental or serial entry of information into the working erasable memory. The program steps may refer directly to a counter register to obtain the desired information and do not have to refer to input buffers. Overflows from one counter may be used as inputs to another. A further property of this system is that the time available for normal program steps is reduced linearly by the amount of counter activity present at any given time.

Program interruption also occurs between program steps. An interruption consists of storing the contents of the program counter and transferring control to a fixed location. Each interrupt option has a different location associated with it. Interrupting programs may not be interrupted, but interrupt requests are not lost, and are processed as soon as the earlier interrupted program is resumed.

2.3.6 Memory

The memory of the computer consists of a coincident current ferrite core memory for erasable storage and, for program storage, a nondestructive read out (core

rope) memory of the transformer type that has the information wired in. The only logical difference between the two memories is the inability to change the contents of the rope memory by program steps. The rope memory is manufactured (information wired in) in a modular form, containing approximately 6K words, that can be plugged into the computer. The computer has a maximum capacity of six of these modules, thus providing for a total capacity of approximately 36K words of fixed memory.

There are two exceptional characteristics of this design. First, the erasable memory was designed to be insensitive to temperatures varying between 0 to 70 degrees C, and second, the rope memory permitted high density. Thus a large memory capacity could be provided in a relatively small volume computer. The insensitivity to temperature permits the computer to be operated with a cold plate temperature range of 0 to 35 degrees C.

2.3.7 Input/Output

The interfaces of the computer were designed to be as flexible as possible in order to provide for the multiplicity of subsystem functions that were controlled or sensed by the computer. The common usage of the computer in the guidance system of the CM and the LM lead to the requirement for flexibility but also provide motivation for limiting the circuit configuration, since the output circuits must operate electrically and functionally in either application.

In most cases the electrical interfaces were of two distinct types; one, a pulse transformer interface for those signals that required relatively fast response or accurate timing, and two, a dc level interface for those that are characterized as status signals or slow response. The only exception to these types of interfaces was the LM hand controller. This was an 800 Hz signal with an amplitude proportional to the angle of deflection of the hand controller. For this case a special analog-to-digital converter was provided in the computer. The primary reason for the two type of interfaces was to provide noise immunity at the interface. The transformer coupling was responsive to high frequency pulses, but because of the transformers the spacecraft wiring could be a balanced line with shielded twisted pairs well protected from induced noise. The dc level signals, that could be sensitive to spurious currents in the grounding paths, were provided with filtering to minimize noise susceptibility. This permitted a simpler spacecraft wiring.

The input/output section includes: logic for storage of status bits (input and output channel registers), logic for timing control, logic for storage of counter pulses and

interrupt commands, and finally the actual interface circuits for the conversion of signal levels. The information transfer is accomplished via the channel registers for all status or discrete information, such as displays or the firing of reaction control jets, and via the counter registers for incremental or serial word transfers.

In addition to the interfaces provided for the transfer of information between the computer and other subsystems, there are many signals required for synhronizing and timing. These include the primary time reference to the spacecraft central timing system at 1.024 MHz, the synchronization signals from the spacecraft that control the computer output to telemetry, and many other computer outputs for timing functions within the guidance system.

2.3.8 Standby Operation

The standby mode of operation was provided to conserve power. The early mission rules called for extended periods of time when the guidance system could be put to sleep. During this standby mode of operation, however, the computer must maintain real time and provide some of the interface timing signals, for example the 1.024 MHz output to the spacecraft central timing system. Since normal program operation and counter incrementing are suspended during standby, a countdown chain was provided in the logic of the computer to maintain time and provide the output timing signals. The timing chain is addressable as an input/output channel. Therefore, when the computer is returned to normal operation, the start up program can update the time counters in the erasable memory.

The standby mode of operation has not been very useful in the APOLLO missions to date, since mission rules leave the system on, or in the case of the LM during translunar flight, power is off. In this case real time and other initial conditions are sent to the computer via update link from ground control.

2.3.9 DSKY Characteristics

The DSKY is designed to function as an integral part of the computer, although it is physically separated. It provides a convenient method of communication between the computer and the operator.

The heart of the language of communication is the concept of Verbs and Nouns. This was designed into displays and operating procedures. The Verb is a two-digit code for what action is to occur. Examples are: loading data into the computer, displaying data from the computer, acknowledging acceptance of results or actions,

and initiating the execution of major computer programs. The Noun is a two-digit code for the operand affected by the action of the Verb. Usually the Noun code refers to a set of data contained in AGC erasable registers. All operation between the operator and system employ the use of these codes.

Commands are entered into the computer from a keyboard of nineteen push buttons. These include the ten decimal digits, plus and minus, the Verb, the Noun, and a number of auxiliary items. No more than one key is depressed at a time, so the nineteen key functions can be encoded into five signals. This is done by a diode matrix in the keyboard section of the DSKY. So that each key depression can be quickly gathered and interpreted, the key code inputs to the computer are of the interrupting type. The key input channel is interrogated by the keyboard interrupt program. This also makes a request to the computer's executive program to process the key code character at the earliest opportunity. A "trap" circuit logically differentiates the leading edge of the key code signal, so that no more than one interrupt is made for each depression of a key. This trap circuit is reset by a signal through all of the normally closed contacts of the keys. The reset signal is presented only when all keys are released.

The special keys or buttons have the following designations and functions:

(1) The CLEAR button. This provides a means for removing errors made during the loading of a data word, but before its acceptance by the computer.

(2) The KEY RELEASE button. This is used along with the KEY RELEASE light. This light, when on, indicates to operator that, because of his activity at the keyboard, some internal AGC activity has been unable to obtain use of the DSKY to display some information to him. Pushing the KEY RELEASE button indicates that operator is willing to relinquish the DSKY in order to observe the internally initiated display.

(3) The RESET button. (Formerly labelled ERROR RESET in Block I). This enables the operator to turn off the error indicators once he has observed them, so that later occurrences of the same error conditions can be indicated.

(4) The ENTER button provides two distinct functions. One, for execution of an already supplied Verb-Noun combination, and two for accepting an already supplied data word when data is being loaded into the AGC. Which function will be provided by depression of the ENTER button is clear from the context of its use.

(5) The Standby/Proceed (PRO) button provides two distinct functions. First, when the computer has been commanded by the proper keyboard entries to enable standby, the depression of the button will cause the computer to go

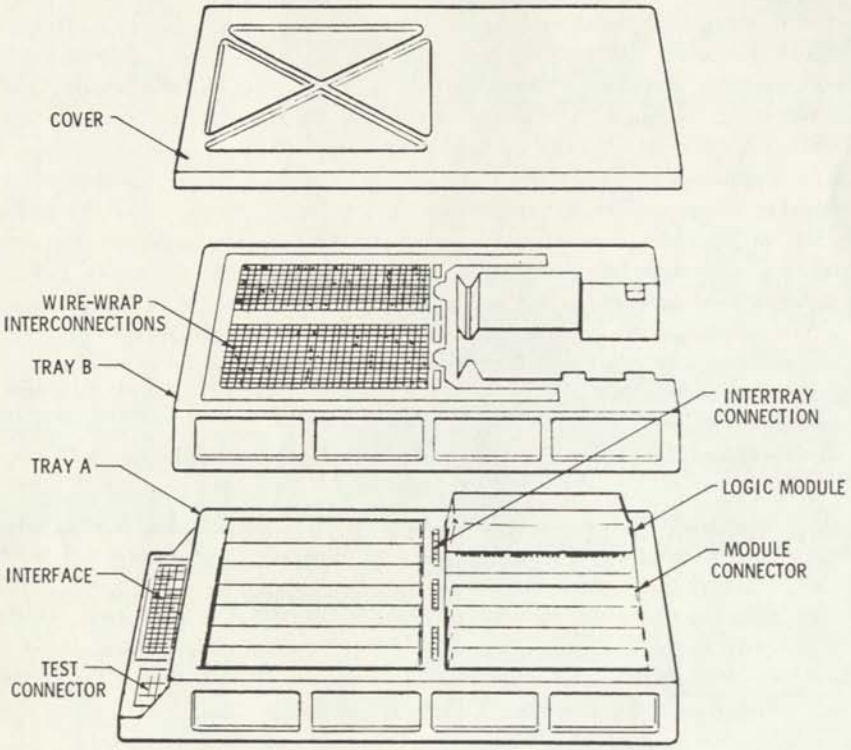


Fig. 2-8 Exploded Artist's Concept of Block II Computer

into standby mode. A subsequent depression of the button will return the computer to normal operation. Second, when the computer is in normal operation the button provides the command to proceed.

The display elements of the DSKY have the following features:

(1) The 5-digit (plus optional sign) data display registers (R1, R2, R3). Numerical data, either displayed to or loaded by the operator, passes through these registers. The signs are appropriate to decimal data, but no signs are used with octal data. The 5-digit basic data word was chosen because, in octal, it allows the 15-bit basic AGC word to be represented exactly; and in decimal, it provides 1 part in 100,000 accuracy that more than accommodates the 1 part in 16,384 accuracy of the basic AGC register contents.

(2) The Verb-Noun flash. This always indicates to the operator that some action is desired of him. The numerical segments of the 2-digit Verb and 2-digit Noun display are alternately turned off and on approximately once per second. This provides a mechanism that readily catches the attention of the operator.

(3) The PROGRAM ALARM light. This indicates errors caused by a program running into a detectable difficulty.

(4) The OPERATOR ERROR light. (In Block I this was called CHECK FAIL light - a term that was not too self-describing). This indicates a procedural error caused by improper operation of the DSKY.

(5) The RESTART light. This indicates one of a class of computer errors, whose common characteristic is that they cause a computer restart or program rollback to the last known safe point. These errors included: computer interrupt activity was too frequent, not frequent enough, or too long in duration; failure of a parity check; computer locked into a one instruction tight loop; failure of the timing scaler or of a counter.

(6) Additional status and caution conditions were provided as indicated in Fig. 2-5 for the CM. In the LM there were four more provided in the panel for a total of fourteen display elements which is the maximum capacity of the panel and DSKY.

2.4 COMPUTER MECHANICAL DESCRIPTION

The mechanical design of the Block II computer utilizes modular construction and wire wrapping for the interconnections of the modules. Figure 2-8 is an exploded artist's concept of the computer. This illustrates the two major subassemblies or trays containing modules and interconnecting wiring. The trays with the covers

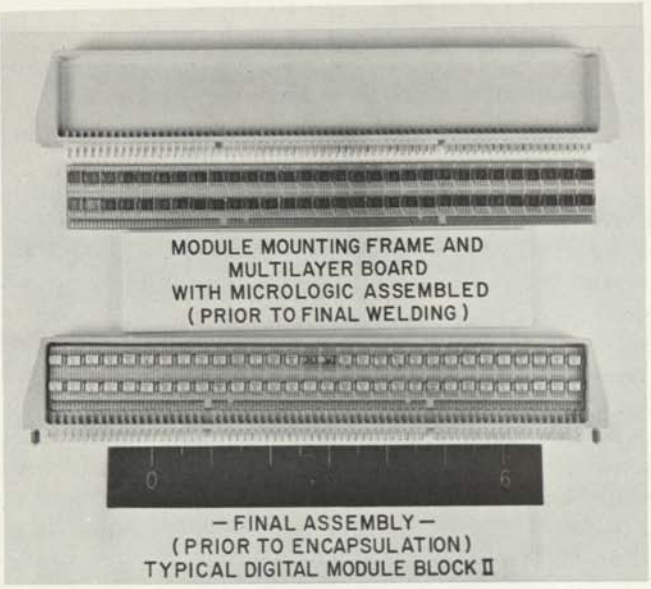


Fig. 2-9 Logic Module Containing 120 Microcircuit Units

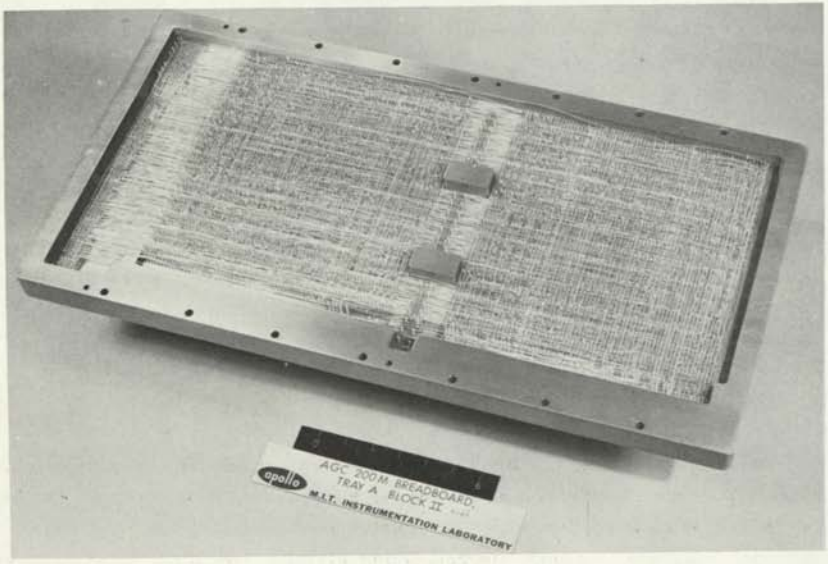


Fig. 2-10 Photograph of Wire-Wrapped Tray Connections

and gaskets provide mechanical support, thermal control via the spacecraft cold plate, environmental seal and shielding from electromagnetic interference. The rope modules are plugged into the structure from outside the sealed case. This permits program changes without breaking the environmental seal.

The module construction is basically welded cordwood type using standard components and integrated circuits. In the case of the 24 logic modules, the integrated circuit gates packaged in flatpacks are welded to multilayer boards for interconnection between gates. See Figure 2-9. The module frames provide mechanical support and thermal control for the components in addition to tray interface connector and jacking screws.

The modules are partitioned between the two trays such that the logic, interface, and power supply are in Tray A. The memory, memory electronics, analog alarm circuits, and oscillator are in Tray B, in addition to the connectors and mechanical support for the tray mounting the six rope modules.

The interconnecting wiring in the trays is accomplished by machine controlled wire wrapping for all interconnections. See Figure 2-10. This technique provides a well controlled and easily reproduced method for making the large numbers of interconnections required. In the computer there are about 15,000 connector pins with an average of more than two connections per pin. After the wiring is complete, the tray is potted to provide mechanical support for the interconnecting wires and connector pins.

2.5 DSKY MECHANICAL DESCRIPTION

The principal parts of the Block II DSKY are the frame, six indicator driver modules, 800 cps power supply, two pluggable display panels and keyboard. The power supply provides high voltage ac power for lighting the electroluminescent display panel. The control of the panel lighting is via the latching relay that provides coding and storage for the commanded display data. The status and caution panel is lighted with filamentary bulbs and also switched via either latching or non-latching relays. In the case of non-latching the information is stored in the AGC output register driving a detailed display light. The keyboard is a special design to permit lighted keys, required actuating force, and small size. All of these elements are interconnected and environmentally sealed by the frame and covers. See Figure 2-5.

SECTION 3.0
COMPUTER DESIGN DESCRIPTION

3.1 INTRODUCTION

The principal features of the electrical and mechanical design of the computer subsystem were shaped by the initial decisions, technological development, and changing requirements. The computer subsystem evolved from these constraints rather than from a fixed specification generated a priori. At various times, computer designs were made available to the associated contractor for production, and the designs were updated as mission requirements evolved. Each of the proposed design changes were thoroughly evaluated to determine the impact on the program. Many proposed design changes were not accepted. A typical instance, in the spring of 1969, involved the proposed increase in the size of erasable memory from 2,000 to 16,000 words. While the increase in erasable memory capacity had many desirable features, development time and cost provided a strong counter argument. A change proposed in late 1969 could not be implemented on an early APOLLO mission, and policy dictated that it could not be made unless absolutely necessary to mission success. The computer as described resulted from this evolutionary process, but the major characteristics were determined during the early phases of the development program.

Table 3-I is a list of major milestones in the development of the computer subsystem. This is intended to give an historical perspective to the various computer designs. Tables 3-II and 3-III summarize Block I and II design and development. These tables give data on design release, production delivery, software release, and flight date. There are many similarities between the Block I and II designs. However, this description of the computer design refers to the Block II configuration except in cases of special interest.

As described in Section 2.0, the computer has three principal functional subdivisions: timing and sequence generation, the central processor, and the memory section. In order to describe the electrical design features of the computer (see Figure 2-4 for block diagram), it is desirable to divide the computer into the following areas:

1. Logic design
2. Oscillator

TABLE 3-I

MAJOR MILESTONES

August 1961	MIT contract.
Spring 1962	GN&C Industrial Support contractors.
November 1962	MIT Proposal to change design to IC logic.
January 1963	AGC4 engineering model operating (IC logic).
Spring 1963	Program planning for LM concept, introduction of Block II CM.
September 1963	EMI specification imposed on contract.
January 1964	Design release for Block I (0 series) configuration.
May 1964	AGC5 mechanical prototype operating Block I (0 series) configuration.
Summer 1964	Mechanical redesign to meet moisture proofing requirement.
Summer 1964	Implementation meetings, established CM and LM requirements.
Summer 1964	Program planning to retrofit Block I (0 series) for moisture proofing.
August 1964	NASA approval for Block II design with common computer for CM and LM.
September 1964	Design release for Block I moisture proof configuration.
September 1964	AGC6 production prototype Block I (0 series) delivered.
January 1965	LM interface control documents signed.
January 1965	AGC7 production Block I (0 series) delivered.
February 1965	AGC120 Block I moisture proof configuration delivered.
Summer 1965	Design impact for EMI requirements identified for Block I and Block II (cable shielding and power ground).
July 1965	Design release for Block II.
July 1965	LM and CM power interface control documents signed permitting common 28V power return and case ground.

TABLE 3-I (Cont)

MAJOR MILESTONES

August 1965	Block I spacecraft EMI testing.
October 1965	CM interface control documents signed.
Fall 1965	Weight reduction study, identify use of magnesium module headers and case for Block II.
November 1965	Block II production prototype delivered non-operating (PPC-1).
December 1965	NASA critical design review (CDR).
December 1965	Block II mechanical prototype operating (200M).
January 1966	Identify requirement for Block II change before production of flight models.
February 1966	Block II design release for flight configuration (C1).
July 1966	Block II flight qualification model delivered (C1).
Fall 1966	Identify requirements for Block II configuration change for reliability improvements (effectivity C8).
Fall 1966	Block I production complete.
Summer 1969	Block II production complete.

TABLE 3-II

BLOCK I DEVELOPMENT SUMMARY

ACTIVITIES	DESIGN RELEASE	COMPUTER NUMBER	PRODUCTION DELIVERY	SOFTWARE RELEASE	FLIGHT DATE
MIT/IL System Test	Jan. 1964	5	May 1964	July 1964	
Production Prototype	Jan. 1964	6	Sept. 1964		
Qualification Test	Sept. 1964	110A	Aug. 1965		
FLIGHT SYSTEMS	July 1965				
AS202		117	Nov. 1965	Jan. 1966	Aug. 1966
APOLLO 4		SP2	Sept. 1966	Nov. 1966	Nov. 1967
APOLLO 6		SP1	Jan. 1966	Nov. 1966	Apr. 1968

TABLE 3-III

BLOCK II DEVELOPMENT SUMMARY

ACTIVITY	DESIGN RELEASE	COMPUTER NUMBER	PRODUCTION DELIVERY	SOFTWARE RELEASE	FLIGHT DATE
MIT/IL System Test	July 1965	200M	Dec. 1965	Mar. 1966	
Production Prototype	July 1965	PPC-1	Nov. 1965		
Qualification Test	Feb. 1966	C1	July 1966		
FLIGHT SYSTEMS	Oct. 1966				
APOLLO 5		C11	Apr. 1967	Mar. 1967	Jan. 1968
APOLLO 7		C13	June 1967	Feb. 1968	Oct. 1968
APOLLO 8		C17	Aug. 1967	Aug. 1968	Dec. 1968
APOLLO 9		CM C21 LM C16	Sept. 1967 Aug. 1967	Oct. 1968 Oct. 1968	Apr. 1969
APOLLO 10		CM C24 LM C15	Oct. 1967 Aug. 1967	Apr. 1969 Apr. 1969	May 1969
APOLLO 11		CM C28 LM C26	Dec. 1967 Nov. 1967	Apr. 1969 June 1969	July 1969
APOLLO 12		CM C18 LM C10	Oct. 1967 Mar. 1967	July 1969 Aug. 1969	Nov. 1969
APOLLO 13		CM C37 LM C30	Aug. 1968 May 1968	Dec. 1969 Feb. 1970	Mar. 1970
APOLLO 14		CM C29 LM C33	Feb. 1968 May 1968	May 1970 Sept. 1970	Jan. 1971

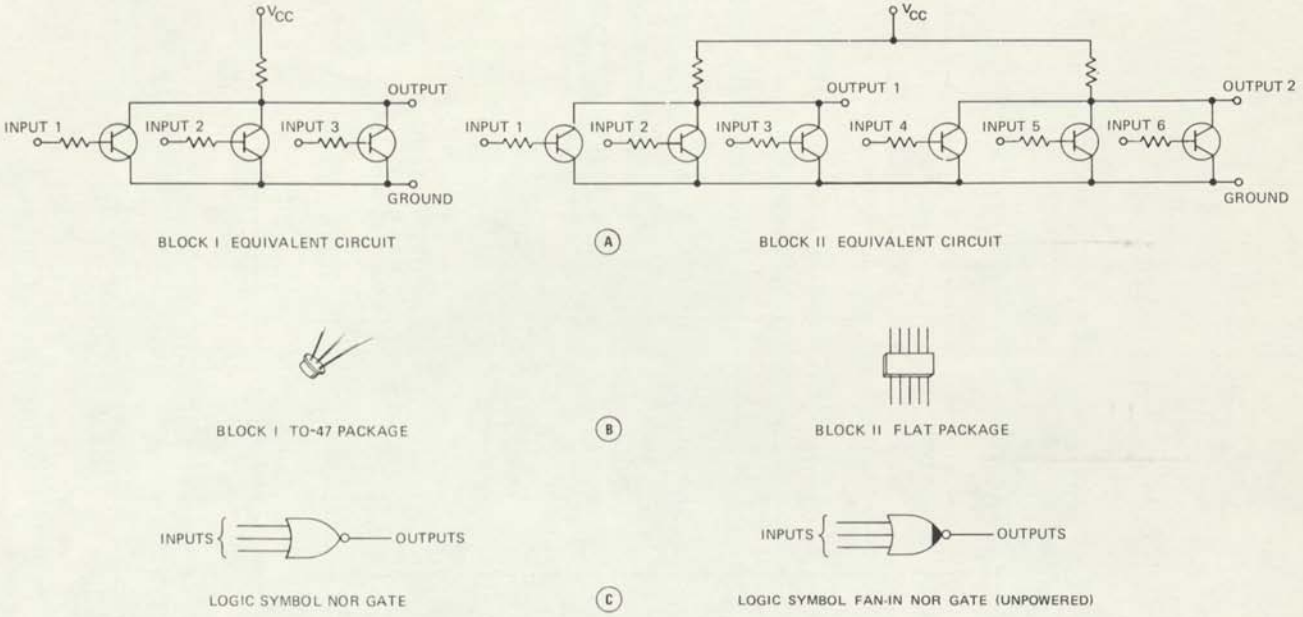


Fig. 3-1 Block I and Block II Integrated Circuit Logic Gates

3. Power supplies
4. Memory circuits
5. Alarms and malfunction detection
6. Interfaces (both input/output methods and external components)
7. Display and keyboard (DSKY) circuitry.

The mechanical design features will then be described in two sections, one concerning the computer and one the DSKY.

3.2 LOGIC DESIGN

The digital logic design of the computer is described within the framework of the principal functional elements required for signal processing. The functions implemented in the logic section of the computer are timing and sequence generation, central and special registers, the arithmetic section, instruction decoding, and memory addressing.

3.2.1 Logic Circuit Element

A logic circuit was required that was capable of synthesizing all switching operations, and that was simple enough to be controllable, testable, and producible. The circuit chosen was an integrated circuit NOR gate. This employs a configuration known as modified direct-coupled transistor logic (DCTL). Three transistors in parallel, along with four resistors, form a three-input gate with a fan-out capability of approximately 5, and an average propagation delay of about 20 nanoseconds, while dissipating about 12 mW of power. In the Block I computer, this gate was manufactured on a single chip of silicon and packaged in a TO-47 package. A Block II modification of this design resulted in a new unit with approximately the same specification, except for a power dissipation of 5 mW per gate and two gates per silicon chip mounted in a 10-lead flat package. Figure 3-1 illustrates the Block I and II integrated circuit gates, the package configuration and the logic symbols.

The importance of using a single circuit should not be underestimated. Thousands of logic gates are employed in each computer, thus high reliability is essential for every gate. It was assumed early in the development that reliability of this new component could best be attained by standardization, and could only be demonstrated by the evaluation of large samples. Had a second type of logic microcircuit been employed in the computer, the number of logic elements could have been reduced by about 20 percent. But it is clear that to have done so would have been false economy from the point of view of reliability, for neither of the two circuits would

have accumulated sufficient operating history to demonstrate the high mean time between failures with the confidence level of a single NOR circuit.

Logic equations expressed in the familiar AND, OR, and NOT notation may readily be realized with NOR operators. The NOR function of the three variables is $N(x,y,z) = \bar{x}\,\bar{y}\,\bar{z} = \overline{x+y+z}$ an AND function is $A(x,y,z) = x\,y\,z$ and an OR function is $O(x,y,z) = x + y + z$ by comparison, $N(x,y,z) = A(\bar{x},\bar{y},\bar{z}) = \overline{O}(x,y,z)$ The NOT operation, or complementation, is the NOR function of one variable, i.e., $\bar{x} = N(x)$. Complex Boolean expressions ordinarily arise only in connection with nonsequential, or combinational aspects of the computer logic. With combination of NOR operations any of these expressions may be realized.

Sequential operations require storage, and the basic logic storage element is the flip-flop. Two NOR gates may be used to form a flip-flop if the output of each is an input to the other, and if all other inputs are normally zero. If one of these other inputs is momentarily made equal to 1, the flip-flop is forced into a state that may be defined as a 1. Whereas, if a free input on the opposite gate is made equal to 1, the 0 state is obtained. Most frequently, the condition for setting a flip-flop in a particular state is that two or more other signals simultaneously assume prescribed values. Detection of such coincidence requires a NOR operation at the input of the flip-flop plus any NOR operation required to invert (complement) the inputs.

It is frequently necessary to implement NOR functions of more than three variables, and also to drive more than five inputs with a single output. For these reasons, NOR gates may be combined so as to increase either the input capacity (fan-in), the output capacity (fan-out), or both. Fan-in is increased by connecting the outputs of unpowered gates to the output of a powered gate. This provides a fan-in of three times the total number of gates. Fan-out is increased by connecting the outputs of powered gates together. Both fan-in and fan-out are increased, but the fan-in is not available because it is necessary to have each input signal connected to as many inputs in common as there are powered gates connected together. This is done in order to distribute the collector currents and thus be able to saturate the transistors. By simultaneous application of these techniques, however, it is possible to increase both fan-in and fan-out at the same time.

An illustrative example of the NOR logic in the computer is provided by the operation of the flip-flop registers in the central processor. Digits are transferred from one register to another through a common set of lines called the write buses. The central register flip-flops are selected by read and write pulses applied to gates

that either set or interrogate the flip-flop of the corresponding register. Figure 3-2 shows a hypothetical set of three flip-flops similar to those in one bit column of the computer's central register section. Information transfers between the registers are controlled by three clocked action pulses: read, write, and clear. Thus the $\overline{\text{WRITE BUS}}$ is normally in the 1 state, and changes to 0, while transferring a 1. Suppose REG 1 contains a 1, i.e., the top gate of its flip-flop has an output of 0. At the time that the $\overline{\text{READ 1}}$ signal goes to 0 from its normal 1 state, the output of the read gate, CONTENT 1, becomes a 1. This propagates through a read bus fan-in and an inverter and fan-out amplifier to make $\overline{\text{WRITE BUS}}$ become 0. Suppose that $\overline{\text{WRITE 2}}$ is made 0 concurrently with $\overline{\text{READ 1}}$. Then the coincidence of 0's at the write gate of REG 2 generates a 1 at the SET 2 input, thus setting the bit to 1.

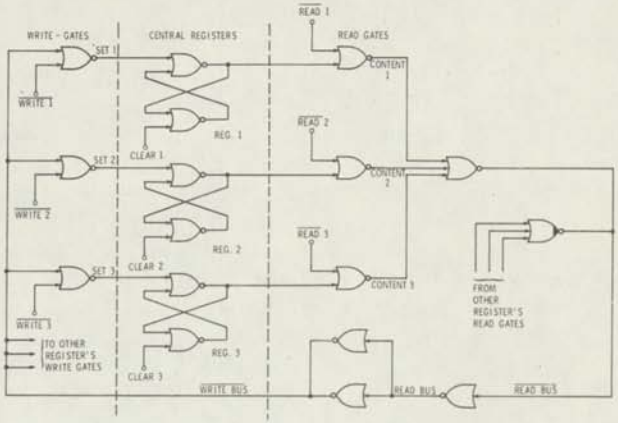


Fig. 3-2 Equivalent Logic, NOR Circuits in Computer Central Registers

If REG 1 had contained a 0, the write bus would have remained at 1, and no setting input would have appeared at the upper gate of REG 2. The CLEAR 2 pulse, that always occurs during the first half of $\overline{\text{WRITE 2}}$, would have forced the flip-flop to the 0 state, where it would remain; whereas when a 1 is transferred, the SET 2 signal persists after the CLEAR 2, and thus forces the register back to the 1 state. Thus the simultaneous occurrence of $\overline{\text{READ 1}}$, $\overline{\text{WRITE 2}}$, and the short CLEAR 2 pulses transfers the content of REG 1 to REG 2. Only the content of REG 2 may be altered in the process. REG 1 and REG 3 retain their original contents. An instance of gates being used to increase fan-in is shown where several CONTENT signals are mixed together to form the signal $\overline{\text{READ BUS}}$. An increase in fan-out is achieved by the two gates connected in parallel to form the signal $\overline{\text{WRITE BUS}}$.

3.2.2 Logic Timing and Sequencing

The timing circuits of the computer consist of a ring counter, a scaler, a time pulse generator, and other circuits that generate sub-phases of the time pulses,

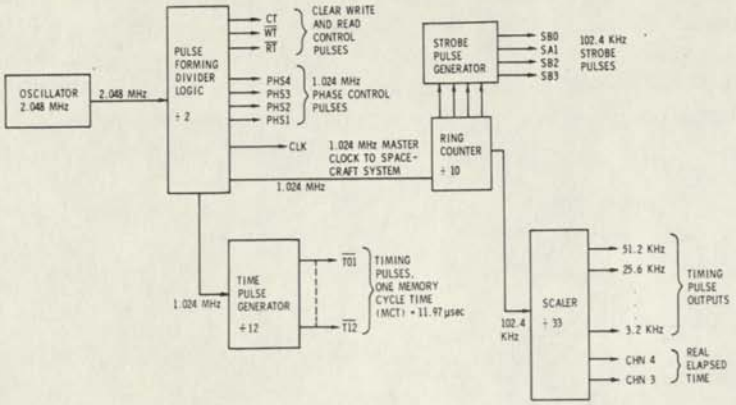


Fig. 3-3 Timing Logic Functional Diagram

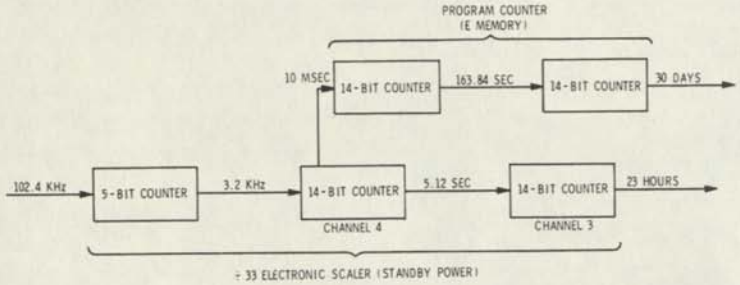


Fig. 3-4 Programming Counter and Scaler Configuration

developed from a 2.048-MHz oscillator, that is the primary timing source (see Section 3.3). Figure 3-3 is a functional diagram of the computer timing circuits.

The pulse forming logic generates the basic control pulses and the spacecraft master clock reference. The spacecraft reference signal of 1.024-MHz is obtained by dividing the 2.048-MHz oscillator output in a single-stage divider. This reference signal is also used, together with the oscillator output, to generate control pulses, clear, write, read, and four phases of the 1.024-MHz signal for use in sequencing activity within time pulses.

A divide-by-12 circuit generates the 12 separate pulses of 0.977 μ sec duration that occur sequentially within a memory-cycle time (MCT) of 11.7 μ sec. A shift register implementation is used in which a single 1 is created and shifted through the 12 positions of the register.

The other frequencies needed for the computer and interface requirements are 102.4 kHz and its binary submultiples. The 102.4-kHz signal is generated in a 10-step ring counter, whose input is the 1.024-MHz signal. The ring counter also generates auxiliary outputs used to construct strobe signals for phase and pulse length control of various interface timing pulses, roughly 3 msec duration, at different positions within the basic 9.76 μsec interval of the 102.4-kHz signal. The binary submultiples of the 102.4-kHz signal are generated by a binary scaler, or counting chain, 33 stages in length. This scaler, along with the rest of the timing, is operative when power is applied to the computer regardless of whether it is in the Standby or Operate mode. See Figure 3-4. It thus serves as a measure of real elapsed time, and the high order 28 bits can be interrogated under program control (I/O channels 3 and 4) to form a double-precision positive integer signifying the number of elapsed quanta of 1/3200 second. In normal operation, this integer repeats with a period of about 23.30 hours, and therefore can be used to update the program time counter following a mode when the computer is in Standby.

The computer sequence generator is the equivalent of a wired memory that is implemented in the logic design. The sequence generator memory is addressed by time pulses, by instruction codes, by a memory cycle stage counter, and by tests of priority activity. The output of the wired memory is a sequence of control pulses that are formed by a cross-point generator as a logic product of the appropriate time pulses and instruction codes. The control pulses provide the gating necessary to perform the operations required during any one instruction. Appended to the sequence generator are circuits for resolving competitive requests for interruptive action. These are organized in a priority structure, where the highest priority

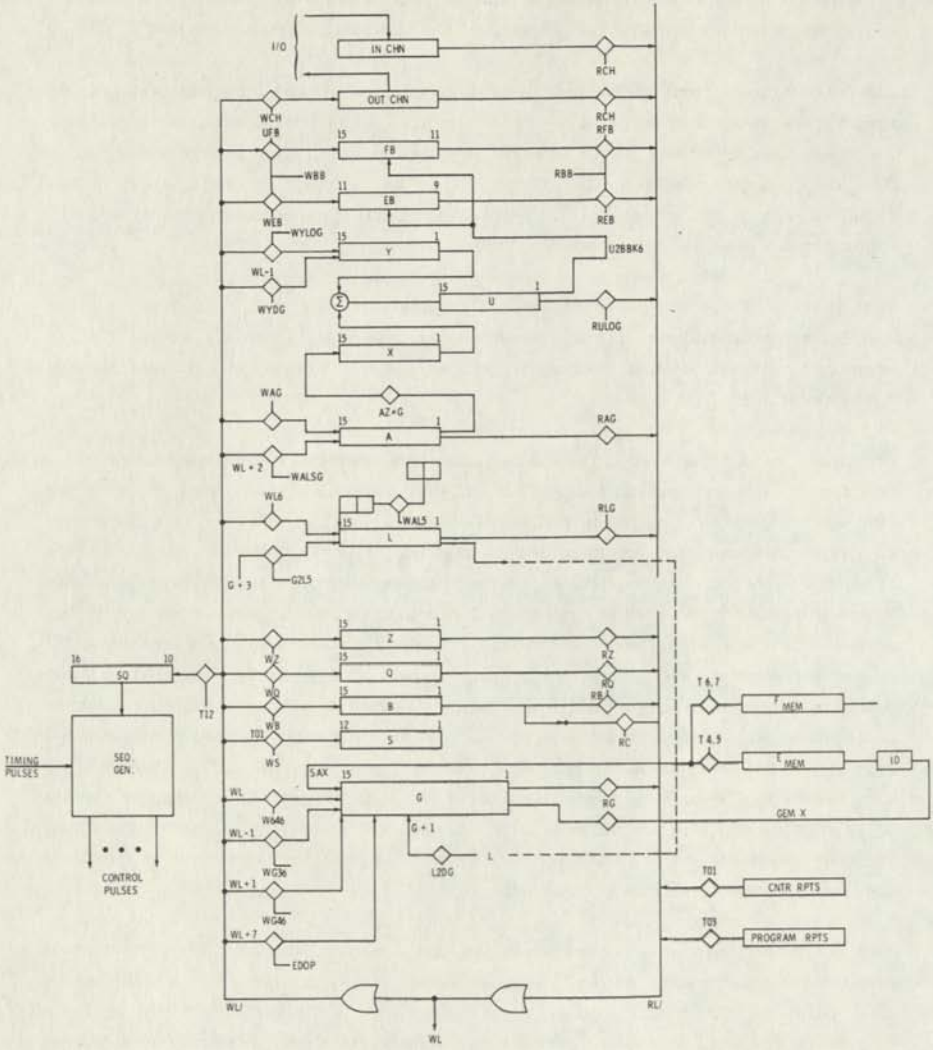


Fig. 3-5 Central Processor

request is serviced first regardless of when it is received. Separate priority circuits serve the counter increment (cycle steal) and program interrupt signals. Cycle steals occur between all instructions. Program interrupt can be initiated at the end of nearly all instructions except when an interrupt is already in progress or when inhibited under program control.

3.2.3 Central Processor

The logic in the central processor includes the flip-flop registers, arithmetic element, interface registers, memory addressing logic, and other logic required for data manipulation. See Figure 3-5.

The special and central registers include addressable and unaddressable flip-flop registers. These include the upper and lower accumulators (A and L), memory address (S) and bank registers (EB and FB), program counter (Z), memory buffer (G), and auxiliary registers (X, Y, Q, SQ, and B). Only two of these registers can be considered non-special, that is, having no function other than temporary 15-bit storage. These are the program counter Z and the return address auxiliary register Q. Auxiliary register B (contains the next instruction) has an inverted output denoted C as well as a direct output. Auxiliary register, X and Y, are temporary storage for the adder. Auxiliary register, SQ, stores the instruction code. The upper and lower accumulators, A and L, perform shifting and overflow storage functions. Memory buffer register, G, performs shifting and cycling functions, and controls inhibit digit drivers in the erasable memory. The memory address register, S, controls memory selection circuits. Bank registers, EB and FB, likewise control memory selection, and are both accessible via a common address denoted BB as well as individually.

Computer arithmetic is based on a single adder, which is used for addition, subtraction, multiplication, and division. A few auxiliary logic functions are incorporated in order to expedite the latter two operations in the Block II computer; these operations were omitted in Block I. The requirement for more computing speed in Block II dictate the use of more logic to speed up the arithmetic operations.

The Block I adder stored the two operands in flip-flop registers (X and Y), and developed, for each bit position, the binary sum (U) of X, Y, and an incoming carry, and also developed the outgoing carry. The logic was arranged so as to limit the number of gate stages between incoming and outgoing carries to 2, which is the minimum possible using NOR logic alone. The 16-bit adder thus required 32 gate propagation delay times, or about 960 nanoseconds for carry propagation alone,

assuming a worst-case 30 nanosecond average gate delay. Taking into account the propagation through buses, gates, and summing circuitry with adequate timing margin, three pulse times (977 nanoseconds each) were necessary for each use of the adder. During multiplication this was the limiting speed factor. The Block I multiply instruction used addition 14 times, once for each bit of the multiplier operand except for sign. Two additions were accomplished per memory cycle, resulting in an eight memory-cycle time sequence.

In the Block II design, logic changes were made to increase the speed of the multiply instruction from 117 μsec (as in Block I) to 46.8 μsec. These changes included the introduction of carry skip, treatment of two multiplier bits at once, and dedicated wire word transfers.

The first logic change (carry skip) decreases the worst-case carry propagation delay by anticipating the output wherever four successive stages, starting with an even-numbered stage, are in receipt of an incoming carry. Carry-prone means that one or both operand bits are 1. When a group of successive stages are carry-prone and a carry enters the lowest order of the group, there is necessarily a carry out of the highest order stage of the group. With a very low cost in additional gates, the worst-case add time was reduced to under a pulse time (977 nanoseconds), so that the sum could be read at the pulse time following the addition.

The second logic change entailed modifying the multiplier function of the operand in the multiplication process. Where one multiplier bit controlled whether or not the multiplicand was to be added to the partial product (as in Block I), two multiplier bits control whether the multiplicand is added 0, 1, 2, or 3 times to the partial product in Block II. The 0 and 1 multiples are readily available by suppressing or allowing addition of the multiplicand. The 2-multiple is obtained by adding the multiplicand shifted left one position, which requires a special write-gate set on the Y register. To obtain the 3-multiple, an apparently indirect means is used of subtracting the multiplicand and remembering to add in a 1-greater multiple during the next addition cycle. This becomes a 3-multiple because that addition in the next cycle is weighted four times the equivalent addition in this cycle, since two bits are handled at once. Therefore, this procedure is tantamount to adding four times and subtracting once to make a net addition of three times the multiplicand. Subtraction is easily handled by storing the multiplicand in the B register, where its complement, and hence negative, can be read to the read bus via the inverted output (C). The carry from one cycle to another can result in increasing a 3-multiple addition to a 4-multiple addition, in which case there is no action during that cycle, but there is a carry to the next cycle.

The third logic change was the incorporation of dedicated wiring between the A and X registers and between the L and G registers to allow simultaneous loading of X and Y, and simultaneous shifting of A and L, such that an entire memory-cycle time could be reduced to two pulse times. The end result of the three revisions is that multiply requires three memory-cycle times in Block II as compared with eight in Block I. This comparison is actually somewhat incomplete, as it omits the fact that multiplication is preceded in each machine by an extend operation of two memory-cycle times in Block I and one in Block II. Thus the real-time comparison is 10 cycles versus 4 cycles, a speed increase by a factor of 2.5.

The divide instruction speed was not as easily increased as the multiply instruction. However, the faster adder and dedicated transfers resulted in a reduction from 18 to 8 memory-cycle times. One of the requirements for the control pulse sequence for Block II division is for the removal of carry propagations in the adder. The carry anticipation circuits do not expedite this action, and logic is therefore furnished to suppress every fourth existing carry at the appropriate time, so that this action can be accomplished within one pulse time as can addition.

Parallel information transfer is effected in these special and central registers by a bus system as well as a certain amount of dedicated transfer circuitry. Generally, each of the registers consists of a gated flip-flop in each occupied bit position. The flip-flop ordinarily has a read gate that fans into the common read bus and has one or more write gates that fan into the flip-flop from the common write buses (one bus for each bit position, shifting operations therefore require more write gates) or from other registers. The read and write buses have identical information, the former driving the latter through amplifying logic. See Figure 3-2. Some special registers (X register) are not connected to the bus but are loaded by dedicated wiring. In the case of the X register, loading is directly from A and feeds the adder logic.

Other than the special and central flip-flop registers, the logic contains a number of other registers and performs special logical operations when special erasable memory registers are addressed. The additional registers include the input/output channel registers and the G register for the memory interfaces. The input/output channel registers 1 through 16_8 and 30_8 through 35_8 are addressable with input/output instructions only, thus resolving the overlap between the channel and the main memory addresses.

For erasable memory addresses, 20_8 through 23_8, the word being transferred into the G register is not sent directly, but is modified by a special gating network.

TABLE 3-IV

SPECIAL ADDRESSABLE REGISTERS

OCTAL PSEUDO ADDRESS	REGISTER NAME	REMARKS	TYPE
00000	A		Flip-Flop Registers
00001	L	(also channel 01)	
00002	Q	(also channel 02)	
00003	EB	Erasable Bank Register	
00004	FB	Fixed Bank Register	
00005	Z	Program Counter	
00006	BB	Both Bank Registers	
00007	--	Zeros	
00010	ARUPT	xRUPT = Storage for x	Erasable Memory
00011	LRUPT	during Interrupt;	
00012	QRUPT	ZRUPT & BRUPT	
00013	(spare)	stored automatically.	
00014	(spare)		
00015	ZRUPT		
00016	BBRUPT		
00017	BRUPT	Stores Next Instruction	
00020	CYR	Cycle Right 1 Bit	
00021	SR	Shift Right 1 Bit	
00022	CYL	Cycle Left 1 Bit	
00023	EDOP	Shift Bits 8 thru 14 seven places right.	

Table 3-IV lists the functions of these special registers. Note that the first eight registers (0_8 - 7_8) in the erasable memory are not accessible, because they are reserved for the central flip-flop registers. The next eight (10_8 - 17_8) are reserved for storage during interrupts (see Section 3.2.5).

3.2.4 Address and Instruction Decoding

An instruction word of the computer contains the address and operation code for a given computer operation (see Figure 3-6). Bits 1 through 12 are stored in the S register and bits 10 through 15 are stored in the SQ register. The multiple use of bits 10, 11, and 12 provides for extend operation codes for restricted memory address field. The contents of S, SQ and other central registers determine memory address and the operation to be performed.

The contents of address register S, erasable bank register EB, fixed bank register FB, and fixed extension F EXT (SUPERBNK) bit in output channel 7 are necessary to address the complete memory of the computer. For example, the 2048 erasable registers are accessed by a combination of the 10 least significant bits of the address field in the S register and the three-bit EB register. The combination of all 12 bits of S, the five bits of FB, and the fixed extension bit are used to address the fixed memory. Table 3-V lists the octal addresses of all fixed and erasable memory locations with the contents of the S register and bank bits necessary to access the locations.

The operations performed by the computer are determined by decoding the contents of SQ bits, 10 through 15 and bit 16, the extend bit, that is set to 1 by an extend instruction (TC6), preceding the operation for which it is required. The extend bit is reset by every instruction except index (NDX). As indicated in Figure 3-6, bits 10, 11, and 12 of the address field of an instruction word are used to expand the operation code by restricting the address field to erasable memory or, for bit 10, input/output channel instructions. That is, the instruction decoding circuits limit the memory address decoding circuits to erasable memory or channel addresses during the operations that make use of bits 10 through 12. Table 3-VI lists the instructions by mnemonic code and the corresponding contents of the SQ register.

3.2.5 Interrupt Structure

In addition to the instructions listed in Table 3-VI, there are a number of involuntary sequences, not under the programmer's control, that can break into or interrupt the normal sequence of instructions. These sequences are triggered either by

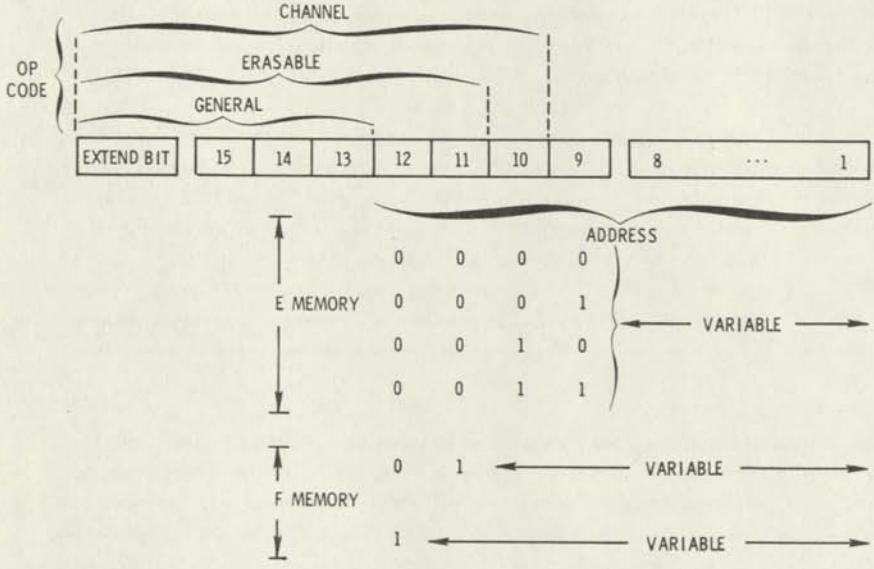


Fig. 3-6 Computer Instruction Word

TABLE 3-V

FIXED AND ERASABLE ADDRESSING LOCATIONS

	OCTAL ADDRESS	F EXT	FB					EB			S										
		1	5	4	3	2	1	3	2	1	12	11	10	9	8	7	6	5	4	3	2	1
2048 Erasable Registers	00000 - 01377	x[1]	x	x	x	x	x	x	x	x	0	0			0000 - 1377[2]							
	00000 - 00377	.			.			0	0	0	0	0	1	1	0000 - 0377							
	00400 - 00777	.			.			0	0	1	0	0	1	1	0000 - 0377							
	.	.			.			.							.							
	.	.			.			.							.							
	.	.			.			.							.							
	03400 - 03777	x	x	x	x	x	x	1	1	1	0	0	1	1	0000 - 0377							
36864 Fixed Memory Registers	04000 - 07777	x	0	0	0	0	0	.			1				0000 - 7777							
	10000 - 11777	.	0	0	0	0	0	.			0	1			0000 - 1777							
	12000 - 13777	.	0	0	0	0	1	.			0	1			.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	66000 - 67777	x	1	0	1	1	1	.			0	1			.							
	70000 - 71777	0	1	1	0	0	0	.			0	1			.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	106000 - 107777	0	1	1	1	1	1	.			.				.							
	110000 - 111777	1	1	1	0	0	0	.			.				.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	.	.			.			.			.				.							
	116000 - 117777	1	1	1	0	1	1	x	x	x	0	1			0000 - 1777							

NOTES

1. x means the contents of the bit is ignored by the address decoding logic.
2. Octal equivalent of variable part of S.

TABLE 3-VI

INSTRUCTION CODES

NO.	CODE[1]	MEMORY	SQ 16	15	14	13	12	11	10
A.	SEQUENCE CHANGING								
1.	TC Relint (3)[3] Inlint (4) Extend (6)	All	0	0	0	0	x	x	x[2]
2.	TCF	F	0	0	0	0	[1 to 3]		x
3.	CCS	E	0	0	0	1	0	0	x
4.	BZF	F	1	0	0	1	[1 to 3]		x
5.	BZMF	F	1	1	1	0	[1 to 3]		x
6.	EDRUPT	Channel	1	0	0	0	1	1	1
B.	FETCHING & STORING								
1.	CA	All	0	0	1	1	x	x	x
2.	CS	All	0	1	0	0	x	x	x
3.	DCA	All	1	0	1	1	x	x	x
4.	DCS	All	1	1	0	0	x	x	x
5.	TS	E	0	1	0	1	1	0	x
6.	XCH	E	0	1	0	1	1	1	x
7.	DXCH	E	0	1	0	1	0	1	x
8.	LXCH	E	0	0	1	0	0	1	x
9.	QXCH	E	1	0	1	0	0	1	x
C.	INSTRUCTION MODIFICATION								
1.	NDX Resume (17)	E	0	1	0	1	0	0	x
2.	NDX	All	1	1	0	1	x	x	x
D.	ARITHMETIC & LOGIC								
1.	AD	All	0	1	1	0	x	x	x
2.	SU	E	1	1	1	0	0	0	x
3:	ADS	E	0	0	1	0	1	1	x

TABLE 3-VI (Cont)

INSTRUCTION CODES

NO.	CODE[1]	MEMORY	SQ 16	15	14	13	12	11	10
4.	MSU	E	1	0	1	0	0	0	x
5.	INCR	E	0	0	1	0	1	0	x
6.	AUG	E	1	0	1	0	1	0	x
7.	DIM	E	1	0	1	0	1	1	x
8.	DAS	E	0	0	1	0	0	0	x
9.	MASK	All	0	1	1	1	x	x	x
10.	MP	All	1	1	1	1	x	x	x
11.	DV	E	1	0	0	1	0	0	x
E.	INPUT OUTPUT								
1.	READ	Channel	1	0	0	0	0	0	0
2.	WRITE	Channel	1	0	0	0	0	0	1
3.	RAND	Channel	1	0	0	0	0	1	0
4.	WAND	Channel	1	0	0	0	0	1	1
5.	ROR	Channel	1	0	0	0	1	0	0
6.	WOR	Channel	1	0	0	0	1	0	1
7.	RXOR	Channel	1	0	0	0	1	1	0

NOTES:

1. See Table 2-IV for the definition of the code.
2. x means the contents of the bit is ignored by the instruction decoding logic.
3. Relint (3), etc.
   Special operations are performed which are address-dependent.

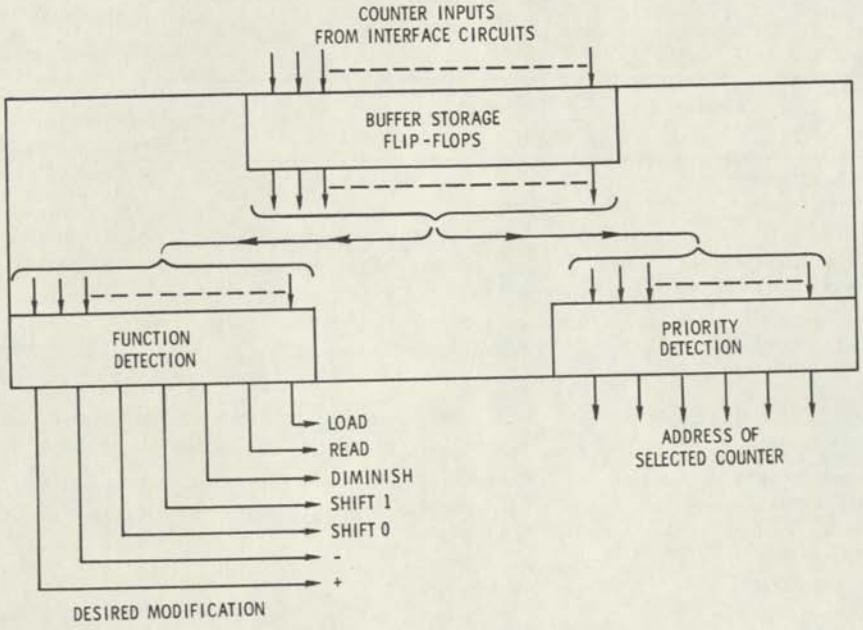


Fig. 3-7 Structure of Computer Priority Circuit

computer interface activity or by certain overflow conditions within the computer. The interrupts are of two distinct types that are quite different. One, counter interrupts may take place between any two instructions and consume one memory cycle time to update the counter being addresssed. Two, program interrupts occur between any two instructions, if no interrupt is in process, and provide the function of transferring program control to a predetermined address in fixed memory. A different address is provided for each of the interrupt options. The length of time the computer remains in the interrupt mode depends upon the operations that must be performed during the interrupt.

3.2.5.1 Counter Interrupts

Counter interrupts can take place between any two instructions. Requests for a counter operation are stored in the counter priority circuit. The priority logic (see Figure 3-7) is examined at the end of every instruction to see if any counter operations have been requested. If not, the next instruction is executed directly. If a request is present, a memory cycle is executed that consists of reading out the word stored in the counter register, performing the operation and storing the results back in the register of origin. All outstanding counter requests are stored and processed before proceeding to the next instruction. Priority for the satisfaction of one or more requests is based on the value of the counter's address. (0024_8 has the highest priority, and 0060_8 has the lowest). This type of interrupt provides for asynchronous incremental or serial entry of information into the working erasable memory. The normal instructions can refer directly to a counter register to obtain the desired information. Overflows from one counter may be used as inputs to another. A further property of this system is that the time available for normal program steps is reduced linearly by the amount of counter activity available at any given time.

There are 29 counter interrupts in the computer. These are associated with 29 erasable memory addresses (0024_8 - 0060_8) that may contain counter-type information. Table 3-VII lists these counters by function, memory location, and purpose. The seven "involuntary" instruction or operation associated with these counters is performed when the appropriate counter interrupt is received. A counter interrupt must specify the counter address and the operation to be performed. These are determined by the dedicated interconnections from the external signal (such as positive or negative changes to the value of a counter) to the appropriate address and operation. The seven involuntary instructions, and the counter address to which they apply, are listed as follows:

TABLE 3-VII

SPECIAL PURPOSE COUNTERS

COUNTER	OCTAL ADDRESS	INSTRUCTION	PURPOSE
TIME2	00024	PINC(1)	High order part of Main Time counter; stores high order part of reference (clock) time
TIME1	00025	PINC	Low order of Main Time counter; stores low order part of reference (clock) time.
TIME3	00026	PINC	Task control timer; controls the operation of utility program WAITLIST
TIME4	00027	PINC	Output Control; controls the operation of rupt routine T4RUPT
TIME5	00030	PINC	DAP Timer; controls the operation of the thrust vector control program through rupt routine T5RUPT
TIME6	00031	DINC	DAP Timer; controls the operation of the RCS for short firings
CDUX	00032	PCDU, MCDU	IMU CDUX counter
CDUY	00033	PCDU, MCDU	IMU CDUY counter
CDUZ	00034	PCDU, MCDU	IMU CDUZ counter
CDUT	00035	PCDU, MCDU	CM = Optics trunnion CDU LM = RR trunnion CDU
CDUS	00036	PCDU, MCDU	CM = Optics Shaft CDU LM = RR Shaft CDU
PIPAX	00037	PINC, MINC	PIPA, V_X counter
PIPAY	00040	PINC, MINC	PIPA, V_Y counter
PIPAZ	00041	PINC, MINC	PIPA, V_Z counter
Q-RHCCTR	00042	PINC, MINC	Pitch RHC counter ($BMAG_X$ in CM)
P-RHCCTR	00043	PINC, MINC	Yaw RHC counter ($BMAG_Y$ in CM)

(1) See Table 2-V for definition of the instruction.

TABLE 3-VII (Cont)

SPECIAL PURPOSE COUNTERS

COUNTER	OCTAL ADDRESS	INSTRUCTION	PURPOSE
R-RHCCTR	00044	PINC, MINC	Roll RHC counter ($BMAG_Z$ in CM)
INLINK	00045	SHINC, SHANC	Input counter for converting incoming serial uplink (or crosslink) information into parallel information
RNRAD	00046	SHINC, SHANC	Input counter for converting incoming serial rendezvous radar or landing radar information in the LM and VHF ranging information in CM into parallel information.
GYROCMD	00047	DINC	Output counter used to control pulse bursts to X, Y, and Z gyros.
CDUXCMD	00050	DINC	Output counter used to control pulse bursts driving IMU X axis CDU.
CDUYCMD	00051	DINC	Output counter used to control pulse bursts driving IMU Y axis CDU.
CDUZCMD	00052	DINC	Output counter used to control pulse bursts driving IMU Z axis CDU.
CDUTCMD	00053	DINC	Output counter used to control pulse bursts driving the RR trunnion CDU.
CDUSCMD	00054	DINC	Output counter used to control pulse bursts driving the RR shaft CDU.
THRUST	00055	DINC	Output counter used to control pulse bursts driving the THRUST control of the LM.
EMS	00056	DINC	Output counter used to control pulse bursts driving the entry monitor system of the CM.
OUTLINK	00057	SHINC	Output counter for converting parallel outgoing information into serial information.
ALTM	00060	SHINC	Output counter for converting parallel information going out to the altitude and attitude rate meters into serial information.

1. DINC, applying to cell 0031_8 (used as timing counter) and cells 0047_8 - 0056_8 (used for generating incremental computer outputs). If the contents of the cell are positive, non-zero, they are decremented by 1 and positive output pulses are provided; if the contents are negative non-zero, the contents are incremented by 1 (i.e., magnitude decreased by 1) and negative output pulses are provided.

2. MCDU, applying to cells 0032_8 - 0036_8 (used for negative incremental inputs). This instruction subtracts 1 (in 2's complement) from the contents of the cell.

3. PCDU, applying to cells 0032_8 - 0036_8 (used for positive incremental inputs). This instruction adds 1 (in 2's complement) to the contents of the cell.

4. MINC, applying to cells 0037_8 - 0044_8 (used for negative incremental inputs). This instruction subtracts 1 (in 1's complement) from the contents of the cell.

5. PINC, applying to cells 0024_8 - 0030_8 (used as timing counters) and cells 0037_8 - 0044_8 (used for positive incremental inputs). This instruction adds 1 (in 1's complement) to the contents of the cell.

6. SHANC, applying to cells 0045_8 - 0046_8 (used for converting serial input to parallel). This instruction shifts the contents of the cell left by one place, and then adds one. It is used for a binary 1 of a serial input bit stream.

7. SHINC, applying to cells 0045_8 - 0046_8 (used for converting serial input to parallel) and to cells 0057_8 - 0060_8 (used for converting parallel to serial outputs). This instruction shifts the contents of the cell left by one place without adding to the contents. It is used for binary 0 of a serial input bit stream or to generate a serial output bit stream from the cell overflow bit. That is, an overflow generates an output pulse on a "1" line, and the lack of an overflow generates a pulse on a "0" line.

3.2.5.2 Program Interrupts

Program interrupt, the second type of interrupt sequence, can occur between any two instructions of a program that is not already an interrupt program. An interruption consists of storing the contents of the program counter (Z), the content of the next instruction register (B), and then transferring control to a predetermined location. The interrupt program is responsible for storing the contents of any other central register that contains data that may be destroyed by the interrupting program. See Table 3-IV. Each interrupt option has a different fixed memory address associated

with it. Interrupting programs may not be interrupted, but other interrupt requests are retained and processed in the order of their priority as soon as the earlier interrupted program is resumed.

There are eleven program interrupts incorporated into the computer design. Provided certain conditions are satisfied, most interrupts cause the performance of the program in operation to be suspended, the contents of certain registers to be saved, and the next instruction (i.e., the instruction at the special address that is dedicated to the interrupt requested) to be executed.

The interrupt is mechanized through the involuntary instruction RUPT, which takes 3 MCT to perform. Other software sequences such as EDRUPT (an "extended order") can introduce an interrupt. Resumption of the interrupted program is triggered by the special purpose instruction RESUME. The individual program interrupts with their mnemonic titles, starting addresses, causes, and functions are listed below in the order of their priority:

1. T6RUPT, starting address 4004_8, is generated by the next DINC after TIME 6 (counter cell 003_8) has been reduced to -0. It is conventionally used to control the timing of spacecraft jet commands to the nearest 1/1600 second.

2. T5RUPT, starting address 4010_8, is generated by an overflow of TIME 5 (counter cell 0030_8). This interrupt is conventionally used to control cycling of computations associated with the digital autopilots. Time quantization of TIME 5 counter is 10 milliseconds.

3. T3RUPT, starting address 4014_8, is generated by an overflow of TIME 3 (counter cell 0026_8). It is conventionally used to control the performance of waitlist task. (See Section 4.2.3). Time quantization of TIME 3 counter is 10 milliseconds.

4. T4RUPT, starting address 4020_8, is generated by an overflow of TIME 4 (counter cell 0027_8). It is conventionally used to control the cycling of periodic input/output functions such as the driving of the DSKY displays. Time quantization of TIME 4 counter is 10 milliseconds.

5. KEYRUPT1, starting address 4024_8, is generated by the depression of a DSKY key and detected from the bits of Channel 15 (main panel DSKY on the command module and the lunar module DSKY). It is used by software to initiate processing of a keyboard input.

6. KEYRUPT2, starting address 4030_8, is generated for the command module by depression of a navigation DSKY key or depression of an optics mark button and detected from the bits of Channel 16. For the lunar module it is generated by depression of the optics mark button or by out of detent contacts of the rate of descent switch. It is used by software to initiate processing of keyboard (DSKY) inputs, optics marking, and rate of descent control.

7. UPRUPT, starting address 4034_8, is generated by completion of the shifting of UPLINK data into counter cell 0045_8. It is used by the software to start processing the UPLINK data word.

8. DLKRPT, starting address 4040_8, is generated by the end pulse that is sent to the computer from the telemetry subsystem to signal the completion of a data transmission from output channels 34_8 and 35_8. It is used by the software to load new data into these two output channels.

9. RADAR RUPT, starting address 4044_8, is generated by completion of the shifting of radar data into counter cell 0046_8. It is used by the software to start processing the radar input data.

10. HAND CONTROL RUPT, starting address 4050_8, is generated by the out of detent switch of the rotational and translational hand controller and is detected from the bits of Channel 31 and 32. This interrupt is used by the software to start processing the inputs from the hand controller.

11. GOPROG, starting address 4000_8, is caused in response to various hardware alarms that produce a hardware restart (see Section 3.6).

3.3 OSCILLATOR CIRCUIT

The prime source of timing for the computer, computer interfaces, and spacecraft central timing is a temperature-compensated, crystal-controlled oscillator operating at a frequency of 2.048 MHz. A temperature-compensating circuit was chosen instead of a temperature-controlled environment in order to conserve operating power and volume, in spite of the fact that temperature compensation requires more detailed specifications, assembly, and testing procedures. The design was capable of realizing a temperature stability that was at least a factor of 10 better than the system requirements over a temperature range of $25^{\circ}C$. The long term stability was another important design requirement, since calibration of the oscillator was not possible with the computer installed in the spacecraft. The stability or drift was specified

at ±1.7 parts per 10^9 per day. Operational experience has demonstrated the adequacy of the design in regard to both temperature and long term drift.

3.4 POWER SUPPLIES

The computer was designed to use two voltages: one for logic power supply (4V) and one for memory drive circuits (14V). If it were not for the excellent base cutoff and collector saturation voltage characteristics of modern semiconductor components, it would have been necessary to include an additional bias voltage. In the Block I design, the logic voltage was 3 Volts and the memory voltage was 13.0 Volts. The memory circuits operated on the 10-Volt difference between the two voltages. By referencing the memory drivers to 3.0 Volts, a bias was provided between logic and memory circuits that increased the noise immunity of the memory interface circuits in the high current environment. In the Block II configuration, the logic supply was increased to 4.0 Volts and the memory to 14.0 Volts.

The key to noise immunity in the Block II computer is the result of provision for low inductance in the ground return path. In the Block I configuration, the requirement for dc isolation of logic signal ground from spacecraft structure precluded satisfactorily low inductance in ground distribution. In the Block II configuration, the structure was used as a ground distribution path, greatly reducing the noise in the ground return within the computer. As a result of the reduced ground noise in Block II, some memory circuits are referenced to ground instead of the logic voltage, thus making available a higher supply voltage in the inductive loads of the memory.

The same general technique is used in both Block I and II to generate logic and memory voltages from the prime source of spacecraft power, this being 28 Vdc with a ±5 Volt tolerance.

Figure 3-8 is a simplified block diagram of the +4 (or +14) Volt power supplies. The circuit is basically a dc-to-dc converter of very high efficiency that employs pulse modulation principles and feedback loop to maintain regulation of load despite the large variation of the power input from the spacecraft. As seen in the diagram, 28 Vdc from the spacecraft fuel cell is applied to the power input circuits. These circuits contain diodes to decouple the two spacecraft power buses and an LC filter for isolation of noise generated either internal or external to the computer. The output provides 28 Vdc to the computer interfaces and the alarm circuits (see Section 3.6). 28 Vdc is applied to the power switch, and also to a preregulator to develop stable voltages for the reference and comparator circuits of the supply.

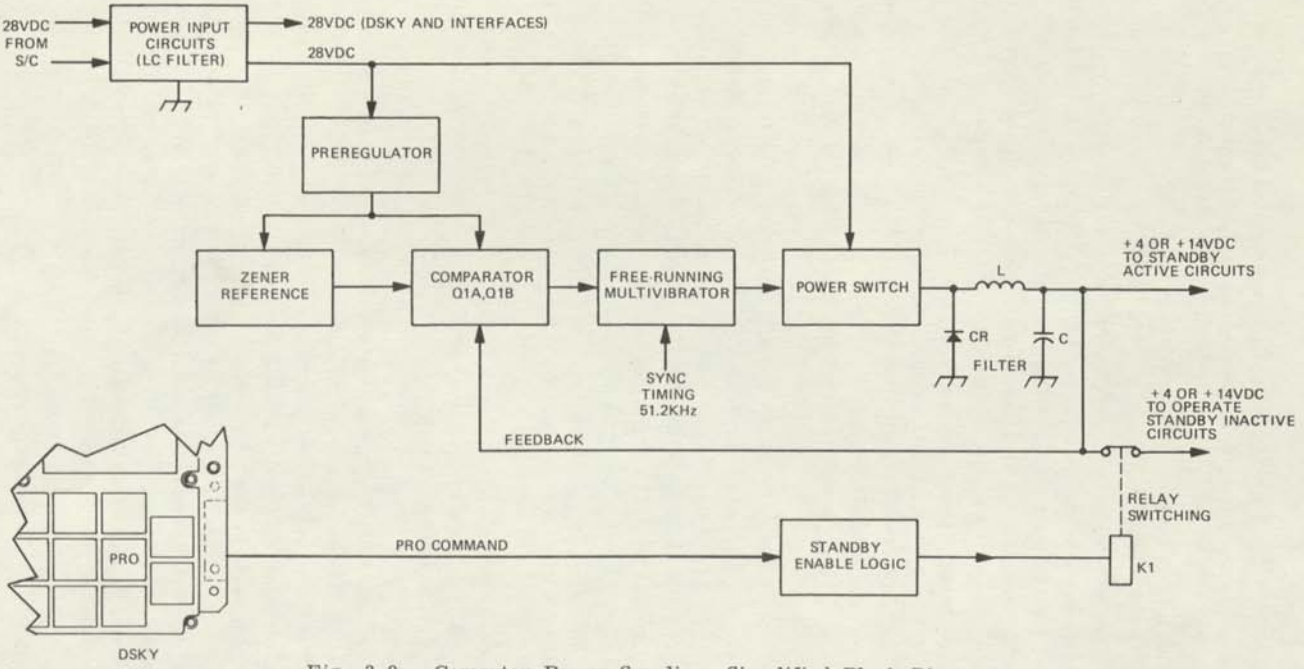


Fig. 3-8 Computer Power Supplies, Simplified Block Diagram

The comparator is a differential amplifier using a zener diode regulator as the reference. Any difference between the feedback from the output and the reference voltage modifies the pulse width of the free-running multivibrator that is synchronized to the computer clock by a 51.2 KHz timing signal. The output of the multivibrator switches the 28 Vdc on and off. The resulting pulsed output is then applied to a filter network consisting of a diode-inductor-capacitor combination that smooths the pulses of current to form a regulated dc voltage. The computer requires two identical modules, one each for the two different voltages. The location of the modules in the computer tray determines whether the output is 4 or 14 Volts by the external interconnections to the zener regulator.

In the standby mode, the only operating computer circuits are the oscillator, some interfaces, scaler, clock divider circuits, and the power supplies, since these circuits are necessary for the maintenance of real-time and synchronization signals to other spacecraft systems. The remaining computer circuits are inactivated by the standby switching circuits. A sequence of DSKY operations instruct the program of the computer to prepare for standby and enables the standby logic. Subsequent depression of the PRO key on the DSKY generates a STBY command signal in the command logic.[1] This signal is routed to the standby switching circuits, causing relay K1 to open the operating supply bus. Reactivation of the computer requires another depression of the PRO key.

The power supply design follows the general rules for minimum volume and power consumption, but lacks ground isolation between the 28 Vdc power input and the power return of the various output signals. The lack of ground isolation is a violation of the electromagnetic interference specification that illustrates the problem of changing requirements. For example, late introduction of electromagnetic interference requirements, schedule pressures, and the desire to keep the volume minimal precluded a design change that would have produced a ground isolated supply. When the requirement for ground isolation was imposed, design development for an isolated supply was initiated, but the design was not completed beyond the breadboard stage because of uncertainties in the ability to package the power supply within the allotted space.

3.5 MEMORY CIRCUITS

As indicated in the functional description in Section 2.0, the computer memory consists of a 2048-word erasable memory and a 36,384 fixed memory. The electrical design details of these circuits are described in the following paragraphs.

---

1. The standby command signal is enabled as part of Power Down program P06.

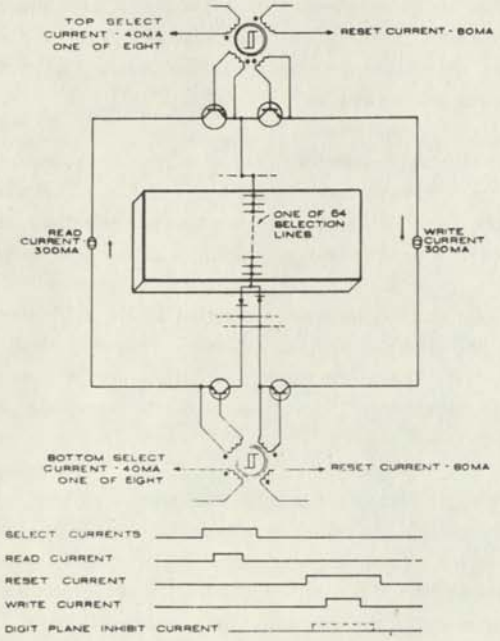


Fig. 3-9 Erasable Memory Current Switching

3.5.1 Erasable Memory

The erasable memory was inherited from its core-transistor logic ancestor. It is a conventional coincident-current ferrite core array whose ferrite compound yields a combination of high squareness and a comparatively low sensitivity to temperature. Moreover, the silicon transistor circuits, that drive the memory, normally vary with temperature in such a way as to match the requirements of the cores over a wide range, from 0° to 70°C, thus requiring no special circuit for temperature compensation. Coincident current selection affords an economy in selection circuitry at the expense of speed in comparison with linear (word) selection. This is advantageous to the computer, where the memory cycle time is already long, largely due to the fixed memory. The 2048-word array is wired in 32 x 64 planes with no splices in the wires for highest reliability. The planes are folded to fit into a 9 cubic inch module along with two diodes for each select line. Bidirectional currents are generated in each selection wire by a double-ended transistor switching network. The selection of 1 wire in 32 is made by 12 switch circuits in an 8 x 4 array; the selection of 1 wire in 64 is made in an 8 x 8 array. The operation of the switching network is illustrated in Figure 3-9. The core driver transistors are driven by magnetic cores, that offer two advantages: small size and storage of address for data regeneration. Again, this circuit economy is realized at the expense of speed. The timing of the currents, that operate the switch cores, is based on the duration of the write current in the memory array, and is 2 μsec. Two current drivers with controlled rise times, one for reading and one for writing, are used on each of the two drive select networks. Sixteen more such drivers are used to drive the digit lines that control the writing phase of the memory cycle. Current amplitude is governed by the forward voltage drop e_2 in Figure 3-10 across a silicon junction and emitter resistor R_2, so that temperature compensation for temperature-dependent coercive force of the ferrite core is achieved without any circuit complications.

The output signal from the memory cores has an amplitude of about 50 millivolts. Transformer coupling to the sense amplifiers provides a common mode noise rejection and a voltage gain of 2. The sense amplifiers (see Figure 3-11) have a differential first stage operated in a linear (class A) mode. A second stage provides threshold discrimination, rectification, and gating or strobing. Three reference voltages are generated for the sense amplifiers by a circuit whose temperature characteristics compensate for amplitude and noise changes in the memory.

The integrated circuit sense amplifiers realize a number of advantages inherent in single-chip semiconductor circuits. Differential amplifiers pose a special problem in component matching, both internally to achieve balance in a single amplifier and

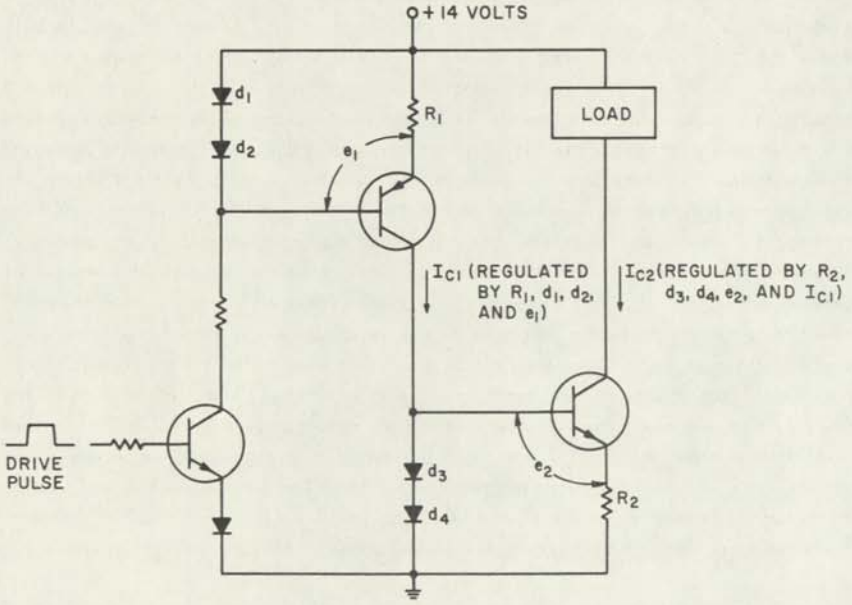


Fig. 3-10 Regulated Pulsed Current Driver Circuit

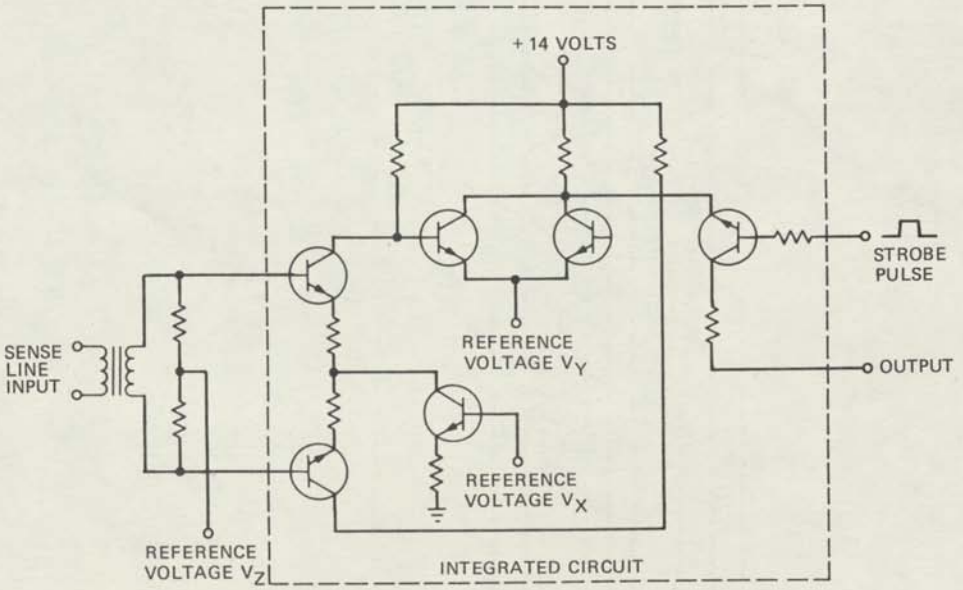
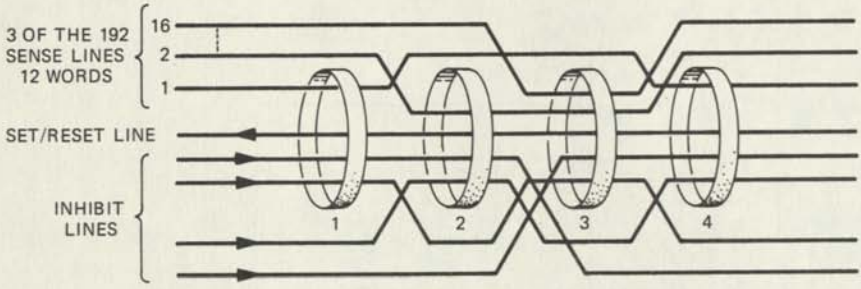


Fig. 3-11 Sense Amplifier Circuit



TRUTH TABLE

Core No. \ Sense Line No.	1	2 . . .	16
1	1	0	0
2	0	1	0
3	0	1	1
4	1	0	0

Fig. 3-12 Core Rope Simplified Schematic

among a group of amplifiers in achieving uniform behavior for common reference voltages. In discrete-component amplifiers a great deal of time and effort are involved in specifying and selecting matching sets of circuit components. In an integrated circuit, however, balance is readily achieved owing to the extremely close match between transistors on the same silicon chip. A similar situation holds for uniformity from one amplifier to another, the difference being easily compensated for by external trimming resistors and reference voltages. The small size of the integrated circuit amplifier is advantageous in obtaining temperature tracking between sense amplifiers and memory, since it is not difficult to keep them at a temperature close to that of the memory cores. Where sense amplifiers have historically been the "weak link" of computer memory systems, the integrated sense amplifier has already been proven to be at least on a par with the rest of the memory electronics.

3.5.2 Fixed Memory

The computer fixed memory is of the transformer type and was developed at MIT/IL. It is designated a "core rope" memory owing to the physical resemblance of early models to lengths of rope. Incorporated into its wiring structure is an address decoding property. Because of this internal address decoding, the cycle time of the core rope is not as short as that of some other transformer memories whose address decoding is external. The resulting bit density is extremely high—approximately 1500 bits per cubic inch, including all driving and sensing electronics, interconnections, and packaging hardware. This high density storage is achieved by storing a large number of bits in each magnetic core. A stored bit is a 1 whenever a sense wire threads a core, and is a 0 whenever it fails to thread a core. Figure 3-12 is a simplified core rope schematic indicating address decoding (inhibit lines) and sense line wiring. The total number of bits is the number of cores multiplied by the total number of sense lines. The memory is composed of six modules. Each module contains 512 cores and 192 sense lines and hence contains 192 x 512, or 98,304 bits of information. Since this information is permanently wired in during manufacture, the memory is nondestructable except by physical destruction or by failure of one or more of a number of semiconductor diodes whose functions are described below.

The means by which a single core in a module with 512 cores is caused to switch (addressed) is an extension to that illustrated in Figure 3-12. A switching current is applied that attempts to set 128 cores (set/reset line). Four such current lines serve a 512-core module. Inhibit currents are applied simultaneously to either the first or second half of each group of 128 cores. Two inhibit lines exist for this

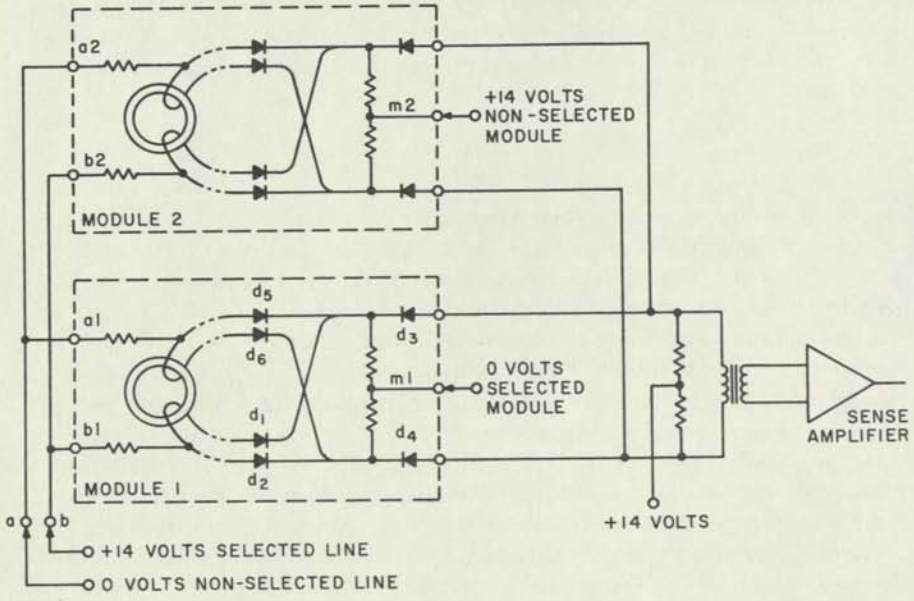


Fig. 3-13 Rope Sense Line Switching Simplified Diagram

purpose. Another inhibit current is applied to either the first or second half of each half-group. Two more inhibit lines exist for this purpose. Six more pairs of inhibit lines exist for the purpose of reducing the uninhibited core groups successively by halves, until only one core is left uninhibited. One member of each inhibit pair carries current at a time. There are eight pairs in all to select among 2^7 cores, of which seven pairs correspond to the seven low order address bits. The eighth pair is logically redundant, being selected by the parity of the address. The redundancy is used to reduce the amount of current required in each inhibit line. After the selected core has switched and the inhibit currents are removed, a reset current is passed through all cores. Only the core that was just set will change state, and the sense amplifiers may be gated on during either the set or the reset part of the cycle to read information out of the memory. The noise level during reset is lower than during set for a number of reasons, but the access time (the time it takes to read the memory after the address is available) is longer. Both ways have been used. The Block II design uses the longer access time and must produce the address earlier as a result.

In the operation of the rope memory, a single core is switched, thereby inducing a voltage in every sense line that threads the core. Only one word is read at a time, so that of the 192 sense lines, only 16 are connected to the sense amplifiers to detect that have voltage and therefore store 1's. Each core stores 12 words, and within each module, a switching network is included in order to transmit no more than 1 of the 12 to the module's output terminals. The principle of the switching network is illustrated in Figure 3-13. It consists of diodes and resistors connected so as to block the sense line's output when sense line diodes are reverse-biased, as in the case of d_5 and d_6; and to transmit it when the sense lines diodes are forward-biased, as in the case of d_1 and d_2. A second-stage switch composed of d_3 and d_4 is used to select one of the six module outputs to be transmitted to the sense amplifers. Only the selected line in the selected module is transmitted. All others are blocked by one or two sets of reverse-biased diodes. All these selection diodes are physically located in the rope modules to minimize the number of terminals necessary for each module. The application of selection voltages to the line and module select terminals is a part of the address decoding that is external to the rope. The balance is internal.

3.6 ALARMS AND FAULT DETECTION

The computer self-checking circuits were designed to provide a fault detection capability. The outputs of the detection circuits generate a restart or warning indication for display and cause the computer to transfer control to address 4000_8

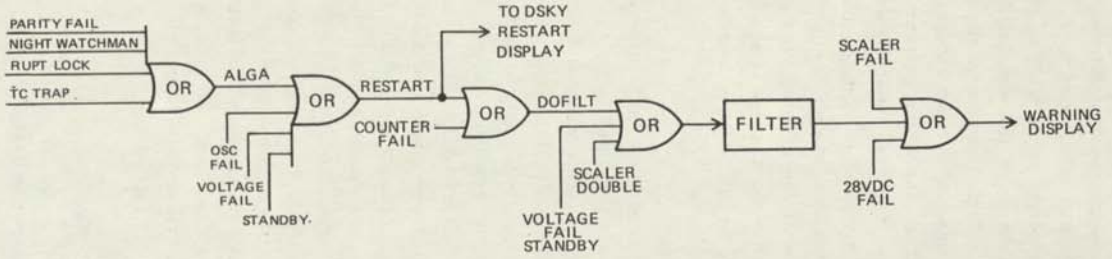


Fig. 3-14 Failure Detection Logic

as soon as the restart condition disappears. The restart display can be cleared by depressing the error reset (RSET) key. The primary purpose, therefore, is to provide detection of faults and recovery by restarting the software. If the fault is a hard failure, the restart display cannot be cleared by depressing the RSET key. This procedure is adequate for operational type fault detections, since the only purpose operationally is to indicate the necessity to switch to a backup mode of operation.

Late in the computer development and operational usage, a special module, called the restart monitor, was designed to plug into the test connector. This module would store the individual indications of fault available at the test connector in a register that was addressable via software (Channel 77). The contents could then be read out for more complete diagnostic information. The primary concern in considering the restart module is spacecraft checkout, where the diagnostic capability is very limited. Experience has shown that many failure indications, whether induced by external stimuli such as power transients, procedural errors, or induced by internal software or hardware problems, have insufficient information for diagnosis. The restart monitor module would provide additional information by breaking down the cause of the restart indication into the various faults, the OR of which forms a restart.

The fault detection circuits comprise two categories: those that are derived logically, and those that are derived using analog-type detection circuitry. The former circuitry is distributed within the logic modules of the computer and the latter in the alarm module. Figure 3-14 shows the functional interconnection of these circuits to provide the various failure control signals internal to the computer and failure display outputs to the DSKY. The conditions that cause the circuits to indicate failure are defined as follows:

1. Parity Fail — Occurs if any accessed word in fixed or erasable memory, whose address is 10_8 or greater, contains an even number of 1's.

2. Night Watchman — Occurs if computer software should fail to access address 67_8 at least once every 0.64 second.

3. Rupt Lock — Occurs if interrupts do not occur frequently enough, or if any interrupt lasts too long. Interrupts must occur at least every 140 milliseconds and must not last any longer than 140 milliseconds.

4. TC Trap — Occurs if TC or TCF instructions are run too frequently or not frequently enough. TC or TCF instructions must occur at least every 5 milliseconds but not continuously for a 5-millisecond period.

5. OSC Fail — Occurs if the oscillator stops or if the computer is placed in Standby. When computer is put into Operate mode and the oscillator starts, there is a 250-millisecond delay before an inhibit to computer operation is removed.

6. Voltage Fail — Occurs when the 4, 14, or 28 Volt supplies go outside a predetermined set of limits for a period of time between 150 to 470 μsec. The limits and timing were set such that the computer can survive a power transient without loss of memory data.

7. Standby — Occurs when the computer is commanded to go into Standby. The signal causes a restart and turns off the +4 and +14 Volt supplies, thus putting the computer into a low power mode where only the scaler and a few auxiliary signals are operative.

Items 1 through 7 above comprise the signals that internally transfer control to address 4000_8 and externally illuminate the restart display on the DSKY.

8. Counter Fail — Occurs if counter increments occur continuously for 625 μsec or else fail to occur following a counter increment request.

9. Voltage Fail/Standby — Occurs if there is power failure during the standby mode of operation.

10. Scaler Double — Occurs if the stage 10 (100 pps) scaler operates at double frequency. Scaler Double with Scaler Fail provides a check on the timing for all logic alarms.

Restart and signals 8 through 10 are ORed, then filtered before being used as one input to the warning display.

11. Scaler Fail — Occurs if stage 17 (the stage in the scaler whose output is a 1.28-second period) fails to produce pulses, thus providing a final gross check on the timing for all logic alarms.

12. 28V Fail — Occurs if all power is lost in the computer. Loss of power opens a relay to indicate the warning condition.

The warning is then the OR of signals 11, 12, and the above filter signals as indicated in Figure 3-14. Other failure detectors and indicators are provided within the logic of the computer as follows:

13. PIPA Fail — Occurs if no pulses arrive from a PIPA during a 312.5 msec period, if both plus and minus pulses occur, or if a long time elapses without at least one plus pulse and at least one minus pulse. (A long time lapse is a period of at least 1.28 sec).

14. Uplink Too Fast — A trap circuit prevents uplink data pulses from reaching the counter priority chain with a period less than 156 μsec. Channel 33 bit 11 is set indicating that the occurrence of data pulses is too fast.

15. Downlink Too Fast — A trap circuit prevents downlink end pulses from causing interrupt any more frequently than 10 milliseconds. Channel 33 bit 12 is set indicating that the occurrence of end pulses is too fast.

Several G&N system alarm indicators are inputs to input channels of the computer for software operations and for computer generated display. These are:

1. IMU CDU FAIL ..... Channel 30, Bit 12
2. IMU FAIL ..... Channel 30, Bit 13
3. TEMP. IN LIMITS ..... Channel 30, Bit 15
4. OPTIC CDU FAIL ..... Channel 30, Bit 07

3.7 INTERFACE METHODS

Information transfer between the computer and its environment uses a substantial portion of the computer's hardware and consumes a significant fraction of the computation time. Figure 3-15 shows the guidance, navigation and control interconnections in the command module, and Figure 3-16 does the same for the lunar module. These figures illustrate the complexity of this interface with respect to the variety of equipment types involved and their attendant interface signals. In addition, the subsystem equipment types had a variety of data formats that were convenient or peculiar to the subsystem. To accommodate the various data formats, the interfaces were designed around the logical operations provided by the incrementing and shifting capabilities of the counters (see Paragraph 3.2.5) and the capability of reading and writing into the channels (see Paragraph 3.2.3). The counters were used for sending or receiving incremental signals and for parallel-to-serial conversion of whole word

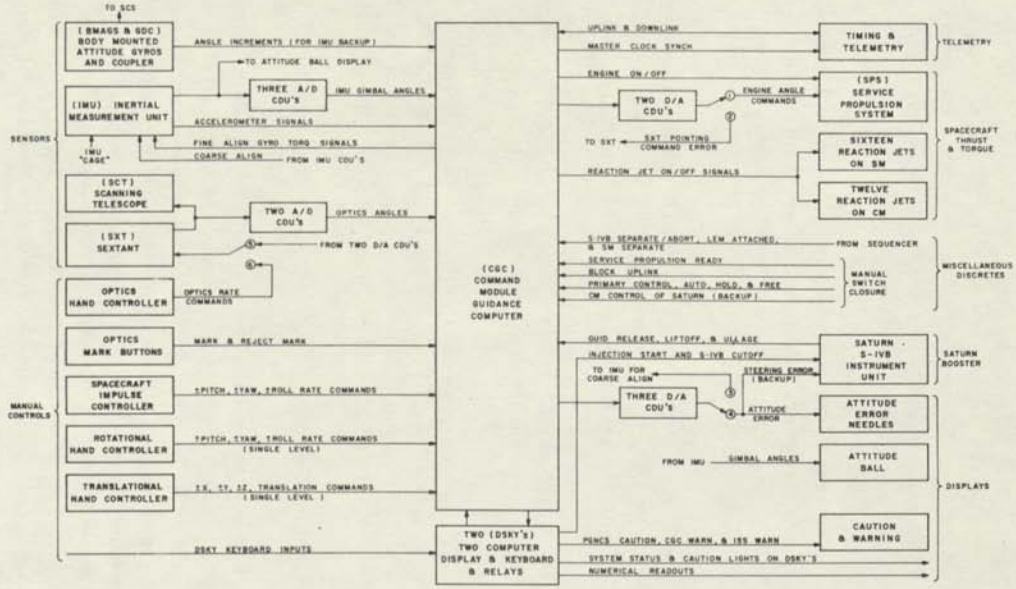


Fig. 3-15 GN&C Interconnections in Command Module

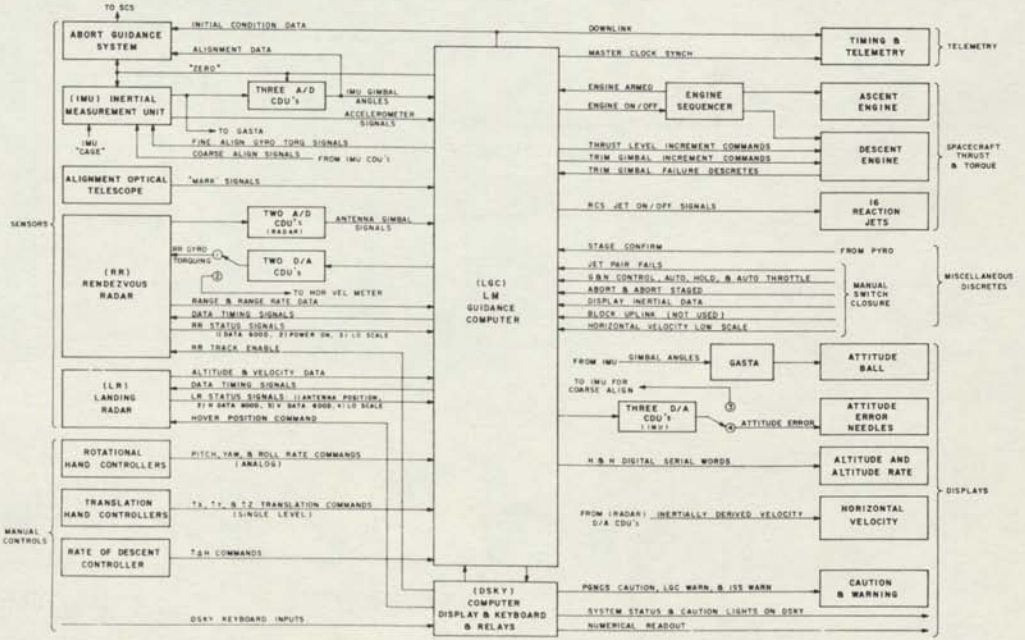


Fig. 3-16 GN&C Interconnections in Lunar Module

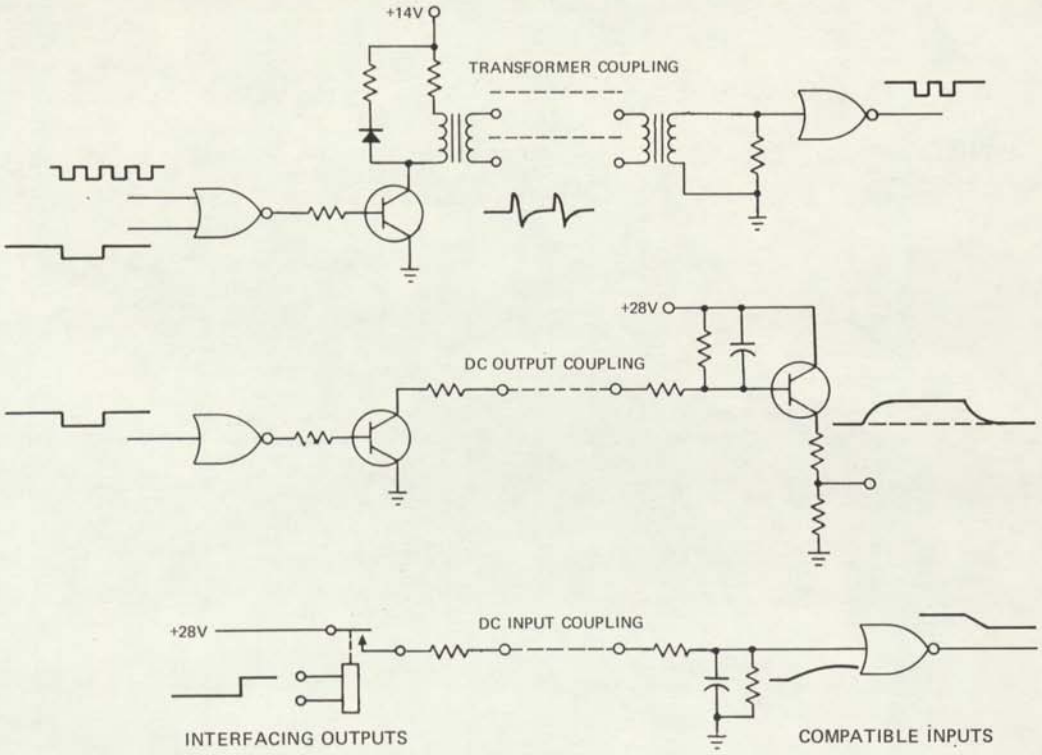


Fig. 3-17 Examples of Interface Circuits

transfers. The channels were used for storage of discrete data bits and in some cases parallel transfers of whole words up to 15 bits in length as indicated in Table 3-VIII.

3.7.1 Circuits

To accommodate the multiple circuit configurations and the requirements of multiple applications of computer interfaces, a concerted attempt was made on the part of the computer design and system integration groups to reduce the number of different circuits involved in the interface to four basic types.

Figure 3-17 illustrates the four basic interface circuits, their interconnection to the computer logic, and the compatible source or receiver. The four (two outputs and two inputs) shown interconnected with the logic symbol for a NOR gate are the circuit types contained within the computer. The other outputs or inputs are the compatible source or receiver.

The transformer-coupled circuits were used in the interfaces, where timing was critical, or where data were transferred in the form of short pulses. Examples of these were the timing signals to the spacecraft and other GN&C subsystems.

The direct-coupled interfaces were used where slow response (1 millisecond) data were being transferred. Most discrete commands are of this type. This circuit was used for its simplicity and minimum number of components.

The DSKY provided relay contact closures for a few cases. The computer/DSKY interface made use of the direct-coupled circuits.

3.7.2 Data Format Conversion

Incremental information transfer is similar to serial information transfer in that a sequence of pulses is transmitted over a single channel. It differs in that each pulse represents the same value, or weight, as opposed to serial transfer where two adjacent pulses differ in weight by a factor of two. The incremental receiver counts pulses to form a word, where a serial receiver shifts pulses to form a word. Both types of receivers are employed in the computer. In each case except the downlink, the implementation is accomplished using the cycle-steal operation of the special counter registers.

TABLE 3-VIII

CHANNEL REGISTER FUNCTIONS

OCTAL ADDRESS	NO. OF BITS	PURPOSE
1	16	Identical to L register (Table 3-IV)
2	16	Identical to Q register
3	14	High order time scaler (Fig. 3-4)
4	14	Low order time scaler
5	8	Outbits, pitch and yaw RCS Jets
6	8	Outbits, roll RCS Jets
7	3	Fixed memory extension bits (Table 3-V)
10	15	Outbits, DSKY display command
11	15	Outbits, DSKY status lights, mode relays and engine control
12	15	Outbits, GN&C and LM radar mode control also LM engine gimbal trim
13	15	Outbits, LM radar selection, CM range unit and AGC interface mode control
14	15	Outbits, output counter control for IMU, LM thrust and LM altitude meters
15	5	Inbits, LM and CM Main DSKY keyboard
16	7	Inbits, CM Navigation DSKY keyboard, optics mark, and LM rate of descent
30	15	Inbits, GN&C and Spacecraft mode
31	15	Inbits, handcontrollers and Spacecraft mode
32	15	Inbits, CM thrust impulse control and LM thruster fail
33	15	Inbits, CM optics, LM radar and failure detection
34	16	Outbits, downlink first word
35	16	Outbits, downlink second word
77	9	Inbits, restart monitor bits (Section 3.6)

The downlink and the uplink are two examples of whole word serial data transfer. These interfaces provide the ground control data link for both prelaunch and inflight operations. The mechanization of the two interfaces differs due to the fact that the downlink data requires diagnostic, status, and mission-related information, whereas the uplink consists primarily of commands. A much greater quantity of data is therefore downlinked to the ground than is uplinked from the ground. The downlink, having a high data rate, is designed to use a minimum of computer computational time, whereas the uplink having a low data rate uses the counter registers. The special circuits of the downlink were loaded with two words at a time via channel addresses 34 and 35. The logic of the downlink interface serializes and synchronizes the 32 bits and provides an 8-bit code. The complete operation is subject to command signals provided by the spacecraft central timing system. At the end of a transmission, the timing system provides an end pulse that terminates the transmission and signals the computer via interrupt in order to prepare for the next two-word transmission.

Incremental information transfer was adopted as a means of analog data transmission in order to maintain high precision with relatively simple techniques of A/D conversion. In conversion of gimbal angles and optic angles, an intermediate transformation is accomplished as part of the A/D conversion in the coupling data units. Refer to Figure 3-18. The inputs to the coupling data unit (CDU) are the electrical resolvers. The outputs are positive or negative increments of angle that are added via the counter register of the computer. The CDU angle counter and the computer counter are initialized as part of system erection. Incremental changes in the gimbal angle are then summed in the computer to maintain the gimbal angle as indicated by the resolver. Incremental transfer is also used for angle commands from the computer to the gyros, the CDUs, thrust control, and certain display functions in the spacecraft. Figure 3-19 illustrates the techniques for gyro torquing, where a pulse of torquing current of duration Δt generates a change in angle of $\Delta\theta$ in the gyro. Figure 3-20 illustrates the same technique in the CDUs to generate a dc output with a magnitude proportional to the $\sum Ni$, where Ni is the sequence of numbers loaded into the counter register.

3.7.3 Interface Example

The radar-computer interface comprehensively illustrates all the techniques employed in interface design. The computer provides gating, timing, and synchronization for the various radar-derived quantities to be measured and read into the computer memory. These signals, as shown in Figure 3-21, use the transformer-coupled interface because timing is critical when compared with the

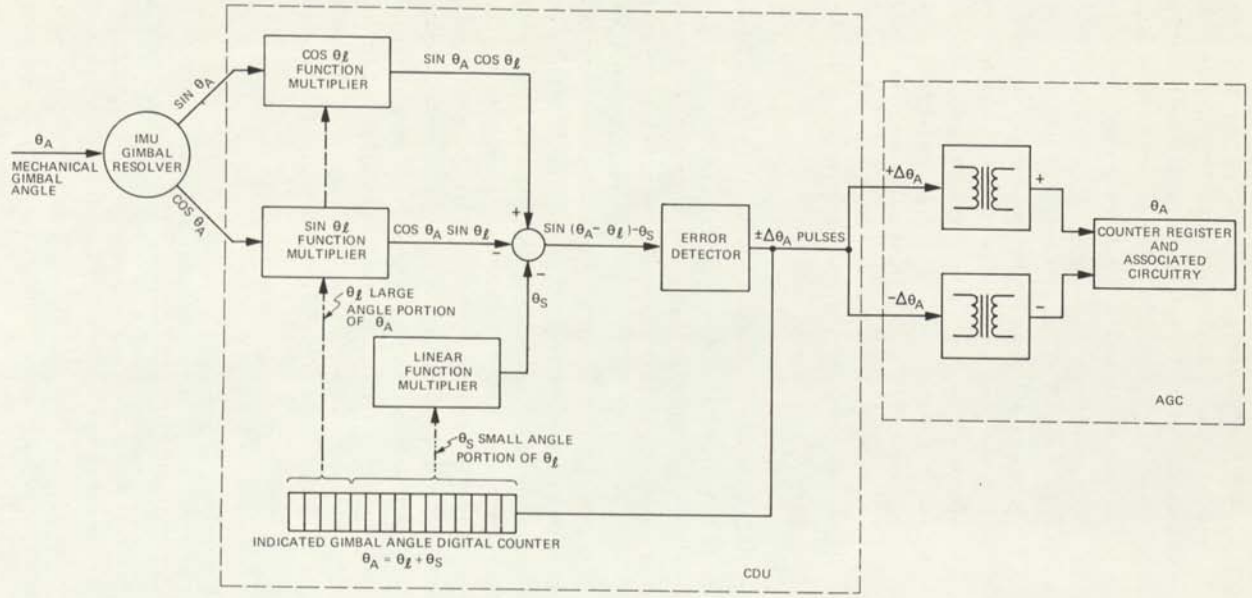


Fig. 3-18 Gimbal Resolver Coupling Data Unit - Computer Interface

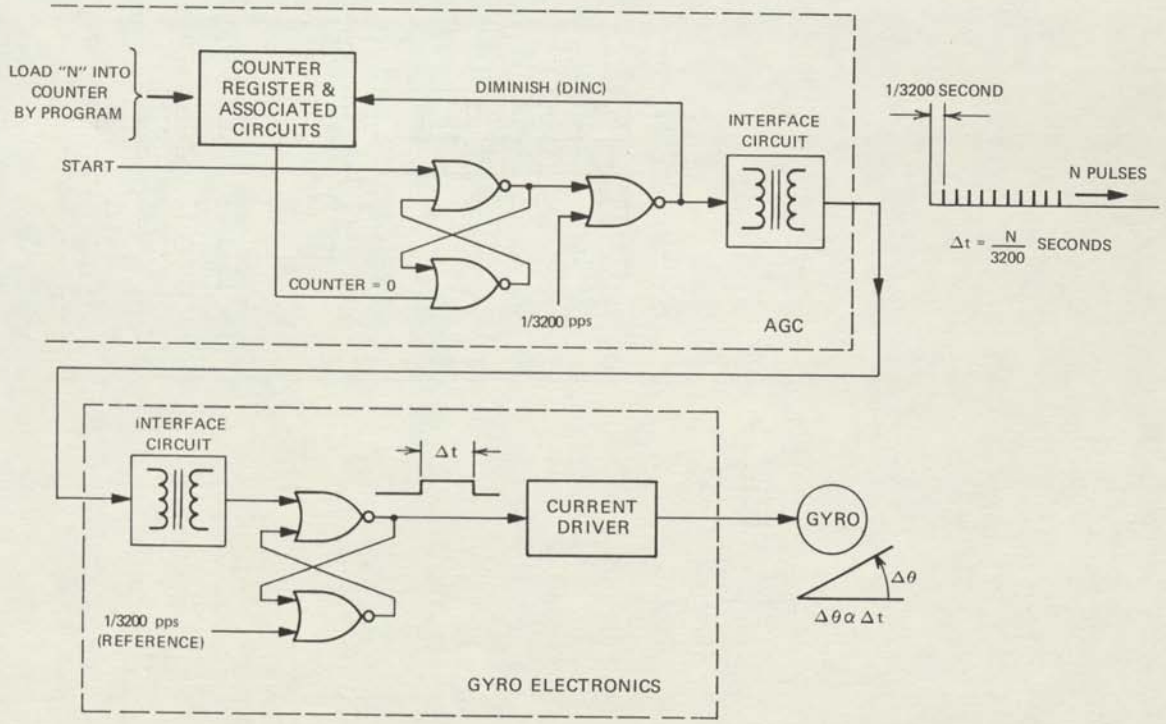


Fig. 3-19 Digital to Analog Conversion by Time Duration of Precision Current

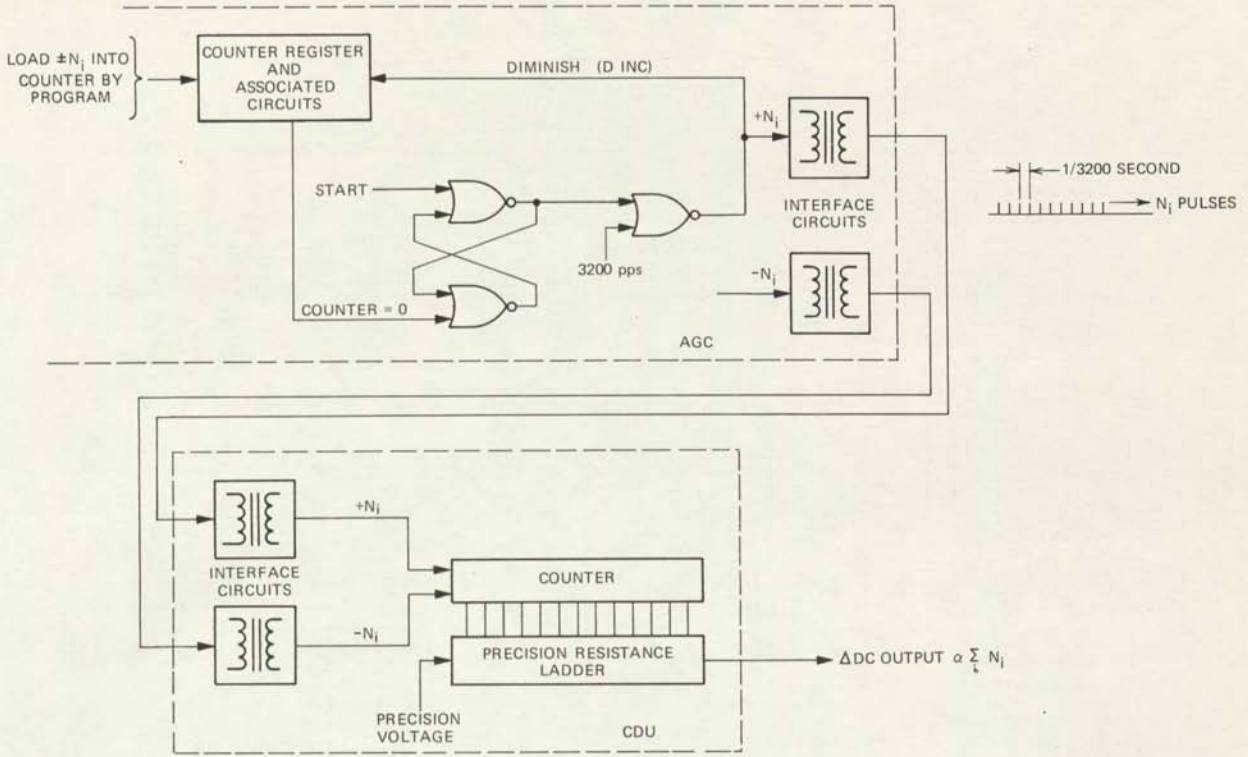


Fig. 3-20 Digital to Analog Conversion by Resistance Ladder

radar mode control inputs and mode command outputs. Since the mode signals are not time-critical, they use the less complicated dc interfaces.

To read the radar data, a sequence is initiated under program control by selecting one of the six radar parameters through the setting of appropriate bits (1 to 3) in output channel 13. Setting bit 4 of channel 13 activates the gating and sequencing circuits, which then transmit the necessary outputs to the radar. (See Figure 3-21). The sequence is completed when the data are read into the counter register of the computer, and the sequencing circuits provide a RADAR RUPT signal to alert the program that data are available.

Figure 3-22 is a timing diagram showing the output and control signals internal to the computer. The sequence is initiated by the program at a random time with respect to the timing signals (T_5) that start the sequence. Following the setting of bit 4 in channel 13, the logic generates 256 gate pulses that the radar uses in conjunction with the continuous gate reset pulses to generate an accurate 80-millisecond gate. During this time, the radar performs a measurement and loads the data into a shift register. The data are shifted out and into the computer radar counter using the 15 radar sync pulses. The completion of the reading is signaled by a RADAR RUPT generated in the computer logic. The RADAR RUPT also resets the activity bits and sequencing circuits in preparation for the next command to read data. Since the complete cycle takes approximately 90 milliseconds, the program is limited to no more than 10 readings per second.

3.8 DSKY DESCRIPTION

The display and keyboard, or DSKY, is in some respects similar to an integral component part of the computer, yet it is operated through the interface circuits as an external subsystem and is mounted remotely on the display panel of the spacecraft. The DSKY provides the man-machine interface for commands to the computer through a keyboard and a set of displays for the transfer of information from the computer to the human operator.

The principal part of the DSKY display is a set of three numeric light registers, as shown in Figure 3-23. Each register contains five decimal digits consisting of segmented electroluminescent lights. Five decimal digits are used, so that a computer word of 15 bits can be displayed in either decimal or octal number systems. In addition, there are three two-digit numerical displays, that 1) indicate the major

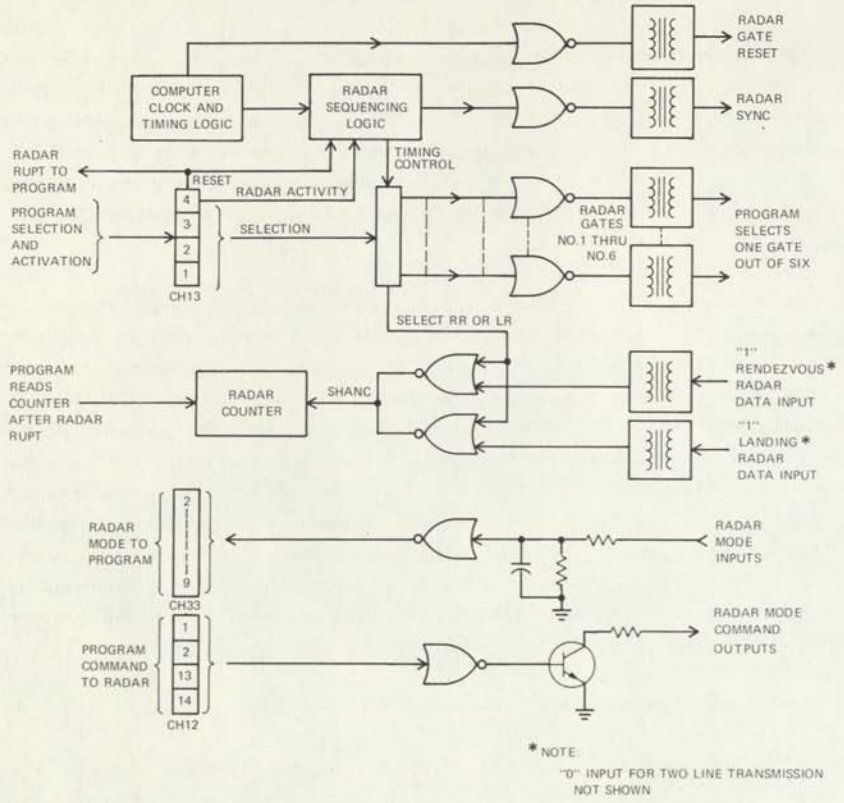


Fig. 3-21 Radar-Computer Interface Functional Diagram

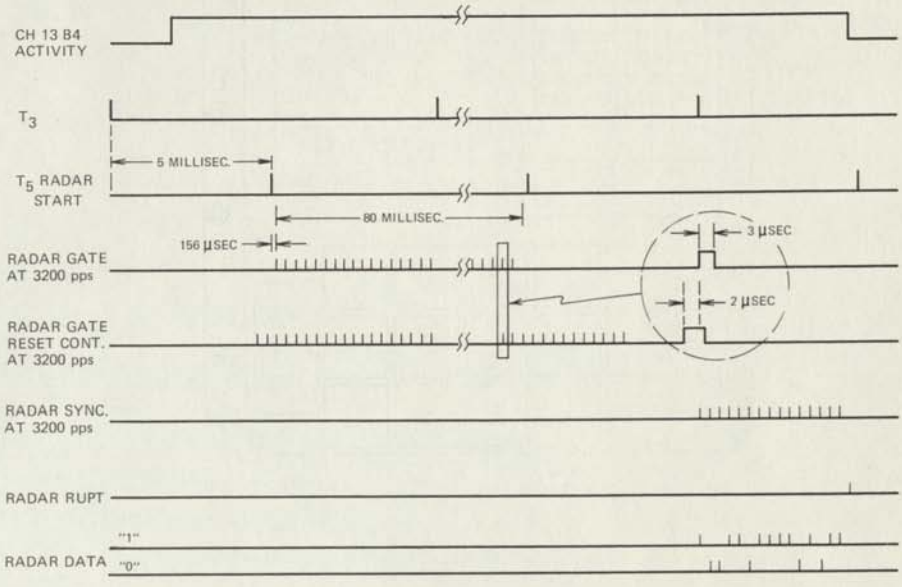


Fig. 3-22 Radar-Computer Interface Timing Diagram

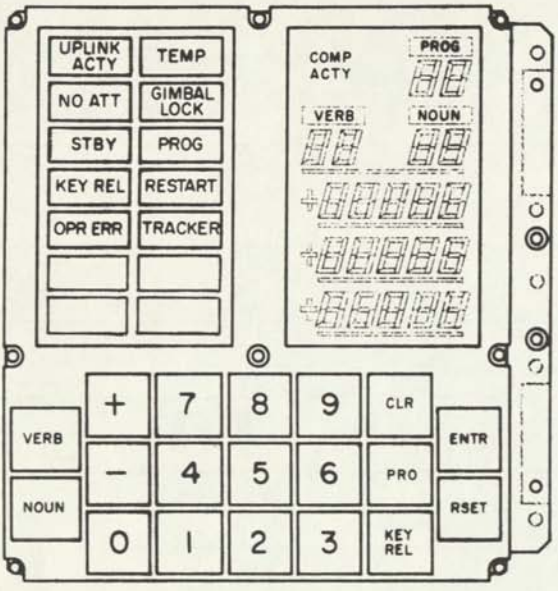


Fig. 3-23 DSKY Display and Control Panel

program in progress, 2) indicate the verb code (which shows what action is to be taken), and 3) indicate the noun code (which shows what action is to be applied).

Electroluminescent lights are used in the DSKY because they are small, easy to read, and require relatively little power. The lights are driven by an 800-Hz amplifier in the DSKY with the gain or output amplitude controlled by the spacecraft light intensity control. The display is switched under computer control by miniature latching relays, whose contacts are used for the decoding logic between the computers five-bit binary data and the seven light segments of each decimal digit in the display. See Figure 3-24. In addition, the latching relays provide memory with no consumption of power following the switching action.

The relays are selected and switched using a 15-bit parallel direct-coupled interface controlled by channel 10. See Section 3.7. The relay selection is accomplished by a double-ended selection matrix. Four of the 15 bits are used for selecting 1 row out of 12 (relay word line), and the remaining 11 bits are used to set or reset the 11 latching relays in the selected row. See Figure 3-25. Five relays are used for each decimal digit. Therefore, the 11 relays control two decimal digits and one other discrete function such as displaying the polarity of a decimal number.

The DSKY also provides status displays, as indicated in Figure 3-23. This display panel is lighted using filament type bulbs, so that white and yellow color codes can be displayed signifying status and caution signals. The power to light these bulbs is provided by the spacecraft but is switched via the relays in the DSKY. Some of these relays are the latching type and are switched by the matrix selection, others are nonlatching relays that are controlled by dedicated channel bits or logic circuitry such as the restart logic. See Section 3.6.

The DSKY contains a total of 120 latching relays and 12 nonlatching relays to provide for the display functions described. A few of these relays, that are not used for DSKY display, provide contact closure interfaces between the DSKY and the spacecraft subsystems. The computer warning and the inertial subsystem warning indications to the spacecraft caution and warning panel are examples of this type.

The keyboard has 19 keys, as illustrated in Figure 3-23, that provide the communication between the operator and the computer. Commands and requests are made in the form of sentences, each with an object and an action, such as "display velocity" or "load desired angle." The first is typical of a command from man to

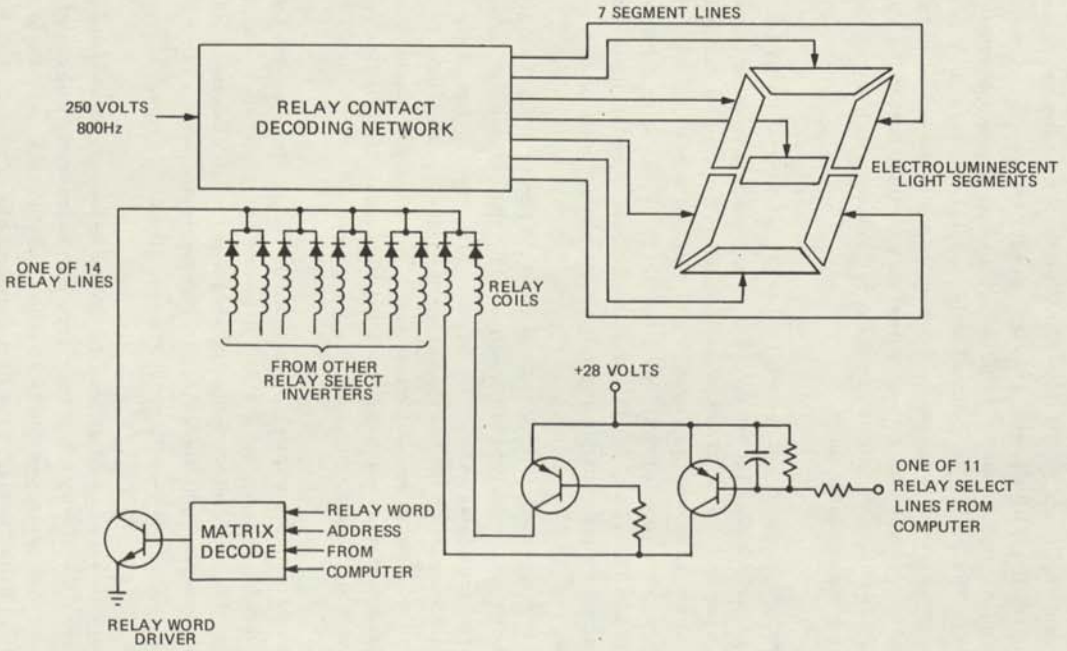


Fig. 3-24 DSKY Display Circuits - Simplified Diagram

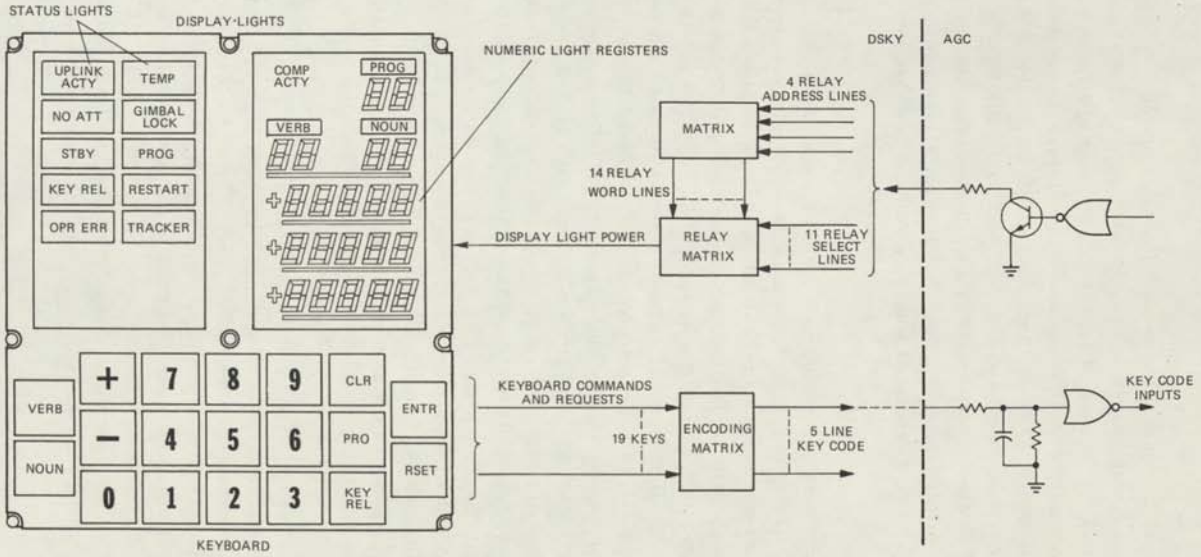


Fig. 3-25 DSKY Block Diagram

machine; the second is typical of a request from machine to man. The DSKY is designed to transmit such simple commands and requests made up of a limited vocabulary of "verbs" or "nouns." These verbs and nouns are displayed by number rather than by written word. To command the computer, the operator depresses the VERB key followed by two octal digit keys. This enters the desired verb into the computer, where it is stored and also sent back to the DSKY to be displayed in the verb lights. The operator next enters the desired noun in similar fashion using the NOUN key, and it is displayed in the noun lights. When the verb and noun are specified, the ENTR key is depressed, whereupon the computer begins to take action on the command.

When the computer requests action from the operator, a verb and a noun are displayed in the lights, and a relay is closed that causes the verb and noun lights to flash on and off, so as to attract the operator's attention and inform him that the verb and noun are of computer origin. To illustrate the usefulness of the requesting mode, consider the procedure for loading a set of three desired angles for the inertial measurement unit (IMU) gimbals. The operator keys in the verb and noun numbers for "load 3 components, IMU gimbal angles." When the ENTR key is depressed, the computer requests that the first angle be keyed in by flashing and changing the verb and noun lights to read "load first component, IMU gimbal angles." Now the operator keys in the angle digits and, as he does so, the digits appear in light register number 1. When all five digits have been keyed, the ENTR key is depressed. The verb and noun lights continue to flash, but call for the second component; and when it has been keyed and entered, they call for the third component. When the third component has been entered, the flashing stops, indicating that all requests have been responded to. If a mistake in keying is observed, the CLEAR key allows the operator to change any of the three angles previously keyed until the third angle has been entered.

The interface between the DSKY keyboard and the computer employs the standard dc interfaces in the computer. See Figure 3-25. The DSKY key depressions are coded in the DSKY into a five-bit word. In the command module application each of the two DSKYs interfaces with a separate channel of the computer. The main panel DSKY interfaces with channel 15, bits 1-5, and the navigation panel DSKY interfaces with channel 16, bits 1-5. Each input provides an interrupt such that the computer program does not have to scan these bits for keyboard activity. See Section 3.2.5.2.

3.9 COMPUTER MECHANICAL DESCRIPTION

The mechanical construction of the computer was determined in part by the following design requirements:

1. Pluggable module construction using welded internal interconnections
2. Replacement of fixed memory modules when the computer is installed in the spacecraft
3. Complete environmental sealing of the assembled computer
4. Thermal control provided by a cold plate structure in the spacecraft
5. Electrical interconnections performed by automatic machine wirewrap techniques using wrapost connectors.

The construction feature common to both the AGC and DSKY is the use of connectors and jacking screws. Connector construction more or less dictated the computer form factor and many of the other design features.

This method of electrical connector construction is used to allow automatic machine wirewrapping of electrical interconnections. Solid male and female "wrapost"-type contacts (Malco connectors) are inserted in plastic insulators, that in turn are installed in a drilled metal plate. The contacts are arranged in columns and rows to form connectors. In general, the male contacts are installed in the housing of a plug-in module, and the female contacts are installed in the connector plate that supports the modules and provides for the interconnection of the modules. Installing the contacts and insulators in a metal plate provides the rigidity and close tolerance spacing necessary to accommodate automatic wirewrapping of the contacts by a Gardner-Denver wirewrap machine.

The wrapost end of the male and female contacts is square in cross-section. Wrapping the wire around the sharp corners of the wrapost forms a cold-welded joint that provides secure electrical and mechanical connections. The female contacts and wrapost are made of gold-plated beryllium-copper to facilitate the formation of the cold-weld joint. The male contacts are made of gold-plated nickel alloy to facilitate welding for interconnection within the module.

Connection between the male and female contacts is maintained by the spring action of the female contact. A rectangular shaped male contact is inserted between two prongs on the female contact. Maximum insertion force is 10 ounces and minimum retention force is 3 ounces.

Jacking screws are used to install and remove electronic modules and to join and separate subassemblies that have integral mating connectors. In most cases the

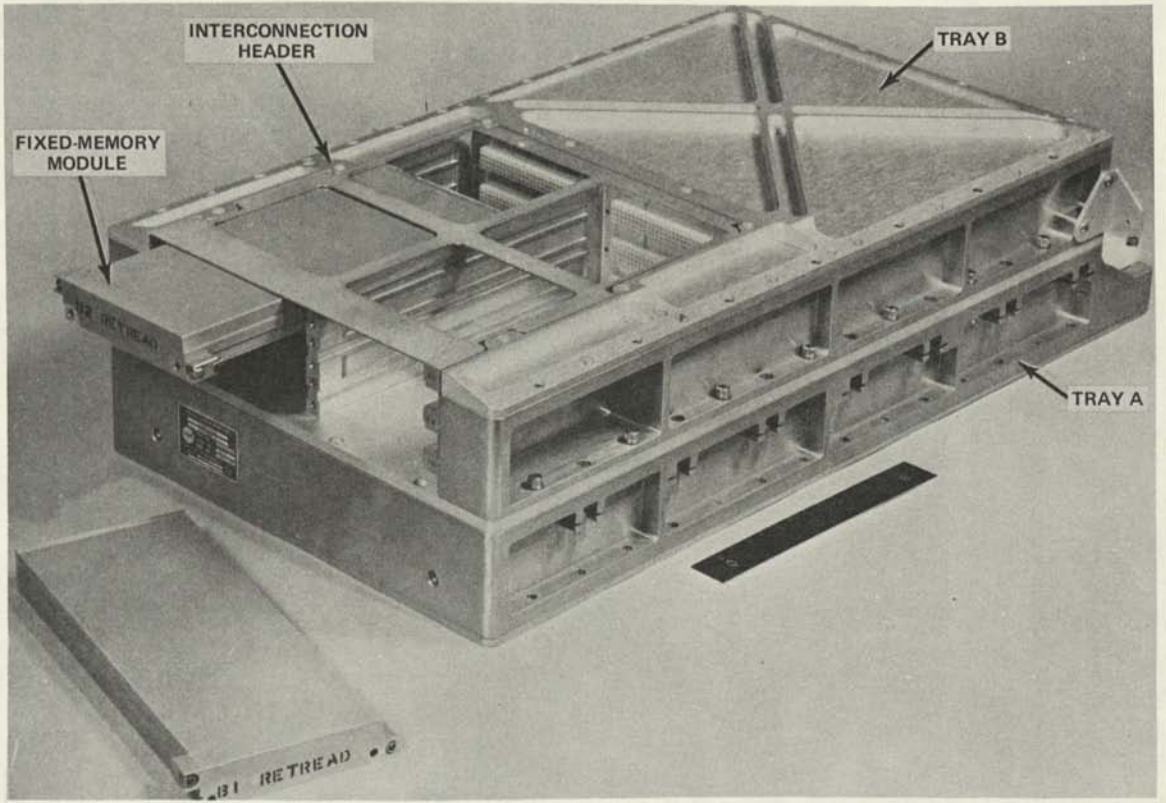


Fig. 3-26 Computer Assembly

screws are captive and are threaded into steel inserts (Helicoils) in the mating surface. The function of Helicoils is to provide repairability if threads are damaged.

3.9.1 Detailed Description

The following detailed description is a discussion of the mechanical features of the computer. It is arranged by component part. The descriptions are presented in a logical sequence but not necessarily in order of disassembly. Figure 3-26 shows an assembled computer. Figure 3-27 is an exploded view showing the relationship of the component parts. The main components of the computer are listed below.

Tray B cover
Tray B wired assembly
Interconnection header assembly
Mid-spacer
Tray A wired assembly
Tray A cover
Plug-in circuit modules

3.9.1.1 Tray A Wired Assembly

Tray A wired assembly consists of the tray, alignment pins, wrapost-type contacts, insulators, grounding sleeves, wire and potting material. Basically, the tray serves as a large electrical interconnecting device that permits all interconnection to be accomplished by machine wirewrapping. It is a rectangular structure machined of magnesium and divided into two compartments, when viewed from the module side (see Figure 3-28). The divider between the compartments contains the connectors that interface with tray B. A recessed shelf at one end of the tray contains an input/output connector and a test connector. The compartment next to the shelf contains connectors for 15 modules, these being a power supply module, 5 interface modules and 9 logic modules. The other compartment contains connectors for 15 logic modules and a power supply module.

Figure 3-29 shows the wirewrap side of the tray. A total of 7468 contacts are used for the module connectors. Of this total, 385 contacts are inserted in metal grounding sleeves rather than plastic insulators. These contacts are electrically bonded to the tray that is used as a ground plane for all circuits. The input/output connector and test connector require 504 contacts of which 10 are grounded. The connectors that interface with tray B require 240 contacts, none of which is grounded. These

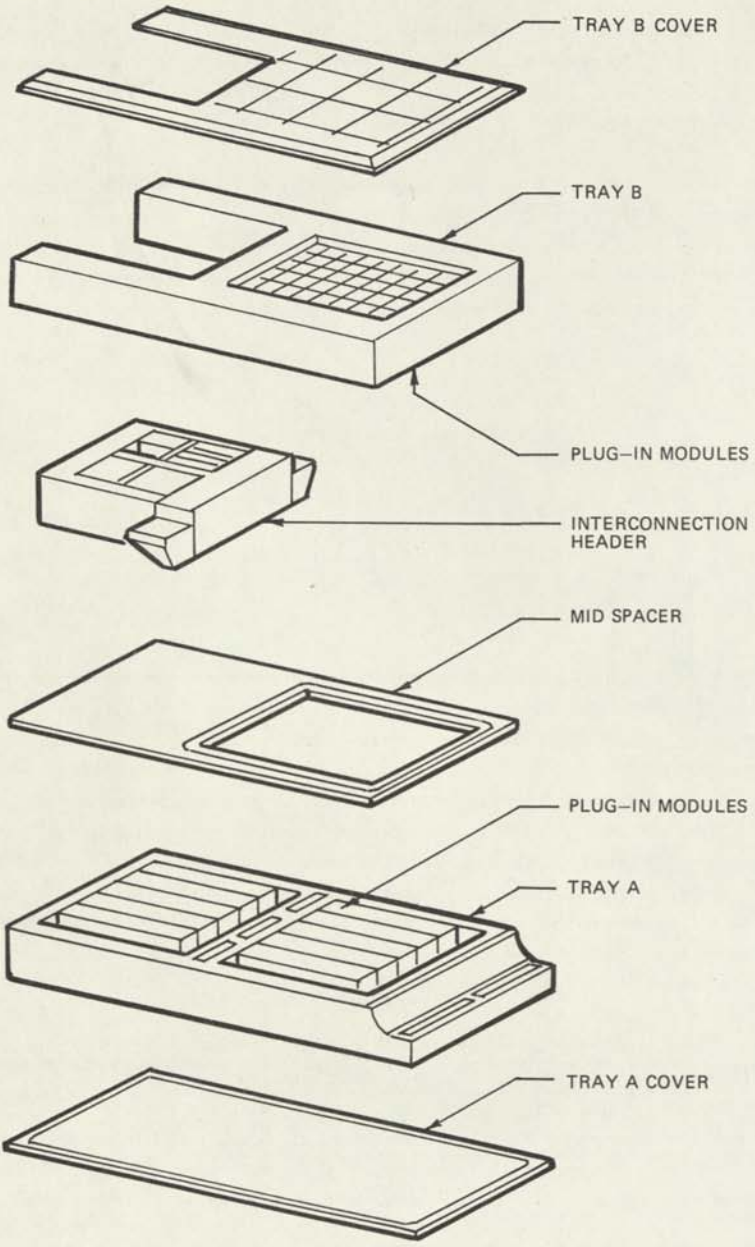


Fig. 3-27 Exploded View of Computer

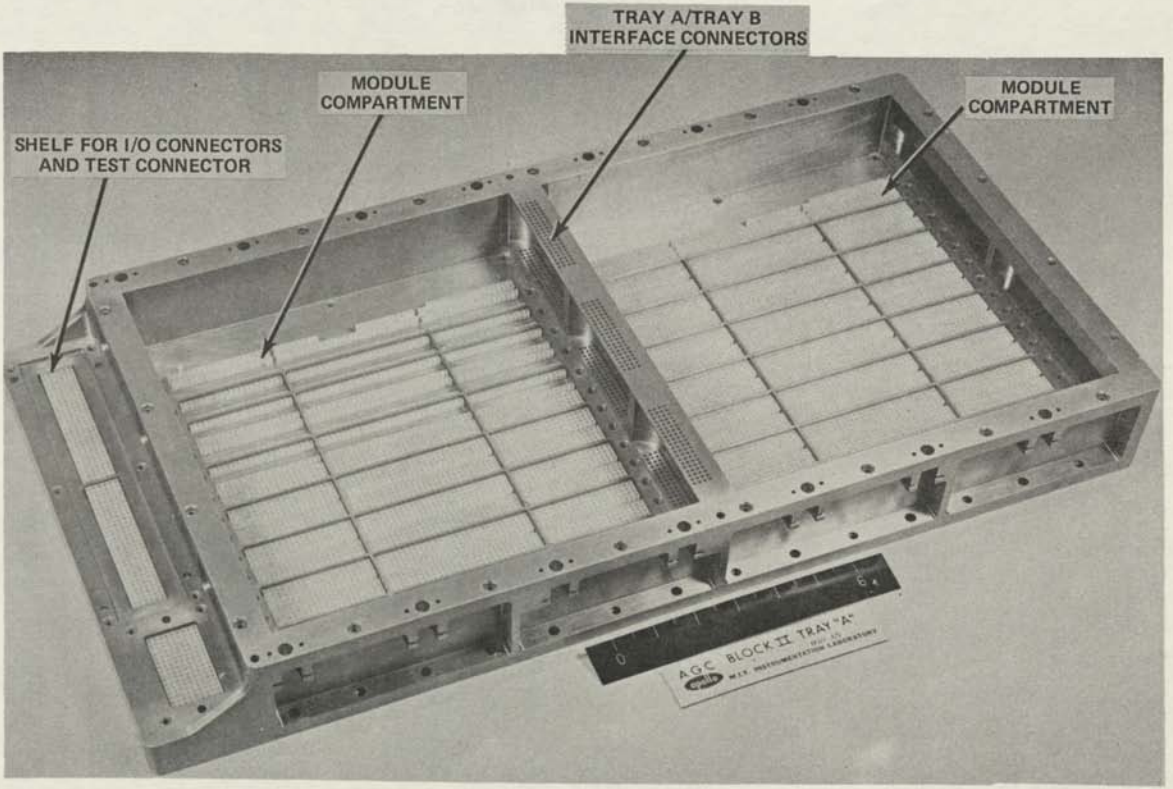


Fig. 3-28 Module Side of Tray A

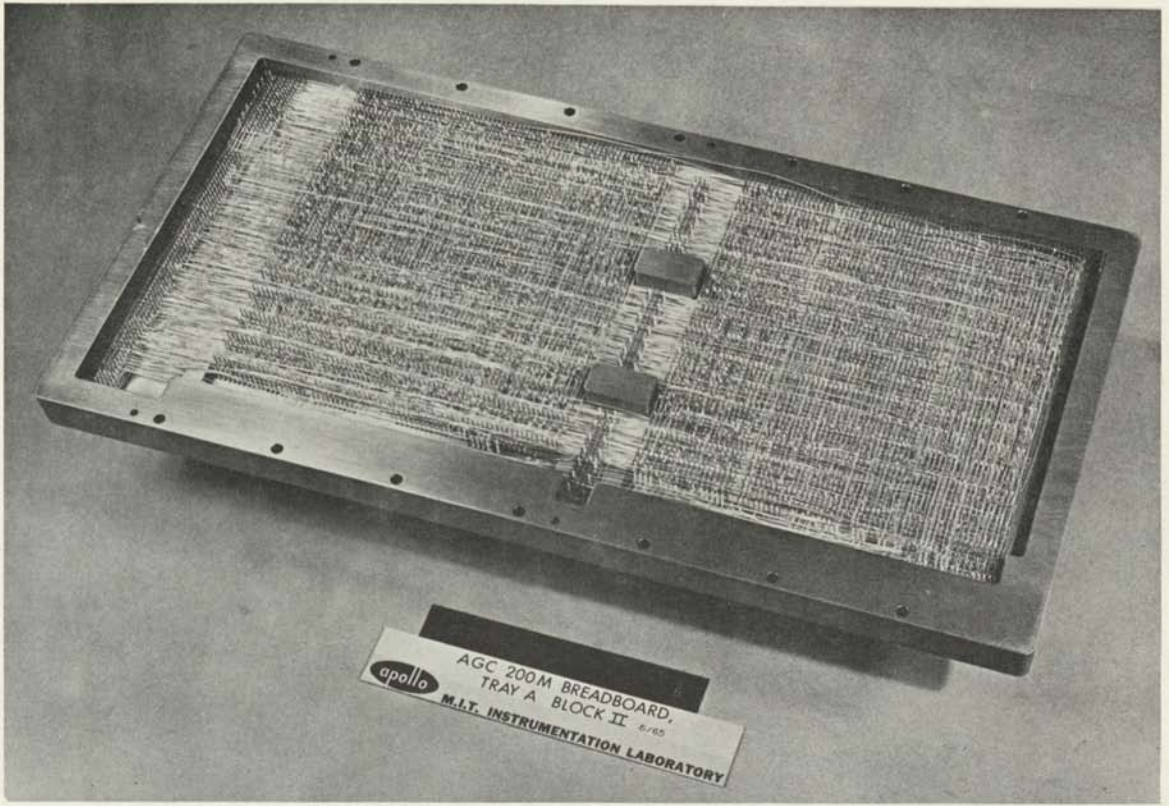


Fig. 3-29 Wire-Wrap Side of Tray A

contacts are long enough to pass through the two sections of the tray between the wirewrap field and connector interface at the top of the modules.

Copper bus strips are installed after the tray has been wirewrapped. These improve dc power distribution from the power supply modules to the logic modules. Two strips are used in each compartment to provide a very low resistance path from the 4-volt power supply module to the logic modules.

Four dowel pins are installed in the flange on the wirewrap side of the tray to align the cover during installation. Six dowel pins are installed in the shelf to align the mating input/output and test connectors. No dowel pins were used for module alignment since space was limited in both the modules and tray. In the case of modules, careful assembly by factory personnel is required to prevent connector damage.

3.9.1.2 Tray B Wired Assembly

Tray B wired assembly consists of the tray, alignment pins, wrapost-type contacts, insulators, grounding sleeves, wire and potting material. Basically the structure is machined from magnesium, so as to serve as a large electrical interconnecting device, using machine wirewrapping and a mounting structure for the interconnection header assembly. Approximately one half the tray area provides one compartment for modules. The other half of the tray provides mounting support and connectors for the interconnection header assembly that in turn provides support and connectors for the rope modules. The compartment contains connectors for 11 modules. These are an erasable memory module and 10 modules associated with control circuits for the erasable memory and fixed memory. The elevated mid-section contains the connectors that interface with tray A.

Connector construction is identical to the method used in tray A. A total of 1465 female contacts are used for the module connectors. Of this total, 71 contacts are inserted in metal grounding sleeves rather than plastic insulators to provide a ground plan for all circuits. Each of the interconnection header connectors contain 75 female contacts, one of which is the grounding type. A total of 240 male contacts, none being grounded, are used in the connectors that interface with tray A. As in tray A, these are long contacts in order to pass between the wirewrap field and connector interface.

Four dowel pins are installed in the flange on the wirewrap side of the tray to align the cover during installation. Four dowel pins are installed in the flange on the

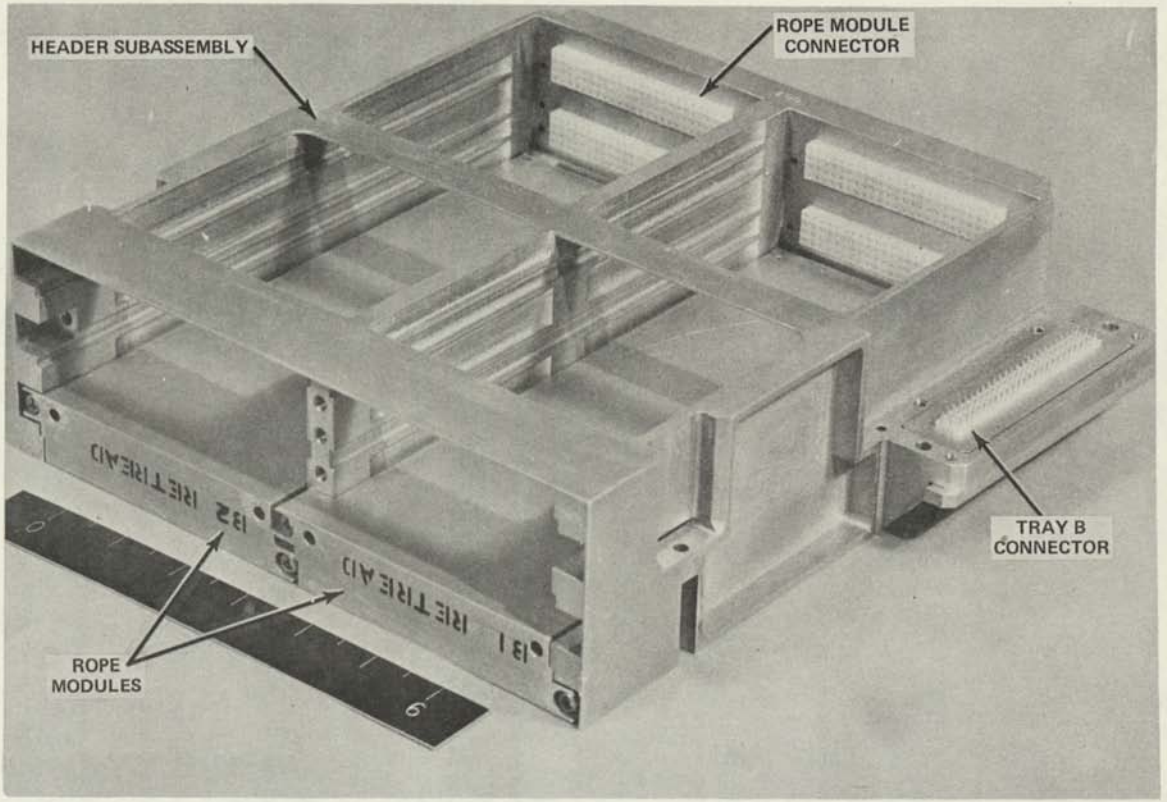


Fig. 3-30 Interconnection Header Assembly

module side to align the mid-spacer during assembly. These pins extend through the mid-spacer and engage mating holes in tray A during assembly of the computer.

An access hole is drilled in the vertical wall of the tray to allow tuning of the crystal oscillator when the module is installed. Normally, the hole is sealed with a threaded plug and O-ring. A pressure valve is installed in the vertical wall of the tray to allow the assembled computer to be filled with dry nitrogen.

3.9.1.3 Interconnection Header Assembly

The interconnection header assembly houses six fixed memory modules and interconnects them with tray B. It consists of a header and connectors, as shown in the Figure 3-30. The header is a one piece intricate aluminum structure. It has two sides, a back, and a center strip that divides it into two sections. The header has no top and no front. Two transverse strips interconnect the sides and center strip to provide structural rigidity with a minimum of weight. Both surfaces of the center strip and the inner surface of each side are grooved to form tracks for installing the fixed memory modules. Helicoil inserts are installed in the front end of the center strip and in both ends to retain the module mounting screws. Two integral flange type supports with connectors are used to mount the header to tray B. Two dowel pins are installed in each support for alignment between tray B and the header connector assemblies. A total of 576 female wrapost-type contacts and insulators are installed in the back section to form six module connectors of 96 contacts each.

3.9.1.4 Mid-Spacer and Tray Covers

The mid-spacer and both tray covers are made of magnesium. Screw holes in each part and matching areas of the trays are specially treated to ensure that electrical bonding is maintained throughout the structure of the assembled computer.

Both surfaces of the mid-spacer and the inside surface of each cover contain an integral butyl rubber seal. The seal is bonded to a groove in the flange area and projects beyond the flange surface. During assembly, the rubber is compressed and forms an environmental seal at the mating surfaces.

3.9.1.5 Module Construction Features

In general, two types of module construction are used. One method applies only to logic modules that are exclusively integrated circuit flatpacks. The other modules used a cordwood construction with welded interconnects. In the E-memory and

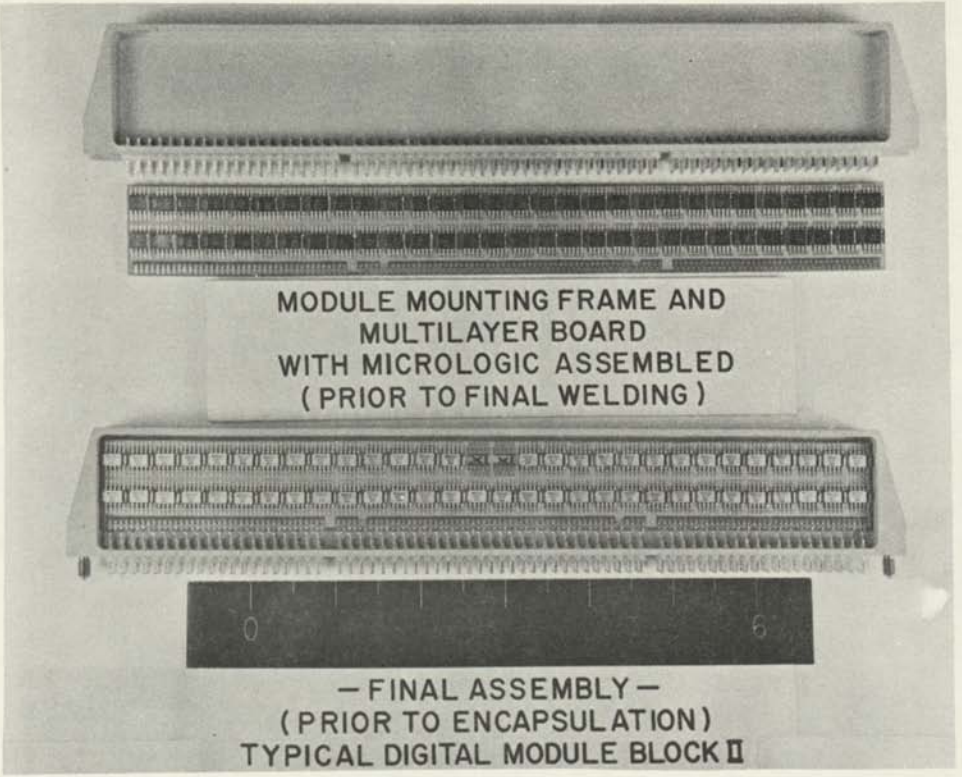


Fig. 3-31 Typical Logic Module Construction

rope memories, where fine copper wire is used, a combination of interconnection methods are used, i.e., soldering for the copper wire and welding of other components.

Logic module construction is shown in Figure 3-31. The top part of the photo shows the housing or header assembly with the Malco contacts installed. Just below the header is a multilayer circuit board with the flatpack integrated circuits attached. The flatpacks are parallel-gap welded to pads on the board. In order to provide a reliable pad for welding, alnifer, a "sandwich" of aluminum, iron and nickel was used. The lower part of the photo shows a module with the board installed in the header. Interconnections between the board and the Malco contacts are made with welded flat nickel ribbon wire.

Figure 3-32 shows the other method of construction, this being termed welded cordwood. Components are installed either directly in holes in a solid header or in holes in a plastic subassembly that fits in a header. Modules that dissipate high power, like the power supplies, use the solid metal header. Interconnections between components and between components and Malco contacts are made with welded nickel wire. In most cases, a mylar strip is placed in the header over the component leads prior to interconnection. Component identification and interconnect paths are marked on the film.

The memory module construction differs from the other modules because of the special requirement for the assembly and wiring of the cores. The rope memory is the most unique in that the sense line wiring pattern determines the contents of the memory. The inhibit wiring is standard for all memories and is installed first. The sense line wiring is installed with the aid of a tape controlled machine that determines the routing of the wires for the operator, thereby translating the requirements of the software to the routing of the wires within the module. The actual threading of the wires through, or bypassing, the cores is accomplished by the operator according to a guide that is positioned by the machine. See Fig. 3-33. The balance of the module assembly follows the standard construction techniques for the other cordwood modules. Figure 3-34 is a photograph of an unpotted rope module showing the cores, inhibit and sense wiring, and the cordwood construction for the resistor and diode gating circuits.

3.9.1.6 Assembly Features

Some of the features of the computer subassemblies include the potting materials, surface coating of metal parts, and assembly of the complete computer.

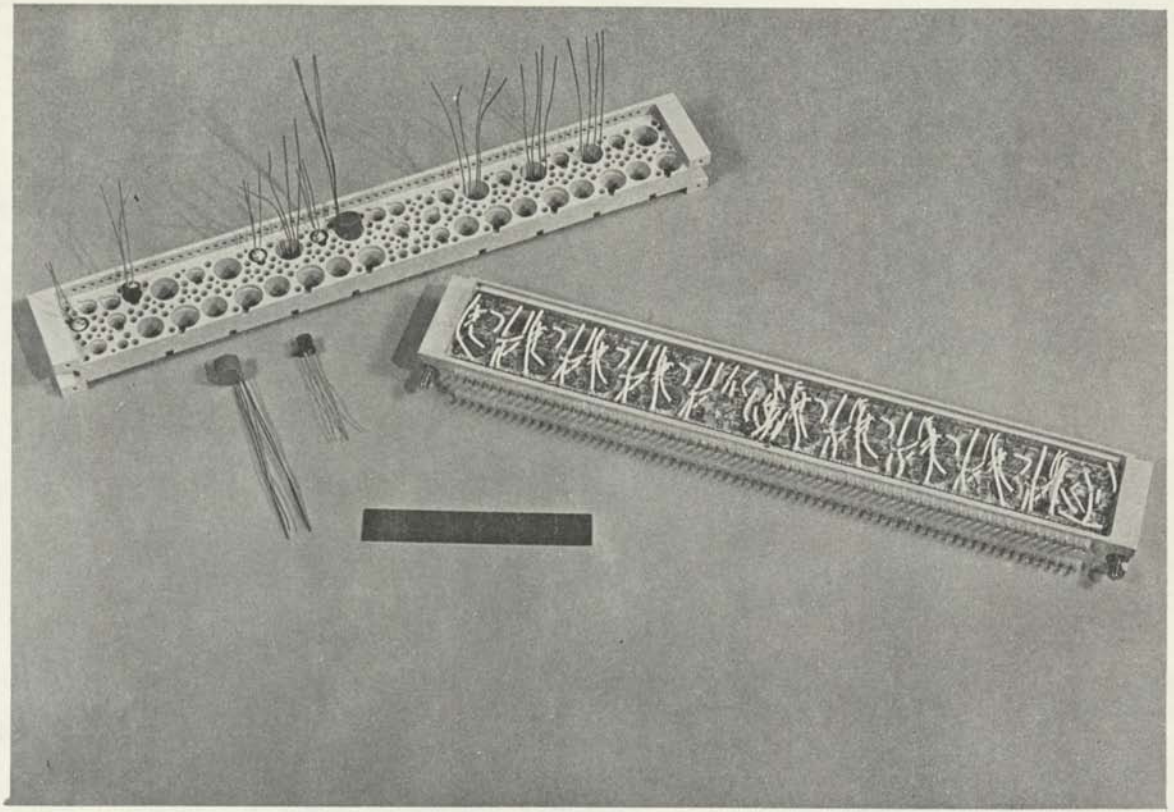


Fig. 3-32 Typical Welded Cordwood Construction

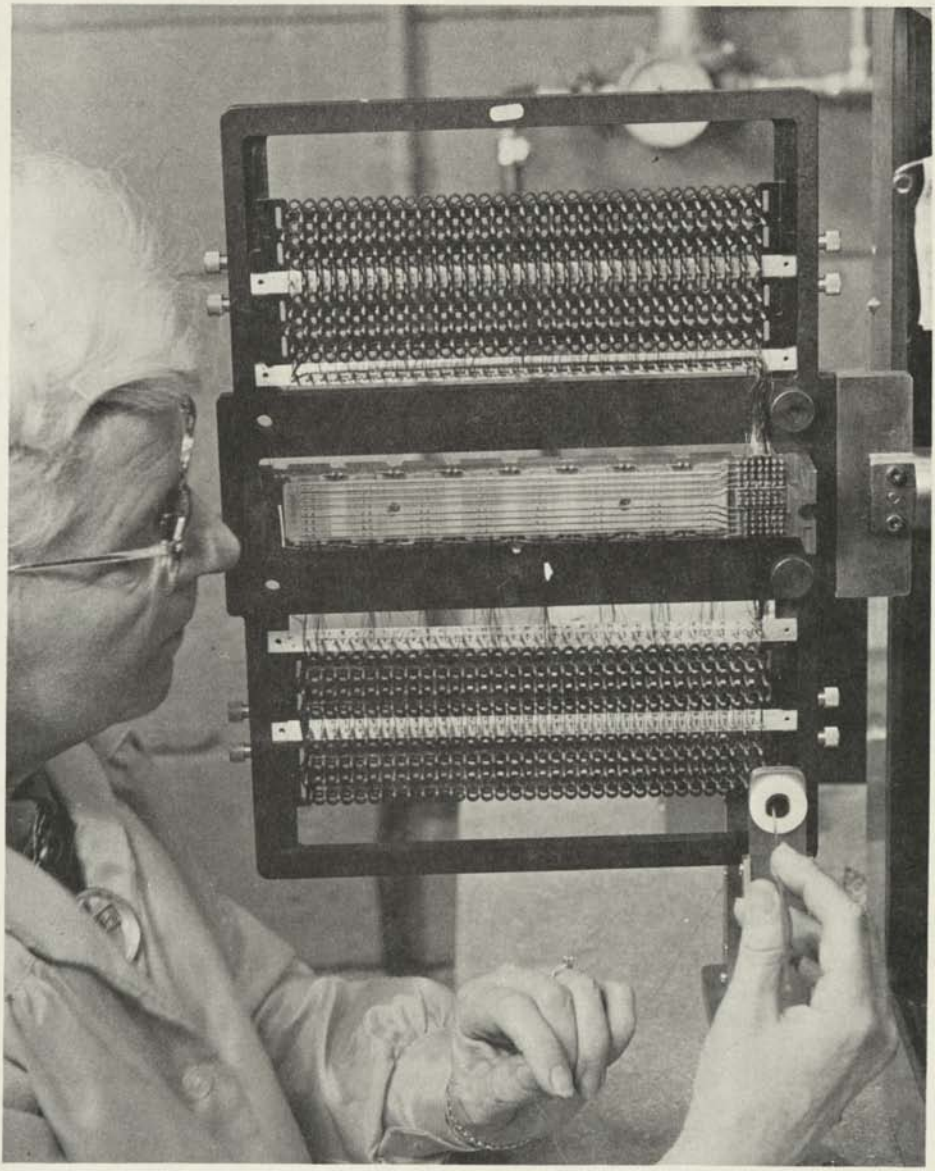


Fig. 3-33 Core Rope Module Sense Line Wiring

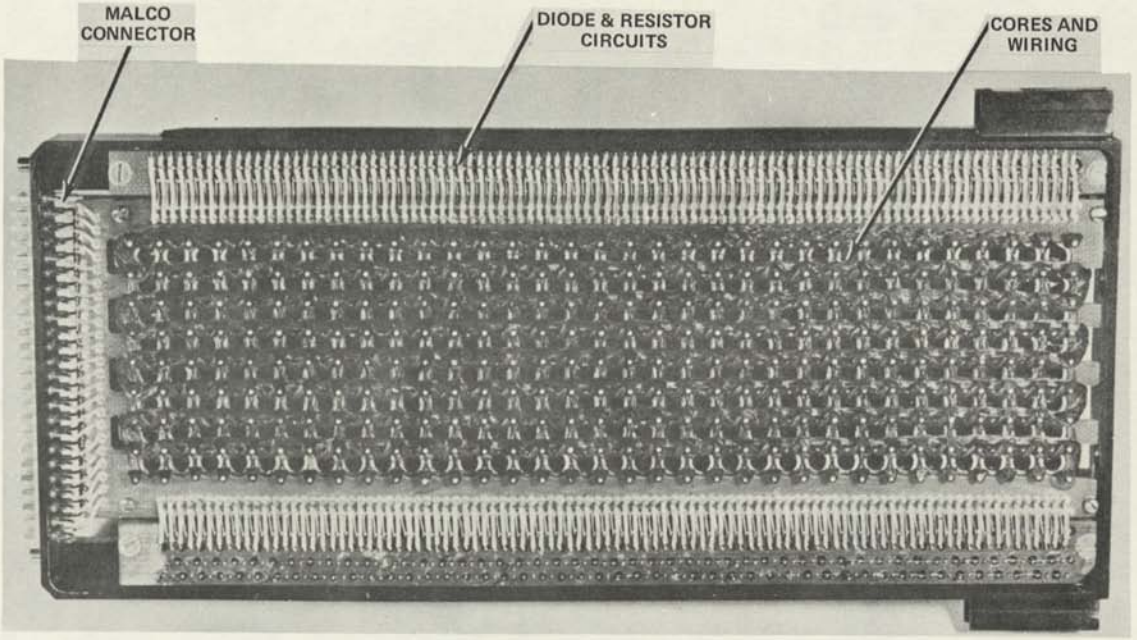


Fig. 3-34 Core Rope Module

The potting of modules, trays, and connector wells provide for the mechanical support of wires and components. In general, foam potting is used for these applications. In the case of Malco connectors that provide tray interconnection from outside to inside the computer, like the system interface connector, a hard potting material is used to seal the area before the remainder of the tray is foam potted. Fragile components, like the ferrite cores of the E-Memory module, are prepotted in a soft potting material like RTV-11 or, in the case of the E-memory, Sylgard 184. A change from RTV-11 to Sylgard 184 potting of the E-memory was introduced to correct the wire breakage problem, discussed in Section 7.2.2. In this case the RTV-11 did not have sufficient damping to prevent movement of the core stacks during vibration. The finished modules and trays then were completely potted using a material of the lowest density that would meet the criteria of sealing against moisture where necessary, and of providing the mechanical support of components and wiring.

Since magnesium was used to save weight wherever possible, it was necessary to protect against corrosive action. Modules or surfaces inside the computer were anodized. All surfaces exposed to the spacecraft environment were painted in addition to the anodize. The paints were required to be odor free and to provide a low emissivity surface for spacecraft thermal control. Surfaces, visible when installed in the CM, were required to match the dark gray of the CM interior. Screw holes, and the surfaces around the holes, are not painted but are treated with conducting anodize to insure electrical bonding of interfaces between trays, spacers, covers and computer to spacecraft cold plate. In the case of the computer, spacecraft mounting hardware, steel bolts with teflon-coated shanks and aluminum washers under the bolt heads were required to prevent corrosion because mounting hardware is composed of dissimilar metals (magnesium and steel).

The interconnection header assembly is installed in the cut out area of tray B. The connectors on the header mate with the connectors on the tray B shelves. Installation is accomplished by two jacking screws that are retained in each of the flange type supports on the header. The module is installed by two flange mounted jacking screws that are threaded into the end surface of the sides and center strip of the header.

All tray modules are installed, and the mid-spacer is placed on the module side of tray B with open section of the spacer framing the module compartment. The spacer is aligned with four dowel pins and is secured by two screws in the tray center section. Tray B, with mid-spacer intact, is assembled to tray A. Alignment is ensured by the four dowel pins on tray B. Screws are inserted in the tray B flange

and threaded into the tray A flange. Covers are installed on the wirewrap side of both trays with high torque steel screws and lock washers. Four additional screws are used to secure and complete the seal of the mid-spacer to tray A. They are inserted into counter-sunk holes in the mid-spacer and are threaded into the tray A flange that is adjacent to the front of the interconnection header assembly. All exposed screw heads are coated with Silicone rubber sealing compounds to prevent corrosion. The six fixed memory modules can then be installed into the interconnection header completing the computer assembly.

Prior to assembly of the trays, vibration pads are installed in some key areas. Three pads are installed between the modules in tray A and tray B in the open area of the mid-spacer. The center vibration pad consists of a silicone rubber sheet that is shimmed to match the spacing between modules. This unit measures approximately 2-1/2" by 1-3/4" and is bonded to the top surface of erasable memory module B12. This vibration pad stiffens the structure and thus reduces the vibration transmitted to the E-memory module. The shimmed pad was a change from a single cellular pad and was introduced as a partial fix for the fatigue failures in the wiring in the erasable memory module. All other vibration pads are cellular silicone rubber approximately 1/8" thick.

After assembly, the computer is leak tested and filled with dry nitrogen to a pressure of 2 ± 0.5 psig.

3.9.1.7 Thermal Design

Temperature control of the computer was achieved through conduction to the cold plate structure of the spacecraft. Radiational cooling was minimized by the choice of finishes to meet the requirements of spacecraft thermal control. Under some conditions, the surfaces surrounding the computer were at a higher temperature than the computer thus causing additional heat loads instead of providing radiational cooling. In every case however, the effects of thermal radiation could be ignored in the thermal design of the computer.

Since the total power consumption of the computer is relatively low, the thermal control was mainly one of distributing the heat load in the computer and providing conduction paths to the cold plate. For example, module locations in the tray were carefully selected. The two power supplies were located at one wall in tray A, where a short path and extra metal could be provided for the heat conduction to the cold plate. The E-memory, memory drivers, and sense amplifiers are located in the center of tray B to provide temperature tracking of the temperature compensating

circuits and the memory cores. The conduction paths were provided from the electrical component to the base of the modules and then into the wirewrap plate, where the heat fans out to the sides of the trays, and thus down the walls of the tray A cover to interface with the cold plate in the CM and with cold rails in the LM. In the case of the two switching transistors (NPN and PNP) the thermal design included specifying a special package. The package was the standard TO-18 case size but with a solid metal header for decreased junction to case temperature rise. At the time of the Block II mechanical design, the solid metal header was not available in the TO-18 case size but had been used by semiconductor manufacturers on other similar cases. Thus the thermal design provided conduction from the element dissipating heat, such as the transistor chip, through all the mechanical interfaces to the cold plate.

The goals of the thermal design effort were: first, to ensure that the temperature of components and especially semiconductors remained below 100°C under worst-case conditions. The second goal was to provide a reasonably uniform thermal environment between modules like the memory electronics and logic modules. A temperature gradient between logic modules would reduce the operating margins of the logic. Thermal measurements on the finished computer have verified that these goals were met. The measured temperature difference between logic modules was less than 5°C and therefore neglectable. The temperature rise through the structure to the hottest components was low enough to maintain junction temperatures well below 100°C.

3.10 DISPLAY AND KEYBOARD UNIT (DSKY) MECHANICAL DESCRIPTION

The mechanical construction of the display and keyboard unit (DSKY) was determined in part by the following design requirements:

1. Overall form factor is governed by spacecraft mounting requirements and human factor requirements concerning the size of indicators and placement of the keys
2. The human factor requirement for spacecraft color, lighting illuminated keys, and actuating force of the keys
3. Complete environmental sealing of the assembled DSKY
4. Replacement of alarm indicator and digital indicator from the front. Both indicators are hermetically sealed at the mounting surface.
5. Replacement of the movable parts of the keys from the front.

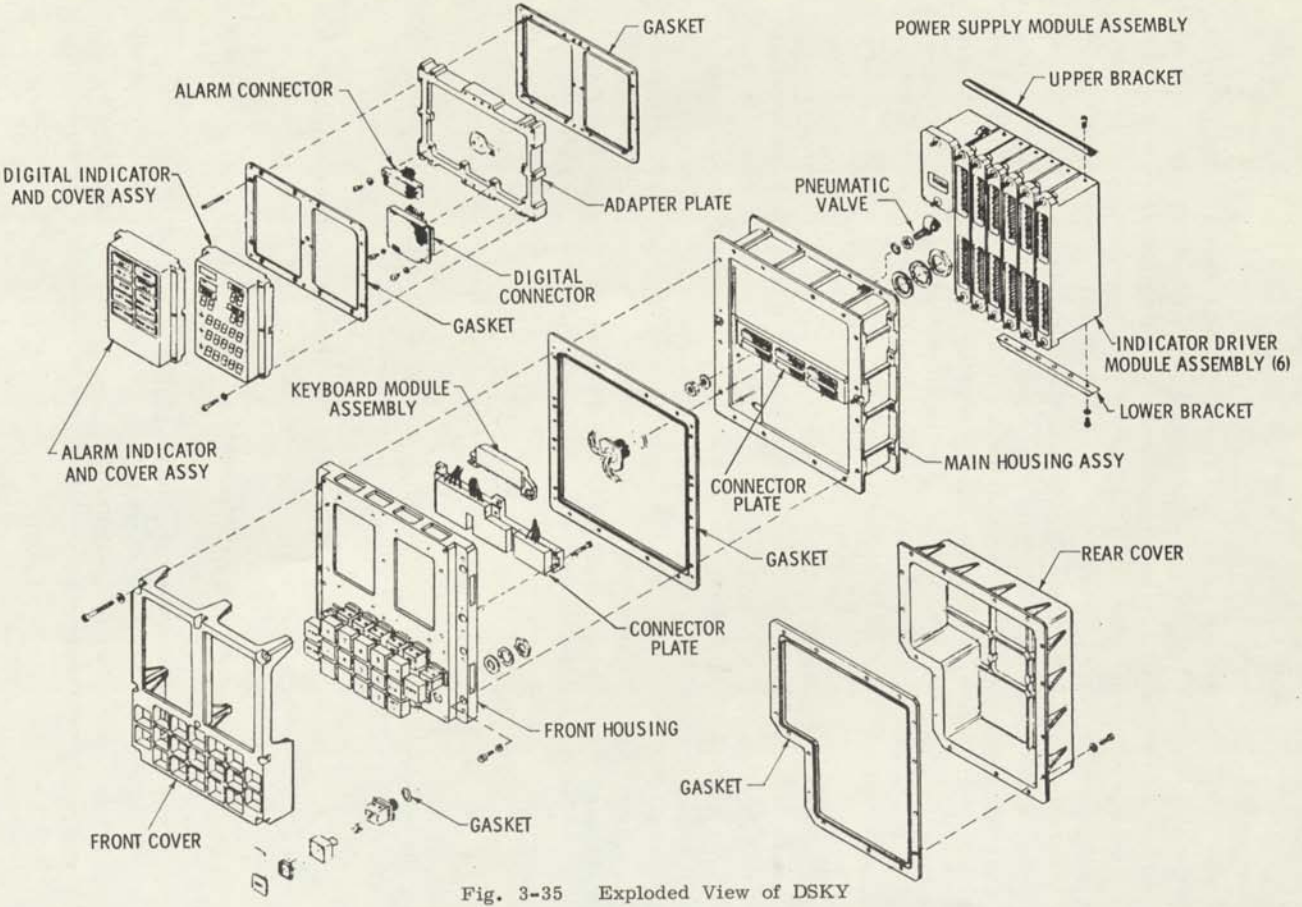


Fig. 3-35 Exploded View of DSKY

3.10.1 Detailed Description

The following detailed description discusses the mechanical features of the DSKY. It is arranged by component part. The descriptions are presented in a logical sequence but not necessarily in order of disassembly. Figure 3-35 is an exploded view of the DSKY that shows the relationship of the component parts.

The main components of the DSKY are listed below:

Front housing assembly
Main housing assembly
Alarm indicator and cover assembly
Digital indicator and cover assembly
Indicator driver modules (6)
Power supply module
Keyboard module
Front and rear covers

The DSKY assembly has two major subsections—the front housing subassembly and the main housing subassembly. The joint between the front and main housing are separated and sealed by an aluminum gasket that has an integral butyl rubber on each bearing surface. The housings are bolted together with 16 screws. Two jacking screws are provided to separate the subassemblies and 212-pin Malco connectors that provide for the electrical connections between them.

3.10.1.1 Front Housing Assembly (see Figures 3-36 and 3-37)

The front housing assembly contains the keyboard pushbutton switches, keyboard encoding module, housing, connectors and wiring.

Nineteen holes are drilled in the lower half of the housing for installation of the pushbutton switch assemblies that comprise the keyboard. The switch is installed from the front of the housing. A flange on the switch assembly contacts the front surface of the housing. An O-ring seal is installed in a groove in this flange contact surface. The threaded portion of the switch assembly extends through the housing, where a hex nut and lockwasher hold the switch assembly tight against the housing. The O-ring is compressed and seals the hole in the housing. The portion of the switch assembly, that extends through the housing, contains the five-pin electrical connector. A detailed discussion of the pushbutton switch is contained in paragraph 3.10.1.5

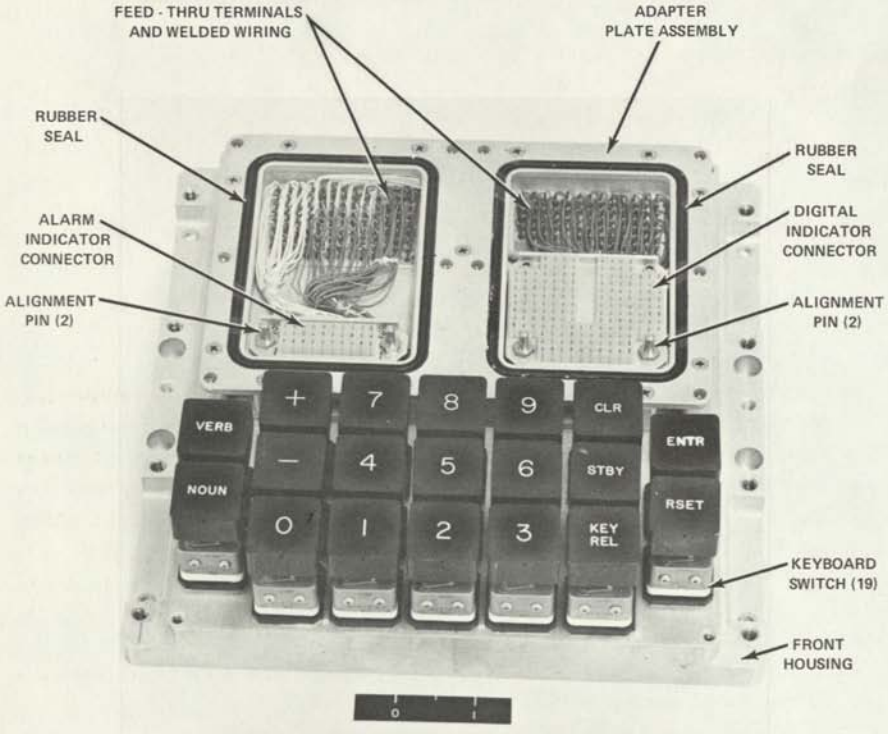


Fig. 3-36 Front Side of Front Housing Assembly

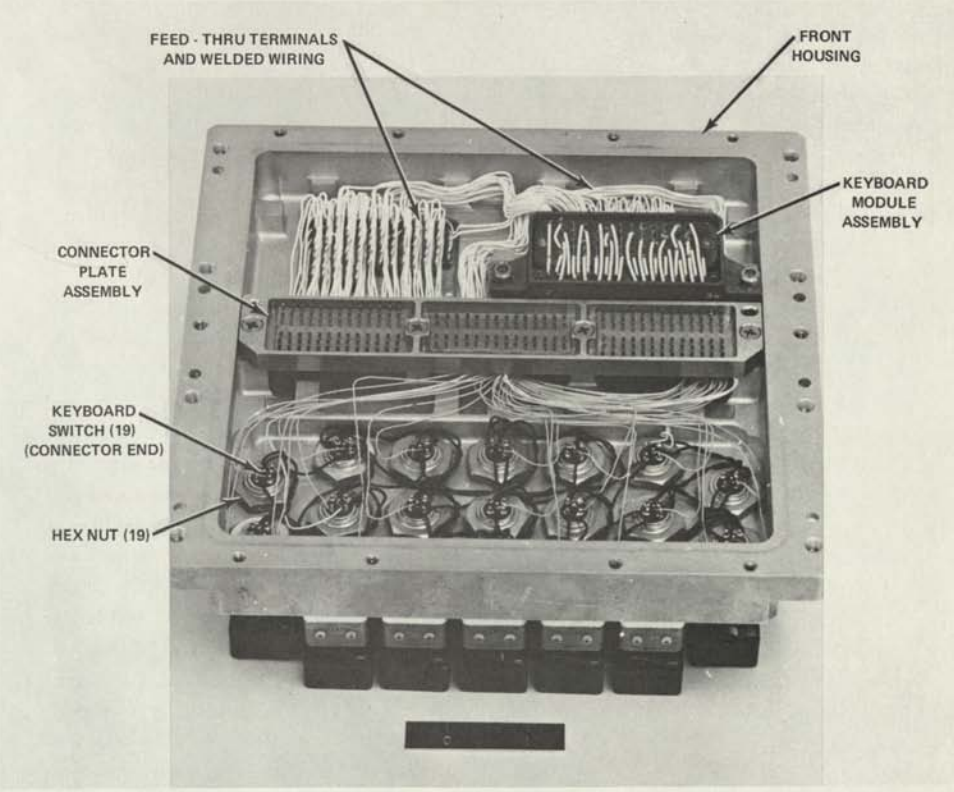


Fig. 3-37 Rear Side of Main Housing Assembly

The adapter plate assembly, with front and rear gaskets, is mounted on the upper half of the front surface of the housing with 13 screws. The assembly provides the seals and mounting surface for the alarm and digital indicators. Feed-through terminals are installed in the adapter plate to which wires are welded on both ends. This allows connections to be made from connectors in the rear of the front housing to the alarm and digital connectors on the adapter plate.

The connector for the alarm indicator consists of 30 female contacts installed in an aluminum plate. The connector plate is held to the adapter plate by two alignment pins and washers, that pass through holes in the connector plate, and are threaded into inserts in the adapter plate. They engage two mating holes in the alarm indicator housing to ensure alignment of the electrical contacts during assembly of the alarm indicator.

The connector for the digital indicator consists of 160 female contacts installed in an aluminum plate. Two alignment pins and two socket head screws are used to attach the connector plate to the adapter plate.

The Malco connectors on the rear of the front housing are contained in a large adapter plate that is secured to the housing with six screws. One connector consists of 34 female contacts that mate with the contacts on the keyboard encoding module assembly. The other connector consists of 212 male contacts arranged in six sections. The mating half of this connector is mounted on the front of the main housing.

The alarm indicator, digital indicator, and front cover are not part of the front housing assembly, but this is a logical place to discuss their installation. A detailed discussion of the indicators is contained in paragraphs3.10.1.3 and 3.10.1.4. Each indicator is attached to the adapter plate front gasket with six jacking screws that pass through the gasket and are threaded into the adapter plate. The rear surface of each indicator is hermetically sealed by a butyl rubber gasket. The front cover fits over the indicators and keyboard and is attached to the front housing with eight screws. It encloses the sides and bottom of the keyboard. The cover exposes only the top surface of the keys and fills the space between them to prevent simultaneous depression of two keys.

3.10.1.2 Main Housing Assembly (see Figures 3-38 and 3-39)

The main housing assembly contains six indicator driver modules, power supply module, housing, interassembly connector plate wiring harness, input/output connector and pneumatic fill valve.

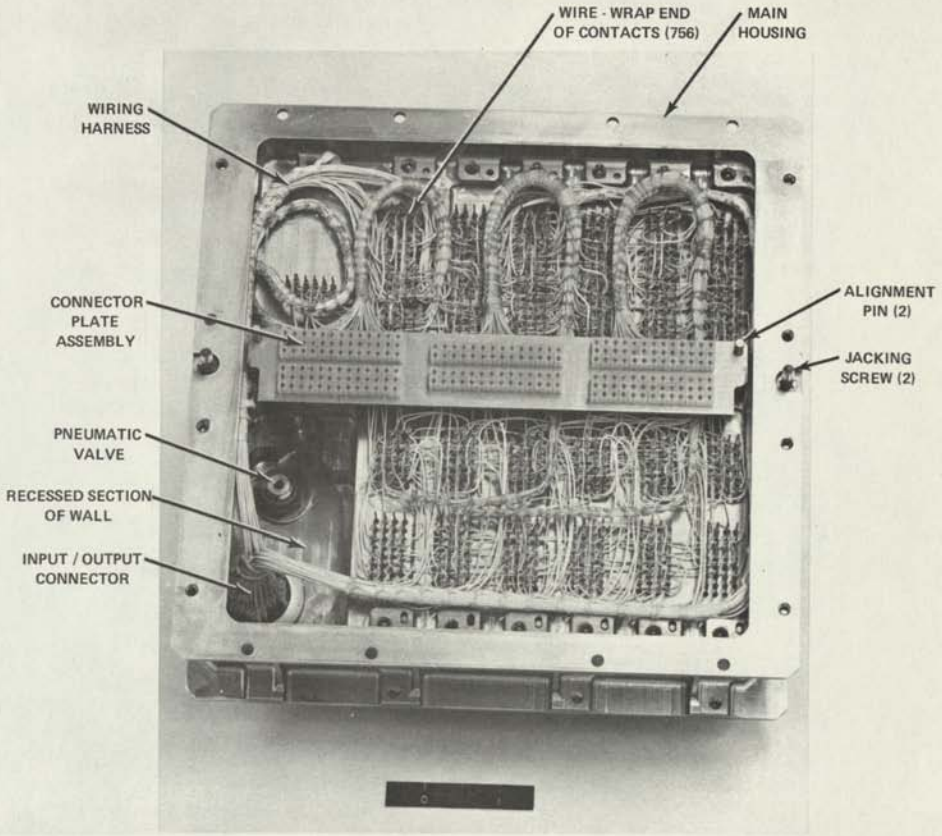


Fig. 3-38 Front Side of Main Housing Assembly

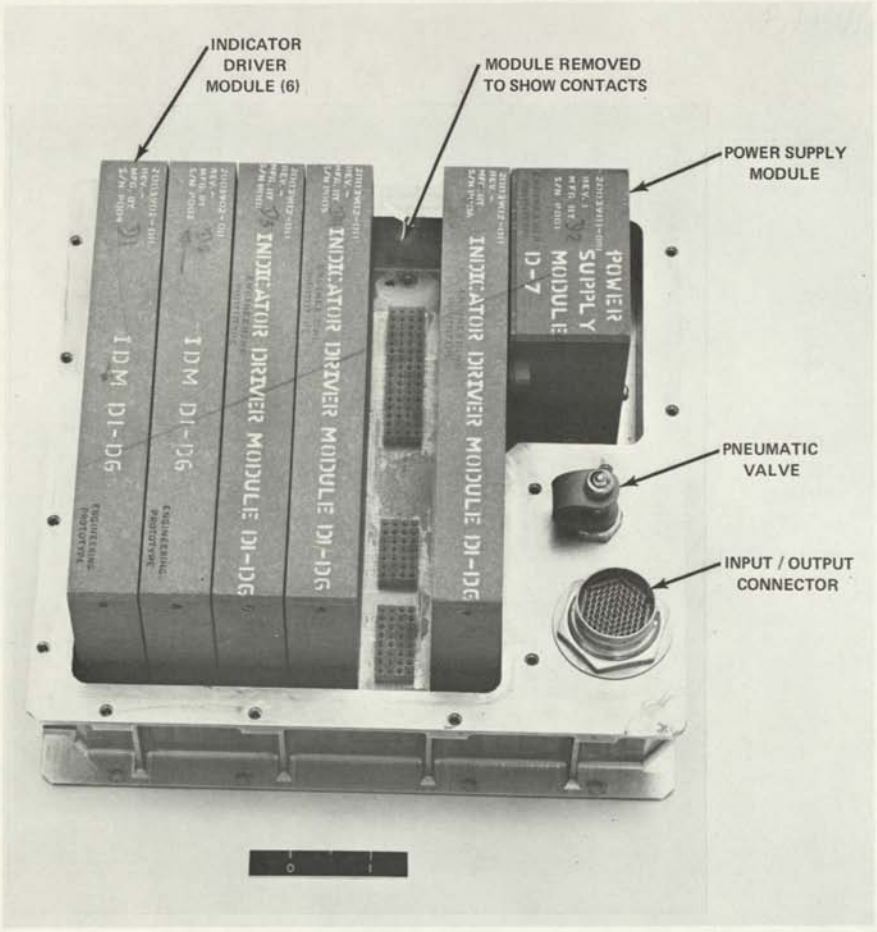


Fig. 3-39 Rear Side of Main Housing Assembly

The housing provides for module mounting and interconnection between the 756 female Malco contacts. The contacts are arranged in six columns of three rows each, with each column representing the connector for an indicator driver module, that is plugged into the rear housing compartment. Twelve contacts in the upper left area represent the connector for the power supply module. A section of the housing wall is offset to the rear with a 91-pin input/output connector installed in the wall section. The connector was required to be compatible with the spacecraft interface and is the only one in the computer subsystem. A wiring harness runs between this connector and the contacts in the main housing.

A connector plate, consisting of 212 female contacts plus a wiring harness, is mounted on the front side of the main housing. The plate is attached to the housing wall with four screws, that pass through the wall from the rear and thread into standoff ribs on the rear of the plate. The wiring harness is in three sections and runs between the contacts in the connector plate and the contacts in the housing wall. This connector and the mating connector on the front housing are aligned by two pins in the main housing plate that engage mating holes in the front housing plate.

A pneumatic filler valve is installed in the rear of the offset section of the housing wall. The rear cover does not enclose the valve, so that it is accessible after the DSKY is assembled. A hole is drilled in the vertical wall of the offset section to form a passage between the forward and rear housing compartments.

Although not part of the main housing assembly, six indicator driver modules and a power supply module are installed in the rear compartment of the housing. Each module plugs into a connector in the housing wall and is retained by two jacking screws. Two alignment pins in the module housing engage mating holes in the main housing wall to ensure proper installation. Additional rigidity is supplied to the installed modules by two L-shaped aluminum brackets. One spans the upper rear edge of the indicator driver modules and power supply module; the other spans the lower rear edge of the indicator modules. The brackets are secured by screws that are threaded into the module housings.

Each indicator driver module assembly consists of a header assembly, covers, decoding circuit assembly, and relay circuit assembly. The header assembly is an aluminum housing with 12 male contacts installed in one end. These contacts plug into matching female contacts in the wall of the main housing. The module housing is divided into two compartments. The smaller compartment houses the decoding circuit assembly. The relay circuit assembly is installed in the larger compartment. Covers are installed on both sides of the housing. The power supply module assembly

consists of a header assembly, covers and circuit components. The circuit components are packaged using welded-wire cordwood construction. The header assembly is an aluminum housing with 12 male contacts installed in one end. The mating female contacts are installed in the wall of the main housing. The power supply module and a typical indicator driver module are shown in Figures 3-40 and 3-41.

The rear cover gasket is placed between the rear cover flange and the main housing rear flange. A butyl rubber seal is bonded to each side of the gasket. The rear cover is placed on the gasket and both parts are attached to the housing rear flange with 16 screws. The gasket provides a hermetic seal, but the pneumatic valve and input/output connector are not covered.

The assembled DSKY was leak tested to ensure that the various seals provided complete environmental isolation. An operational DSKY is filled with a mixture of 87 percent nitrogen, 9 percent helium and 4 percent air at approximately 15 psi. The gas is introduced through the pneumatic filler valve.

3.10.1.3 Alarm Indicator and Cover Assembly

The alarm indicator and cover assembly consists of the alarm indicator, cover panel, gasket, frame, jacking screws, washers, and retaining rings. (See Figure 3-42). The frame, gasket and panel are bonded together with silicon adhesive to form the cover assembly.

The cover panel is a flat piece of annealed glass measuring approximately 3 inches high by 2-1/2 inches wide by 1/16 inch thick. Both surfaces are coated to reduce reflection. One surface is beveled on all four edges. The frame is made of an aluminum alloy in the form of a rectangular box with the rear of the frame open and flanged on both sides. Three holes are drilled in each flange to accept jacking screws. The front of the frame contains a rectangular opening with a beveled edge. The panel is bonded to the inside of the front surface of the frame with its beveled edge in contact with the frame. The molded silicone rubber gasket is placed inside the frame and is bonded to the panel and the frame. The cover assembly was added to the initial design as part of the flammability requirement, since the material of the alarm indicator was flammable in the spacecraft environment.

The cover assembly (frame, panel and gasket) is placed over the alarm indicator. The indicator is a sealed unit that consists of a plastic face plate, filament-type bulbs, wiring, connector pins and insulators contained in an aluminum housing. The indication contains 14 legend areas that are each illuminated by 3 bulbs wired

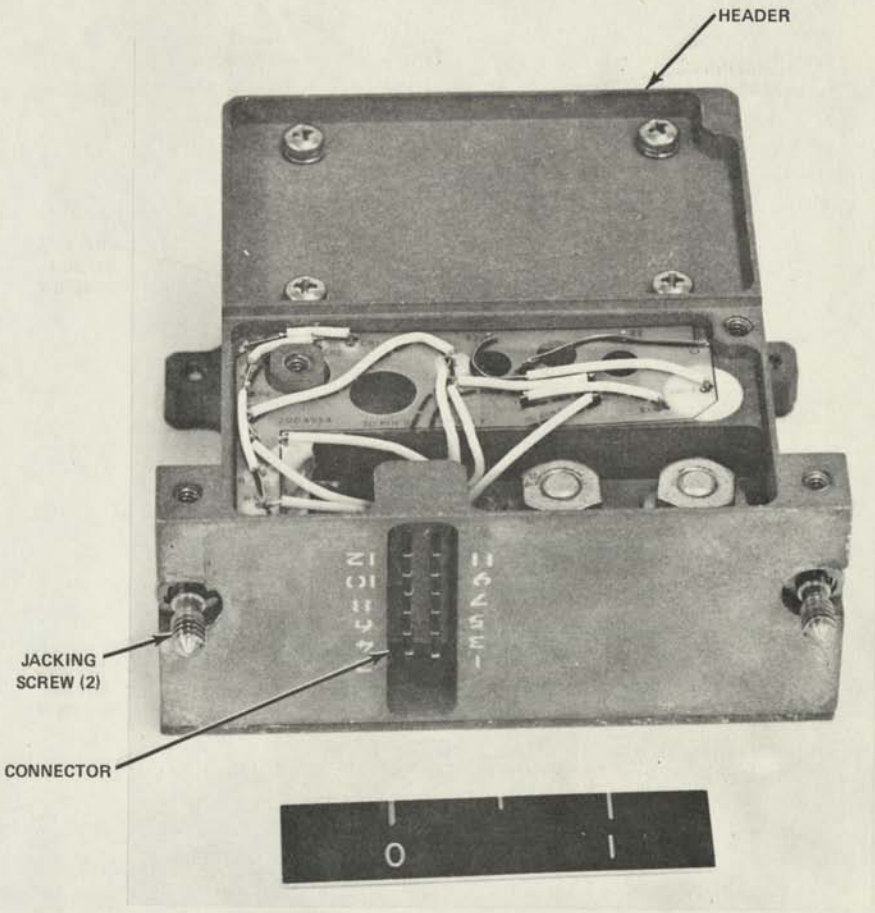


Fig. 3-40 Power Supply Module

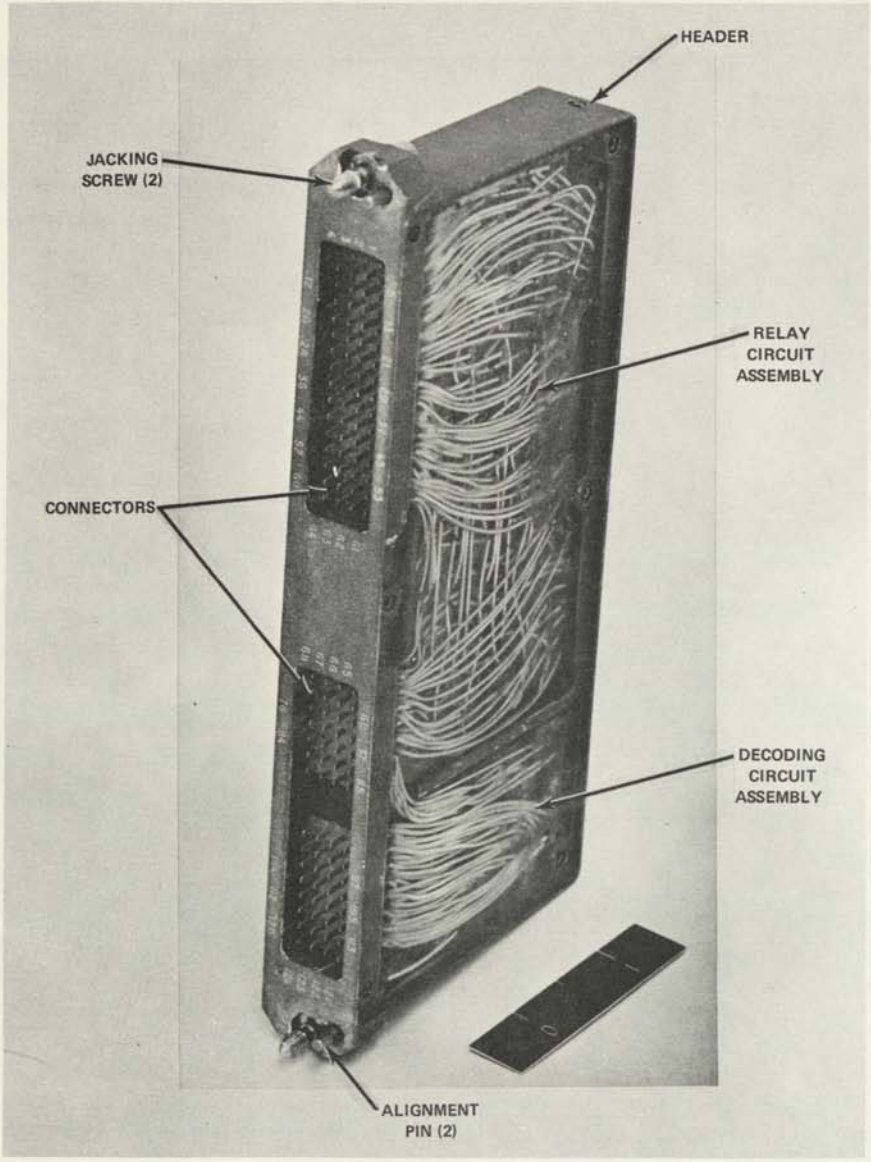


Fig. 3-41 Typical Indicator Driver Module

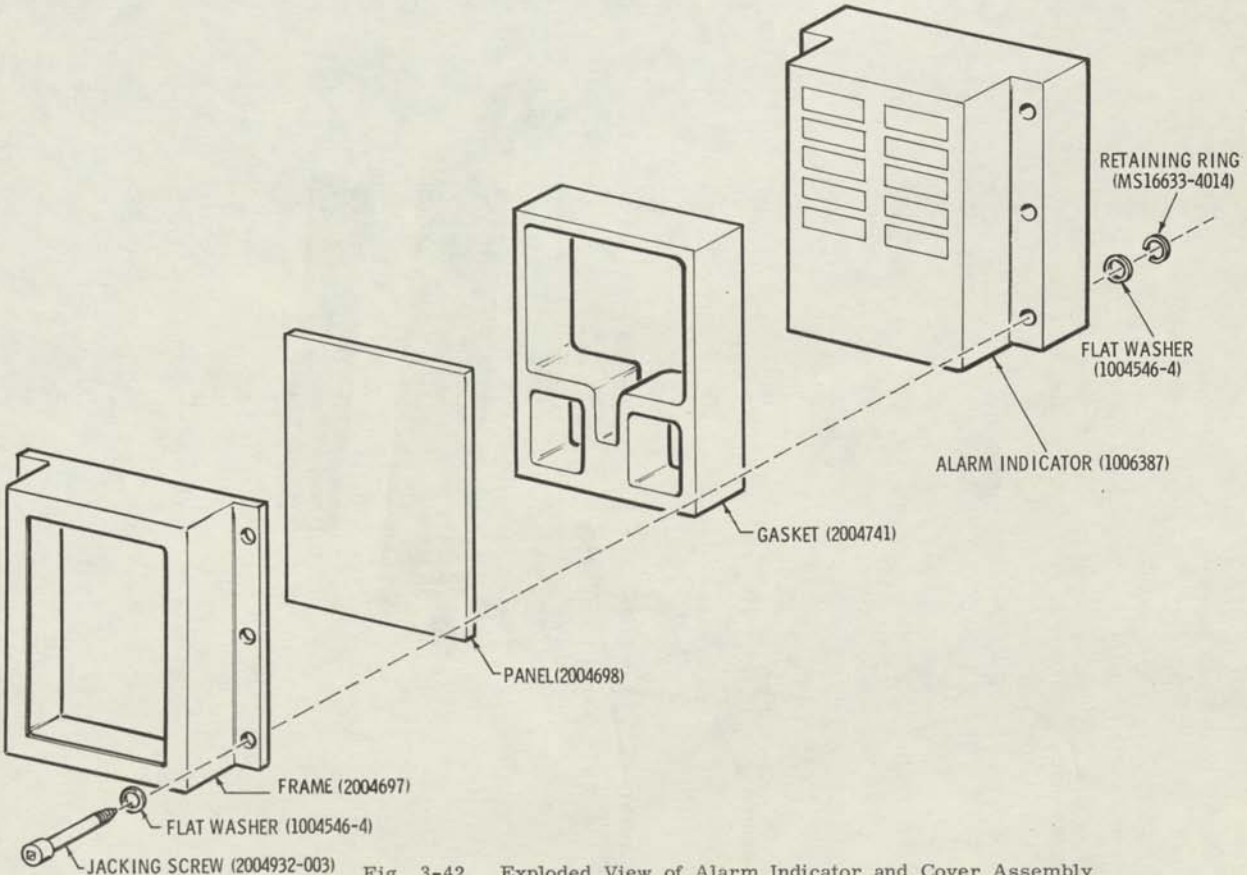


Fig. 3-42 Exploded View of Alarm Indicator and Cover Assembly

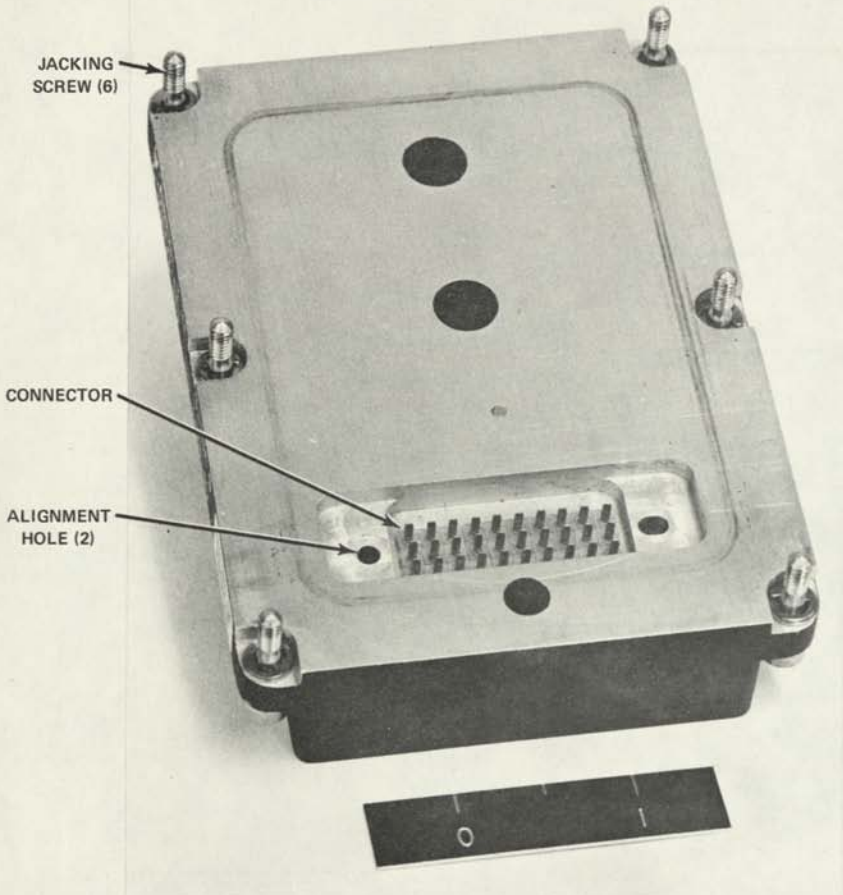


Fig. 3-43 Rear Side of Alarm Indicator

in parallel. A recessed area in the rear surface of the housing contains 30 connector pins. (See Figure 3-43). Each pin is installed in a plastic insulator that is inserted in a hole in the housing. The insulator seals the hole through which the pin extends. The rear of the housing is flanged on both sides. Holes are drilled in the flanges to match those in the frame. The cover assembly fits over the indicator, and the two pieces are held together loosely by 6 jacking screws, 6 retaining rings and 12 flat washers. The jacking screws are retained in the flanges, and in turn they loosely hold the cover assembly and alarm indicator together. Subsequent installation of the jacking screws in the adapter plate force the cover assembly and indicator together to form a sealed unit.

3.10.1.4 Digital Indicator and Cover Assembly

The digital indicator and cover assembly consists of the digital indicator, cover panel, frame, jacking screws, washers, and retaining rings. (See Figure 3-44).

The indicator is a sealed unit that consists of a glass face plate, electroluminescent units, wiring, connector pins, and insulators contained in an aluminum housing. A recessed area in the rear surface of the housing contains 160 connector pins. (See Figure 3-45). Each pin is installed in a plastic insulator that is inserted in a hole in the housing. The insulator seals the hole through which the pin extends. The rear of the housing is flanged on both sides, and three holes are drilled in each flange to accept jacking screws.

The cover panel is a flat piece of laminated glass measuring approximately 4-1/4 inches high by 2-1/2 inches wide by 1/8 inch thick. The front surface is coated to reduce reflection. The panel is bonded to the glass face of the indicator with a thermosetting optical adhesive. It protects the indicator face against damage by impact. A bead of silicone compound is applied along the edge of the panel to seal the panel-to-indicator interface.

The frame is made of aluminum in the form of a rectangular box. The front of the frame contains a rectangular opening with a beveled edge. The rear of the frame is open and flanged on both sides. Holes are drilled in the flanges to match those in the indicator housing. The frame fits over the indicator, and the two pieces are held together loosely by 6 jacking screws, 6 retaining rings and 24 flat washers. The jacking screws are retained in the flanges, and in turn they loosely hold the frame and digital indicator together. Subsequent installation of the jacking screws in the adapter plate, hold the frame and indicator firmly together.

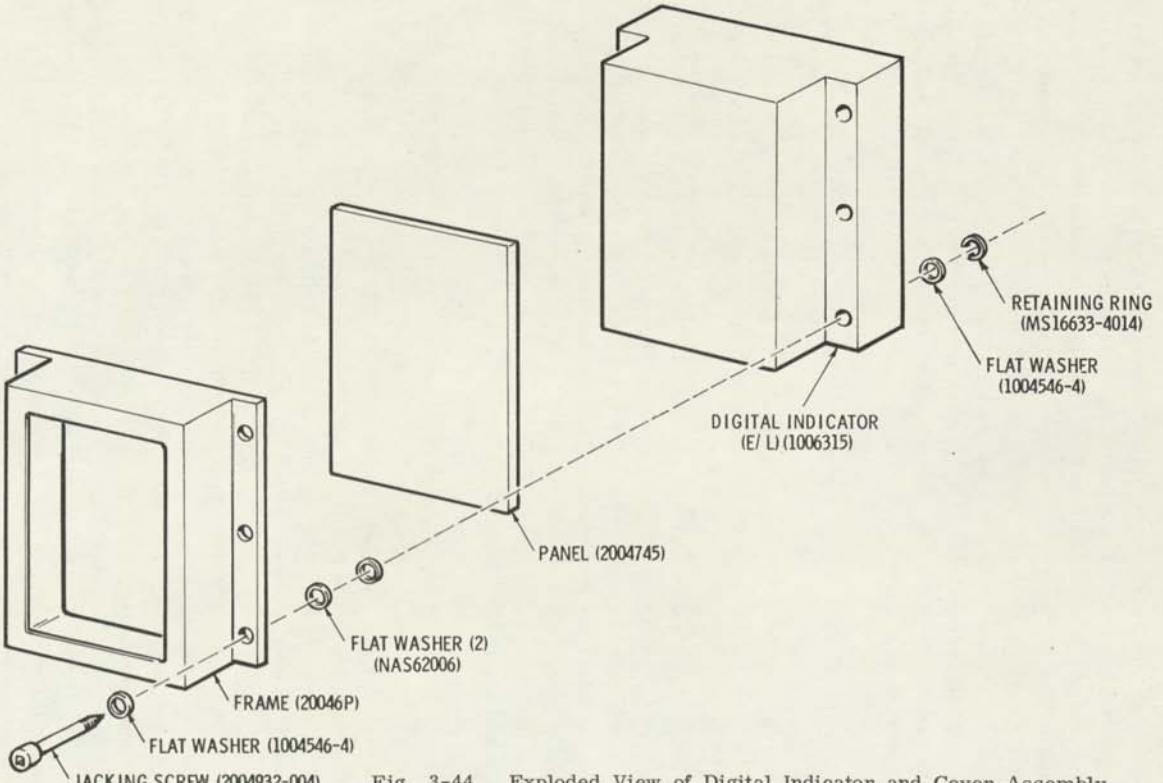


Fig. 3-44 Exploded View of Digital Indicator and Cover Assembly

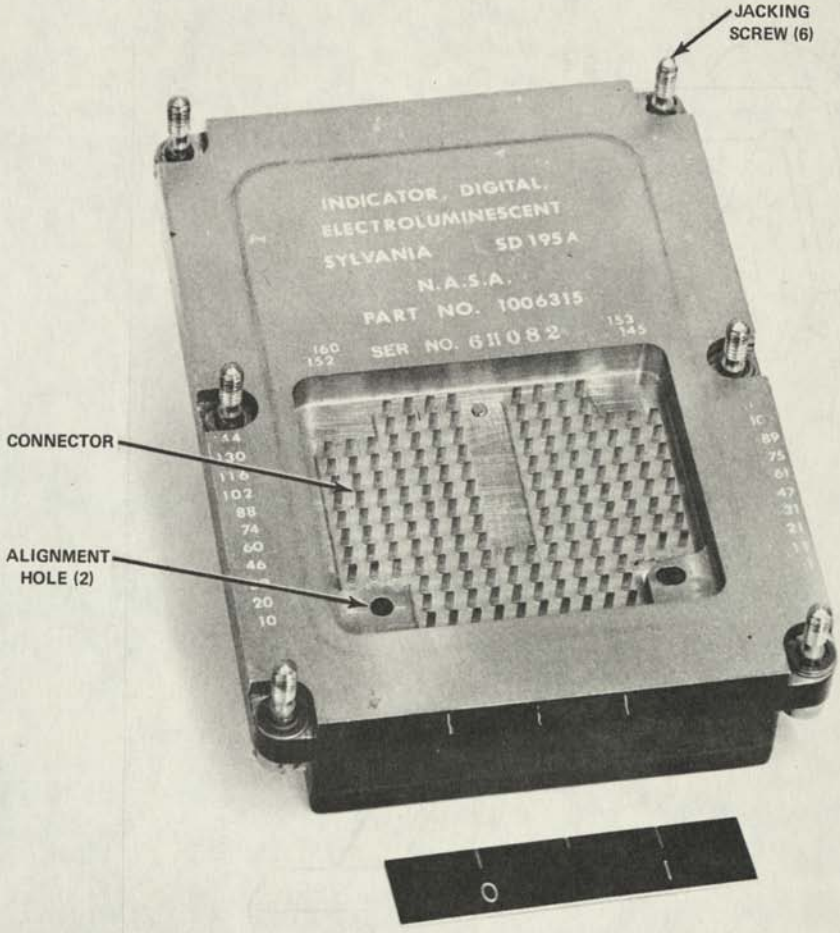


Fig. 3-45 Rear Side of Digital Indicator

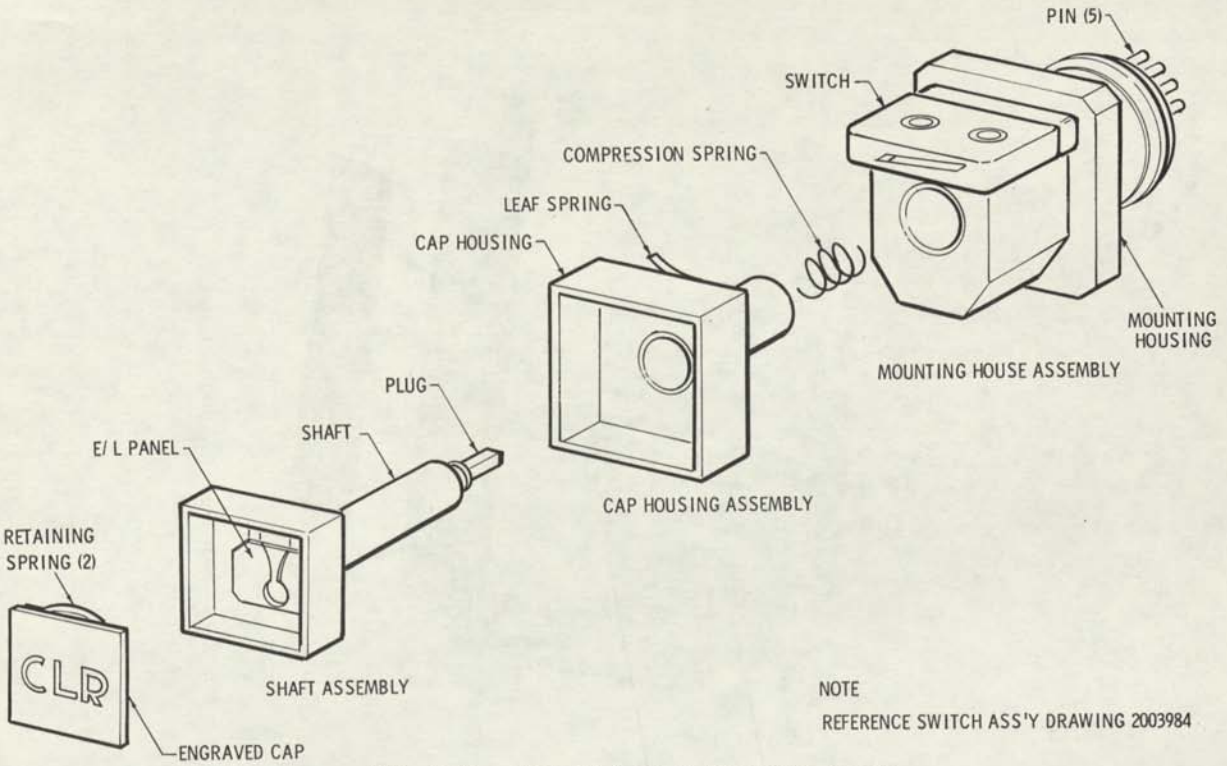


Fig. 3-46 Exploded View of Pushbutton Switch

3.10.1.5 Pushbutton Switch

An exploded view of a typical pushbutton switch is shown (See Figure 3-46). It consists of three functional assemblies: mounting houseing, cap housing and shaft.

The mounting housing assembly consists of the switch, housing, header, and disc. It serves as the mounting structure for the assembled pushbutton switch. The connector consists of five feed-through pins imbedded in disk shaped glass insulator. An integral metal ring surrounds the edge of the disc. This assembly is called a header, and it is soldered to the end of the housing. Soldered wires connect three switch terminals to three connector pins. The wires run from the terminals through a hole in the housing flange and proceed internally to the pins. Non-insulated wires from the remaining two connector pins terminate at a plastic disc that is bonded to the inside of the housing. These wires contact two wires on the shaft when it is inserted in the housing.

The shaft assembly consists of the shaft, plug, cable, and electroluminescent panel (EL). The aluminum shaft is a hollow cylinder with a square head that houses the EL panel and cable. The cable consists of two flat copper conductors sandwiched between insulators with solder pads on each end and is bonded to a recessed area in the head. The EL panel is placed over the cable and bonded to the head. Two metal tabs on the EL panel are soldered to pads on the cable. The plastic plug is inserted in the end of the shaft and bonded in place. Two wires are soldered to pads on the cable in the shaft head and run inside the shaft to the outside of the plug. Epoxy cement is used to hold the non-insulated end of the wires to the plug. When the push button switch is assembled, these wires contact the two wires in the mounting housing disc. This completes the circuit between the EL panel and the connector pins.

The shaft assembly is inserted in the cap housing assembly. A coil spring is placed in the mounting housing bore. The shaft/cap housing unit is inserted in the bore. The shaft, which extends beyond the cap housing, is bottomed and held in place with two set screws. They are threaded into the mounting housing and engage slots in the shaft. The compression coil spring forces the cap housing against the underside of the shaft head. The switch is not actuated in this position. Depressing the cap housing forces it down against the spring. Cap housing movement of approximately 3/16 inch is required to actuate the switch. An additional movement of 1/16 inch is allowed before the cap housing bottoms on the mounting housing. A leaf spring on the underside of the cap housing contacts a lever on the switch to cause actuation.

An engraved cap is installed on the cap housing and is held by two retaining springs. Actuating the switch energizes the EL panel and illuminates the cap.

SECTION 4.0
COMPUTER SOFTWARE

4.1 INTRODUCTION

The computer software developed concurrently with the computer hardware design manufacture. Software capable of testing the hardware was required as soon as hardware was available. In order to provide this testing hardware, many of the DSKY and utility operational programs were developed. Table 4-I is a listing of all significant program assemblies that were released for development testing of computer. The first, ECLIPSE, was designed to test the rope module manufacturing processes and also for use as an aid to computer testing. Earlier "breadboarded" releases of ECLIPSE were manufactured for computer debugging late in 1963.

In general, APOLLO computer programming* can be characterized as machine language programming of extremely high accuracy that exceeds or matches the hardware reliability. The programming was performed with limited support tools in an environment of concurrent mission design for hardware that was of limited capacity. The programs produced represented a high degree of organization in terms of tightly-packed instructions and data. Major program redesign is difficult, in general, but in APOLLO, this difficulty was increased as a result of the use of a wired memory. It was necessary to ensure that the program was correct the first time, because manufacture of new modules was both time consuming and costly. The pressure to meet scheduled release dates, on the other hand, did force the release-to-manufacture before program manufacture was complete. The interaction between the pressure to release and the pressure to thoroughly verify can best be illustrated by the results shown in Table 4-II. This table illustrates two major points: First, there was a strong desire to release programs early, so that most of the spacecraft checkout could be accomplished using mission type software. For example, program assembly 2021110-011 (SUNDANCE 292) was released in April 1968 for a mission that flew in April 1969 (APOLLO 9). Second, there was a strong desire to use the latest version of the program in the operational flight. A very interesting example was the Luminary program for Apollo 11. The program assembly (Luminary 69) was released in Nov. 1968. It was to be used for Apollo 10 and 11. In April 1969 one module was revised for the Apollo 10 mission but also Luminary 97 was released for Apollo 11. In May, three modules of Luminary 97 were revised to form Luminary 99. In June just one month before flight one of the modules was revised again. All of this was accomplished without an impact on

* For a more complete account see reference No. 12.

TABLE 4-I

LISTING OF COMPUTER PROGRAM ASSEMBLIES

COMPUTER PROGRAM ASSEMBLY NUMBER	COLLOQUIAL NAME	DATE RELEASED TO MFGR.	USE
1003203-011	ECLIPSE	Feb. 1964	Computer Test
1021100-011	ARTEMIS	March 1964	Computer Factory Test
1021100-011	MOONGLOW (Eclipse Rev.)	May 1964	Computer Test
1021102-011	SUNRISE	July 1964	System Test
1021104-011	ARES	Jan. 1965	Computer Factory Test
1021106-021	CORONA	Jan. 1966	APOLLO 2 Unmanned High Apogee Suborbital
1021107-031	SUNSPOT	July 1966	APOLLO 3 Manned Earth Orbital (not flown) BL I.
1021108-021	SOLRUM 55	Nov. 1966	APOLLO 4, 6 Unmanned High Apogee Suborbital
2021100-011	RETRED 44	July 1965	Computer Test BL II Prototype
2021101-011	AURORA	March 1966	System Test LM
2021102-011	VENUS	Jan. 1966	Computer Factory Test
2021103-011	RETRED 50	Dec. 1965	System Test LM
2021104-051	SUNDIAL-E	May 1966	System Test CM
2021105-021	NEWSPEAK	May 1966	Computer Factory Test
2021105-041	TWOSPEAK	May 1966	Computer Factory Test
2021107-011	LAMESH	June 1967	Computer Factory Test

TABLE 4-II

COMPUTER PROGRAM ASSEMBLIES RELEASED FOR APOLLO MISSIONS

Computer Program Assembly No.	Colloquial Name	Module Nos. Modified	Date Released to Mfgr.	Spacecraft	Launch Date	Mission
2021106-011	Sunburst		Feb. 1967	LM		
-021	Sunburst 116	B1, B3, B5	Apr. 1967	LM		
-031	Sunburst 120	B3	Oct. 1967	LM	Jan. 1968	APOLLO 5
2021108-021	Sundisk 282		Feb. 1968	CM	Oct. 1968	APOLLO 7
2021111-021	Colossus 236		Aug. 1968	CM		
-031	Colossus 237	B5	Aug. 1968	CM	Dec. 1968	APOLLO 8
2021111-041	Colossus 249		Oct. 1968	CM	Apr. 1969	APOLLO 9
2021110-011	Sundance 292		Apr. 1968	LM		
-021	Sundance 302	B1, B2, B3, B4, B5, B6	July 1968	LM		
-031	Sundance 306	B1, B2, B3, B4, B5, B6	Oct. 1968	LM	Apr. 1969	APOLLO 9
2021113-011	Comanche 44		Feb. 1969	CM		
-021	Comanche 45	B3	Mar. 1969	CM		
-041	Manche 45 Rev. 2	B2	Apr. 1969	CM	May 1969	APOLLO 10
2021112-011	Luminary 69		Nov. 1968	LM		
-031	Lum. 69 Rev. 2	B2	Apr. 1969	LM	May 1969	APOLLO 10
2021113-031	Comanche 51		Mar. 1969	CM		
-051	Comanche 55	B2	Apr. 1969	CM	July 1969	APOLLO 11
2021112-041	Luminary 97		Apr. 1969	LM		
-051	Luminary 99	B1, B2, B5	May 1969	LM		
-061	Lum. 99 Rev. 1	B1	June 1969	LM	July 1969	APOLLO 11
2021113-061	Comanche 67		July 1969	CM	Nov. 1969	APOLLO 12
2021112-071	Luminary 116		Aug. 1969	LM	Nov. 1969	APOLLO 12
2021113-071	Comanche 72		Oct. 1969	CM		
-081	Manche 72 Rev. 3	B2	Dec. 1969	CM	Mar. 1970	APOLLO 13
2021112-081	Luminary 130		Nov. 1969	LM		
-091	Luminary 131	B5	Dec. 1969	LM		
-111	Lum. 131 Rev. 9	B5	Jan. 1970	LM		
-121	Lum. 131 Rev. 1	B5	Feb. 1970	LM	Mar. 1970	APOLLO 13
2021113-091	Comanche 108		May 1970	CM	Jan. 1971	APOLLO 14
2021112-131	Luminary 163		May 1970	LM		
-141	Luminary 173	B1, B2, B3, B4, B5, B6	June 1970	LM		
-151	Luminary 178	B1, B2, B3, B4, B5, B6	Sept. 1970	LM	Jan. 1971	APOLLO 14

TABLE 4-III

PARTIAL LIST OF INTERPRETIVE OPERATORS

Operator	Average Execution Time, Milliseconds
DP Add	0.66
DP Subtract	0.66
DP Multiply	1.1
DP Divide	2.5
DP Sine	5.6
DP Cosine	5.8
DP Arc Sine	9.3
DP Arc Cosine	9.1
DP Square Root	1.9
DP Square	0.76
DP Vector Add	0.92
DP Vector Subtract	0.92
DP Vector x Matrix	9.0
DP Matrix x Vector	9.0
DP Vector x Scaler	3.3
DP Vector Cross Product	5.0
DP Vector Dot Product	3.1
DP = Double Precision	

spacecraft testing even though the normal manufacturing cycle for a module was 45 days.

The programs contained in ECLIPSE were a collection of routines, whose main purpose was to exercise an actual computer and DSKY. Since the only software checkout tools were the all-digital simulator, these programs provided a test bed for the software-hardware interface and an early test of the software development tools. The main program sections included in ECLIPSE were the utility, DSKY, and selfchecking programs. The fundamental function of each of these programs has not changed since their development, and is outlined in the following paragraphs.

4.2 UTILITY PROGRAMS

Utility functions are performed by program sections that coordinate and synchronize computer activities to guarantee orderly and timely execution of required operations. These functions control the operation of lunar mission tasks on either a priority or a real-time basis. The utility functions also translate an "interpretive" language to basic machine language that allows complex mathematical operations such as matrix multiplication, vector addition, dot product, and cross product computations to be performed within the framework of compact routines. In addition, the utility functions save the contents of registers A and Q during an interrupt condition, and enable data retrieval and control transfer between isolated banks in the fixed switchable portion of fixed memory.

4.2.1 Interpreter

Most of the AGC programs relevant to guidance and navigation are written in a parenthesis-free pseudocode notation for economy of storage. In a short-word computer, such a notation is especially valuable, for it permits up to 32,768 addresses to be accessible in a single word without sacrificing efficiency in program storage. This notation is encoded and stored in the computer as a list of data words. The AGC program called the "interpreter", translates this list into a sequence of subroutine linkages which result in the execution of the pseudocode program. A pseudocode program consists of lists called "equations". Each equation consists of a string of operators followed by a string of addresses to be used by the operators. Two operators are stored in an AGC word, each one being 7 bits long. A partial list of operators appears in Table 4-III.

Use of the interpreter accomplishes a saving in instruction storage over programs generated in an automatic compiler, and it affords the programmer a rapid and concise form of program expression that liberates him from the time consuming job of programming in basic machine language. In so doing it expands the instruction set into a comprehensive mathematical language accommodating matrix and vector operations upon numbers of 28 bits plus sign. This is made possible at the modest cost of a few hundred words of program storage and the cost of about an order of magnitude in execution time over comparable long word computers. The computer was designed with the idea that its weight, size, and power consumption were costly items. Mission requirements warrant a hardware compromise of word length with a minimum of 15 bits, and an instruction repertory of 33 instructions with which to work. The result, therefore, is a fairly simple machine with limited abilities. While the computer hardware provides for manipulation of single and double precision quantities, frequent need arose to handle multiprecision quantities such as trigonometric operations, vector and matrix operations, and extensive scaler operations. Thus, in order to fulfill the system requirements planned for the lunar missions, it was necessary to expand the capabilities of the computer through the use of the software.

One method of accomplishing this expansion would have been through a collection of subroutines. By creating within the computer a large library of subroutines that perform various higher level arithmetic and language operations, the burden of having to code complicated operations in extensive sequences of basic machine language would have been avoided. This approach had two distinct disadvantages, however. First, since programmers would be calling subroutines often, a large amount of memory would be occupied with frequently repeated calling sequences. Secondly, much of memory would be taken up with temporary storage for contents of registers that were needed for later processing.

To solve this memory wastage problem caused by frequent use of calling sequences, an entirely new mnemonic language was created in which each mnemonic corresponds to a subroutine. Since, in many cases, the new mnemonic instructions require no addresses, a "packed" instruction format was designed that stores two seven-bit operation codes in one word of memory and any required address constants in the two following words. To interpret the special mnemonic language, a central subroutine was designed that encodes the instruction formats using common temporary storage, and executes the required subroutine sequence of computer instructions.

Therefore, in addition to the machine language instructions described in Section 2, Table 2-IV, the software is provided with an interpretive language which permits mission peculiar memory requirements to be reduced, and in most cases sharply reduces the coding labor necessary to implement the required software functions.

The penalty paid for the interpretive language is execution time. A double-precision add order in basic machine language takes about 35 microseconds to be executed, while the analogous interpretive language order has been estimated to take about 660 microseconds. For most of the guidance oriented computations (such as orbital integration), the increase in execution time is not disadvantageous when weighed against the resulting reduced memory requirements. In other cases (such as digital autopilot calculations) the execution time penalty cannot be tolerated, and therefore these computations are performed, using machine language coding, in spite of the increased memory requirements that may result.

4.2.2 Executive

All AGC programs operate under control of the Executive routine except those that are executed in the interrupt mode. Executive controlled programs are called "jobs" as distinct from so-called "tasks", that are controlled by the wait-list routine and completed during interrupt time. The functions of the Executive are to control priority of jobs and to permit time sharing of erasable storage.

Jobs are usually initiated during interrupt by a task program or a keyboard program. The job is specified by its starting address and another number giving it a priority ranking. As the job runs, it periodically checks to see if another job of higher priority is waiting to be executed. If so, control is transferred away, until the first job again becomes the one with highest priority. No more than 20 milliseconds may elapse between these periodic priority checks.

When a job is geared to the occurrence of certain external events and must wait a period of time until an event occurs, it may be suspended or "put to sleep". The job's temporary storage is left intact during the period of inactivity. When the anticipated event occurs, the job is "awakened" by transfer of control to an address that may be different from its starting address. If a job of higher priority is in progress, the "awakening" will be postponed until it ends.

When a job is finished, it transfers control to a terminating sequence that releases its temporary storage to be used by another job. Approximately ten jobs may be scheduled for execution or in partial stages of completion at a time.

4.2.3 Waitlist

The function of the Waitlist routine is to provide timing control for other program sections. Waitlist tasks are run in the interrupt mode, and must be of short duration, 4 milliseconds or less. If an interrupt program were to run longer, it could cause an excessive delay in other interrupts waiting to be serviced, since one interrupt program inhibits all others until it calls for resumption of the normal program.

The Waitlist program derives its timing from one of the counter registers in the AGC. The counter priority stage, that controls this counter, is driven by a periodic pulse train from the computer's clock and scaler such that it is incremented every 10 milliseconds. When the counter overflows, than interrupt occurs that calls the Waitlist program. Before the interrupting program resumes the normal program, it presets the counter so as to overflow after a desired number of 10 millisecond periods up to a limit of 12,000 for a maximum delay of 2 minutes.

If the Waitlist is to initiate a lengthy computation, then the task will initiate an Executive routine call, so that the computation is performed as a job during non-interrupted time.

4.3 DISPLAY AND KEYBOARD[11]

The programs associated with operation of the display and keyboard units are intrinsic to the use of the computer in the APOLLO guidance and navigation system. These programs are long, but their duty cycle is low, so that their use of the time budget is reasonably small.

Key depressions initiate an interrupt program that samples the key code, makes a job request to the Executive, and then resumes. When this job is initiated, it examines the code and makes numerous branches based on past and present codes to select the appropriate action. Nearly always, a modification of the light registers in the display is called for. A periodic interrupt program similar to Waitlist, but occurring at fixed time intervals, performs the required display interface manipulations, after it has been initiated by the job. More complex situations occur as a result of lengthy processing of data and periodic re-activation of a display function. For example, it is possible to call for a periodic decimal display of a binary quantity, for which the Waitlist is required to awaken a sampling and display job every second. This job samples the desired register or registers and makes the conversion to decimal according to the appropriate scaling for the quantity, i.e, whether it is an angle, a fraction, an integer, etc., and where the decimal point is located.

The display and keyboard programs are highly sophisticated routines to which a certain amount of computer hardware is expressly dedicated for the sake of efficiency. They also make full use of the Executive and Waitlist functions to furnish a highly responsive and flexible medium of communication.

4.4 DEVELOPMENT OF COMPUTER SELF-CHECK PROGRAM

The self-check effort was originally undertaken to provide a laboratory program to aid in debugging the computer design. The goal was to exercise as much of the machine as possible through use of the program to verify correct operation. No attempt was made to provide diagnostic information when an error was detected. However, the program was divided into functional sections, and when an error occurred, a record was provided indicating which part was running at the time. After the computer design was confirmed, self-check served as a confidence tool. A successful run through all of its routines indicated that no hardware malfunction existed.

4.4.1 Block I Versions

4.4.1.1 Eclipse

The first computer self-checking program (included in ECLIPSE) was an early effort at verification of the operation of the computer instruction set. Each instruction was exercised and its operation checked for predetermined results. This test did not attempt to verify the operation of each control pulse in the computer, but rather to verify nominal operation at the instruction level. Some attempt was made to check instructions referencing the three classes of memory: fixed, erasable, special and central.

4.4.1.2 Sunrise

The self-checking programs, included in SUNRISE and following programs, were significantly different from earlier versions. The main change in emphasis was that the checking program exercised and verified the computer operation at the control pulse level, rather than at the instruction level.

The main program sections included in SUNRISE are listed and described below.

1. Check of Pulses – Most of the control pulses in a given instruction are used every time that instruction is used. However, the functions, that some of these pulses perform, are not utilized until some time later. A systematic method is used to check the existence of pulses that perform such functions. As an example, the pulses, that write a data or instruction word back into erasable after it has been used, are not checked, until that data or instruction word is used again.

Some of the control pulses serve no useful purpose. They appear in various memory cycles because they are utilized on the same line in other memory cycles. It is more economical, with respect to the physical construction of the computer, to let them appear where they were not required than to omit them.

There are two memory cycles and a few control pulses that cannot be checked without introducing signals external to the computer. Specifically, the control pulse in the SHINC and SHANC memory cycles and the WOVR pulse in the MINC memory cycle are not checked. It is also not possible to check the WOVI pulse on line 11 of the NDX1 memory cycle.

No particular effort was made to check the pulses connected with the S and Z registers and the NISQ pulse. Some of these pulses are used in every memory cycle, and the fact that SELFCHK is successfully completed assures the existence of these pulses.

2. Check of Special and Central Registers and Erasable Registers – This section of self check ensures that the A, B, C, Q registers and the output of the adder have all the possible 16 bit combinations pass through them at least once. At the same time this is being processed, all 15 bit combinations are written into and called from an erasable register. Next, the erasable memory from 1774_8 down through 60_8 is checked. It is not necessary to check erasable registers 1777_8, 1776_8, and 1775_8 at this time, because they have been used many times previously during the normal operation of SELFCHK. Then the special erasable registers from octal 57 down through octal 20 are addressed to see if a parity error occurs. Finally, the cycle and shift registers are checked by putting a combination of alternate 0s and 1s in these registers and ensuring the correct operation is performed.

3. Check of Multiply and Divide Arithmetic Function – There are four multiply loops in the multiply subroutine. The two main purposes of this subroutine are to form all the different combinations of adds possible in the multiply instruction (1 to 14) and to change the value of the word to be added from minimum to maximum for each combination of add. The total time of the multiply routine takes approximately 30 seconds.

The four divide subroutines form different combinations of subtractions while varying the value of the word to be subtracted. It takes approximately 0.012 seconds to go through all the four divide subroutines. However, SELFCHK keeps the computer in the divide subroutines for approximately 20 seconds.

The multiply and divide subroutines are a good arithmetic check of the computer. Therefore, the long activity time of these subroutines may be utilized to check normal operation of the computer along with asynchronous and synchronous interface signals.

4. Operation Procedures and Malfunction Indications – SELFCHK runs as part of the background idle loop and thus has the lowest possible priority. The number loaded into the SMODE register via the DSKY controls the manner in which SELFCHK will be run in the background idle loop. The options are:

   a) SELFCHK runs continuously through all of its routines, if there is no job of higher priority.
   b) Only the pulses and erasable check are performed.
   c) Only the pulses check is performed.
   d) SELFCHK is off.

   The three SCOUNT registers may be utilized to monitor the operation of SELFCHK. Register, SCOUNT, is incremented upon the successful completion of the PULSES part of SELFCHK. Register, SCOUNT + 1, is incremented upon the successful completion of the ERASABLE + SC part of SELFCHK. Register, SCOUNT + 2, is incremented upon the successful completion of the MP + DV part of SELFCHK. The three registers may be displayed using the DSKY.

   If SELFCHK should locate a malfunction, the following sequence of events occurs:

1. The contents of the Q register are placed in the SFAIL register.
2. The program alarm light on the DSKY illuminates.
3. An alarm code is placed in R1 of the DSKY and placed in the FAILREG register.
4. The ERCOUNT register is incremented by +1.
5. The SELFCHK routine is started at the beginning.

If a second malfunction is located, all of the previous steps are repeated. It is therefore possible to leave the SELFCHK routine on for long periods. One can keep track of the number of malfunctions, and their location in the three main parts of SELFCHK, by looking at the four count registers. The SFAIL register contains the error address +1 of the last malfunction.

4.4.1.3 Final Block I Version[1]

The final Block I SELFCHK contained a check of fixed memory in addition to the sections described above.

1. Check of Rope Memory – The routine for checking the correct contents of a rope is called ROPECHK. Its purpose is twofold. First, it is a check on the computer. It makes sure all current drivers, sense amplifiers, and associated circuitry used in connection with the fixed memory are operating properly. Second, it is a check on the rope itself. It makes sure none of the sense or inhibit lines have become shorted or opened (essentially guarantees contents of rope are correct and can be read correctly by the computer).

   The sum of each bank should be the same as its bank number in the low order bits of the computer. A bugger word CHECKSUM constant is added to the normal sum of the bank as the last word to be added. This bugger word forces the sum of the bank to be plus or minus the sum of the bank. As an example, the sum of bank 33 octal may be 00033 or 77744. When bank is full, the "bugger" word is the last address in the bank. If the bank is not full, two or more successive TC SELF words indicate the end of the summing process for that bank. The "bugger" word then immediately follows the last TC SELF word. Of course, all addresses in a bank up to the bugger word have to contain words of good parity.

2. SHOW-BANKSUM Routine – SHOW-BANKSUM consists of a routine called SHOWSUM. This routine essentially does the same thing that the routine ROPECHK does; that is, add up the sum of separate banks in the rope. ROPECHK makes sure the sum of the bank is plus or minus its own bank number, while SHOWSUM displays the sum of the bank in R1 of the DSKY irrespective of what the sum may be. SHOWSUM also displays the bank number and the bugger word in R2 and R3 of the DSKY at the same time. The sum of the bank and bank number in R1 and R2 are shown as the least significant bits instead of bits 11-15 (the actual bank bits in the computer). Again it is worthwhile mentioning that the sum of a bank may be plus or minus its bank number. That is, bank 5 may be 00005 or 77772.

   Undoubtedly the greatest use of this routine will be in restoring the confidence of personnel in the computer and in verifying that the correct rope modules for a particular mission are actually the ones in the computer package. Since there are 24 bugger words for each rope, it is very unlikely that ropes for two different missions would have the same 24 bugger words.

4.4.2 SELFCHK in Block II[2]

The development of the Block II AGC SELFCHK was similar to that of the Block I version. Early efforts were directed at checking at the instruction level, as was the case in Block I. Subsequently, a thorough test at the control pulse level was developed. The functional characteristics of the final version were similar to the final Block I program. Naturally the differences between the Block I and II computer affected the details of the program. In particular, the test of the control pulses was rewritten for the expanded set of Block II instructions. The erasable memory check was modified to handle the expanded erasable memory and its associated bank organization. The fixed memory check was modified to check the expanded fixed memory and its superbanks.

A routine added in the Block II version is the DSKY check. This routine lights up all the DSKY electroluminescent elements by displaying 9 through 0 in the Verb, Noun, Prog, R1, R2, and R3 registers. It also turns on the computer activity light and the verb-noun flash.

The operating procedure for Block II allowed more flexibility in controlling which routines were executed by SELFCHK. An expanded set of numbers to be loaded into SMODE were defined to permit the operator to initiate selected routines, some of which were previously grouped under a single option number.

The alarm procedure for Block II was made more flexible than that of Block I. Depending on the contents of SMODE, following a detected error, SELFCHK will either continue at the next address after the error line, start at the beginning of the test being run, or stop.

Certain sections of SELFCHK were deleted, as the Block II mission evolved due to the limited amount of fixed memory available. Specifically, the extensive multiply and divide checks were not present in SUNBURST, SUNDISK, and following programs. The pulses check and the DSKY check were not included in SUNDANCE, COLOSSUS, LUMINARY, and following programs.

4.5 FACTORY TEST ROPES

The factory test ropes were developed by Raytheon Company to sell-off AGCs, as they were produced. In general, the factory test ropes furnished a degree of diagnostic capability not present in the SELFCHK programs developed at MIT/IL. The SELFCHK programs were, however, incorporated into the factory test ropes.

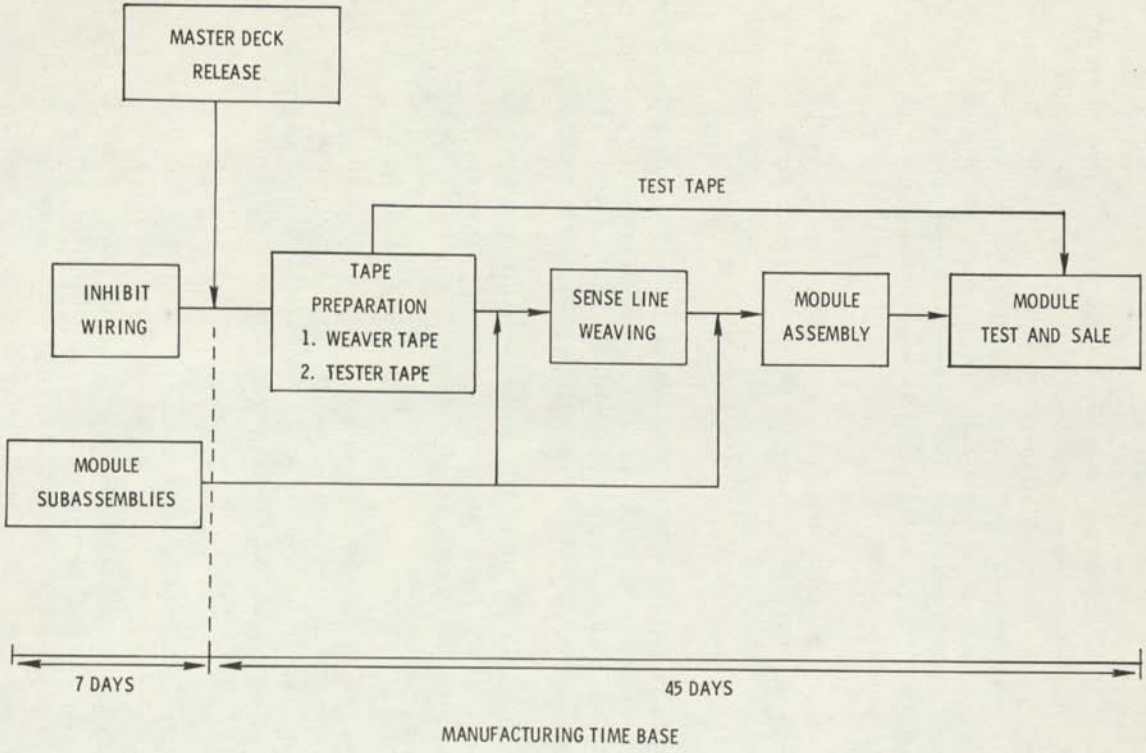


Fig. 4-1 Manufacturing Flow Plan

ARTEMIS and ARES were the names of the Block I factory test ropes; NEWSPEAK and LAMESH, the Block II versions. The degree of diagnostic capability increased as usage experience developed. LAMESH, being the latest Block II version, had considerable flexibility and diagnostic capability.

4.6 FIXED MEMORY FABRICATION

The APOLLO mission program that is to be manufactured as rope modules is specified in the Guidance Systems Operations Plan (GSOP). The GSOP is designed to cover in practical detail all the elements necessary for coding the mission programs. Following program verification of the requirements specified in the GSOP, the program is ready to be released for manufacturing of the fixed memory modules. One of the key elements in the fabrication process is the methods of verifying that the contents of the memory module are a bit for bit duplication of the binary output of the program assembly.

4.6.1 Program Release

Upon receipt of written direction to release a mission program, a deck of "director cards", key punched with appropriate orders is submitted to the computer. These cards direct the computer to generate, list and verify a master deck magnetic tape for delivery to manufacturing. The tape specifies the sense line wiring of each memory module and module position, when installed into a computer. These include module numbers B1, B2, B3, B4, B5, and B6. This completely specifies the bit by bit contents of each module and the location or address of these contents.

The manufacturing process starts by processing the master tape to produce two punched mylar tapes; a braid (core rope weaver) tape and a checker (core rope tester) tape. The braid tape actuates the core rope weaving machine that controls the weaving of the sense line wires in the memory module. The woven memory is tested along with the checker tape to verify the contents of the memory following fabrication. The simplified flow plan and time base for the fabrication cycle is shown in Figure 4-1.

4.6.2 Module Data Verification

The verification method depends primarily upon parity tests and verification of the memory bank sum check and sum check constant. As described previously, the last word in each memory bank is a constant that makes the sum of all the words of the bank equal to plus or minus the bank number. The self test, when running in

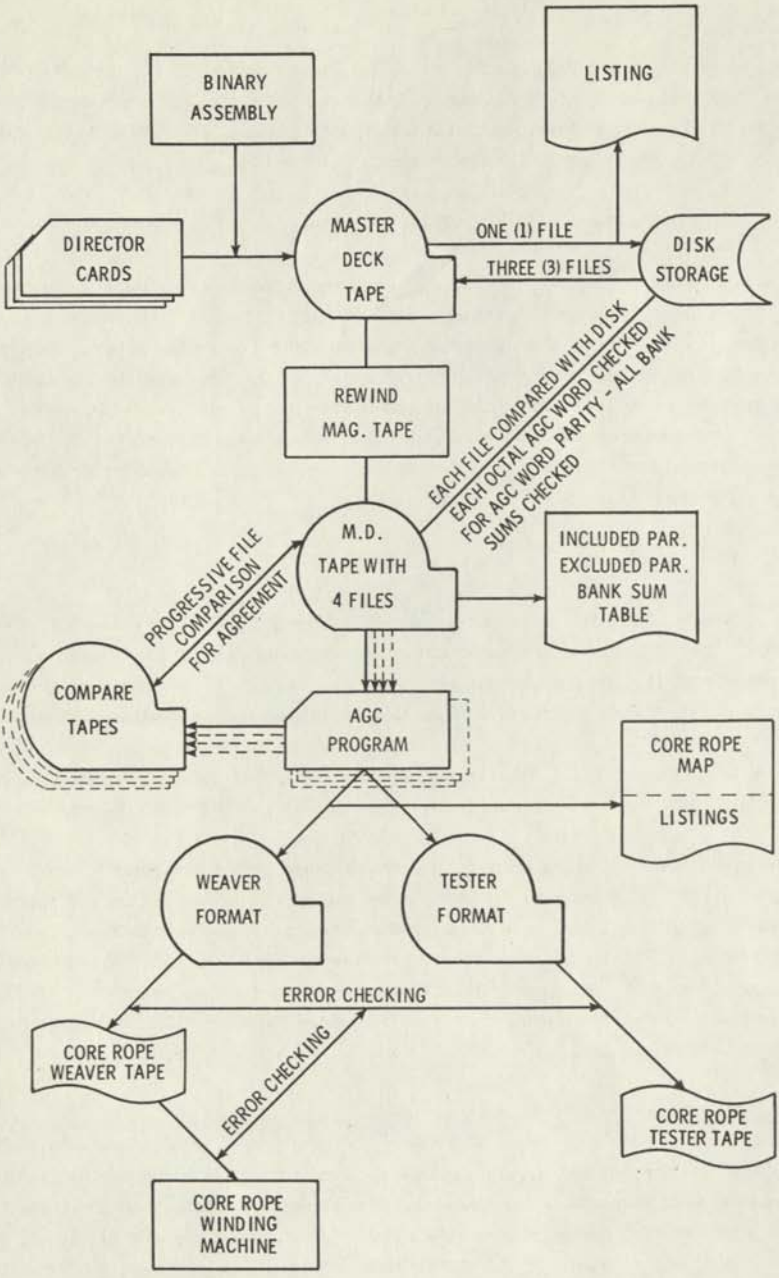


Fig. 4-2 Functional Diagram of Data Processing and Verification during Manufacturing

an AGC program, is capable of summing the contents of each word in the bank and displaying the results for a test of the memory. This technique then is applicable for verification of data at any point in the manufacturing process. It is used in addition to parity tests and the more standard bit by bit comparisons of data during operations such as generating the master deck tape or generating weaver tapes from the master deck tape. The check sum testing closes the testing loop from any point, where it is used, back to the binary record of the program assembly. Figure 4-2 is a functional diagram of the data processing and verification part of the manufacturing cycle.

SECTION 5.0
GROUND SUPPORT EQUIPMENT

5.1 INTRODUCTION

The development of the computer ground support equipment (GSE) paralleled the computer development. There were two classes of GSE. One was the engineering tools that were developed to support the hardware and software design, and two, the production GSE that was developed to support the manufacturing and field testing. The latter, although similar functionally to the engineering tools, was designed and manufactured by the support contractors. Figure 5-1 is a photograph of this production GSE or Computer Test Set which represents the major portion of the equipment required for computer selloff and field testing. The following sections describes the engineering tools developed at MIT/IL to support the hardware and software development. Figures 1-3 and 1-4 illustrate the early versions of these equipments.

Late in the production phase of the hardware and software, it became apparent that the GSE equipments were not adequate for the analysis of transient anomalies. For example, a transient failure during computer vibration or during software verification was very difficult to analyze, since neither the computer nor the GSE provided the necessary test cooperative features. As a result, MIT/IL designed a system called Trace (see Section 5.3.3) to aid software debugging and in parallel, additional factory test equipment was designed to aid in the analysis of hardware transient failures. Each of these equipments were similar to a system called Coroner which was designed and built very early in the Apollo program. The function of the Coroner was to store data for a short period of time before and after a transient failure. Due to a lack of interest and technical difficulties, the Coroner development was dropped. The value of a capability like Coroner was not fully appreciated until the production phase of the development.

5.2 DEVELOPMENT OF GSE

This section describes the ground support equipment (GSE) used to check the operation of the computer subsystem. Checkout of the computer can be divided into two areas:

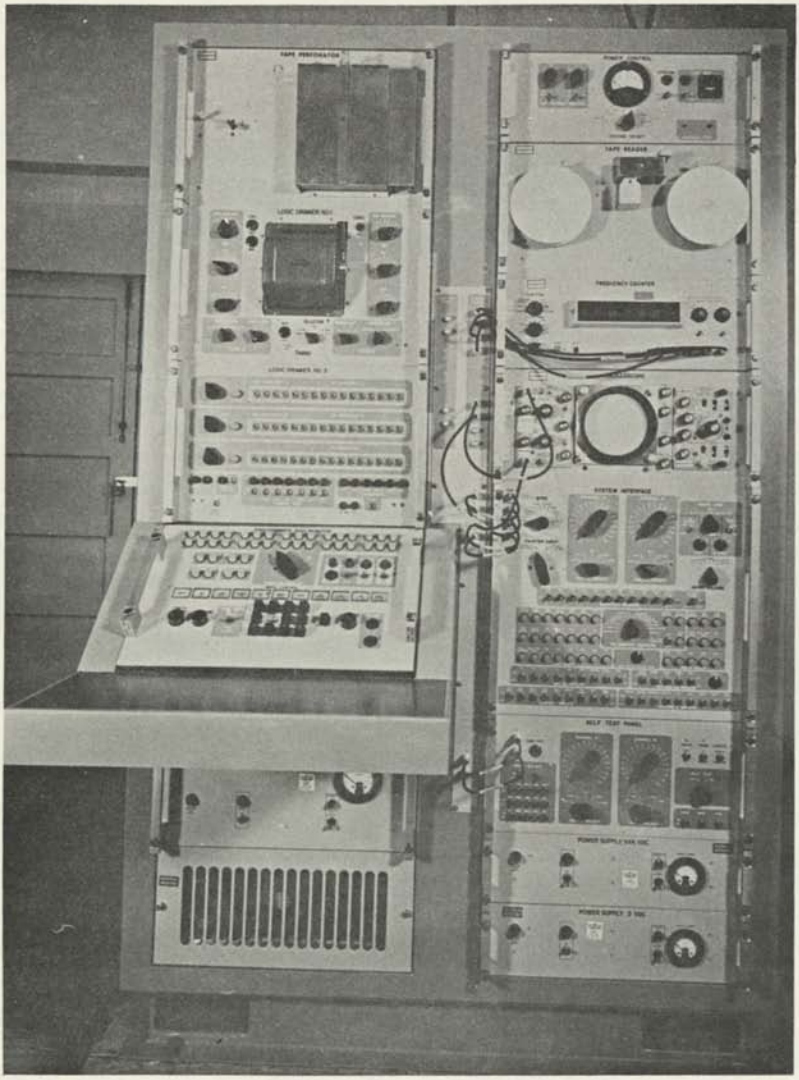


Fig. 5-1 Computer Test Set

1. Internal Operation – proper execution of operation codes, interrupt and counter increment processing, etc.
2. Interface – operation of the interface lines between the AGC subsystem and other subsystems.

Although there is some interaction between the two areas, two separate pieces of equipment were used at MIT/IL to provide the full checkout capability. The reasons for this are partly historical and partly because of the "system environment" in which computers are used at MIT/IL. (The Computer Test Set incorporates both capabilities. Its operation is basically the same as the "AGC Monitor" and "Interface Test Console" discussed in the following two sections).

5.2.1 Monitor

During the computer development phase a means of verifying the "central processor" portion of the AGC was required. The first monitor was built to serve this purpose. It provides a continuous display of the accumulator (A), instruction location counter (Z), memory address register (S), memory data buffer (G), operation code (SQ), and several status bits. In addition, the "W" register can be used to display the contents of any register or memory location, or to display the contents of the data bus (Write Lines) during any particular portion of the micro-program sequence. This is accomplished by providing external access to the write lines and the various control pulses in the computer. These signals were made available at the test connector of the computer. By providing duplicate registers in the monitor and supplying write line and control pulse information from the AGC, the contents of the registers internal to the AGC can be copied and displayed

By adding comparison and start/stop logic, the monitor can be used to stop program execution at a certain point or to "step" through a particular sequence. This capabiliity, along with the display, allowed the engineer to debug the central processor portion of the AGC.

Present monitors have an expanded display panel and increased comparison circuitry due to the increased complexity of the AGC and the use of the monitor as a software debugging aide. A more complete description is contained in sub-section 5.3.1.1.

5.2.2 Interface Test Console

The Interface Test Console (ITC) was designed to check the interface signals of the computer. The electrical parameters of the various interface networks were

specified and tested during the development phase of the AGC. The ITC is used to verify and test the interfaces at the computer subsystem level.

"Static" testing of the interface signals is performed to verify the hardware. This is usually accomplished by supplying a repetitive pattern for pulse inputs and programming repetitive patterns for pulse outputs. Checkout of discrete inputs and outputs is accomplished manually. The operation of the ITC is essentially identical to the operation of the "interface" portion of the Computer Test Set (CTS).

The ITC can also be used to simulate interface data for other spacecraft systems as a software debugging aide. This will be discussed further in Section 5.3.2.

5.2.3 Portable Monitor (Block I)

During spacecraft testing of the early versions of Block I test ropes, computer "lock-up" problems were encountered. The symptom of a "lock-up" was usually that of ignoring all input stimuli. The only known cure was to remove the computer from the spacecraft and to correct the situation via a Computer Test Set in the GN&C Lab. (Later findings showed that large electromagnetic disturbances could cause and also cure the "lock-ups". This "cure", obviously, was not the most desirable).

A "Portable Monitor" was designed to diagnose and cure the computer problems in the spacecraft, eliminating the need to remove the AGC from the spacecraft. The capabilities and operation of the Portable Monitor are similar to the AGC monitor. Due to physical constraints only one data register and associated display are provided, but its operation is similar to the monitor's "W" register. For example, it can copy any selected register or memory location. One additional feature is included in the Portable Monitor that is not included in other test gear. This is the capability of automatically loading all of erasable memory with a specified constant. This capability was incorporated to cure the "lock-up" problems. (It was determined that the "lock-ups" were due to erroneous "data" in the AGCs erasable memory, and that a cure was to "zero" erasable memory).

5.2.4 "Blue Box" (Block II)

The switchover to the Block II AGC posed a new problem in analyzing malfunctions. Both the Block I and Block II AGCs contain hardware alarm circuits that check for various abnormal conditions. In the Block I system occurrence of these alarms would illuminate corresponding indicators on the NAV DSKY. The Block II system,

however, illuminates only one common RESTART indicator. In addition, some alarms do not cause a restart and are monitored only by the AGC Warning circuit. Since the warning integrator "averages" its inputs, single or low repetition rate alarms of a certain class could occur and not be noticed.

Individual occurrences of each alarm can be monitored at the AGC's test connector. The "Blue Box" was designed to record alarm conditions when other equipment, such as an AGC Monitor or Computer Test Set, are not available.

The "Blue Box" consists of flip-flops and indicator lights used to record and display all alarms, two voltmeters, control "pots" to monitor and vary the AGC'S 4 and 14 volt power supplies, and indicator lights to display several of the status bits and controls to disable some functions of the AGC (interrupts, counter increments, restarts). Although the "Blue Box" cannot "freeze" the AGC upon occurrence of a restart, its ability to display the various alarms and status bits is quite useful in analyzing malfunctions, when other more sophisticated test equipment is not available.

5.2.5 Restart Monitor

The Restart Monitor was designed for use in spacecraft testing and flight operation where access to the AGC's test connector was not practical. Its function is to monitor the alarm outputs of the test connector. Since the test connector is not readily accessible when the AGC is installed in the spacecraft, the indicator lights used in the "Blue Box" were replaced by circuitry that allowed the computer to interrogate and reset the "indicators" under program control. This is accomplished by utilizing one of the spare input/output "channels" of the AGC.

Occurrence of an alarm will set a corresponding flip-flop in the Restart Monitor. The status of these flip-flops can be interrogated by performing a channel read instruction on channel 77. The flip-flops are reset by an channel write instruction on channel 77. This allows the various alarms to be monitored and displayed via the DSKY and/or telemetry.

5.3 DEVELOPMENT OF SOFTWARE CHECKOUT AIDS

5.3.1 Core Rope Simulators

Since the AGC employs "ropes" for program memory, a method of providing a rapid-turn-around program memory is required to support software debugging. To meet this requirement a core rope simulator (CRS) is utilized.

Through appropriate interfacing between the AGC and the core rope simulator, data that would normally be contained in a set of "ropes" is transmitted from the CRS to the AGC. From the software point of view, there is no difference in performance between using actual "ropes" or CRS data.

There are several different versions of core rope simulators in use with varying interface methods and additional capabilities. However, all versions utilize a read/write ferrite core memory to store AGC program data. The memory is cycled in a "read-restore" mode (effectively a non-destructive readout cycle), when accessed by the AGC, thus simulating the "fixed" content of the "ropes". Any word may be written into, however, via controls on the CRS. This allows individual words, or the whole assembly, of a program to be readily changed.

5.3.1.1 AGC Monitor

"AGC Monitor" is the name of the CRS most commonly used at MIT/IL. Its name is a "carry-over" from the early days when the Monitor did not have core rope simulator capability (see Section 5.2.1). A brief summary of its main hardware features follows:

1. Up to 40K of read-write memory available to simulate the ropes (the AGC presently uses 36K).
2. Paper tape reader and/or magnetic tape transport used to "bulk load" the CRS memory and the AGC erasable memory if desired. All of either memory or selected portions can be "dumped" onto magnetic tape for later processing.
3. Display hardware and logic to properly interface the CRS and AGC.

Functional capabilities of the AGC monitor are:

1. Simulate portions or all of the AGC "rope" memory.
2. Continuous display of the address registers (S, FBANK, FEXT, EBANK), double precision accumulator (A,L), instruction location counter (Z), operation code (SQ) and status bits, and memory contents (G).
3. Display, either continuously or selectively, any register or memory location in the "W" register of the monitor.
4. "Single step" program execution.
5. Stop program execution, when specified conditions are met (execution of a particular instruction, particular contents of a register or memory location, state of various status bits, or any combination of the proceedings). This capability is often used to allow the program to run to a certain point where the capability, mentioned in item 4, is used to "step" the program through a questionable area.

6. Load or read any memory location without disturbing program execution.

Capabilities 2 through 6 are not required for core rope simulator but are available as diagnostic aids in debugging hardware and software problems.

5.3.1.2 Portafam

The Portafam is a compact core rope simulator intended for use in areas, where physical constraints (such as inside the spacecraft) preclude the use of a "standard" CRS.

Physically the Portafam consists of three subunits:

1. An IBM compatible, seven track, tape transport to provide a means of bulk loading memory.
2. An interface adapter, that occupies the space normally used by the AGC "ropes", to convert AGC "rope" signals into logic levels and vice-versa.
3. A 48K of read-write memory and control logic to interface and communicate between the AGC, tape, and memory.

Functionally the Portafam is composed of three subsystems:

1. Memory Subsystem – 48K words by 16 bits/word ferrite core memory and associated drive electronics. 36K is used to store AGC program data for CRS purposes. The remaining 12K, called "trace memory", is used as a "pushdown" stack for storing diagnostic information.
2. Tape Subsystem – consisting of the tape transport and associated control electronics, and logic to format data transfers between the tape subsystem and the memory subsystem. Provision is made for two separate program assemblies to be contained on the tape, allowing a "mission" or "special" program to be loaded without physically interchanging tapes.

   All tape records utilize both lateral (vertical) and longitudinal parity along with a two bit sequence code included in each byte of the tape. Three simplex copies of a program are written on the tape to provide a means of error recovery. This format was adopted because errors due to tape imperfections could cause multiple errors rather than a single bit error. The alternatives, such as multiple error correction (such as Hamming codes), did not appear to justify the extra logic or effective loss in packing density.
3. "Master Control" Electronics – This controls the communication between the tape subsystem, the memory subsystem, the AGC, and the operator display and control panel.

There are four basic operations performed by Portafam: simulation of AGC rope data, verify loading of the memory, storing diagnostic information in the "trace" memory, and retrieval of the "trace" memory information. Simulation of rope data is accomplished by monitoring the AGC's rope connector for fixed memory accesses. When fixed memory data is required by the AGC, the information is retrieved by the corresponding location in Portafam memory and transferred to the AGC via the interface adapter.

Verification of loading of the memory can be initiated manually or by detection of an AGC Restart. Since it is not possible to infer the cause of a Restart from the rope connector, all Restarts are assumed to be the result of erroneous data in the Portafam memory. The tape unit is started and each location of the data memory is compared against the tape data, with all discrepancies being recorded in the "trace" memory. The sequence performed in this operation is briefly as follows: An AGC data word is assembled from tape. This information is compared against the information in the corresponding Portafam memory location. If a discrepancy is detected, the tape information is, usually, loaded into the memory. The same memory location is again readout and compared to determine, if a "hard" failure is present. If any errors are detected in the above sequence, three words are written into the "trace" memory that contain the memory address, type of error, and the data in error.

The diagnostic information stored in the "trace" memory consists of the memory address accessed each AGC memory cycle. Since erasable memory activity cannot be completely monitored at the rope connector, complete information on these cycles cannot be provided. Any discrepancies between tape data and contents of the memory are also recorded.

The "trace" memory information can be retrieved in several ways: manual readout via the control panel, "dumped" onto tape by manual request or automatically upon detection of a Restart, or it can be accessed by the AGC via a particular address sequence.

5.3.2 Interface Simulators

5.3.2.1 Interface Test Console (ITC)

In addition to providing "static" checkout capability of all AGC interface signals (Section 5.2.2), the ITC possesses the capability to simulate, in "real time", the telemetry interface to the AGC. This capability includes:

1. Reading K-start tapes and simulating the timing of the spacecraft checkout Uplink computer.
2. Downlink interrogation at the 50 word per second rate. The ability to record Downlink transmissions at a 10 word per second rate.

The K-start capability is used to verify tapes used for ground checkout, and provides a convenient means of input for special tests.

The Downlink recording capability is used to verify the AGC Downlink programming and provides a means of collecting data during special tests.

5.3.2.2 Uplink-Downlink Exercisers

The various Uplink-Downlink exercisers were built to provide telemetry interfacing for the several GN&C systems (at one point there were three systems on test: Block I, CM, and LM). The Uplink portion is, basically, the K-start reading capability mentioned in the last section. One of the Downlink units has the capability of recording information at the 50 word per second rate.

5.3.3 Trace System

Additional test cooperative feature is provided by the Trace system. This system aids software testing and debugging by recording the contents of selected AGC registers in a "pushdown" list which is frozen following an indication of a computer problem, such as restart or computational error. The Trace system includes a Core Rope Simulator, Coroner, Tape Recorder, and analytical software as shown in Figure 5-2. The Core Rope Simulator (CRS) provides the normal erasable memory simulation of the computer fixed memory. The Coroner provides for decoding and storage of selected data during the operation of the computer. The contents of the Coroner memory can be dumped onto the tape recorder for analysis off line.

5.3.3.1 Coroner Characteristics

The Coroner's function is to arrange incoming AGC register data into various formats according to the conditions programmed on the Coroner control panel and to store the data in the Coroner's "pushdown" memory. The memory capacity of the Coroner is 8192 x 64 bit words. In order to achieve maximum flexibility with this memory several modes of operation were provided which permits selectivity in the data recorded. Some of the selectivity features include the following:

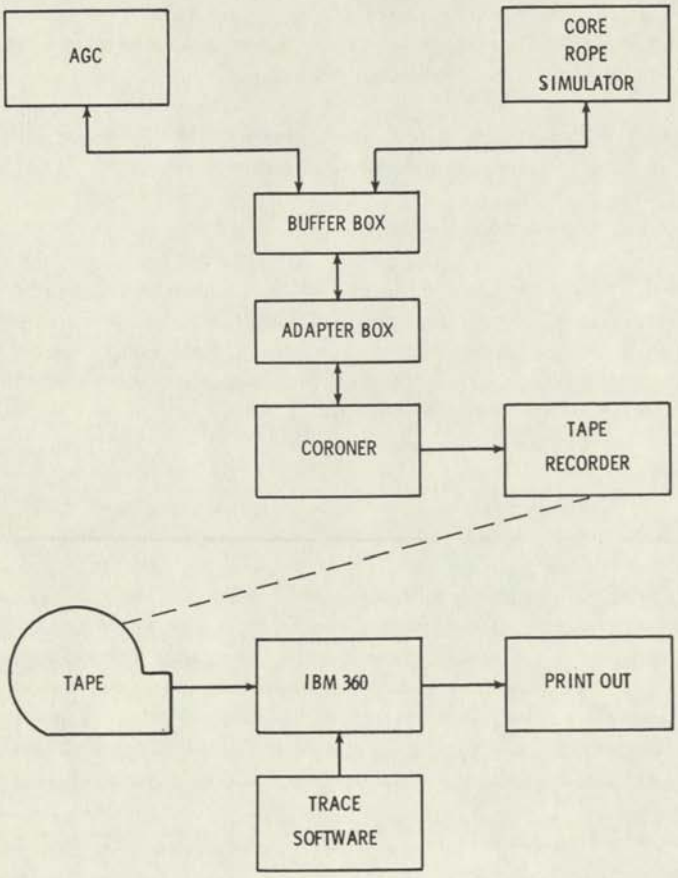


Fig. 5-2 Trace System Block Diagram

a. Provision to ignore interrupts and counter increments.
b. Provision to record only interrupts, increments, selected bank registers, and branching operations.
c. Provision to record data between selected program points or when specified locations are addressed.

The Coroner provides the ability to stop the computer on various conditions such as:

a. Coroner memory is full.
b. A selected number of memory cycles is complete.
c. A selected number of memory cycles after the indication of an error.

These various modes provide for considerable flexibility for the analysis of software bugs and operating conditions. The data which is collected during a trace operation may be displayed one location at a time for "on-line" analysis, or in the dump mode the contents of the Coroner memory and the settings of the Coroner controls are recorded on magnetic tape for "off-line" analysis.

5.3.3.2 Trace Analytical Software

The analytical software was designed to operate on the IBM 360 series computer and print out the magnetic tape data in various formats to aid in debugging. There are eleven possible formats for the printout. The following two modes illustrate the contents of the printout:

1. Partial Edit mode prints a line of output corresponding to each memory cycle in which data from the computer is stored in Coroner. The information printed depends upon the Coroner mode data which is included in the printout. The data is printed in octal code.
2. Basic Instruction mode prints a line of output for every computer instruction recorded by Coroner. The information printed includes the Coroner mode, the contents of the computer S register, the instruction mnemonic, the operand address (K), the contents of K after execution of the instruction, contents of the accumulator and lower accumulator after execution, contents of Q register and the contents of B bank.

SECTION 6.0
SYSTEMS INTEGRATION

6.1 INTRODUCTION

Integration of the digital computer with the APOLLO guidance system entailed the design and specification of electrical interfaces, the verification of electromagnetic compatibility, and the solution of problems encountered during field testing. Because of the multiplicity of interfaces required for both command module and lunar module systems, spacecraft systems integration presented a significant task for the interface designers. This was because a reasonable amount of standard circuitry was required for simplicity in the computer, and also to provide for multiple application of the circuits in the two different spacecraft configurations. The interfaces between the computer and the spacecraft systems were specified in considerable detail in the Interface Control Documents (ICDs). However, the interfaces between the computer and the guidance system were not as rigidly controlled by specification. But because the same circuits were used to interface both the guidance components and the spacecraft systems, the ICDs were applicable for both applications.

Along with the systems integration effort was the requirement for verifying the electromagnetic compatibility (EMC) of the computer with other subsystems. This effort acquired greater significance than the requirement to satisfy the military specifications governing EMC, primarily because published specifications were inadequate for the demands of a digital system. The specification limits, for example, were unnecessarily stringent in such areas as immunity to system noise levels on digital signal lines. Conversely, the specifications were not stringent enough in those areas that required the computer to be immune from electrical disturbances and radiation.

Finally, considerable effort was expended as a result of experience in the field. Many problems were extremely difficult to analyze. In general, these problems fell into three categories: operator or procedural errors, hardware design problems, and software difficulties. The computer and displays provided excellent detection of anomalies, but the diagnostic capability was limited. During spacecraft testing, the only diagnostic information available was provided by the display programs for a single snapshot of data followed by a restart. Additional, but very limited diagnostic information was obtained by the analysis of down telemetry data. The down telemetry

was a fixed format of system data transmitted at a rate of 100 words per second and repeated every two seconds. The display data was included in the down telemetry list. The analysis of this data and the knowledge of the computational sequence provided the only information to verify the cause of anomalies. Some transient faults took considerable effort before they were satisfactorily explained. Not until late in the program (Spring, 1970) was the computer modified by adding a restart monitor to provide more information on the cause of errors. Refer to Section 5.2.5 for more detail.

6.2 DEVELOPMENT AND SPECIFICATION OF INTERFACES

The variety and multiplicity of interfaces necessitated the application of certain fundamental objectives to the interface design. These objectives engendered a philosophy basic to the interface electronics and can be summarized as follows:

1. Standardize as much as possible.
2. Use of simple circuit configurations.
3. Development of wide but uniform specification limits.
4. Utilization of low pulse repetition rates and duty cycles.

There are very sound reasons for these objectives, but they did result in an inflexible approach to negotiating the Interface Control Documents (ICDs). The computer interfaces with a large number of spacecraft systems in both the lunar module and the command module, most of these systems were designed by separate groups of contractors who desired custom interfaces to suit their own applications. The merits of the MIT/IL standardization goal were clearly recognized and strongly supported by NASA/MSC. This made standardization possible even though many other APOLLO contracts and subcontractors were affected by the results.

The facts that the LM and CM computers were identical and that the computer electrical interface design was frozen before many of the other subsystems, strained the early ICD negotiations somewhat but did simplify the associated decision process, since the number of choices were limited. Dual use of the computer might have seemed extravagant, since there were interface functions utilized in the lunar module but not in the command module and vice versa. This extravagance, however, was felt to be worthwhile, since it simplified manufacture, production testing, EMC testing and qualification. Later it simplified the negotiation of new interfaces in each application, such as the addition of uplink to the LM and the addition of VHF ranging to the CM. The latter was designed to make use of the radar interfaces of the LM application.

Essentially, the Block II configuration has four types of interfaces: A transformer-coupled input circuit, a transformer- coupled output circuit, a dc-coupled input circuit, and a dc-coupled output circuit. The display unit contained a fifth type of interface, namely relay contacts. (Another interface circuit, a micrologic interface, was used for testing purposes but was not utilized during flight.) The Block I circuits were basically the same as Block II, except that more use was made of relay contact closures in the Block I DSKY.

By keeping the number of different types of interface circuits to a minimum, factory test equipment was simplified somewhat. But more important was the increased reliability afforded by having a high utiliziation of standard circuits. The standard circuits were therefore subjected to more hours of operation thereby producing more significant information on circuit limitations and operational peculiarities. The factory test specifications, while tighter than the ICD specifications, were uniform to facilitate testing. In practice, however, the actual loads for the interface circuits were different from the factory test loads. Verification was accomplished by correlations tests. While an attempt was made to negotiate identical loads on the interfaces, and therefore, specifications, it was extremely difficult in some cases because of individual differences.

The wide tolerances in the interface specifications afforded compatibility with a great many circuits and also reduced the criticality of the interface design. The possibility of utilizing a specific parameter (such as pulse rise time) was made more difficult because of the wide variation allowed in the specification. The objective was to ensure that the interface would not be sensitive to pulse shape and would only be required to distinguish the presence or absence of a pulse. Although this objective was desirable, it could not always be achieved because a few subsystems required more critical timing, and therefore made use of the rise and fall times of the pulses.

Another basic objective was to keep the data rates low. This created problems in ICD negotiations but for the most part was successful. Since the AGC processes pulses by cycle stealing, with many inputs being processed at the same time, saturation of the computer was possible. A limit of 3.2 KHz for I/O counters was established. The low rate had a positive effect by constraining designers who tend to specify higher data rates than necessary. In addition, the low data rate tended to reduce power consumption.

TABLE 6-I

AGC INTERFACE CIRCUITS

BLOCK II	BLOCK I
55 X Circuits (3) *	38 X Circuits
33 Y Circuits	30 Y Circuits
68 C Circuits (7) *	28 T Circuits
74 D Circuits (3) *	26 D Circuits
3 A Circuits	12 K Circuits
11 DSKY relay outputs	47 DSKY relay outputs

( )* Number of circuits not available at the interface because of pin liminations.

TABLE 6-II

TEST CONNECTORS (MICROLOGIC INTERFACES)

BLOCK II	BLOCK I
81 Outputs	76 Outputs
34 Inputs	25 Inputs

6.2.1 Circuit Types

In Block II there are four types of multiple usage interface circuits and also a special analog-to-digital (A/D) converter in the computer with a limited number of relay closures available from the display unit. The special A/D converter ("A" circuit) was used for the proportional hand controller inputs in the lunar module. The four multiple-usage types were a transformer-coupled input "Y", a transformer-coupled output "X", a dc-coupled input "D" and a dc-coupled output "C". The "D" and "C" circuits are designed for 28 volt dc signals, and the "Y" and "X" transformer-coupled circuits are designed for pulse data. Table 6-I shows the number of circuits of each type in Block II and Block I. The main differences between Block I and Block II were: one, the reduction in the number of relay outputs in the display unit to improve reliability, and two, the increase in the overall number of circuits in the AGC due to increased utilization of the computer such as the digital autopilot, etc. The design description in Section 3.7 treats the various interface functions in more detail.

There was an additional interface (test connector interface mentioned briefly above) that was designed for use with the factory test equipment. This interface was designed with the philosophy of putting into factory test or ground support equipment all circuitry not required during the flight. This would reduce the amount of circuitry in the computer. The result was a logic level interface with the ground support equipment that had a minimum drive and a minimum noise rejection capabiliity. This interface design was a source of many transient failures due to noise in both the operational and ground checkout configurations. The noise problems in the operational configuration were solved by providing a connector cover which shielded and shorted the interface signals to ground. The ground checkout problems were minimized by using extreme care in shielding and grounding. Table 6-II lists the number of these circuits.

In addition to the interfaces listed above, whenever the computer was required to supply a 28 volt output, a resistor was inserted in series to eliminate the possibility that an external load might short and overload the computer circuit breaker or internal wiring. There were 24 of these circuits designated "R" circuits. Their function was to supply power to functional elements like switches. The signal return was an input to a "D" circuit. The only 28 volt circuit, not employing this series resistor, was the power supply for the DSKY.

Since the same computer was used in both the lunar module and command module, each interface was defined by a number as well as by name to avoid confusion.

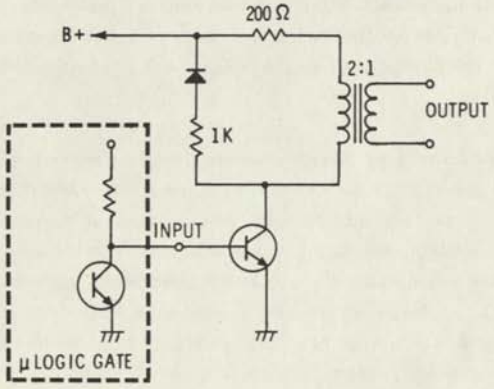


Fig. 6-1 Transformer Coupled Output Circuit

The numbering system was advantageous in that the electrical design could be identified and specified before the functional design. Use of the numbering system permitted release of drawings and discussion of signals that had not yet been named or defined functionally.

6.2.1.1 Transformer-Coupled Interface Circuits

Transformer coupling was used to provide dc isolation in those interfaces particularly susceptible to noise and EMI. Specifically, those interfaces where fast response pulse signals were required. These signals tended to be more susceptible to noise because their amplitudes were lower and their speed of response faster than the dc-circuits. The transformer output circuit was designed to have signal characteristics compatible with the transformer input circuit.

6.2.1.1.1 Transformer-Coupled Output Circuit (X Circuit)

The output circuit (X circuit) was designed to utilize a minimum of parts and to be driven by the output of a logic gate. The circuit is shown in Figure 6-1. The transistor and transformer are the minimum number of parts for circuit operation. The other parts are required for circuit protection when operating under a variety of conditions. The 220 ohm series resistor is a compromise between a low output impedance, approximately 50 ohms, and protection against a short circuit on the output. The resistor-diode circuit, in the primary, was added for circuit protection when operating with no load. The actual value of the resistor was a compromise between the collector voltage generated by the inductive kick when the transistor turns off and a value to limit the current through the transformer. The pulse waveform will vary depending upon pulse length, repetition rate and load. As a result the various applications were simulated to verify the electrical specifications before the ICDs were signed off.

The components used in the design were those selected as standard throughout the computer. That is, the input and output transformers are the same, the diode is the standard switching diode, and the transistor is the standard NPN. The transformer was specially designed to take pulse widths up to seven microseconds at 10 volts and was specified both by parameter values and pulse output characteristics at a 100 KHz repetition rate in order to insure uniformity of the production component.

The use of these parts and the desire for a standard pulse output circuit resulted in a design that was not optimized for every application. This is a rather typical example of a trade-off between electrical performance and the requirements of

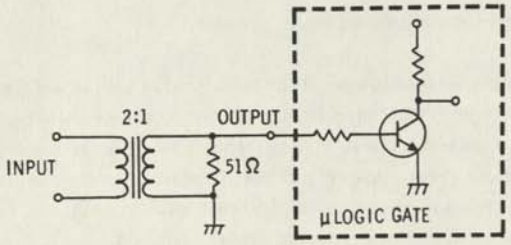


Fig. 6-2 Transformer Coupled Input Circuit

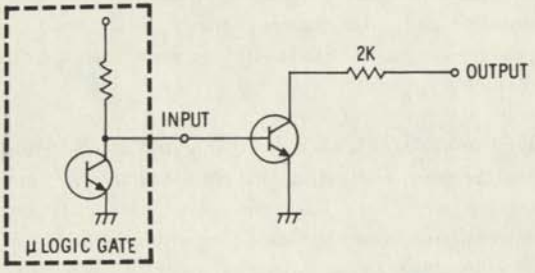


Fig. 6-3 DC Output Circuit

standardization. For some of the high frequency outputs, such as the clock synchronizing signal, a better waveform would have resulted with a specially designed transformer. However, if each of the many outputs of the computer were allowed the freedom of individually selected components, the overall design would have been much more complex and many special circuits would have been required, thus limiting the commonality between CM and LM applications.

6.2.1.1.2 Transformer-Coupled Input Circuit (Y Circuit)

The circuit shown in Figure 6-2 was considered the simplest possible circuit for this application; it requires a 50 ohm resistor and transformer. The turns-ratio chosen was a 2:1 step-down so as to increase the noise rejection capability at the input to the logic by a factor of two. The 50 ohms is a compromise between an input impedance of 200 ohms that allows a reasonable signal to be developed and yet minimize the effects due to mismatch of the line impedance. In Block I a resistor of 470 ohms was required in series with the micrologic input to keep the input impedance constant for both on and off condition of the micrologic gate. This resistor was omitted in Block II, since the low power micrologic had a higher input impedance.

6.2.1.2 Direct Coupled Interface Circuits

The dc interface circuits, like the ac coupled circuits, were designed with a minimum number of components but with much higher noise rejection. The signals that they interfaced with are in general noisier, and the response required is slower than those signals used with the ac circuitry. Accordingly, the dc circuits were designed with heavy filtering for noise rejection.

These circuits resulted in better reliability, fewer interface connector pins, fewer wires in the harnesses than the ac circuit and hopefully the low power, large signals, high noise immunity interface would reduce electromagnetic compatibility problems.

6.2.1.2.1 DC Output Circuit (C Circuit)

This circuit is shown in Figure 6-3, and here again there is a minimum of components. The circuit consists of a transistor and a resistor for current limiting. Basically this circuit was designed with characteristics such that it could replace the relay type function, that is a high impedance when not conducting and a low impedance when conducting. The "C" circuit's simplicity is an advantage over a relay, but the absence of isolation in the ground return required special consideration. However, the replacement was desirable. The relays took up more volume, were less reliable, and used more pins on the interface.

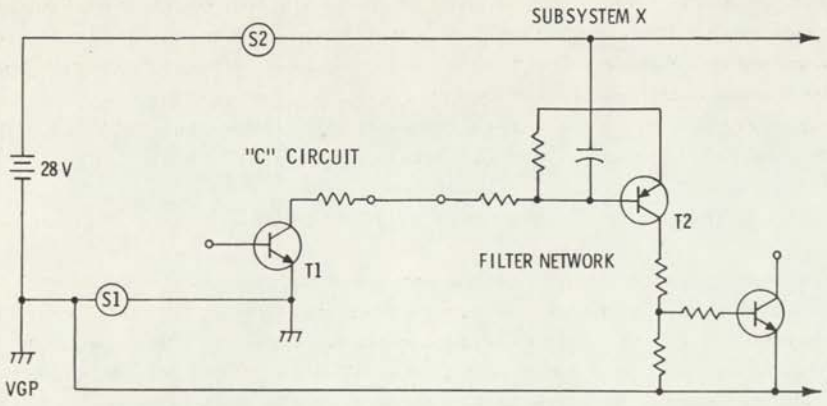


Fig. 6-4 "C" Circuit Application

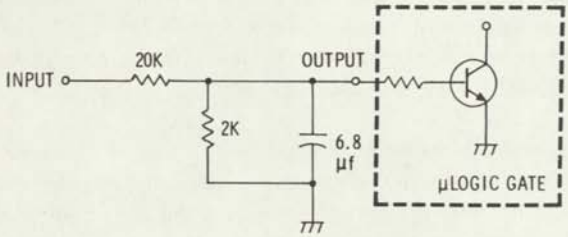


Fig. 6-5 Discrete (D) Input Circuit

The intended use of the "C" circuit is shown in Figure 6-4. This figure indicates that the voltage developed across the interface circuit when nonconducting is 28 volts plus S1 and S2. S1 and S2 are noise sources in the AGC power return and the subsystem power line respectively. The noise could also be induced in the loop composed of "C" circuit, S1 ground, the battery, the S2 path and the filter network. EMI considerations require the signal return to be included in the shield with the signal conductor. However, the S2 loop necessarily contains other currents, such as the transistor T2 collector current. If the emitter return of T1 was included inside of the shield, one would not only have the base current of T1 included, but due to the long emitter base path noise would be injected into the base of T1 and amplified. Thus the interface has its own problems and cannot help but violate a literal interpretation of the EMI ground rules.

The "C" circuit puts the burden of filtering on the receiving end. In principle the interface is noisy, since S1 is an uncontrolled noise that would add to the signal when T1 is conducting, and when T1 is off, voltage induced between the signal line and power line is a source of noise. The circuits, shown in Figure 6-4, with adequate time constants in the filter network would accomplish the required noise rejection. To insure that subsystems were designed with adequate filtering, the "C" circuit was specified as a very noisy source.

For the circuit application shown in Figure 6-4, the noise rejection specifications were:

T1 on: +50 volts, if noise pulse does not exceed 0.5 milliseconds width at maximum repetition rate of 10 pps.

T1 off: -50 volts, if noise pulse does not exceed 1 millisecond width maximum repetition rate 10 pps.

6.2.1.2.2 DC Input Circuit (D Circuit)

This circuit is shown in Figure 6-5, and the number of components again are minimized. The resistor divider network is necessary to convert the 28 volt input to a level compatible with micrologic circuitry. The actual values of resistors and capacitor were a compromise between many requirements. Primarily the circuit had to be noise tolerant and drive at least two logic gates. A noise specification similar to that of the "C" circuit also applied for noise rejection of the receiving network.

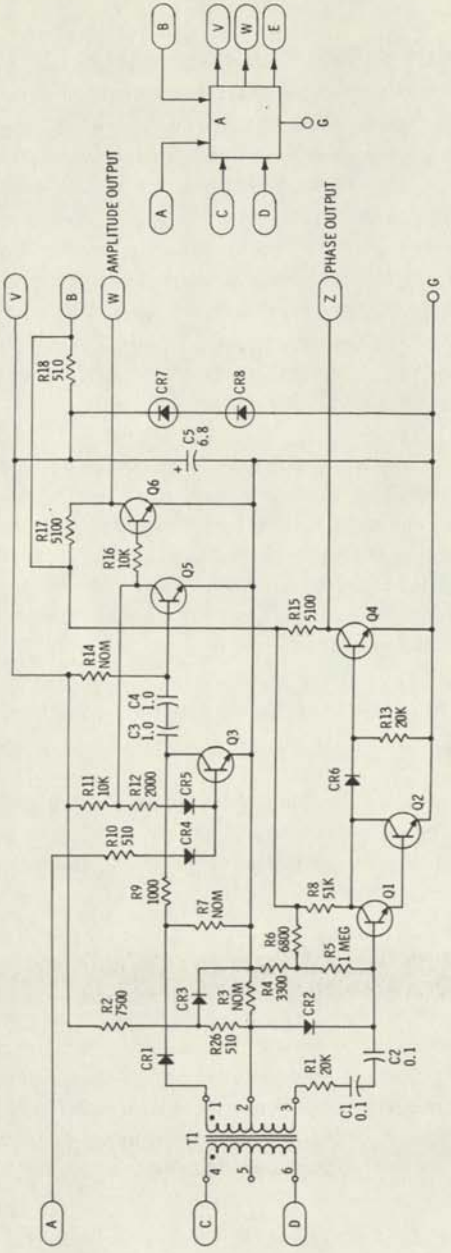


Fig. 6-6 "A" Circuit

6.2.1.3 Special Circuits

Although the circuits described above covered most of the AGC interfaces, there are some situations that were unique. These are described below.

6.2.1.3.1 Hand Controller Interface

In Block II a special interface circuit was designed for the hand controller in the LM. This circuit was called the "A" circuit (Figure 6-6), and three were required; one each for pitch, roll, and yaw. These provide an analog to digital conversion of the three degrees of freedom of the hand controller. The conversion is accomplished by generating a pulse length proportional to the peak amplitude of the input. The circuit in Figure 6-6 can be divided into two parts. The upper half of the transformer secondary is used to charge capacitors C3 and C4 to the peak of the input voltage, and on the application of a pulse to pin A, to hold Q5 off for a length of time proportional to the voltage across C3 and C4. The bottom half of the transformer secondary is used to detect phase. The 800 cps signal is squared up, and the output at pin P strobed at the appropriate time to indicate the phase.

The input is an 800 cps amplitude modulated signal, and the outputs are: one, a signal on pin W, that disappears for a time proportional to the amplitude of the input signal, and two, a signal on pin E that can be interpreted as indicating the phase of the input. When the amplitude signal disappears, it allows a 3200 pps pulse train to be gated into an AGC counter. The number of counts is proportional to the amplitude of the 800 cps input signal. The amplitude is proportional to the angle through which the hand controller is moved, and the maximum linear deflection corresponds to a count of 32 (decimal), thus dividing the hand controller deflection into approximately 32 equal increments.

The input is utilized by the program setting a bit in an output channel that results in a pulse on pin A starting the gating process. The input can be measured a maximum of 33 times a second, as one has to allow time for the capacitors C3 and C4 to charge up to the input amplitude. The phase signal, pin E, is strobed just before the gating sequence starts. The excitation to the hand controller is derived from an 800 cps supply in the GN&C. This in turn is generated by pulses from the AGC, so that the computer knows what the excitation phase of the hand controller should be and can therefore determine the polarity of the hand controller deflection.

The phase signal, pin E, is a squared up version of the input. One of these signals and an 800 Hz reference signal were brought out to the computer test connector, so

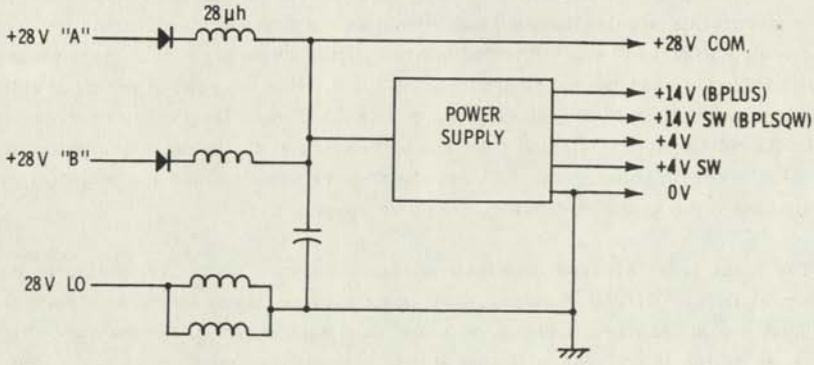


Fig. 6-7 28 Volt DC Input Power

that one could check the phasing from the AGC through the GN&C 800 Hz amplifier and LM hand controller loop. Pin V was brought out, so that one could monitor and record the zener voltage during module testing.

6.2.1.3.2 DC Power Interface

This interface is shown in Figure 6-7. The Command Module has two redundant 28 volt buses. To isolate these buses from one another, series diodes were placed in each high.

In the LM there was only one power bus for the LGC. The external power wiring connected the two diodes in parallel. The diodes provided another function, that is they prevented accidentially applied negative voltage from damaging the computer.

The combination of the chokes and capacitors in the power input lines provided filtering for noise on the power lines whether generated internally and conducted out or vice versa. This design evolved after considerable EMC testing of the Block I computer. These tests included the audio and transient susceptibility tests that are part of the EMC specifications. These also included RF transients induced by electrostatic discharges (spark test) to the computer case and cabling. In Block I the configuration of the filters was similar to that shown in Figure 6-7, but initially the computer signal ground was isolated from the computer case. The results of the transient testing indicated the signal ground to case isolation was detrimental. The distributed capacity between the wiring and the case was quite large and provided return paths for transient currents between the power lines and case. It thus made the computer quite sensitive to spark type transients, even though it was insensitive to the transients between power lines. Many filter configurations were tested but in order to make a significant change in the test results, the computer signal ground and case was connected at multiple points. In the Block II design the case was designed to be the ground return with a minimum of wiring between ground connector pins for redundancy. This change with the filters shown made the computer immune to power line transients much greater than 50 volts and electrostatic discharges to the case and cabling shields.

In Block II mechanical design the power supply modules, the interface modules, system connector and the interconnecting wiring were all located physically in order to maximize the computer's immunity to externally generated interference.

6.2.2 Interface Control Documents (ICDs)

One of the main objectives at MIT/IL was to have a single electrical ICD for the computer in each spacecraft. Since the computer interfaces with many different subsystems, this was a difficult objective. It did require much coordination but was felt to be worth the effort. The program management at NASA/MSC recognized the potential advantages and directed the spacecraft contractors to cooperate in the standardization of documentation and use of the standard interfaces. The problem of negotiating these standard interfaces with all the various subsystems in each of the spacecrafts would have been impossible without the strong backing and support of the NASA management.

Initially it was desired to have functional descriptions included in the ICDs. Because of the lead time required to build the computer and due to the fact that the software would be written somewhat later, the initial ICDs did not contain the functional descriptions. Some of these were added later, and parameters affecting the software were defined in other documents. There were some signals negotiated in the ICDs whose functions were not used by the programmers. Also, a number of signal functions were defined after the computer design was released.

Another objective of the ICDs was to have the actual interface circuit schematics included in the document for information only. This would help minimize some of the ambiguity contained in the specifications, and would give the electromagnetic compatibility test engineers information on how to simulate the various interface circuits for their EMI tests. It also allowed one to recognize that when there were problems on a particular interface which other interfaces might have similar problems. Having the circuits on both sides of the interface permitted engineering verification of the signal characteristics before spacecraft integration tests. In some cases this objective was not realized until too late in the program to be useful. The reason for the information being late was primarily contractual. That is, the spacecraft subcontracting chain was too long to respond very fast.

6.2.3 Summary

The typical interface functions are described in Section 3.7 and in the ICDs for the CM (MH01-01380-216) and the LM (LIS370-10004).

As can be seen, the main objectives of the interface design was simplicity and reliability. Nevertheless, one had to pay tribute to the component manufacturers for making the objectives possible with the quality and flexibility of their products.

A typical example was the NPN transistor that was used in the "X" and "C" circuits. The "X" circuit required a high frequency switching transistor, and the "C" circuit required a high voltage low leakage transistor. These extremes are usually considered incompatible.

It should be noted that the dc interfaces violated the EMI specifications. If simplicity is to be obtained, there is no other choice. With the provision of heavy noise filtering, entailing slow response and low frequency, the intent of the EMI specification can be realized with simple circuits. To meet the EMI specification, the power supply would require some form of dc isolation, so that 0 volts could be tied to structure. The interface circuits require dc isolation, so that the current in the signal ground is in fact the return signal current. The signal and return can then be a twisted pair.

6.3 ELECTROMAGNETIC INTERFERENCE

This section will discuss the electromagnetic interference (EMI) philosophy, the specification problems, the interference problems encountered and the tests run. It also contains a list of milestones that may help define the interrelationship between the computer development schedule and the class of design problems associated with electromagnetic interference.

6.3.1 EMI Philosophy

Electromagnetic interference problems characteristically appear fairly late in the system development cycle, they are masked by a plethora of unrelated symptoms and in digital equipment they result in intermittent failure. In a word EMI problems can be nightmares which have severe impact on a development program. In order to minimize this impact various approaches have been used. One approach has been to conform almost blindly to an EMI specification and control plan and then continually tighten up the limits when a higher confidence level is required. One characteristic of this approach is that no deviation or waivers be granted, and that a hard nosed attitude be presented toward all such requests. The results of this attempt at strict conformance is the development of many design conflicts which impact things like system weight, power and reliability.

When a design engineer is faced with a conflict between EMI requirements and other system constraints, he usually doesn't bother to consult the electromagnetic compatibility (EMC) specialist because he knows the answer will be in strict conformance to the specification. The EMC group tries to control the design through

organization charts, authority statements, and EMC sign-offs. The design engineer proceeds more or less independent of the EMI requirements and presents the EMC group with a completed design that can't be changed because of a schedule or cost impact, or the responsibility is passed to a subcontractor who has a more limited view of the complete system problem. It should be noted that in most cases neither side realizes that they are participating in such a game.

The approach taken by MIT/IL during the computer development was that the EMI problem, like other design and reliability problems, requires engineering judgement and decisions that cannot be delegated or resolved by a specification. The primary objective was to make effective the intent and relevance of the EMI specification. Conformity to the specifications was considered to be weighted equally with other design constraints like weight, power and reliability. In the case of the computer's susceptibility to interference, considerable design effort was expended to make the computer less sensitive to interference then that required by the specification. Special tests were introduced to explore the susceptibility. These test methods and the designs necessary to protect the computer will be described below.

6.3.2 Milestones

The interaction between the dates related to the contractual requirement for electromagnetic interference specification, hardware design release and EMI testing are indicated in the list of milestones.

1.	Sept. '63	MIT negotiates EMI consultant subcontract.
2.	Sept. '63	NASA adds contract addendum requiring GN&C system to meet MIL-I-26600 and EMI-10A electromagnetic interference specifications.
3.	Jan. '64	Design release of Block I (0 series) (see Figure 1-6.)
4.	Jan. '64	EMI consultant on the job.
5.	Spring '64	Studies of the intent and purpose of the EMI specifications begin.
6.	Apr. '64	EMI meeting at KSC including prime contractors.

7.	Mar. '64	Power lines transient tests run on Block 0 and rack mounted configuration.
8.	Sept. '64	Design release Block I (see Figure 1-8).
9.	Oct. '64	EMI tests on interface circuits.
10.	Dec. '64	EMI test plan for AGC completed.
11.	Dec. '64	Transient susceptibility tests run on Block I AGC and GN&C.
12.	Feb.-Mar. '65	EMI problems with System 5 in MIT/IL Systems Test Lab using Block I (0 series) and Block I computers (initiated testing with controlled electrostatic discharges.)
13.	July '65	Design Release Block II (see Figure 1-9).
14.	Aug. '65	Block I spacecraft EMI tests run at CM contractor.
15.	Fall '65	Reviewed MSFC EMI Spec 279.
16.	Feb. '66	Raytheon EMI tests run on Block I.
17.	Mar. '66	MIT EMI specification limits defined.
18.	Apr. '66	Ac Electronics EMI Tests on Block I GN&C system.
19.	May '66	Limited conducted and transient tests run on Block II GN&C system at LM contractor.
20	Spring '67	AC Electronics EMI qualification test on Block II using a CM GN&C system.

6.3.3 Specification Problem

The main problems with the EMI specification were due to either ambiguous statements or test requirements that are not applicable to a system like the GN&C with a digital control computer. The latter set of problems were probably due to the fact that the specification was originally written for a particular system where it was meaningful, but in the attempt to apply it more generally, it lost its relevance. EMI specialists assisted in the interpretation of the specification but establishing the relevance of the tests and reducing the associated requirements to computer design requirements was the most difficult part of the development. That is, the specification provided very little aid in determining good design practice for digital equipment.

6.3.3.1 Susceptibility Testing

The objective of susceptibility testing is to determine whether the equipment under test can operate satisfactorily when subjected to electromagnetic interference somewhat more severe than that expected during normal operating conditions.

6.3.3.1.1 Power Line Transient Susceptibility

The first problem encountered had to do with the interpretation of the 50 volt transient. The purpose of this test is to apply a 50 volt 10 microsecond wide pulse to the power input of the equipment under test. If the equipment has a large capacitative input, how could a 50 volt transient be developed? The capacitor would short it out. What actually happened in such tests was that the transient was developed across the power line lead inductance. The fact that the specification does not control the energy that the transient generator could supply to the test item, and that the power source might be shorting the transient out, posed problems. The AGC had an inductive input, so the transient could be developed quite easily. The GN&C system had a capacitive input. When testing the GN&C, an open circuit amplitude was used to set up the transient generator, where the open circuit was really a 250 microhenry choke in series with the 28 volt power supply. The reason for the "open circuit" interpretation was that the amount of energy in the transient was able to be controlled. The correct interpretation of the test was important, since the GN&C had a voltage quadrupler that could be a problem.

For the computer the specification was interpreted as being applicable to all combinations of power leads and structure. However, this interpretation was not generally accepted. For the computer it was important, since, in operational

environments, transients can be developed in all combinations.

6.3.3.1.2 Conducted Audio Susceptibility

The purpose of the test is to insure that the subsystem can operate, while an audio signal modulates the power bus. The audio modulation is accomplished by inserting a transformer secondary in series with the 28 volt power bus. While the NASA requirements seemed to specify the open circuit output at the transformer secondary, other military EMC specifications called for measuring the output when loaded by the subsystem under test. None of these specifications made allowances for either the load presented by the subsystem or the waveform distortion that resulted from this loading. These differences made the interpretation of the NASA specification more difficult.

Since the GN&C had a capacitive input, and if the injected signal was measured when loaded by the system, the system could dissipate excessive power in the capacitors at the higher (15 kHz) frequency range of the specification. This problem was resolved by taking into account the capacitor dissipation and by reducing the closed circuit input to prevent damage. As a result, the system was not tested at the specification limits.

In addition to the specification requirement there were randomly distributed inquiries as to how long the AGC (and what was worse the GN&C) could run without power or how low and how long the supply voltage could be without degrading the performance appreciably. The computer was designed to go to sleep when a loss of power was detected, then restart when power returned, but the electronics of the Inertial Measurement Unit could not tolerate loss of voltage for extended periods. In the computer, if the voltage went below a level of 20 volts, the computer would shut itself down and save enough information to allow the program to be restarted.

6.3.3.1.3 Radiated Susceptibility

The test methods used to specify the tolerance limits on radiated susceptibility were found inadequate for testing the guidance computer. The computer was completely insensitive to the radiation specified, but until design changes were made, the computer was very sensitive to electrostatic discharges. One can argue whether electrostatic discharges should be present operationally, but APOLLO experience has confirmed that static discharges were quite common in subsystem test laboratories, spacecraft testing and finally during the launch of APOLLO 12.

6.3.3.2 Interference Testing

The purpose of these tests is to establish a limit on the interference generated by the equipment under test. The assumption inherent in these tests is that a subsystem that meets the specification will not disturb other subsystems with which it is associated.

6.3.3.2.1 Power Line Conducted Interference

This test establishes a limit on the interference that may be injected into the power distribution system as the result of the normal operating condition of the subsystem. A problem arose because of conflicting limits between the ICD and the EMC specifications. The reasonableness of the limits, and which specification was the controlling document, had to be resolved. The situation was resolved by making the ICD requirements controlling, together with the qualification procedues to cover the test methods.

The AGC power supplies were pulse width modulated and synchronized at 51.2 kHz. As a result of high frequency and the synchronization of the two power supplies being out of phase, the conducted interference was relatively easy to filter using simple LC filters. All these precautions helped but were not enough to enable the equipment to pass the EMI specification limits. This was one area where a trade-off was made between a reasonable amount of filtering and relaxing the EMI specifications.

Another problem was that in addition to the EMI specifications, one also had to conform to a Power Interface Control Document, which specified a limit in the voltage or current ripple injected into the power distribution without a firm specification on the source impedance of the power system.

6.3.3.2.2 Signal Leads Conducted Interference

This part of the specification was a major problem of interpretation. In fact even the intent of the specification was vague. The specification should be concerned with interference between signals and controlling shield currents. Instead EMC specialists seem to concentrate on the specification of noise in each signal path when no signal is being transmitted. The signal amplitudes and noise tolerance of any one signal path should be under control of the interface control documents, such that the interface circuits are designed to operate under the signal and noise specified. Then the shielding, grounding, and wire routing should be provided to prevent coupling to another signal path.

The problem of interpretation of the specification develops because the specification shows the measurements being made with a current probe encircling a cable coming out of the system under test. The problem here is to determine whether it is encircling the whole bundle (several wires and their returns) or just a single wire. Quite a few EMI specialists have taken the latter interpretation and put current probes around individual signal leads and then tried to impose the EMI specifications on that measurement. For digital signals this would mean any signal present after the pulse had subsided or transmission was over and would be a specification independent of interface circuit characteristics. This situation is ridiculous because if the noise on a signal path were to be the cause of interference, then the signal, when present, would cause much greater interference.

6.3.3.2.3 Radiated Interference

The radiation testing was straightforward. Here the specification limits were not met. The limits were relaxed as a compromise between excessive radiation and additional shielding. The Block I design was much worse than Block II. The steps taken to reduce the susceptibility of the computer to static discharges also reduced the level of radiated interference. In Block II the requirement for moisture proofing, that resulted in metal covers enclosing the subsystems, also reduced the radiation. The covers were designed for low conductive paths between the mating gasket surfaces, but RF gasketing was not added to the design.

6.3.3.3 Grounding and Shielding

The specification provided little or no help in determining the optimum techniques for grounding and shielding. The approach used in the computer developed from past experience as modified by early susceptibility tests, mainly power line transients and electrostatic discharges.

Initially, the AGC had all its signal leads shielded and the shields returned to zero volts, grounded at the signal source only. This was done so that any shield currents, resulting from unbalanced signal lines, would return to the source through the shield rather than through the structure.

When the sensitivity of the AGC to transients between zero volts and structure and the susceptibility of the AGC to static discharges was determined, the above philosophy changed to that of connecting the AGC zero volts to structure (case) and to the multiple point grounding of all the input and output shields. This resulted in violating the single point ground requirements (making two vehicle ground points).

The logical remedy was a transformer coupled dc-dc power supply to isolate the AGC zero volts from the power bus, but a new power supply design was too late to phase in. The field experience has shown that the advantage of making the AGC insensitive to zero volt-to-structure transients and static discharges more than compensated for the violation of the single point ground specification.

6.3.4 EMI Specification Limits

This section contains the specification limits in the GN&C qualification and also the additional tests that were used as development tools in order to evaluate and reduce the susceptibility of the computer to transient conditions. See Table 6-V. These tests were run on the computer subsystem while operating the GN&C.

The qualification test limits may appear to be very loose, when compared to EMI-10A, but they are felt to be more realistic than the EMI-10A limits for this type of equipment in this kind of application. The actual EMI-10A limits are shown as broken line in the figures. A listing of the figures and tables for the test limits follows together with their titles.

Fig. 6-8 Conducted Interference Test Limits - 30 Hz to 150 KHz

A. Narrowband DC Lines.
B. Narrowband AC Lines.
C. Broadband.

Fig. 6-9 Conducted Interference Test Limits - 150 kHz to 25 MHz

A. Narrowband (Current Probe).
B. Narrowband (Line Stabilization Network).
C. Broadband (Current Probe).
D. Broadband (Line Stabilization Network).

Fig. 6-10 Radiated Interference Test Limits - 15 kHz to 150 KHz

A. Broadband

Fig. 6-11 Radiated Interference Test Limits - 150 kHz to 10 GHz

A. Narrowband.
B. Broadband.

Fig. 6.12 Conducted Audio Susceptibility Test Limits

Table 6-III Conducted Radio Frequency Susceptibility Test Limits

Table 6-IV Radiated Radio Frequency Susceptibility Test Limits.

Table 6-V Transient Susceptibility

A. Conducted.

B. Radiated.

The radiated transient test called out in Table 6-V is designed to simulate interference generated by static discharges. The circuit for the test is shown in Figure 6-13. The spark is calibrated by adjusting the spark gap.

6.3.5 Electromagnetic Interference Problems

Many of the specification problems encountered have already been discussed in Section 6.3.3. The major hardware problem was the sensitivity of the AGC to static discharges and power line transients developed between the power lines and case. These problems could have been disregarded, since there were no requirements to operate in the presence of these interferences. The steps taken to remedy the problems, connecting the computer signal ground to structure and using the connector shells to provide a short ground path for the cable shields, also reduced the self generated internal noise, and reciprocally reduced the RF radiation. When these changes were incorporated into the Block II computer, it was very tolerant to power transients and spark discharges. Another class of problems resulted from the interface design that was used between the computer and GSE. See Section 6.2.1. In the test configuration with the GSE connected, the computer was more sensitive to interference than in the flight configuration. This sensitivity was the result of the types of circuits used inside the computer and the interfacing hardware in the GSE. In the operational mode the computer has a special module that plugs into the test connector. See Section 6.4.2.

6.3.6 EMI Tests

There were a great many tests run on the computer both in a subsystem configuration and with the guidance and navigation system. These are summarized below.

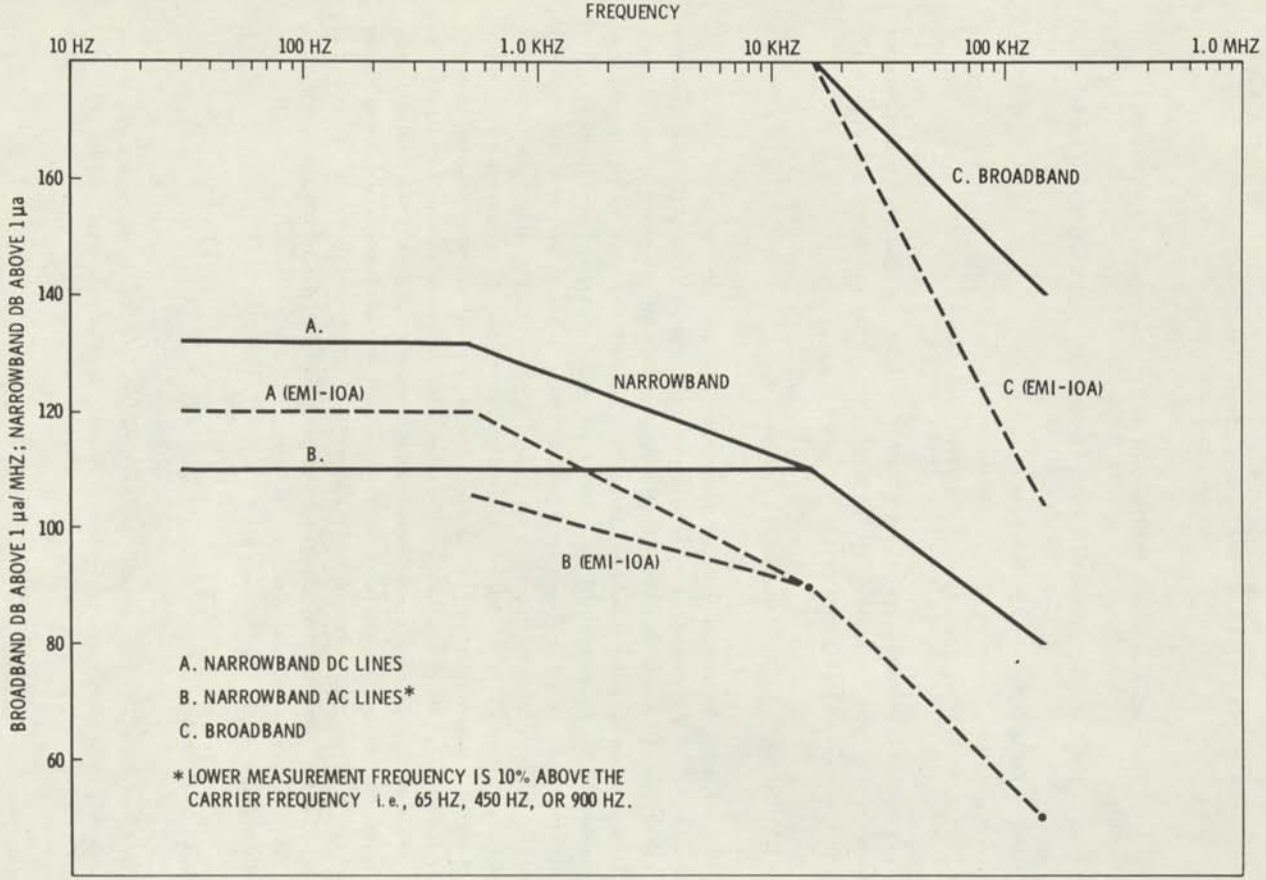


Fig. 6-8 Power Line Conducted Interference Test Limits Using Current Probe

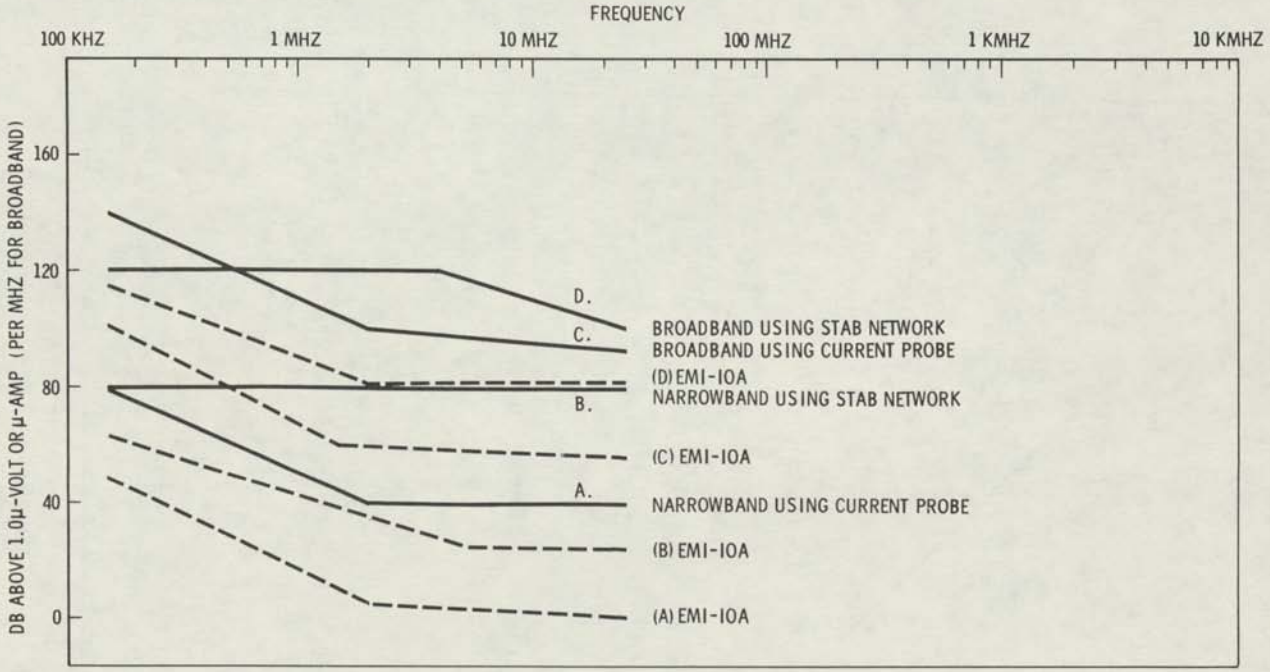


Fig. 6-9 Conducted Interference Test Limits

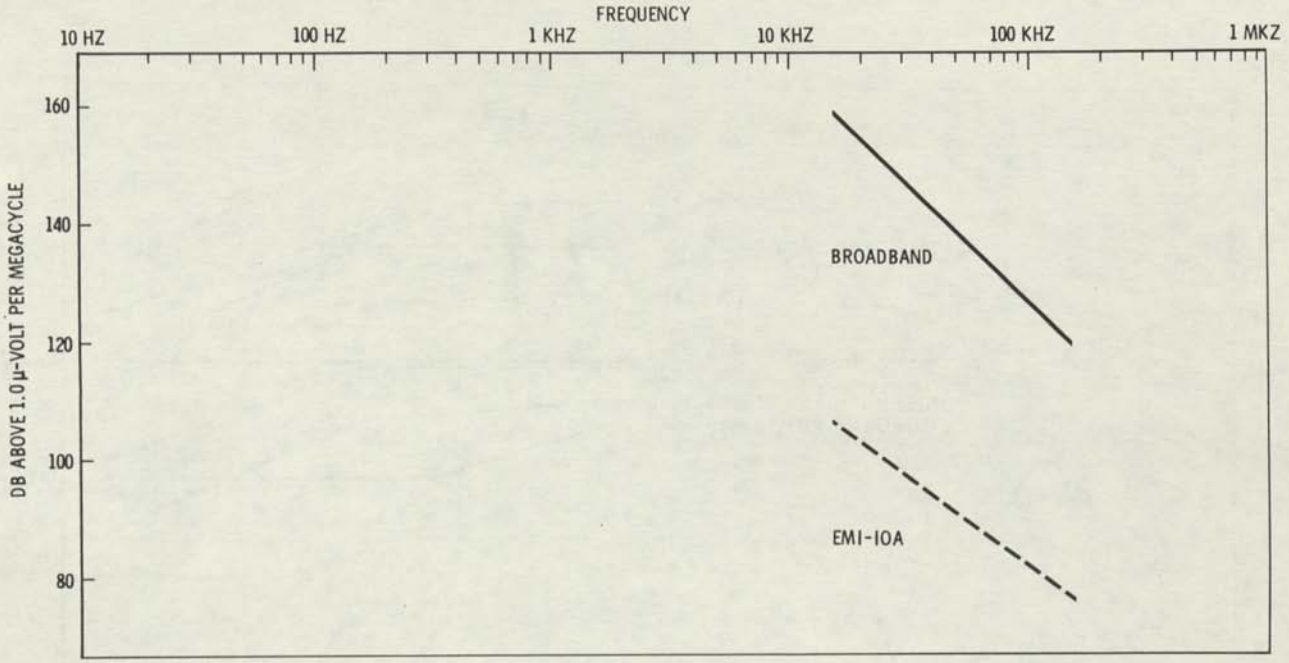


Fig. 6-10 Radiated Interference Test Limits - 15 KHz to 150 KHz

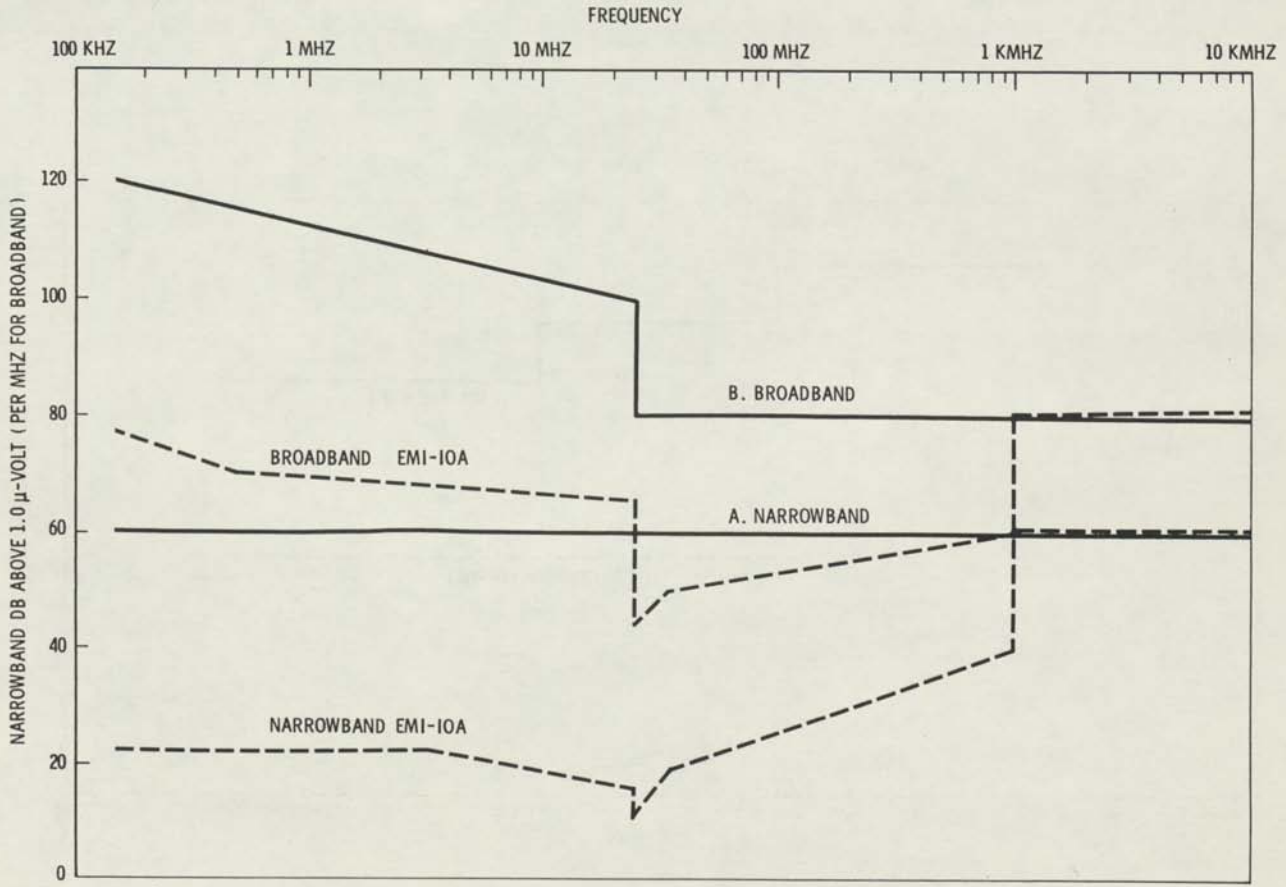


Fig. 6-11 Radiated Interference Test Limits - 150 KHz to 10 KMHz

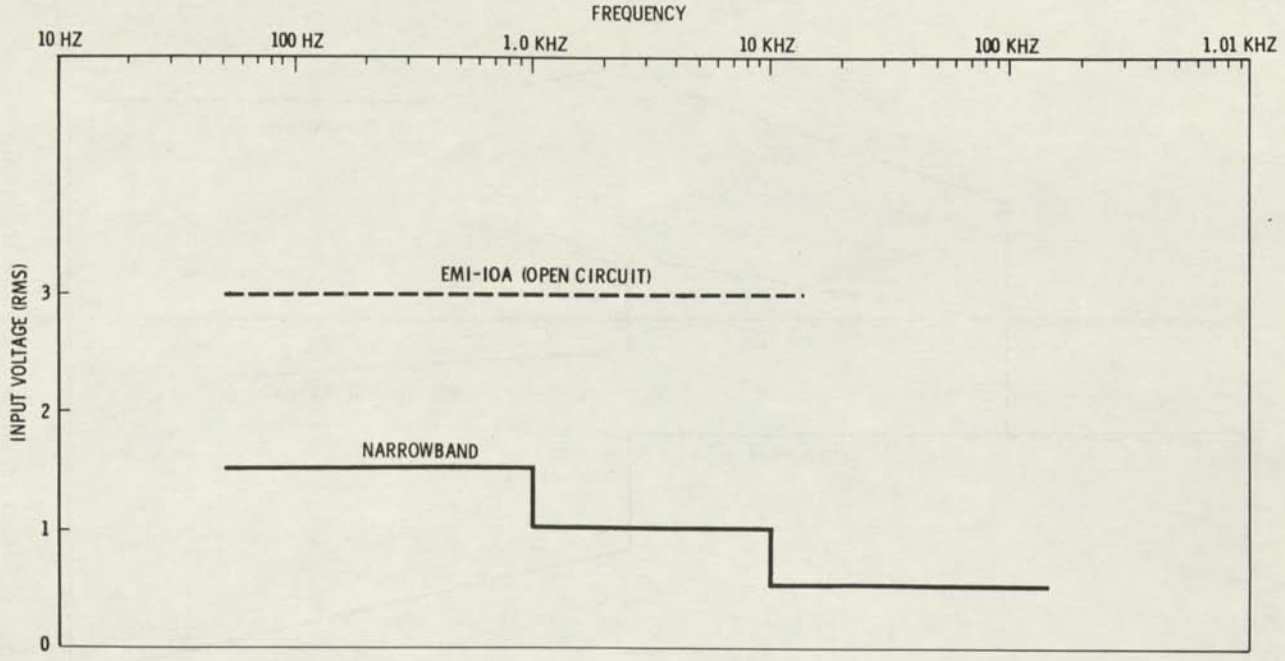


Fig. 6-12 Conducted Audio Susceptibility Test Limits

TABLE 6-III

CONDUCTED RF SUSCEPTIBILITY TEST LIMITS (LSN)

INPUT VOLTAGE*	FREQUENCY RANGE HZ	MODULATION
0.5 VRMS	150 kHz - 1 GHz	30%, 400 Hz
0.5 VRMS	1 GHz - 10 GHz	3 μsec, 400 Hz Rep Rate

TABLE 6-IV

RADIATED RF SUSCEPTIBILITY TEST LIMITS

INPUT VOLTAGE*	FREQUENCY RANGE HZ	MODULATION
0.5 VRMS	15 kHz - 1 GHz	30%, 400 Hz
0.5 VRMS	1 kMHz - GHz	3 μsec, 400 Hz Rep Rate

*0.5 VRMS level is that measured into a 50 ohm load or that read on the test equipment meter.

TABLE 6-V

TRANSIENT TESTING

A. CONDUCTED TRANSIENTS

AMPLITUDE*	POINTS OF APPLICATION**
± 50 Volts	+ 28 Hi, + 28 Lo
± 50 Volts	+ 28 Lo, Case
± 50 Volts	+ 28 Hi, Case

*Loaded with power source only (Use of 200 μh isolation choke is permissible).

**As close to GN&C as possible.

B. RADIATED TRANSIENTS

SPARK GENERATOR	POINTS OF APPLICATION
10 kV Spark	GN&C Structure and Cabling (Various Points)

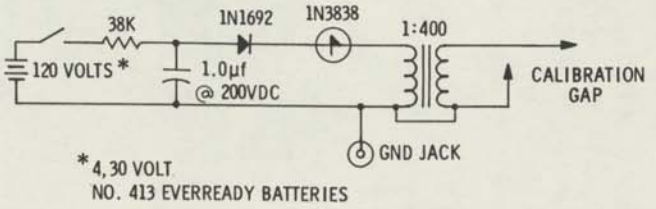


Fig. 6-13 Spark Generator for Radiated Transient Test

The initial tests (Spring '64) were primarily concerned with power line transient and audio susceptibility. As a result of these tests a choke input filter on the power lines was recommended because the computer had to ignore the 50 volt 10 microsecond transient required by the specification, but had to detect a power drop out and put itself to sleep without losing information. These tests also detected the susceptibility of the computer to transients between power lines and case. The Block I configuration introduced the choke input filters on the power lines.

In the Fall of '64 some preliminary GN&C system tests were made, and special tests were run on the computer input and output circuits (C or D circuits) to determine if there were any EMI problems with the single wire I/O configuration. The purpose of these tests were to validate these I/O circuits and the proposed Block II configuration with signal ground and case ground common.

In the Spring of '65 the first occurrence of the static discharge problem was identified on a Block I configuration. Test equipment, which would provide a controlled static discharge, was developed, and a series of tests were run in order to study the problem. This problem was resolved with the multiple point grounding of all the signal shields and improved grounding of the AGC zero volts to structure.

During 1965-66 an EMI committee made up of design engineers representing guidance and navigation and the EMI consultant met and reviewed ICDs, EMI specifications, the Raytheon test plan, and the AC test plans for Block I and Block II. In addition, meetings were held with North American, NASA, Grumman, and AC Electronics. One of the major problems for this committee was specifying the load impedances for the various interface signals. Realizing the difficulty of interpreting specifications, MIT/IL had tried to get the computer interface circuits put into the electrical ICD (for reference only). However, this was only realized fairly late in the program. Part of the problem was that there were several layers of subcontracting, and that the prime contractor did not have this information himself. This not only would have facilitated the simulation of the interface loads for the various EMI tests, but in addition, if a particular circuit was causing a problem, one could make this information available to the other contractors.

Raytheon ran EMI tests on a Block I computer in Feb. of '66, and the results indicated that the AGC failed some specification limits of MIL-I-6181 D.

System EMI tests were run by AC on a Block I GN&C system[1] in the Spring of '66. This system passed using the revised specification limits, although it would not have passed the EMI 10A specification limits.

Preliminary tests were then run (May '66) on a Block II computer and GN&C system at Grumman in the GN&C Test Lab. These were primarily susceptibility tests that the GN&C system and the AGC passed. There were some anomalies noted, but further testing indicated that the problems were procedural or non-repeatable.

In the Spring of '67, the Block II qualification EMI tests were run on GN&C system (CM configuration)[2] to the revised specification limits.

Out-of-spec conditions were critiqued and for the most part attributed to ground equipment problems. Additional tests[3] were run on a Block II system (LM configuration) at MIT in order to answer questions raised by the CM system tests.

6.3.7 Conclusions and Recommendations

There is a definite need for a realistic set of EMI specifications. Since this will never be realized, the only solution is for the EMI and design groups to work together, and where there are incompatibilities an engineering compromise must be made.

Interface control documents, containing the circuits actually under consideration, should be mandatory. The impact of design decisions on both factory test and system compatibility checkout should be kept in mind.

One will need to have some specification limits to work to, but these limits will have to vary as a function of power level involved as well as type of signal. This has already been done to some extent in the present specification. The principal control that the EMC groups should have is to help decide on the reasonableness of

---

1. Block I - 100 series electromagnetic interference and susceptibility qualification test final report TR-1503-1, 30 Sept. 1966, AC Electronics.

2. Block II command module electromagnetic interference and susceptibility qualification test final report TR-1512-8, 6 Nov. 1967, AC Electronics.

3. APOLLO GN&C System Test Group Memo #1044, EMI test results by Richard Sheridan, 31 July 1967.

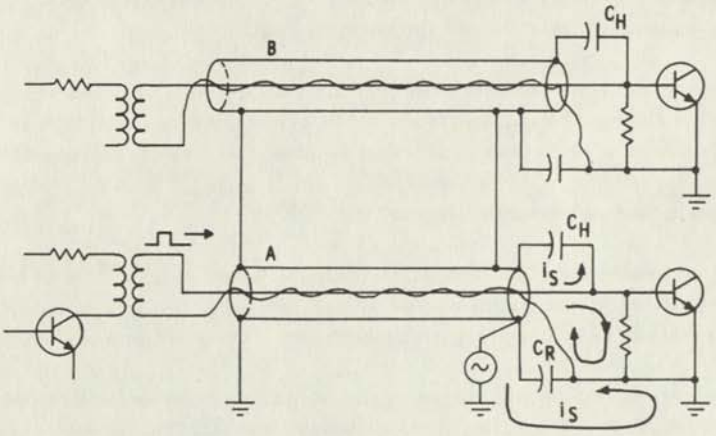


Fig. 6-14 Induced Signals from Unbalance Lines

the weight and reliability factors vs. the EMC. In this area there will have to be more interchange.

The use of transformers and relays to isolate one subsystem from another is reasonable, (in fact MIT itself pushed for this in the early part of this program), but this does have its penalities. Transformers are fairly large, and because one always wants the smallest one possible, they can have reliability problems. Moreover they require two pins on the connectors and two wires in the harness. Also if one does not use transformers on the receiver as well as on the source, one could have unbalanced lines and shield currents that could cause cross coupling (see Figure 6-14). A signal on line A will induce a current in the shield ground, since the capacity C_H of the signal high to the shield is not shorted out, while the capacitance of the return to the shield C_R is shorted. The shield current (i_S), because of excessive shield length, could develop a signal in the shield lead that could cause a signal to be fed into signal B because of a similar unbalanced condition. The situation is worse if one loses the shield ground at either end. Similarly the penalties, when using relays, are size, reliability and EMI problems. A relay interface can use up to 2 or 3 pins in the connector and wires in the cable. Thus one can see the basic conflict between isolating subsystems on the one hand and on the other hand reducing weight, number of signal wires, size of connectors and increasing reliability.

The C and D interface circuits used in the AGC were essentially transistor input and output circuits using a single pin with the power bus furnishing the return. They were designed to require little power, use few components and be noise tolerant. Return lines were available if the receiver or source did want to stay isolated or comply with the requirement to have the return conductor in the same shield as the hot (single) conductor. The absence of ground isolation in these interfaces did not result in an interference or system integration problem.

6.4 FIELD EXPERIENCE

The system integration problems, that were experienced during GN&C and spacecraft checkout, were the most troublesome during the Block I development. As operator experience developed, and as the software and hardware anomalies were eliminated, checkout ran quite smoothly. Since transient or non-repeating type anomalies were the most common, it was extremely difficult to analyze the symptoms and satisfactorily explain the anomaly. Although there were many failures, and all had to be explained, there were only a few that were indications of design faults or software bugs. In general, many of the faults in Block I were the result of electrical transients of

many types. Power-line transients and transient behavior of subsystems during power up and power down were the most common. The interference on signal lines, induced by operation of various switch contacts, were the result of marginal shielding and grounding. In some cases these transient signals were due to coupling within the computer between signal interface and other logic signals. All of these electrical interference problems indicated that the early computers and interface cabling were more sensitive to interference than desirable, even though the system would pass the standard EMI susceptibility specifications. A series of design changes, related to shielding and grounding, eliminated electrical interference problems except those induced by temporary power failures that would cause a V-fail and a restart back to normal computer operation.

6.4.1 Uplink Problem - APOLLO 6 Mission

There was one interference type problem that occurred during the APOLLO 6 mission. The AGC generated frequent uplink alarms both during and in the absence of ground initiated uplink data. During the phase of the mission that required updating of some of the contents of E-Memory, the computer would reject or interrupt the transmissions and signal uplink alarms. The interference conditions made the process of loading data into the computer very difficult.

The alarms were determined to be the result of noise on the uplink interface wiring that the computer would interpret as signal, since the noise amplitude was equal to or greater than signal. The computer circuits would reject pulses under one volt in amplitude. The computer logic and software was designed to test uplink data (normally of pulses with an amplitude of more than 4 volts) for errors in order to protect the AGC operations against erroneous inputs. An error in the data caused by a noise pulse greater than 1 volt would cause the computer to reject the transmission and signal an uplink alarm.

The uplink interface between the radio link and the ground support equipment for prelaunch checkout consists of two signal lines ("1" and "0") transformer coupled that feeds information into one of the E-Memory input counters by performing a memory cycle steal. When a pulse is transmitted on the "0" line, the contents of the uplink register are shifted to the left one place, and a "0" is introduced in the least significant bit. When a pulse is transmitted on the "1" line, the contents of the register are shifted to the left one place, and a "1" is added in the least significant bit. Thus the data word is shifted serially into the uplink register. The format of the data word requires the first bit of the 16 bit data word to be a "1". An uplink interrupt is generated in the computer when a one is shifted from the 15th to 16th

bit position of the register. The interrupt software transfers the data to storage and then clears the register to complete normal transmission of one word.

Two methods were used to protect the AGC from erroneous transmission. The first is logic that tests for input pulses coming too fast, and the second is a coded data signal. The data format has the same bit configuration as the five bit DSKY key codes that allows processing by the same software as that used to process DSKY entries. Each word (15-bit transmission) represents one 5 bit key code (K). The 15 bits contain the code $K\bar{K}K$. The software decodes the transmission by testing for the presence of a key code (K) in the pattern $K\bar{K}K$. In the presence of noise the AGC would reject the data and indicate an uplink alarm.

The occurrence of noise during the mission of APOLLO 6 initiated an intensive investigation that not only located the source of the noise in the spacecraft but also the sensitivity of the routing and shielding of the spacecraft cabling used on this interface. Figure 6-15 is a simplified diagram of the interface and cabling;the umbilical input lines, used during prelaunch checkout, were determined to be the lines that were susceptible to the interfering noise. After launch these lines remained connected to the umbilical and also passed through several connectors within the spacecraft. The radio telemetry/GSE control, shown in Figure 6-15 was added to eliminate the noise pickup after launch. The investigation also determined that many of the checkout problems between the ground support equipment and AGC were not operator errors but were very likely the result of the noise at the interface.

This problem had an interesting side issue that involved the history of the development of the uplink and the coded data. Early in the program, when MIT proposed and designed the interface to use coded words in place of uncoded words, there was a great deal of opposition to this design approach, since the efficiency of transmission was lowered, and there was already extensive coding in the RF link. The argument was that the probability of a bad data transmission being sent to the computer was so small that it would be negligible. In addition erroneous data can be detected, since the computer sends every uplink word down the downlink for verification on the ground. In the light of the noise problem, the lack of the coded data would have made it almost impossible to get a reliable load into the computer. In a case like APOLLO 6 the noise could have caused a catastrophic failure in the AGC, since there would have been no way to prevent the noise from entering the computer.

6.4.2 Test Connector Interface Problem

The problem associated with the test connector was first detected during the early Block I spacecraft integration tests and verified during the EMI studies. The test

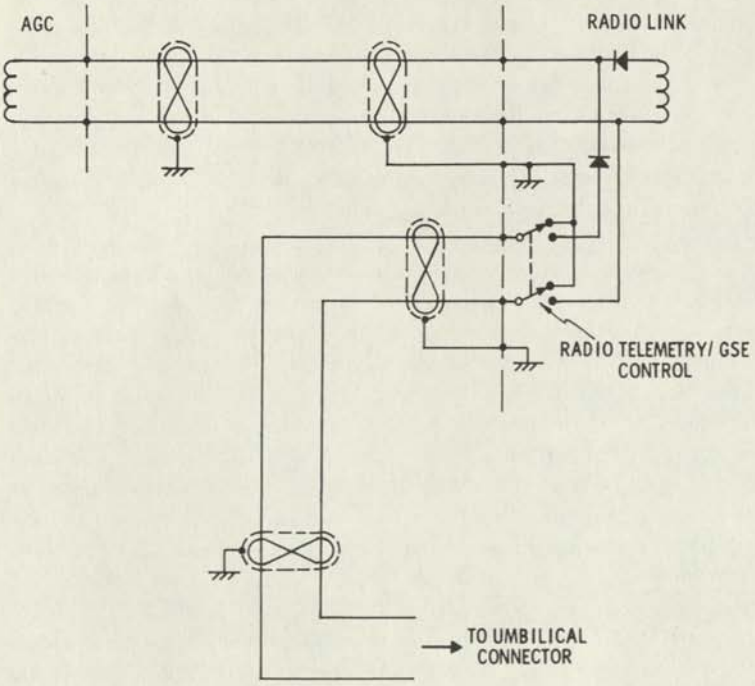


Fig. 6-15 Uplink Interface Circuits

connector interface was shown to be susceptible to electrical interference. Since this interface was designed with a philosophy of minimum fly away hardware, computer input logic signals were carried directly to the test connector. In the spacecraft configuration (the only configuration in which the interface was not used) these logic input wires were unterminated and provided coupling internal to the computer between high level system mode interface signals and computer logic. A special test connector cover was designed for both Block I and Block II computers which grounded these test signal inputs and prevented the coupling. In the mechanical layout of the Block I computer the test connector interface and two of the interface modules were adjacent, thus permitting coupling between the wiring. In the Block II design the interface modules and wiring were located to provide the maximum isolation between interface and logic signal wiring.

6.4.3 TC Trap Problem

A problem, characterized by a TC Trap alarm during Block I spacecraft testing, is typical of the type that is extremely difficult to analyze. When the actual cause of the alarm was determined, it was not the result of an integration problem. The initial symptoms had indicated the problem was the result of noise. A previous problem due to signal coupling between IMU moding signal inputs to the AGC and unused test connector inputs in the computer had just been solved by installing a protective shield and grounding cover on the computer test connector. During subsequent spacecraft testing a TC Trap alarm occurred. There had been no previous known instance of this alarm during this test sequence. The obvious deduction was that the test connector "fix" was causing the problem. Further testing seemed to confirm this deduction, since the TC Trap occurred with a low probability but was never detected when the cover was removed.

A "witch hunt" committee was formed to investigate the problem. The committee did an extensive study to determine the source of noise and the area within the TC Trap alarm circuitry that was sensitive to noise. The study could not identify any source either internal or external to the computer, since there were no moding operations for several seconds before the occurrence of the TC Trap. The study did provide a clue, since it determined that the TC Trap occurred within 20 milliseconds after the computer executed a program called pseudo-verb 36. Analysis of the software provided no additional clues. In fact, it was erroneously concluded that there was no software bug. In order to gain more information for the committee to use, a simulation of the hardware and software configuration was set up in the System Test Lab at MIT. The simulation did result in occasional TC Traps both with and without the test connector cover, thus eliminating the cover as a factor.

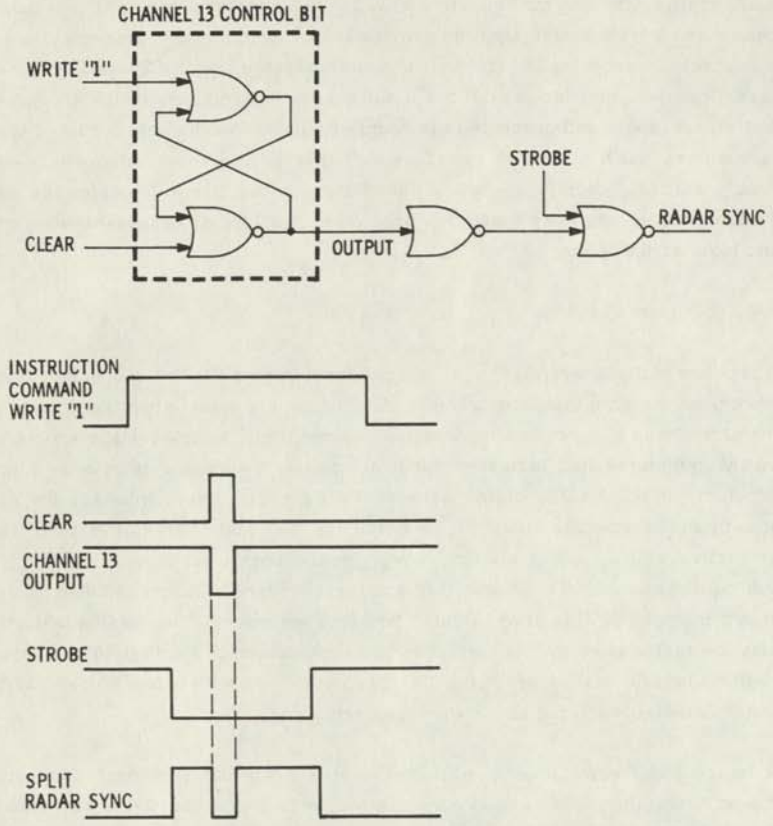


Fig. 6-16 Radar Interface Anomaly

Analysis of the conditions in the software with the aid of the AGC monitor (see Section 5.3.1.1), revealed that the alarm came about due to executive activity coincident with pseudo-verb 36. The problem was the K-start tape (test programs used for spacecraft testing which were loaded into E-Memory) for pseudo-verb 36 improperly used the resident subroutines in the E-Memory. The K-start tape had not been adequately verified in operation with the E-Memory programs.

6.4.4 Radar Problem

The radar problem was identified during testing of LM-5 (APOLLO 11 mission launched July 16) in April and May of 1969. Previous to this time occasional errors in reading radar data were written off as not being mission critical. Therefore the problem was not investigated. The failure in LM-5 repeated several times and excited sufficient concern to stimulate an investigation.

The results of the investigation determined that there was an AGC hardware-software interaction resulting in a split radar sync pulse output. The radar would interpret the split pulse as two data sync bits and as a result would send two data bits to the computer during one bit time. Since the input counters of the computer depend upon cycle stealing, the computer would not read two data pulses that came within a period of less than 12 microsecond. The computer would ignore the second data bit. The result was that one bit of data out of the 15 would be lost, and the remaining lower order bits shifted left (toward the more significant end of the word) by one bit. This analysis explained the symptoms experienced during LM-5 testing.

The split sync pulse was generated when the software wrote into channel 13. See Figure 6-16. The write operation clears and restores the output bits of channel 13 that control the radar gates. The clear and restore action takes about 250 nanoseconds. The timing of the action in the channel due to the performance of that instruction is partially synchronized with the timing strobe for the sync pulse output. The fact that there is some synchronization between the twelve time pulses in a MCT and the strobe pulses provides a beating effect that is coupled with a more random timing of the program steps executing the instruction. The result is a fairly low probability of getting a split pulse. There are many conditions determined by the timing at start-up where the probability of a split pulse approaches zero. This also says that if conditions exist where one split pulse is generated, then the probability of getting additional split pulses is much higher than it would be if all the variables were random.

The remedy for this problem was a change to the software that would prevent the software from using the output channel during the process of reading radar data.

There was a possibility of hardware design change which was rejected, even though it was a rather simple tray wiring change. The hardware design change would require recycling the computers to the manufacturing facility which was not considered expeditious.

The AGC logic design was such that it made the interface susceptible to this transient. If the control for the radar interface had been derived from the other output of the flip-flop in Channel 13, there would not have been a problem, since there is no transient condition on that side of the flip-flop.

6.4.5 Miscellaneous

There was a class of integration problems that resulted from the lack of understanding about how the computer and other subsystem interfaces operated during the power-up sequences. For example:

1. When the uplink equipment was turned on or in some cases when turned off, the equipment would emit one or more pulses. These pulses would remain in the AGC register and would cause the first data transmission to be in error, unless the register was cleared before transmission.

2. When the computer was turned on, the computer would indicate a warning alarm for as long as 20 seconds and would trigger the spacecraft master caution and warning.

3. When the computer was switched between standby and operate, a power transient internal to the computer would modulate the clock sync signals to the spacecraft. Sometimes the modulation would cause the down telemetry to drop out of sync for approximately one second.

These problems were relatively minor in terms of corrective action required but were troublesome to analyze. The corrective action taken was to modify the operating procedures and update the ICD to identify the signal behavior during the transient conditions.

Ironically one of the most troublesome integration problems was the mechanical interface with the two spacecrafts. The configuration of the two cold plates was different so each had a set of unique problems. The LM had two cold rails which provided mechanical support along the edge of the computer. In a vacuum the cover of the computer would deflect slightly due to internal pressure and put additional

stress on the cold rails. The cold rails were designed for minimum weight but had to be strengthened to support this stress, since a change in the computer would have been a weight impact for both CM and LM. The mounting bolts, which were provided with the computer, were being changed almost constantly in order to track the various configurations and account for the mistakes in the Interface Control Documents and bolt specifications. These problems may appear to reflect upon the abilities of the mechanical designers, but that was not the case. They point out the difficulty in documenting and communicating a design when working under the many constraints, including weight, and attempting to coordinate the mechanical requirements between the two spacecraft designs.

Both APOLLO 11 and APOLLO 12 missions had anomalies that are of interest. During the lunar landing phase of APOLLO 11, the computer in the LM signaled an alarm condition several times. These alarms were an indication to the astronauts that the computer was eliminating low priority tasks because it was carrying a computational load in excess of its capacity. The computer was designed and programmed with the capability of performing the high priority tasks first and causing low priority tasks to wait for periods of reduced activity. Several times during the landing the computer had to eliminate low priority tasks and signaled the astronauts of this fact via the alarms.

The over-load condition resulted from the fact that the rendezvous radar was on but was not in the GN&C mode. In this mode the radar angle data was being sent to the GN&C with a phase different than during normal operation. The analog to digital converters in the GN&C system could not lock onto the angle signals. The resulting hunt or dither caused a maximum data rate into the AGC counters that consumed more than 15% of the computational time. The loss of computational time was sufficient to over-load the computer several times during the landing.

The APOLLO 12 anomaly was attributed to lightning striking the vehicle during the first few seconds of launch. The lightning induced temporary power failures in the fuel cell system. The transfer to the backup battery power resulted in a power transient and a condition of V-Fail in the AGC. Subsequent tests on the computer indicated no damage or loss of E-Memory contents during the lightning or power transients.

SECTION 7.0
RELIABILITY

7.1 INTRODUCTION

Basic to the success of the APOLLO guidance computer was the realization that conventional reliability practices were not sufficient to meet the reliability requirement for the computer. An early estimate[3], using fairly optimistic component failure rates and component counts, showed the resulting computer failure rate to be well above that which would be required to meet the computer's apportionment of the mission success probability (P_S = 0.998). Even considering the mission plan for operating the computer only 10 percent of the mission time (for the balance of the mission the computer would be in standby, which in theory reduces the probability of failure), made the failure rate only marginally acceptable. Under these conditions designers could use redundancy techniques or develop more reliable components and manufacturing procedures in order to improve the reliability. In the case of the APOLLO computer various methods of accomplishing the redundancy were studied. However, none could be used and still meet the power, size and weight requirements of the APOLLO mission. The elimination of redundancy provided the sensitivity and motivation for improving reliability at all levels of design, specification, manufacturing and testing. The tight assembly, inspection and test procedures during the manufacturing process detected many problems, each of which were closely monitored, and for which corrective actions were developed. The resulting emphasis on quality has paid off by decreasing the actual failure rates of the computer considerably below the original estimates, even though the component count increased after the original reliability estimates were made.

7.2 COMPONENT DEVELOPMENT

During the early stages of the computer design, an effort was made to constrain the number of different components to a selected few, thereby concentrating the engineering effort required in the area of component development. These constraints were rigidly adhered to and were a constant source of complaints from the circuit design engineers because they felt the limited number of component types constricted their designs excessively. Not only the types of parts were limited but also the range of values. For example, resistors were limited to one type and to a tightly

restricted number of different values. The constraints were reviewed frequently and relaxed as new requirements were justified, but the existence of the constraints accomplished a greater than normal degree of standardization. The benefits that resulted from the effort to standardize were: (1) a reduction in the level of activity needed to specify the components and the level needed to develop testing methods that were capable of continuously monitoring the quality of the components, (2) a reduction in the efforts required to track the manufacturing problems that were related to a component defect or testing procedure, and (3) more important to the reliability of the component was the large volume of procurements that provided increased competition between vendors and greater motivation to meet the reliability requirements.

7.2.1 Component Selection

Component selection was started in parallel with the development of circuit designs. Initially the design engineers were required to specify the general characteristics of the required components and the possible vendors for the component. Then, after a vendor was selected, sample purchases and engineering tests were made. One of the earliest and most important reliability tests was an internal visual examination of the component in order to identify the construction processes used. This visual examination identified weaknesses in the design, helped determine the type of tests that could be used to qualify the part, and provided information necessary to establish process controls. Additional engineering tests, both environmental and electrical, provided the information as feedback to the vendor for product improvement. This process of iteration varied in magnitude for different types of components. Parts like resistors and some condensors required little or no development activity, as only the type of component and the vendor needed to be selected. At the other extreme, the semiconductor components required development activity that lasted well into the design and production of the Block II computer.

The most prominent example of the activity involved in component selction and the value of standardization in minimizing the activity required was the development of the integrated circuit NOR gate. The Block I logic design was accomplished with only one type. The initial Block II design also used one type but had to be changed to two types as a result of logic coupling in the substrate between the two independent gates on the single chip (see Section 7.4.2.3). The resulting types, that were a dual logic gate and a dual expander gate, differed only in interconnection pattern on the chip. Therefore the manufacturing and testing of the gates were otherwise identical, and the engineering effort could be concentrated on the development of a single device.

To select standard transistors and diodes was probably more difficult because of the wider variety of applications. The NPN transistor was a good example of this problem because the range of application varied from the very low current high frequency operation in the oscillator to the high current memory drivers and high voltage relay drivers. This range of application stressed the state-of-the-art in transistor manufacturing, since it required a reasonably high voltage, high current type transistor. But it also required high gain at low currents as well as fast switching and low leakage. This range of applications was satisifed by the development (or selection) of a transistor chip with adequate electrical characteristics that could be mounted in a metal base TO-18 header. The case configuration was selected as the result of thermal design considerations. The metal base TO-18 header provided a package configuration with a low junction to case thermal resistance. Transistors for a relatively few special applications, such as the oscillator which required high gain at low currents, could be selected during computer assembly from the normal production distribution of parameters for this single transistor. This standardized the transistor production, qualification, and testing up to module fabrication. To select a standard PNP transistor was a problem similar to the NPN. Diodes were standardized to one type and selected for special application like the matching of forward voltage drop in the rope sensing circuits.

A few circuit applications could not be met using these standard parts. Most instances were in the power supplies, where very high power and current were required. Comparing the effort of specifying, evaluating, qualifying, and monitoring a low usage component to that of a high usage component illustrates the advantages of standardization. As an example, consider the high current switching transistor used in the pulse width modulated power supply. This component is a single usage item but had vendor and application troubles several times during the computer production. The individual problems with this device consumed as much analysis effort as comparable problems with the high usage component.

7.2.2 Engineering Qualification

Components were qualified differently depending on their criticality and production maturity. Parts like resistors, capacitors, transformers, and low usage semiconductors were handled as illustrated in Figure 7-1. A specification control drawing (SCD) was prepared; a nominal amount of engineering evaluation was conducted; the parts were released for production procurement; and then subjected to the component flight qualification program. These parts had no screen and burn-in requirement other than that which was specified in the specification control drawing (SCD). Critical parts, like the integrated circuits and high usage transistors, followed

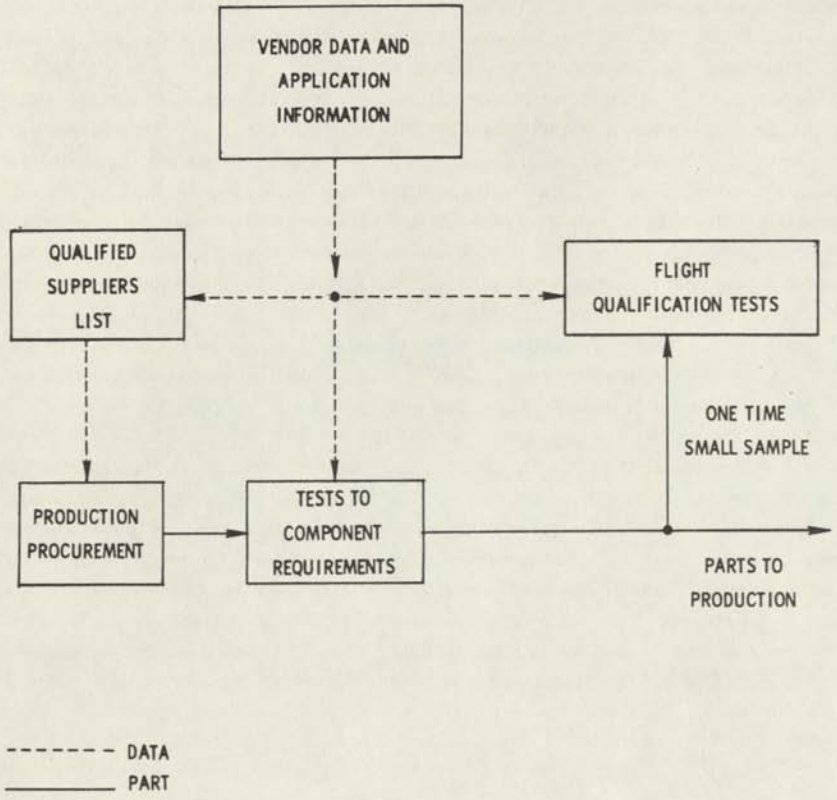


Fig. 7-1 Procedure for Component Selection and Qualification

the more rigorous procedure of engineering qualification and production screening, as illustrated in Figure 7-2. The DSKY relay and the standard diode followed a procedure in between these two extremes where the engineering evaluation and qualification was minimized, but a tightly controlled screening procedure was introduced as a requirement fairly late in the program.

All parts were subjected to testing or data analysis sufficient to establish that the part was qualified for in-flight operation. The qualification of critical components like the integrated circuits required considerable development, since the technology was new and very little history had been developed that would lead to a knowledge of the component reliability.

The engineering qualification process of the critical parts began with an assessment of the vendor's ability to supply devices, the institution of component standardization in designs, the generation of specification control drawings (SCD), and the preliminary study of device failure modes. A block diagram of this preliminary evaluation, that precludes any production procurement, is presented in Figure 7-2. The qualification procurements that supplied parts for the engineering qualification testing and engineering evaluations established confidence in the manufacturer's device processing and provided data on the device failure modes. Conclusions from the failure mode analyses were supplied to the manufacturer who then applied corrective action. This cyclic procedure was continued until the most obvious problems were eliminated. The knowledge of the failure modes and methods of exciting the failure modes were used to design the test environments and rejection criteria of the component screening procedures.

The design of the qualification testing procedure considered the conditions of the component application and the most likely failure mechanisms. Because these tests used small sample sizes, approximately 100 from each manufacturer, only those mechanisms with a reasonably high probability of excitation could be detected, even though the tests and failure analysis were carefully conducted. It was also extremely important that all qualification and engineering testing be performed on devices fabricated from processes as near identical to computer production as possible. The qualification method that was used subjected the devices from various vendors to environmental extremes beyond usage conditions in an attempt to identify failure modes that could occur in normal applications. The method, commonly called the step stress technique, was used in most cases but, since the same lot of devices was subjected to the different stress levels serially, care had to be exercised in the analysis of failures in order to determine the test condition that caused the failures. Based on the results of step stress tests, vendors were selected, and test conditions for screen and burn-in were verified.

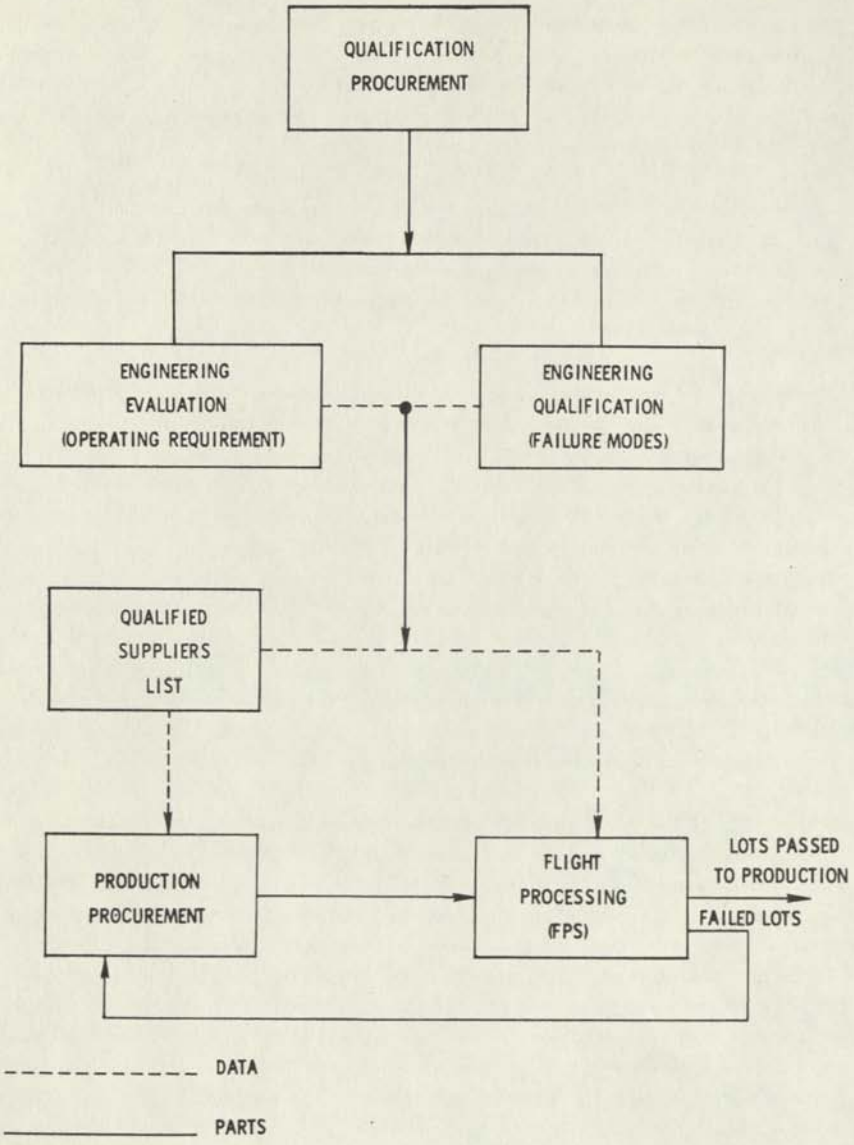


Fig. 7-2 Procedure for Critical Component Selection and Qualification

The engineering evaluations were performed simultaneously with the qualification procedures to determine device speed, fanout capabilities, noise immunity, and operating temperature range. From these evaluations, limits of the electrical parameters for the specification control drawing (SCD) were determined to insure proper device operation in every usage mode and to establish the logical design rules for the computer.

Engineering qualification and evaluation tests determined those vendors capable of supplying the semiconductor part without serious reliability problems. Qualification tests alone were insufficient to determine the ability of a vendor to control his process and continue to deliver a quality product. Large volume production procurement of a high reliability part requires continuous monitoring and process control to insure that the quality demonstrated in the qualification tests is maintained during the production cycle. The procedure requiring continued monitoring of vendor quality and processes was written into the procurement and processing specifications.

7.2.3 Specification

Data generated during the engineering evaluations and qualification tests were used to prepare the specification control document (SCD) for procurement of the more critical parts. Based on the performance of a part during vendor qualification, the qualified supplier list (QSL) was formed to specify the vendor sources acceptable for production parts. Less critical components were specified with vendor supplied data and qualified after production procurement began. See Figure 7-1. Many other components, such as relays, followed a procedure between those depicted in Figures 7-1 and 7-2. That is, a limited number of engineering tests were conducted to determine vendor qualification, but the specifications were derived from vendor supplied data.

7.2.4 Flight Processing Specifications

The Flight Processing Specification (FPS) was developed in response to apparent and real reliability needs. The need for the FPS or its equivalent evolved from a great deal of data and also from sobering history. At the outset of the program there were many component problems. One instance occurred when the reliability group stated that some parts should not be used in fabricating computers. However, because of production schedule pressures, the faulty components were used, and, as predicted, the modules with these defective parts developed failures and had to be scrapped. This constant conflict between production schedules and reliability required that the reliability be better defined with a quantitative measure of the

quality before the component was released to production. A reliability specification similar to the specification control documents (SCD) was required. Then, the quality of parts, on a lot basis, could be evaluated from quantitative data. The Flight Processing Specification (FPS) became the tool that generated quantitative data for determining the quality of a lot of components. It became apparent after considerable experience that the FPS forced component part process control without explicitly stating it, while NPC200-2, "The Quality Program Provision for Space System Contracts", April 1962, stated process control without the ability to enforce it. That is, NPC200-2 required that processes would be documented and not changed without approval. However, the FPS provided vendor motivation because lots would be rejected, if the vendor lost control of the process in such a way that the change was reflected in the visual inspection or product quality.

The FPS was initially developed for the integrated circuit, because as the most numerous component in the computer, the integrated circuit was the predominant component controlling the computer's reliability. It was shown[4] that a computer mean time to failure of 3000 hours was a minimum in order to achieve the necessary probability of mission success. It was also shown[3] that, if a failure rate of 0.02%/10 hours were assigned to the integrated circuit together with realistic failure rates for the other component parts of the computer, the overall reliability of the computer fell considerably short of the mark. It was further stated[3], "Even if we assume an improvement factor of five or even an order of magnitude, system reliability, as we have calculated it, is still not adequate". Extrapolating to the integrated circuits, a failure rate of closer to 0.001%/10 hours was required. The discrete transistor had not demonstrated a reliability this high and even today, the standard product lines of integrated circuits will exhibit failure rates between 0.1 and 0.01%/10 hours. As a result, any program requiring high reliability like APOLLO, must resort to extreme material and process control, captive vendor assembly lines, extensive screen and burn-in, and lot rejection procedure.

From a position of technical director for the APOLLO system the only means available to insure the required reliability was to impose the flight process specifications as a contractual requirement. One benefit of this requirement was that the APOLLO managers became aware of component reliability and actually used the data as a quantitative tool in the management decisions. The main purpose of the FPS was to establish a firm non-varying procedure that would provide data whose significance could be easily understood. One major drawback in most reliability procedures is that without a firm non-varying procedure, it becomes impossible to assess the importance of isolated failures or component anomalies. There must be complete knowledge of the order of testing, the method of testing, and the method of reporting failures to evaluate the significance of the single failure.

Another side effect was briefly discussed previously. APOLLO experience showed that component reliability could be compromised when a higher priority was placed on production schedules, and there was no requirement for documentation that identified the compromise. The reliability required by NPC200-2, although imposed upon the contractor, did not provide the detailed reliability procedures necessary to make the requirement effective. This is not a criticism of NPC200-2. It would be impossible to write a specification that would detail all things for all components. The details of a general specification are the responsibility of the prime contractors. The flight processing specification did indeed contain the detailed description of how to execute the requirements of NPC200-2.

In general the FPS approach turned out to be such an iron clad document that no deviation was possible without a waiver. Although a deluge of controversy followed, and pressure was applied to loosen the requirements, it was felt that every conceivable effort should be expended to provide highest possible quality components for production. A good procedure, therefore, would highlight component problems and not success. If the FPS was to be a good management tool, the deviations and problems must appear for management decision via the waiver route. In contrast, the loosening of the requirements would create fewer waivers and would create the condition where the requirement for reliability was paid for, but not documented, and not necessarily realized. The waiver, indicating the lack of reliability, became part of the data package for a computer and provided documentation for judging the reliability of the computer years after the components were tested.

The rejection criteria and other details of the FPS were not determined arbitrarily. In the case of the integrated circuits, the FPS was formalized after engineering qualification had been completed, after 100,000 integrated circuits had completed screen and burn-in, and after significant field data[5] were available. Only one vendor approached the low failure rate required by the FPS rejection criteria (this was the case for both Block I and Block II, but the vendors were different in the two cases). The acceptance numbers for the Block II FPS were established from the Block I experience. That is, if these acceptance numbers had been used to reject lots in Block I, there would have been no integrated circuit failures. These numbers would have resulted in approximately 20 percent of the lots being rejected. The acceptance numbers were adjusted so that approximately 80 percent of this vendor's lots would pass. The other 20 percent were considered troublesome and proved so by generating failures in computers.[6,7]

Figure 7-3 depicts the general flow of parts and data as required for the flight processing procedure. The devices, procured by lots, proceed through the screen

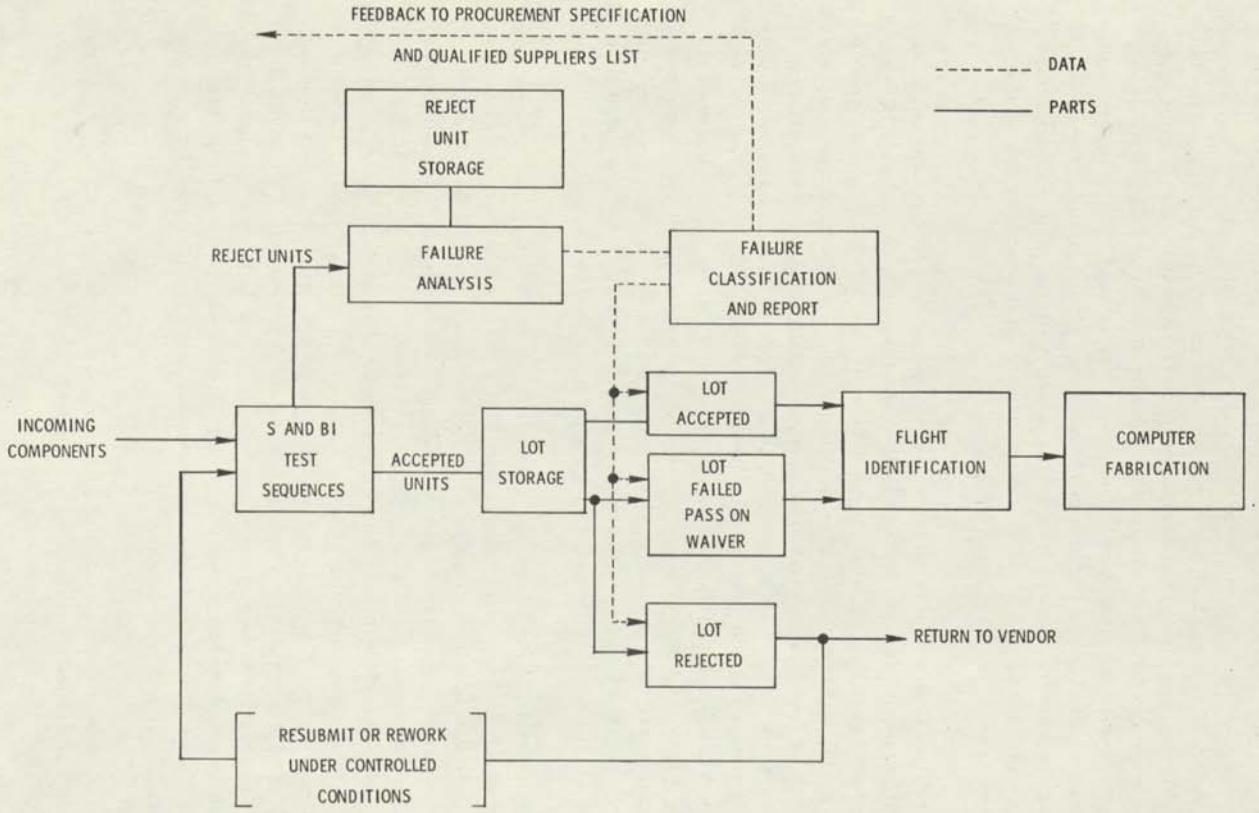


Fig. 7-3 Block Diagram of the Flight Processing Specification Procedure

and burn-in test sequence (Table 7-I) to determine whether the lot is qualified for flight. That is, the FPS procedure is a lot by lot flight qualification in contrast to the more normal procedure, where a part or vendor is qualified by testing a typical production run rather then depending upon process control to insure that the quality is maintained.

After completion of screen and burn-in tests, the lot is stored until failure analysis is completed. After failed units are catalogued, analyzed, and classified to complete the lot assessment, a written report is prepared and, if the lot passed, the devices that passed all tests are identified with a new part number as a flight qualified part and sent to module assembly. A semiconductor part with the flight qualification part number is the only part that can be used in flight qualified computer assemblies. From failure analysis, rejected parts proceed to reject storage, where they will be available for future study. Failed lots are rejected, unless special analysis and consideration qualifies the part for flight computer production by waiver. The waiver was required to be authorized by NASA and to accompany the computer as part of the data package.

The accumulated data, from the screen and burn-in sequences and failure analysis, were used to evaluate vendor production capability, device quality, reliability, and continued status as a qualified supplier.

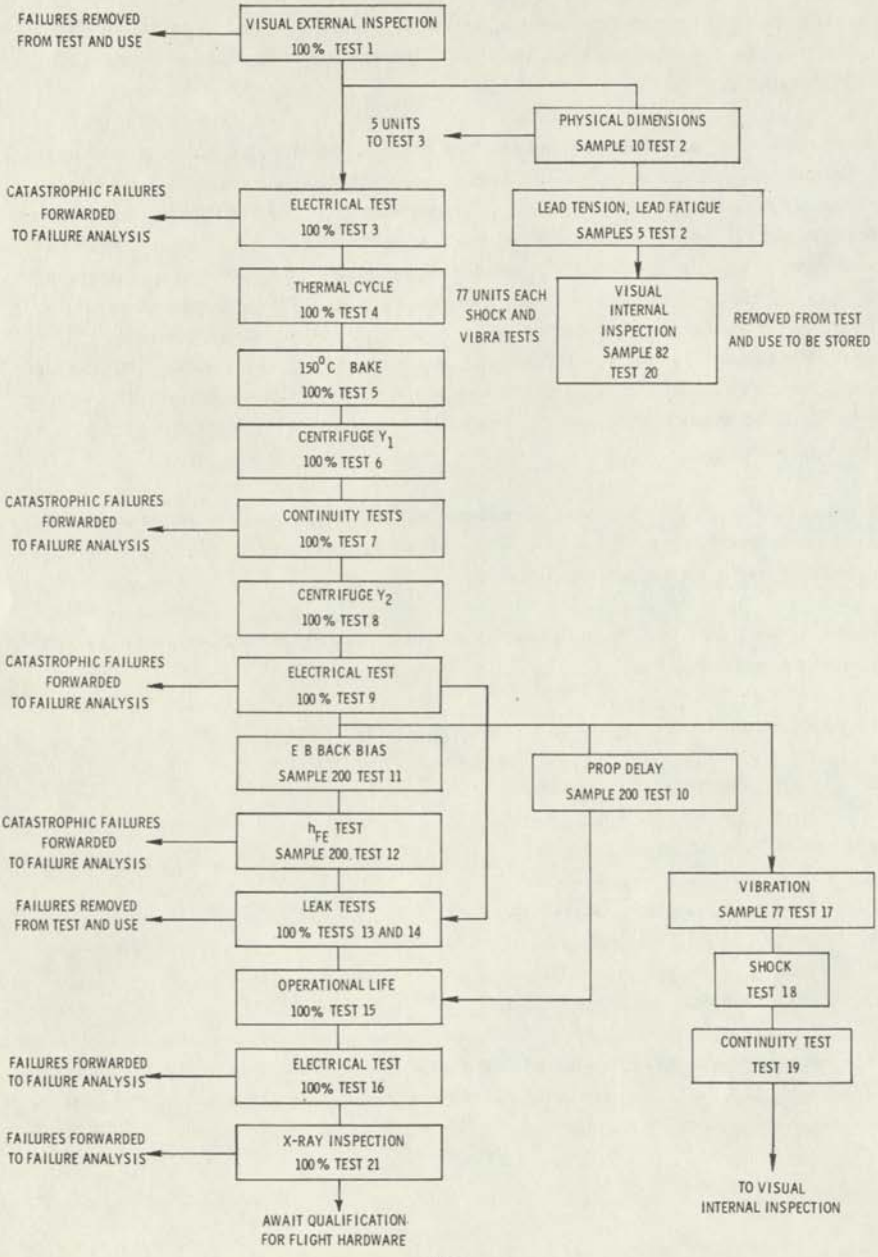
Flight processing specifications that were eventually written for many of the standard computer parts included:

1. Part 1006771 Block I single NOR gates, TO47 package.
2. Part 1006321 Block II dual NOR gates, flatpack.
3. Part 1006323 NPN transistor.
4. Part 1006310 PNP transistor.
5. Part 1006751 Block I diodes.
6. Part 1006399 Block II diodes.
7. Part 1005003 relay non-latching.
8. Part 1005001 relay latching.
9. Part 1006769 sense amplifiers.
10. Part 1006395 multilayer boards

The information in these specifications is exemplified by a description of the group of documents briefly summarizing the salient features of the flight processing for the dual NOR gate. ND1002359 is the flight process specification (FPS) for this

TABLE 7-I

FLOW DIAGRAM FOR THE TEST SEQUENCE



gate. It specifies procedures for lot acceptance resulting in flight qualified parts. In particular, ND1002359 specifies:

1. The operational stress, environmental stress, and the test sequence. This testing procedure is referred to as the screen and burn-in process. The procedure was designed to excite failure modes of components that would be potential failures during the normal stress environment experienced during computer operation. Table 7-I is the flow diagram for the NOR gate.

2. The electrical parameter tests to be performed during the screen and burn-in procedure. The tests were determined during the engineering evaluation and were chosen to detect failures and assure proper computer operation.

3. Definitions of failures. Failures have been defined as catastrophic, several categories of noncatastrophic, and induced.

4. Disposition of failures. The conditions are defined for removing failures from the screen and burn-in procedure and forwarding them to failure analysis or storage if failure analysis is not necessary.

5. Failure mode classification. Failure modes are classified in groups according to screenability and detectability of the failure mode. Five groups were used, ranging from noncatastrophic failure (Group 0) that are 100 percent screenable to failures that could not be identified (Group 4) and are therefore assumed not screenable.

6. Maximum acceptable number of failures per classification. Table 7-II is an example of the acceptance criteria used on the NOR gate. The significance of these numbers may be illustrated by the fact that a lot of 5000 gates exhibiting more than one Group III failure during test 7, 9, and 16 will be rejected. (A Group III failure is an open circuit caused by corrosion, scratches or similar defect in the interconnecting metallization on the semiconductor chip.) However, the lot can pass with one Group III failure and no more than 15 Group I failures during tests 7, 9, and 16. (A Group I failure can be caused by an open bond due to underbonding, a failure mechanism screenable by centrifuge.) Although the very small number of failures allowed may seem excessively limiting, the numbers were developed from the early experience with the gate and were deliberately set low to provide a very sensitive control of quality.

TABLE 7-II

FAILURE ACCEPTANCE CRITERIA FOR NOR GATE

TEST NUMBER	MAXIMUM ALLOWABLE FAILURE IN % OF LOT SIZE				
	GROUP 0	GROUP 1	GROUP 2	GROUP 3	GROUP 4
3	0.5	0.3	0.08	0.04	0
7, 9, & 16	0.5	0.3	0.04	0.02	0

7. Maximum acceptable number of failures for non-electrical tests such as leak test, lead fatigue, etc.

8. A report for each flight qualified lot. The report must contain the complete history of the lot with the specific data and analysis required for flight qualification. The report also contains a complete accounting and disposition of all parts in the lot.

The specification, ND1002257, defines the rejection criteria for internal visual inspection of the NOR gate. ND1002257 serves a dual purpose. It is applied by the device manufacturer during a 100% preseal inspection for removal of defective parts, and by the customer on a sample basis as a destructive test for lot acceptance as part of the requirements of the FPS. Some of the problems in a lot may be detected only by destructive internal visual inspection and certain failure modes can only be observed after the sealed and branded device has been exposed to operational and environmental stresses.

The internal visual inspection criteria were defined after most failure modes of the APOLLO logic gates were determined. Devices were not rejected merely on the basis of aesthetics, but only when a fault that contributes to a known, potential failure could be visually observed. The rejection criteria of ND1002257 do not attempt to reject all of the visually observed faults contributing to failure because of the difficulty of precisely or quantitatively defining faults that are subject to individual interpretation.

7.3 MANUFACTURING PROCEDURES

7.3.1 Process Control

Strict process controls are used throughout the assembly. Processes like welding, wirewrapping, potting are specified and are under tight control. As an example, in the case of welding all lead materials are controlled. The weld setting of the welding machine is specified for every set of materials to be welded, and the in-process inspection procedures are established. Periodic quality control inspections are made on each welding machine to verify that the machine and the operator are producing weld joints that can pass destructive type tests. The material, size, and shape of electronic component leads are standardized where possible without sacrificing the reliability of the component. The standard lead materials used are

kovar, dumet, and nickel. The interconnection wiring is nickel, thus limiting the number of different kinds of weld joints that must be made during assembly. The fact that the process of welded interconnection lends itself to tight control was one of the primary reasons for its use in the APOLLO computer design.

7.3.2 Final Acceptance

Final acceptance procedures were designed to test the functional capability of the computers and DSKYS in addition to subjecting the assemblies to stresses that would excite potential failure mechanisms. The final assembly was subjected to extreme vibration, temperature, and voltage that were in excess of the maximum mission requirements. The modules are subjected to temperature cycling, operational tests under thermal extreme, and in some cases operational vibration tests to detect design and workmanship defects. Some of the tests that were specified initially were changed to increase their effectiveness as a screen. The history of vibration testing as applied to the detection of component contamination represents an example of how the procedures were changed to increase the effectivity.

Briefly, the history of vibration testing starts with sine vibration that was changed to random. Later the vibration axis of the computer was changed to increase the sensitivity to logic gate contamination, and finally operational vibration of individual logic modules was introduced. The computer long term aging test is an example of decreasing the requirement, since the test was not contributing significantly to the screening of potential failures. The Block I long term aging required 200 hours operating time before sale of a computer. In Block II the requirement was reduced to 100 hours, since the experience during the Block I testing and in field operations indicated that no potential failure mechanisms were being detected by the test.

7.4 MANUFACTURING PROBLEMS

The manufacturing problems during the development and production phase of the program were primarily concerned with obtaining or maintaining a component quality level that might be considered beyond the state-of-the-art for even high reliability components. Some problems were caused by the component design or the manufacturing processes. Other problems were the result of a discrepancy between the component application and its design characteristics. The latter were usually detected during computer assembly and test.

7.4.1 Component Defects

The types of component quality problems experienced during production can be illustrated by problems with the switching diode, the two switching transistors, the NOR gate, and the relays used in the DSKY.

7.4.1.1 Diodes

Three major problems with the switching diode were detected during FPS processing.

1. Junction surface instabilities were detected by increases in reverse leakage current that was a generic problem of the mesa process used early in the program. All the early production orders were given to a single vendor who used the mesa process. The problem was alleviated by removing that vendor from the QSL (Qualification Status List) for that part and by changing to a new specification (SCD) that required a silicon planar construction.

2. A minor problem was intermittent short circuits in the diodes caused by loose conducting particles entrapped within the package. This problem was controlled by visual microscopic inspection (the package was unpainted, clear glass), the lot rejection criteria of the FPS, and contamination control processes during diode manufacture.

3. The most serious diode problem was the variation in forward voltage drop (ΔV) detected by the variables data limit during the FPS testing. The unstable forward voltage would be a critical failure in many of the computer applications that depend upon the inherent stability of this parameter. After several lots were rejected during flight processing, an engineering investigation revealed that the unstable condition was caused by a change in the contact resistance of the chip-to-stud and button-to-stud contacts. Because the problem was a process-dependent rather than a time-dependent failure mechanization, a screen was developed and introduced informally into the vendor's processes. However, during a later production procurement, the vendor screen was not applied as a requirement, and the vendor discontinued its use. As a result the flight processing (FPS) again flagged the problem by rejecting lots with excessive failures. The problem was resolved the second time by the reintroduction of the vendor screen, and eventually the procurement documents were changed requiring the vendor screen.

7.4.1.2 Transistors

All significant transistor problems were related to the internal leads and lead bonds.

1. In the gold aluminum system of leads and bonds, "purple plague", (an aluminum rich, gold-aluminum intermetallic) was a problem from those vendors that were not properly controlling the bonding pressures and temperatures. The vendors whose parts continued to exhibit this failure mode were eliminated from the QSL early in the program.

2. Another set of problems developed when some vendors were permitted to change the process to the all aluminum system of leads and bonds. It was discovered during FPS testing that there was a time-dependent failure mode resulting from motion in the aluminum lead wire when the transistor was switched on and off at a relatively slow rate. The moving lead and the resulting fatigue of the lead was associated with 1-mil aluminum wire and wedge bonds. A process change eliminated the problem by using larger diameter wire or by a different bonding technique that did not constrict the wire near the bond. See Reference 8 for further discussion of this failure mechanism.

. Occasional die attach problems appeared and caused problems in the applications that required low thermal resistance for proper heat conduction. Only the most severe die attach problems were detected during centrifuge test of the FPS. Others caused some failures during manufacturing when the modules were subjected to extreme temperature and voltage operating conditions. Although the failure investigation revealed the test conditions were more severe than intended in the module specification, it also revealed the existence of some parts with defective die bonds. Corrective action required improvement in the vendor's die attach process, but the ability to screen for defective elements during FPS testing could not be improved. A measurement to verify that each unit meets the specified thermal resistance would be a suitable screen but would be difficult and expensive to implement. Therefore no change was made to the procedures. The margin of safety in the thermal design of the computer was adequate, even though transistors may not meet the specified thermal resistance.

7.4.1.3 Block I Single NOR Gate

Although the single gate in the TO-47 package exhibited various vendor problems, only the problems with the major supplier will be discussed. The units from other

suppliers were not used in flight computers. The major problems were bonds opening and loose conducting particles shorting.

1. Open bonds that predominated early in the program, were caused by insufficient application of pressure or temperature during the bonding operation. The manufacturer eventually overcame the bonding problem. However, during final component procurement, and after the manufacturer moved his facilities, a variety of bonding problems reappeared. Since the part was replaced by the Block II gate, there was no corrective action.

2. The occasional shorting caused by conducting particles in the TO47 package was very difficult to eliminate. The shorting material usually originated from the package material itself (i.e., nickel and kovar) and could not be detected with X-Rays. Since there were very few problems that were the result of conducting particles, the seriousness of the more general problem of contamination was not recognized until much later in Block II production.

7.4.1.4 Block II Flatpack Dual NOR Gate

The three major problems with the dual NOR gate were package leaks and leak testing, open bonds caused by a gold rich, aluminum-gold intermetallic and shorting caused by loose conducting particles.

1. The leak test problem occured as a result of failures due to corrosion of the metallization internal to the flatpack. The engineering investigation concluded that the corrosion was induced by agents in those packages that were gross leakers (using a procedure similar to that recommended in MIL-STD-883) and had escaped the leak test procedure. Corrosive agents, e.g. water, that were used in the leak test procedures, were forced into and retained within the package, thus introducing a time-dependent failure mode. Further analysis indicated the problem developed following a slight modification in the package sealing process that created a larger number of gross leakers. With the larger number of leakers and the ineffective test procedure, defective units escaped the leak tests and created the problem. The solution required the development of a more quantitative leak test procedure (method detects a weight increase following pressurization under FC-75 fluid) employing a less corrosive liquid as a detecting media and introduction of additional tests for modules containing parts that were not adequately leak tested.

2. The open bond problem experienced with the logic gates was characterized by cracking in a gold rich phase region of the gold-aluminum formation[9]. Many engineering tests were initiated to determine the cause of the outbreak. The conclusions of the study were that the testing in the FPS was reasonably but not entirely effective, and that the failure mechanism was not time or temperature dependent. The mechanisms that caused the failures were never determined. However, studies indicated the mechanism is related to the atmosphere surrounding the chip during the lidding process. However, two computer failures were of this type. (See field experience Section 7.5).

3. The problem with loose conducting particles in the logic gate developed in severity throughout the production cycle. The change in severity of the problem was due in part to an increased awareness of the problem, and in part as a result of corrective action to alleviate some poor die attach problems. The corrective action was a harder die scrub during die attach that resulted in gold "pile up" around the chip. The "pile up" would break loose thus becoming a source of conductive particles within the package. Other sources are pieces of lead material, gold-tin solder from the cover sealing process and chips of silicon.

   The corrective actions to solve the contamination problems started by introducing vendor internal visual inspection changes in December 1966. By August 1967 MIT/IL had completed a study on the use of X-Rays as a screen and had attempted to change the FPS to provide for a 100 percent X-ray screen. The change was not processed until August 1968 because of many debates about the effectiveness of the screen. To illustrate this lack of an agreement, the following is a quote from one published memo: "to perform 100 percent X-Ray examination of several thousand flatpacks, looking for slight anomalous conditions indicated by white or greyish spots on the film, is not conducive to good efficiency". This attitude prevailed in management, until it became obvious that the time consumed in debugging computers with intermittent failures during vibration was not conducive to good efficiency either. When this became obvious, it was almost too late to X-Ray screen because most of the lots were in module assembly. However, the few remaining lots were processed through X-Ray, and the FPS was changed to specify the procedure.

   The only remaining corrective action possible was the introduction of a module vibration test with the capability of detecting transient failures induced by mobile conducting particles. This module vibration procedure that was

introduced in the early Fall 1968 was effective, since no more failures occurred during computer vibration, but it was also costly and time consuming. Unfortunately, it has been impossible to determine the effectiveness of the X-Ray in reducing the contamination failures in the module or computer vibration. This was because lot identification of X-Rayed versus non-X-Rayed lots and failure analysis of vibration failures were not completed to the point of confirming the failure and identifying whether the failure was X-Rayed. However, the gross failure rate during vibration was lower for those modules using a high percentage of X-Rayed lots.

7.4.1.5 DSKY Relays

Transient failures in the Block I DSKY during vibration testing was the first indication of contamination in the DSKY relays. The indication of failure was the intermittent display of all eights on the DSKY numeric panel. Analysis confirmed that an all eights display could be induced by shorting between the normally open and closed contacts of any one out of over one hundred relays used to switch the numeric displays. Since the application was extremely sensitive to the presence of conducting particles in the relays, a concentrated investigation was mounted to develop corrective action. Block I procurement was complete. Therefore the investigation for Block I concentrated on developing vibration screens at the DSKY assembly level. The Block II investigation included vendor processes and the introduction of relay screening techniques.

A vibration profile was devised that would subject the relays to a vibration environment designed to excite loose particles without damage to the relay. The vibration was successful in detecting additional contamination failure, but it also revealed new open-contact failure modes. Earlier, it had been assumed that open contacts were caused by particles lodged in the armature that prevented contact closure. It was later determined that many of the open contacts were mechanically closed but exhibited high-resistance. The hypotheses presented for the high-contact-resistance failure mode were never sufficiently verified with tests or data to determine the origin and causes of the problem. For instance, the hypotheses did not explain why most of the failures occurred at only one set of contacts.

Because of the defined dilemma between excessive vibration inducing contact failures and the necessity of the vibration screen for contamination, a special engineering test was devised whereby a lot of relays (85 good relays) would be subjected to intervals of vibration until an interval was reached during which no failures occurred. After 16 vibration intervals a zero-failure interval occurred, and the test was stopped.

A data review revealed that the failure distribution up to the fifteeenth interval was completely random. Therefore the question arose whether the zero-failure interval was also a random event, and if the 17th and 18th interval would also generate zero failures, or would additional failures be generated? Also, since only 26 relays survived, the effectivity of the vibration as a screen was questioned. A significant point that developed from the test was that most of the opens occurred at one contact incorporating materials different from the other contact.

Subsequently the relay FPS was imposed, and a vibration screen was introduced at the module level. The FPS specified a vibration screen, internal visual inspection, etc., with lot rejection for lots with excessive failures due to contamination. When fully flight-processed relays were built into the modules, additional vibration failures occurred at the module level, and at least three DSKYs failed during sell-off following screening of all the relays and modules. The only conclusion that can be drawn is that the corrective action taken reduced the frequency of failure at the final assembly but did not eliminate the problem.

The field experience, although good, should not be accepted as a verification for this type of relay processing. The wearout mechanisms of this relay were never evaluated following the vibration environment to which the relay was subjected, and the experience of continued contamination failures during vibration testing is a positive indication that the screens were not 100 percent effective. In addition, there was an indication of contamination in the main panel DSKY of the APOLLO 12 command module before launch. During the mission there was no further indication of failure.

7.4.2 Design Defects

This section deals with manufacturing problems that were the result of marginal design or component application. In particular, the type of design problem that wasn't detected during the engineering or qualification tests of preproduction hardware. Although there were relatively few of these problems, they were of interest because they illustrate where engineering analysis or testing to worst case conditions did not excite the latent failure mechanism. The randomness of the variables that trigger the failure masked the failure mode during all the preproduction and qualification tests.

7.4.2.1 EL Panel

Several EL (electroluminescent) panels experienced cracked glass when subjected to vacuum and temperature environments. The first significant failure occurred

during lunar module thermal vacuum tests. The initial conclusions were that the failure was induced by a blow on the panel surface. Later investigation determined that the glass strength was marginal when subjected to the environment of temperature and vacuum. The vacuum is critical, since the panel is sealed at one atmosphere, and in the vacuum environment the glass must withstand a pressure differential of one atmosphere. Analysis revealed that either invisible surface scratches that weakened the glass or the normal distribution of the strength would be sufficient to result in a small percentage of failures.

Because the marginal design was not discovered until most production units were complete and many were delivered, a fix was developed rather than a new design. The fix was accomplished by bonding safety glass to the surface with an optical cement.

7.4.2.2 Erasable Memory Module

A much more complicated problem developed when there were several failures of the erasable memory modules due to breaks in the #38 copper wire used for internal wiring of the core stacks and from the core stack to module pins. See Figure 7-4 for the cross-sectional view of the module and the location of the failures. Analysis of the breaks concluded that they occurred when the wire was subjected to tensile or fatigue stresses. The investigation determined that the cause of the break was excessive motion of the core stack and module pins within the RTV-11 potting material during vibration testing. The analysis necessary to determine the flight worthiness of parts delivered and to determine a satisfactory solution to the problem was probably one of the most difficult "witch hunts" during the computer development.

A mathematical model[10] of the failure mechanism or wire damage was developed to determine the flight worthiness of computers delivered with these modules. Using this model and the failure experience, it was possible to develop a probability of a failure of a module in the flight environment after being subjected to a known vibration history. As might be expected the results did not produce a ground rule against which to judge the flight worthiness of a particular module. There were many variables in the analysis, the value of which could only be assumed. For example, the vibration environment of the module in flight, the relation between stress in the wire and displacement of the wire, the weighting factor between the damage caused by sine vibration and random vibration, since the module encounters both, and many other variables of similar nature. However, results of the mathematical model provided a knowledge of the relationship between the various variables that was extremely useful for evaluation of the merits of the proposed design changes.

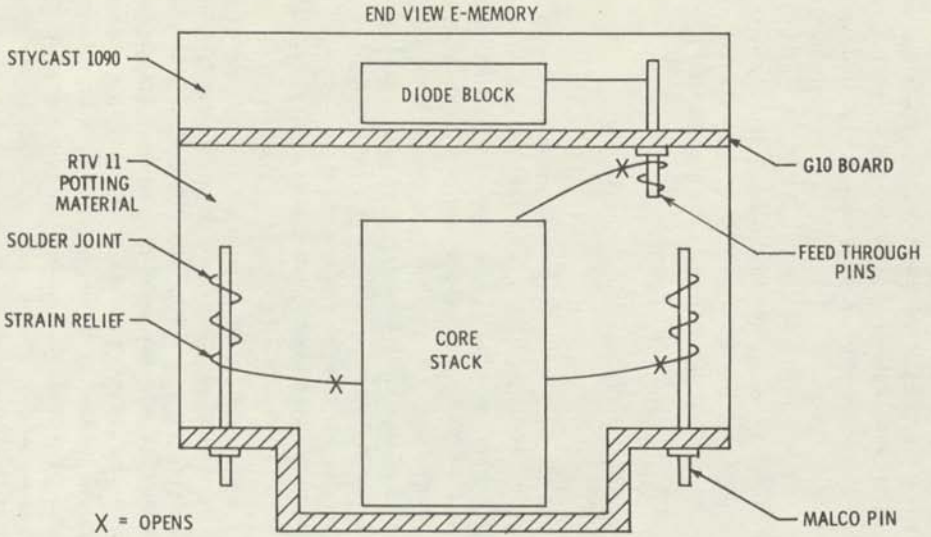


Fig. 7-4 Erasable Memory Module

The design change, that was eventually accepted, was a change in the potting compound from RTV-11 to Sylgard 184. This change resulted in about a factor of four reduction in the damping factor of the potting material that, according to the mathematical model, would reduce the damage by a factor of several hundred. In addition, changes were made in the test procedures that would limit the length of exposure to vibration. Previous to this change the accumulated vibration time on the modules varied between fifteen and several hundred minutes. Since the damage is a linear function of exposure time, the reduction in this parameter did not make a significant change in the accumulated damage. In order to change potting materials an investigation into the chemical, electrical, and mechanical effects was necessary. It was known from previous experience that many desirable potting compounds would crack the ferrite cores used in the memory. This and many other properties of the potting materials had to be evaluated before the design change could be accomplished.

It should be noted that a preproduction E-memory module had been subjected to and had passed a step stress test as an engineering qualification, and that during the design other approaches to packaging the memory were investigated but had failed the engineering qualification. The intent of these tests was to determine the capability of the design and of the subassemblies to withstand the specified vibration levels. The stress levels of vibration and exposure time, that the module was required to pass, were considerably more than the expected exposure during the production cycle. The distribution of random variables associated with the module manufacturing processes apparently masked the failure mode during the engineering qualification tests. The analysis of the production failures as a function of accumulated vibration time indicated that the failures per unit of vibration were almost constant. That is, the failures were not the result of a wearout mechanism. A wearout mechanism would cause the failures per unit of vibration to increase with accumulated vibration. Since the failure probability experience in production was quite low, the probability of detecting the failure mechanism during the engineering qualification tests would also be quite low. To evaluate a module that exhibits a constant probability of failure, a vibration life test would have to be run using several modules in place of the stress test using one module.

7.4.2.3 "Blue Nose"* Problem

The "Blue Nose" problem is an example of a component problem, where a fundamental characteristic of the component was not considered in its applications. The

*"Blue Nose" is an expression in the parlance of the MIT logic designers indicating a logic gate used without power applied, such as a gate used to increase the fan-in as described in Section 3.2.1. It takes its name from the graphical symbol used to denote it.

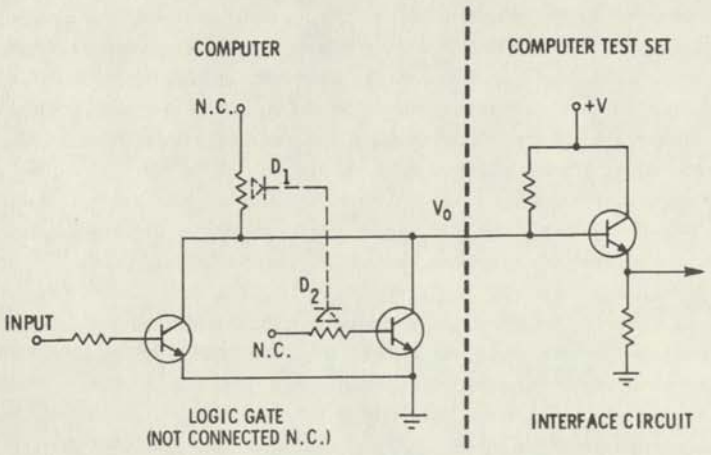


Fig. 7-5 Integrated Circuit Gate Illustrating "Blue Nose" Problem

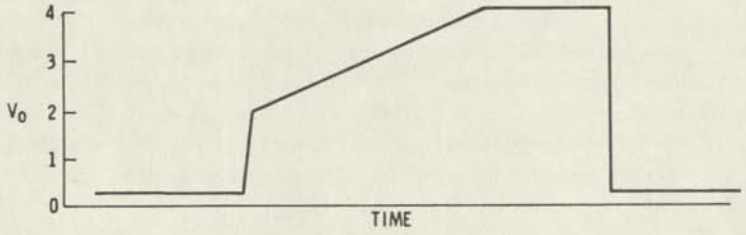


Fig. 7-6 First Pulse of Pulse Train

characteristics of the isolation regions of the integrated circuit NOR gate caused the problem because: (1) the behavior of the isolation regions was not understood during the design, and (2) the engineering evaluations were not detailed enough to expose the existence of marginal conditions. The problem developed late in Block I production in the interface between the computer and computer test set. Figure 7-5 shows the circuit schematic, and the parasitic elements that caused the problem are shown as dotted lines. When V_o rises to about 2 volts, the diode-capacitor coupling occurs through the resistor substrate, diodes D_1 and D_2, to the unused transistor. This coupling is a feedback path that slows the pulse rise time as indicated. The rise time will be a function of the gain of the unused transistor as well as a function of the repetition rate of the driver. Diode D_2 behaves as a capacitor that charges rapidly but discharges slowly, since the reverse impedance of D_1 is in series. The first pulse of a pulse train will be slow, as shown in Figure 7-6, and all succeeding ones faster, if the period between the pulses is small compared to the discharge time of D_2. Since the magnitude of the effect is also dependent upon the gain of the unused transistor, it can be seen why engineering tests may not detect the problem. The condition required to detect the slow rise time is one where the transistors are high gain, and the rise time of the pulse is critical yet the data rate is low. Late in the Block I production a shift in the distribution of the transistor gain to a higher average gain caused this problem to be detected and become very troublesome. The most expeditious solution at that point in production was to select the low gain components for use in the critical locations. Another possible solution, that could not be as easily phased into production, was to ground the unused inputs of the gate.

During Block II production a similar "Blue Nose" problem developed that was made more complex by the fact that there were two gates on one chip with common isolation regions and common power supply. With the Block II gate the resistance coupling between outputs of an unpowered chip resulted in additional signal load on a "1" output signal if the mate was a "0" signal output. The "1" output was required to supply the load current of the two collector resistors to ground via the transistors of the other gate. For a heavily loaded gate this would be excessive. During Block II the problem was solved by grounding all unused logic inputs and making a special version of the logic gate that opened the metallization on the chip between the collector and collector load resistor, thus making a "Blue Nose" gate a nonstandard logic element. Another solution would have been a change from a 10-to a 14-lead package with independent power pins, thus decoupling the collector resistors between the two gates. The minimum impact on test equipment and module design resulting from the metallization change made the special gate a much more attractive solution to the problem than a change in package.

7.4.2.4 Diode Switching Problem

This defect is also an application type problem. It was first detected in July 1968 when a set of rope memory modules failed to operate correctly, when running selfcheck, following installation in the computer of the lunar module #3. A concentrated engineering analysis was necessary to locate the problem and to determine that static matching of the diode forward voltage drop was insufficient for the rope application. In Figure 3-13 of Section 3.5.2 (Simplified Rope Sense Line Switching Diagrams) diode pairs, like D_1 and D_2, are matched for forward voltage drop. As long as the turn-on time of both diodes is approximately equal, the circuit functions according to the design requirements. If the turn-on time is considerably different (on the order of 0.1 μsec), then a voltage difference develops that can be large enough to cause the sense amplifier to read a logic "1" in place of a "0". The failure mode was triggered by a shift in the distribution of the turn-on time of the diodes to slower average value. The memory module specification and test equipment could not detect the fault except under extreme conditions of transient unbalance. Under more normal conditions the transient would die out rapidly. Thus only static unbalance was specified and controlled by the module specifications. The operating conditions between the computer and module test equipment, although designed to be the same, were different enough to cause the transient unbalance to be more sensitive when operating in the computer. As a result, the latent problem was undetected, until the parameter shift of the diodes triggered what was a design deficiency in the module testing and in the use or specification of the diode. The problem was solved by providing a match of the forward voltage drop during the turn-on transient in addition to the static match.

7.5 FIELD EXPERIENCE

The failure history of the computer has demonstrated a failure rate which is considerably lower than predicted early in the development. The early prediction[1] estimated the mean time between failures (MTBF) of about 3000 hrs. A summary of the history (up to Dec. 1971) of 42 flight configuration Block II computers shows 5 field failures which were mission critical. Of more interest are the failure modes that were experienced, the relationship between these failures and the effectivity of the screening procedures used during manufacturing. From the field experience accumlated during the program the probability of mission success has been calculated. This results in P_s=0.996 for the command modules mission which requires a post launch operating time of approximately 200 hrs.

The flight type computers were maintained on strict quality control surveillance. The requirements inposed by the quality control surviellance included maintenance

of complete unit history records, failure reporting and corrective action for all indications of anomalous behavior, a record of all test data, a record of operating time, etc. Failure reporting and corrective action was documented and reported via AFRs (Apollo Failure Reports). Failures of primary interest are those with a "CAUSE" classification of "Part" in the failure reporting system. Failures with a "Cause" classification such as "Secondary", "Induced", "Procedure Error", "Test Error", "Handling" etc. are not considered here although Table III provides a breakdown of the majority of the failures into these classifications. The DSKY failures are less interesting and are not covered in detail since DSKY components are of a largely obsolete technology (pushbutton switches, indicater panels, and relays). The data package for each computer contained all the pertinent data and was maintained current as the computer was tested in preparation for flight.( )R

The failures of primary interest for this section of the report are those that occurred in the field or more precisely while on flight status which starts with the date of acceptance by NASA as determined by the Material Inspection and Receiving Report (DD-250) and ends when the computer was removed from flight status. Table IV defines this period for each computer. The removal from flight status is defined as the end of a flight for those computers which have flown, or when allocated to a ground function such as system simulations, or for the qualification test system when the system completed tests. As indicated in the table, the period of interest starts in the Fall of 1966 when the first computer was delivered for qualification tests, runs through the production period which ended in the Spring of 1969, and through the operational period up to Apollo 15 and terminates at the end of the period of complilation for this report (31 Dec. 1971).

Of the 19 failures listed in Table III which occurred while the computers were on flight status 5 are of particular interest since they are of the type for which no corrective action could be taken. A complete breakdown of the 19 failures is presented in Table 7-V. The first 5 are the failures counted for the prediction of a mission success probability. The other 17 include 10 failures due to contamination in the flatpacks which were detected when a flight status computer was returned to the factory and subjected to a vibration screen more severe than the original acceptance level and an order of magnitude higher than flight levels. See section 7.4.1.4. Failures like this, which were indicated when the computer was being tested at factory test environments more severe than normal mission enviroments, are not counted against the computer for purposes of reliability prediction unless they corroborate field failures, for example, the one contamination failure which occurred during spacecraft integration tests was verified during vibration therefore is counted. The 11th failure (also not counted) was the result of the diode design problem discussed

TABLE 7-III

AFR CAUSE CLASSIFICATION

FAILURE "CAUSE" CLASSIFICATION	AGC	DSKY
Development type dated before 1967	252	67
Procedure and testing errors	199	32
Induced by GSE and Cabling	150	28
Handling and Workmanship	336	42
Electrical Part	182	237
Factory acceptance testing	166	201
On flight status	19	36

TABLE 7-IV

AGC CENSUS

S/N	DD 250 DATE	END DATE	REMARKS	OPTIME HOURS
16 (C-1)	7/25/66	8/23/67	Qualification test system	1776
18 (C-4)	10/20/66	5/16/67	Shipped to Raytheon	274
19 (C-5)	11/19/66	11/27/68	Shipped to MSC	711
20 (C-6)	11/26/66	2/22/69	Shipped to Raytheon.	722
22 (C-2)	8/15/66	7/31/67	Shipped to MSC	122
23 (C-7)	12/7/67	4/26/68	Shipped to Raytheon	107
24 (C-8)	2/7/67	12/31/71	Assigned to APOLLO 17 LM	1549
25 (C-10)	6/27/67	11/20/69	Flew with APOLLO 12 LM	412
26 (C-12)	6/24/67	12/31/71	End of Report Period	1329
27 (C-13)	8/4/67	10/22/68	Flew with APOLLO 7 CM	1545
28 (C-14)	8/23/67	12/31/71	End of Report Period	1286
29 (C-9)	4/5/67	8/2/71	Flew with APOLLO 15 LM	1081
30 (C-11)	6/10/67	1/22/68	Flew with APOLLO 5 LM	987
31 (C-15)	10/12/67	5/23/69	Flew with APOLLO 10 LM	1322
32 (C-16)	9/1/67	3/7/68	Flew with APOLLO 9 LM	1613
33 (C-17)	10/2/67	12/27/68	Flew with APOLLO 8 CM	1471
34 (C-18)	10/11/67	11/24/69	Flew with APOLLO 12 CM	1530
35 (C-19)	9/6/68	12/31/71	End of Report Period	1369
36 (C-20)	4/30/68	12/31/71	End of Report Period	1829
37 (C-21)	2/8/68	3/13/69	Flew with APOLLO 9 CM	1159
38 (C-22)	3/29/68	12/31/71	End of Report Period	1488
39 (C-23)	1/17/69	12/31/71	End of Report Period	1652
40 (C-24)	1/19/68	5/26/69	Flew with APOLLO 10 CM	1206
41 (C-25)	12/15/67	12/31/71	Assigned to APOLLO 16 CM	1219
42 (C-26)	1/16/68	7/21/69	Flew with APOLLO 11 LM	1314
43 (C-27)	2/12/68	12/31/71	Assigned to APOLLO 16 LM	1334

TABLE 7-IV (Cont)

AGC CENSUS

S/N	DD 250 DATE	END DATE	REMARKS	OP TIME HOURS
44 (C-28)	3/25/68	7/24/69	Flew with APOLLO 11 CM	1144
45 (C-29)	2/26/68	2/12/71	Flew with APOLLO 14 CM	1720
46 (C-30)	8/6/68	4/17/70	Flew with APOLLO 13 LM	971
47 (C-31)	1/16/69	12/31/71	End of Report Period	1087
48 (C-32)	4/10/68	12/31/71	End of Report Period	574
49 (C-33)	8/6/68	2/8/71	Flew with APOLLO 14 LM	1184
50 (C-34)	7/25/68	8/6/71	Flew with APOLLO 15 CM	926
51 (C-35)	4/29/69	12/31/71	End of Report Period	511
52 (C-36)	3/31/69	12/31/71	End of Report Period	1254
53 (C-37)	9/25/68	4/17/70	Flew with APOLLO 13 CM	524
54 (C-38)	2/10/69	12/31/71	End of Report Period	876
55 (C-39)	3/26/69	12/31/71	End of Report Period	453
56 (C-40)	5/6/69	12/31/71	Assigned to APOLLO 17 CM	528
57 (C-41)	9/10/69	12/31/71	End of Report Period	584
58 (C-42)	5/13/69	12/31/71	End of Report Period	473
59 (C-43)	5/15/69	12/31/71	End of Report Period	408

in section 7.4.2.4. All flight hardware which is sensitive to this design problem has been purged of the defect. The next failure listed in Table V (also not counted) was a transistor open bond at the post. This was an aluminum wire interconnect bonded to a gold plated post (not the transistor chip) which was open. Analysis indicated there was no evidence of a bond having been made between the wire and the post. None of the previous testing had caused the contact to open. The computer had been on flight status for over a year without indication of this defect and had been returned to the factory as part of a retrofit program to make an unrelated design change. After this retrofit, the failure was first detected when the computer was operating at the upper temperature limit of the thermal cycle. The failure was difficult to isolate since it was not repeatable, but after further diagnostic vibration and testing at the upper temperature limit of thermal cycle, it was again detected and located.

The transformer failure which occurred in June of 1971 is an interesting failure but is not counted against the reliability since the leakage resistance was at least a factor of ten above that which would cause failure in a mission. The history of testing which detected the failure started with a sequence of lightning strikes on the Apollo 15 launch complex about one month before launch. The lightning damaged the GSE power source to the spacecraft. There was no detectable damage to the computer; however, the computer was replaced as a precaution and retested. During a special test of the interfaces this transformer was determined to have excessive leakage. Analysis confirmed the defective transformer but concluded that the defect was not induced by lightning and concluded that the leakage resistance was high enough for proper operation of the system.

The only failure of inteconnections (weld) was in a rope module. This failure was detected during spacecraft vacuum test. Since the rope modules are not enclosed within the computer, the vacuum environment stressed the case and caused the weld to open. Subsequent to this failure the module acceptance tests were modified to include an operating vacuum environment which has detected two more defective welds. Since the introduction of this additional acceptance test spacecraft testing has been free from defects.

The open multilayer board in a logic module was detected when analyzing the module for suspected contamination failures during the module vibration. The computer (C-32) from which this module had been taken was part of a GN&C system which had exhibited an anomolous condition during spacecraft testing. As part of the analysis several of the computer modules were subjected to vibration. This was the only failure detected and it could not be related to the original system failures.

TABLE 7-V

FAILURE CLASSIFICATION

	AFR	DATE	LOCATION	COMPUTER PART TIME		FAILURE MODE
(COUNTED FOR MTBF)	17275	2/6/69	NR	50 (C-34)	Nor Gate	Shorted interconnects Conductive contamination
	17272	1/29/69	NR	43 (C-27)	Nor Gate	Open bond Gold rich, aluminum-gold intermetallic
	23837	11/15/71	DELCO	55 (C-39)	Nor Gate	
	60202	11/20/70	KSC	51 (C-35)	Nor Gate	Open interconnects Corrosion of aluminum
	17291	5/9/69	NR	47 (C-31)	Transformer	Open primary winding Nick in the wire
(NOT COUNTED FOR MTBF)	21973	11/15/78	RAYTHEON	25 (C-10)	Nor Gate	Conducting Contamination Detected during factory test environment more severe than normal mission (vibration).
	20843	6/13/68	RAYTHEON	26 (C-12)	Nor Gate	
	20845	6/15/68	RAYTHEON	26 (C-12)	Nor Gate	
	21560	10/10/68	RAYTHEON	26 (C-12)	Nor Gate	
	20281	12/29/67	RAYTHEON	34 (C-18)	Nor Gate	
	19623	2/26/68	RAYTHEON	34 (C-18)	Nor Gate	
	20491	4/27/68	RAYTHEON	34 (C-18)	Nor Gate	
	20925	7/20/68	RAYTHEON	36 (C-20)	Nor Gate	
	22850	12/15/69	RAYTHEON	38 (C-22)	Nor Gate	
	21919	9/19/69	RAYTHEON	45 (C-29)	Nor Gate	
	22842	11/11/69	RAYTHEON	45 (C-29)	Nor Gate	
	21812	11/19/68	RAYTHEON	25 (C-10)	Transistor	Post Bond Missing Detected during factory test environment more severe than normal mission (thermal).
	19845	7/5/68	KSC	32 (C-16)	Diode	Design Problem Sensitive to forward switching time; all flight hardware purged.
	23759	6/26/71	DELCO	36 C-20)	Transformer	Primary secondary leakage Intermittent high impedance short
	18938	11/15/68	KSC	31 (C-15)	Weld Joint	Intermittent open Rope module exhibited an open during spacecraft vacuum test.

Another class of field failures which are not listed are associated with connectors. The Malco connectors at the interface of the computer and the rope modules are subject to damage during mating of the modules or interface cabling. This class of problem was controlled by careful handling but in summary there was more connector damage then might be expected from this type of connector. Damage is the only class of connector failure that has occurred.

The population of DSKYs considered on flight status was 64 with 36 failures as noted previously. Table VI is a listing of the operating time on the DSKYs. The most interesting class of failures in the DSKY is that which resulted from contamination in the relays. During the manufacturing cycle special vibration screens were developed for testing at the component level, at the module level, and finally at the DSKY level of assembly. See section 7.4.1.5. The experience of continued contamination failures during vibration testing at each level of assembly is a positive indication that the screens were not 100 percent effective. In addition, there was an indication of contamination in the main panel DSKY of the APOLLO 12 command module just before launch. Contamination of any one of 108 relays that operate the electroluminescent panel can cause the panel to read all eights while the relay contacts are shorted by the contamination. The APOLLO 12 DSKY experienced this condition prelaunch but during the mission there was no further indication of failure. Since that experience, a small test program has been developed which will cycle all relays and hopefully clear a failure if it were to occur during flight.

In summary, the contamination in flatpacks and DSKY relays has continued to plague the APOLLO program.

As discussed under Section 7.4.1.4 and 7.4.1.5, the methods for screening components were modified during the production cycle in order to increase screening effectiveness. In the case of the flatpacks, the computers at the end of the production run had the most effective screens which included 100% X-Ray of the components, monitored vibration at the module level, and operating vibration at the computer level. Earlier computers had various combinations of these tests but most of them had only operating vibration at the computer level. Experience has shown both for the DSKY and the AGC that a field return which is subjected to the latest methods of module vibration will very likely have failures due to contamination. One of the computers, after successfully flying a mission, had a contamination failure when it was returned to the factory and its logic modules were subjected to the vibration test. Notice that there is no evidence of contamination failures in flight. The total history (including the testing prior to DD250) of the computers indicates there have been 58 failures resulting from contamination in flatpacks. Most of these occurred during the vibration part of the acceptance test procedures. The 10 failures discussed previously occurred when computers were returned to the factory and

TABLE 7-VI

DSKY CENSUS

S/N	DD 250 DATE	END DATE	REMARKS	OP TIME HOURS
34 (D-1)	12/28/66	10/14/67	Sent to KSC for simulators	129
35 (D-2	12/24/66	1/29/67	Sent to Delco Engrg.	336
36 (D-3)	1/16/67	12/31/71	End of report period	794
37 (D-4)	1/25/67	1/22/68	Flew with APOLLO 5 LM	929
38 (D-5)	4/4/67	10/14/67	Sent to MSC	316
39 (D-6)	2/2/67	7/17/67	Sent to MIT for simulators	638
40 (D-7)	2/16/67	12/31/71	End of report period	610
41 (D-8)	6/27/67	12/31/71	End of report period	1492
42 (D-9)	3/14/67	3/13/69	Flew with APOLLO 9 CM	2010
43 (D-10)	3/14/67	5/26/69	Flew with APOLLO 10 CM	1481
44 (D-11)	6/10/67	12/31/71	End of report period	909
45 (D-12)	6/21/67	12/31/71	End of report period	864
46 (D-13)	6/24/67	2/11/69	Sent to GAC for simulators	576
47 (D-14)	7/27/67	12/31/71	End of report period	2321
48 (D-15)	8/4/67	12/27/68	Flew with APOLLO 8 CM	1732
49 (D-16)	9/1/67	12/31/71	End of report period	3165
50 (D-17)	8/23/67	10/22/68	Flew with APOLLO 7 CM	1436
51 (D-18)	10/12/67	3/7/69	Flew with APOLLO 9 LM	1655
53 (D-20)	1/19/68	7/24/69	Flew with APOLLO 11 CM	1125
54 (D-21)	1/16/68	7/21/69	Flew with APOLLO 11 LM	1278
56 (D-23)	2/12/68	11/20/69	Flew with APOLLO 12 LM	1011
58 (D-25)	12/21/67	12/27/68	Flew with APOLLO 8 CM	1491
59 (D-26)	3/25/68	3/13/69	Flew with APOLLO 9 CM	1391
61 (D-28)	2/26/68	4/17/70	Flew with APOLLO 13 LM	1018
62 (D-29)	2/8/68	5/26/69	Flew with APOLLO 10 CM	1151
63 (D-30)	1/26/68	2/8/71	Flew with APOLLO 14 LM	850
64 (D-31)	1/30/68	10/22/68	Flew with APOLLO 7 CM	1330
65 (D-32)	1/30/68	5/23/69	Flew with APOLLO 10 LM	1293
66 (D-33)	3/25/68	7/24/69	Flew with APOLLO 11 CM	1106
67 (D-34)	3/29/68	11/24/69	Flew with APOLLO 12 CM	1123

TABLE 7-VI (Cont)

DSKY CENSUS

S/N	DD 250 DATE	END DATE	REMARKS	OP TIME HOURS
68 (D-35)	9/25/68	12/31/70	End of report period	351
69 (D-36)	7/25/68	2/12/71	Flew with APOLLO 14 CM	1244
70 (D-37)	4/4/68	12/31/71	End of report period	659
71 (D-38)	4/10/68	12/31/71	End of report period	406
72 (D-39)	4/1/68	4/17/70	Flew with APOLLO 13 CM	1208
73 (D-40)	4/1/68	12/31/70	End of report period	1004
74 (D-41)	3/29/68	11/24/69	Flew with APOLLO 12 CM	1123
75 (D-42)	4/30/68	8/6/71	Flew with APOLLO 15 CM	637
76 (D-43)	4/30/68	12/31/70	End of report period	536
77 (D-44)	7/25/68	2/12/71	Flew with APOLLO 14 CM	1294
78 (D-45)	8/15/69	12/31/71	Assigned to APOLLO 16 LM	1098
79 (D-46)	9/6/68	4/17/70	Flew with APOLLO 13 CM	935
80 (D-47)	9/6/68	12/31/71	End of report period	709
81 (D-48)	7/25/68	12/31/71	End of report period	245
82 (D-49)	11/1/68	12/31/71	End of report period	442
83 (D-50)	11/1/68	12/31/71	Assigned to APOLLO 16 CM	1148
84 (D-51)	1/17/69	12/31/71	End of report period	593
85 (D-52)	12/3/68	12/31/71	Assigned to APOLLO 17 LM	1119
86 (D-53)	1/17/69	12/31/71	End of report period	381
87 (D-54)	8/6/68	12/31/71	End of report period	324
88 (D-55)	3/31/69	12/31/71	End of report period	475
89 (D-56)	9/4/69	12/31/71	End of report period	329
90 (D-57)	9/4/69	12/31/71	Assigned to APOLLO 16 CM	949
91 (D-58)	3/26/69	12/31/71	End of report period	234
92 (D-59)	11/1/68	12/31/71	End of report period	123
93 (D-60)	11/1/68	12/31/71	End of report period	114
94 (D-61)	2/10/69	8/6/71	Flew with APOLLO 15 CM	944
95 (D-62)	2/10/69	12/31/71	End of report period	755
96 (D-63)	12/12/68	8/2/71	Flew with APOLLO 15 LM	1190
97 (D-64)	5/15/69	12/31/71	End of report period	705
98 (D-65)	5/13/69	12/31/71	End of report period	213
99 (D-66)	3/26/69	12/31/71	Assigned to APOLLO 17 CM	707
100 (D-67)	3/19/69	12/31/71	Assigned to APOLLO 17 CM	795
101 (D-68)	4/29/69	12/31/70	End of report period	418

TABLE 7-VII

AGC RELIABILITY STATISTICS

FAILURE ENVIRONMENT		A AGING TIME	B VIBRATION	C THERMAL CYCLE	D NORMAL OPERATION	E FLIGHT
SAMPLE SIZE (NUMBER OF AGC'S)		← 42 →				17
AGGREGATE TIME IN ENVIRONMENT		876,000 HR.	6500 HR.	4200 CYCLES	42,000 HR.	2000 HR.
NUMBER OF FAILURES		3	1	1	0	0
MEAN TIME BETWEEN FAILURES		292,000 HR.	6500 HR.	4200 CYCLES	70,000 HR.*	
APOLLO 14 MISSION	MISSION TIME IN ENVIRONMENT / PROBABILITY OF SUCCESS	CM 200 HR. LM 100 HR. / CM = 0.9993 LM = 0.9996	0.75 HR. / 0.9999	1 CYCLE / 0.9998	CM 200 HR. LM 48 HR. / CM = 0.997 LM = 0.9993	

* ASSUME 0.6 FAILURES FOR MTBF COMPUTATION

$$P_S = e^{-\frac{t}{MTBF}}$$

$$CM\ P_S = (0.9993)\ (0.9999)\ (0.9998)\ (0.997) \approx 0.996$$

$$LM\ P_S = (0.9996)\ (0.9999)\ (0.9998)\ (0.9993) \approx 0.998$$

were subjected to the latest vibration methods of the acceptance test proceedures. Only AFR 17275 (listed in Table V) was related to a failure during operation in the field and was varified by subsequent factory testing.

In general the life cycle of the computer includes assembly and test as part of the manufacturing cycle, followed by GN&C system assembly and test (which is completed when the system is sold to NASA by means of DD250), a period of storage which includes testing to insure operability as a ready spare, installation into the spacecraft followed by a lengthy cycle of prelaunch checkout, and finally a mission. The life cycle is completed for the Command Module System at splashdown. In case of the Lunar Module, the cycle is completed when the operation of the ascent stage of the LM is terminated. In the previous section this cycle was divided into two major periods: first prior to DD250, and second the remaining period defined as flight status. This later period of total age and operating time for each production computer is tabulated in Table IV and is used for determining the reliability statistics.

Table VII classifies the time computers have spent on flight status into each environment and identifies each failure with the environment which induces the failure. The failure environments identified under the five columns are the following: Column A. aging time, which is the total time on flight status; Column B. Vibration, which results from shipment, handling and flight; Column C. thermal cycle, which results from the normal turning power on and off Column D. operation, which is the portion of Column A that the computers have spent in normal operation. The column labeled Flight is that portion of operating time listed under Column D which computers have spent in flight. The aging time and operating time are derived from Table IV. Vibration time is estimated from the records for shipment, handling, etc. The number of thermal cycles is estimated from operating history recorded in each computer's data package.

The failure modes listed in Table V are catagorized in Table VII according to the type of environment which induces that type of failure. The three failures under Column A are those of the four logic gate failure. These failure modes are time dependent but reasonably independent of temperature for the range of normal operation; therefore, these are assigned to the aging time column. The logic gate contamination failure is assigned to vibration. The transformer failure was an open winding which, due to the potted contruction, is stressed by temperature cycling. The failure was intermittent under the conditions of computer warm up. As indicated there are no failures which are classified under operation since the failure rates associated with these four failure modes are not accelerated by the additional environments of temperature, current, voltage, etc. which are imposed by operation.

TABLE 7-VIII

AGC PARTS COUNT

NAME	TOTAL	GENERIC TYPE	SUB-TOTAL
Capacitors	221		
		Solid Tantalum	200
		Ceramic	11
		Glass Dielectric	10
Resistors	2918		
		Wire Wound	111
		Tix Oxide Film	2807
Transistors	550		
		NPN Switching	443
		PNP Switching	94
		Power	13
Diodes	3325		
		Switching	3300
		Zener	25
Transformers	123		
		Pulse	120
		Signal	3
Inductors	108		
Thermistors	4		
Cores, Magnetic	35840		
		Ferrite	32768
		Tape Wound	3072
Integrated Circuits	2826		
		Dual Nor Gate	2460
		Dual Expander	334
		Sense Amplifier	32
Connectors - Pins	19957	Malco	

The MTBF and success probabilities are calculated as indicated in Table VII for both CM and LM computers of the APOLLO 14 mission. The probability of success (P_s) of the mission is the joint probability that both computers survive all environments. From the information in Table VII and the parts count of Table VIII, the failure rate of various components can be calculated. These failure rates may be of interest but the following conclusions that can be derived from the APOLLO experience are of more general interest.

1. The composite MTBF for the computer, when operating in the mission environments for an Apollo Command Module flight of 200 hours, can be computed from the results of Table VII ($P_s=0.996=e^{-200/MTBF}$). This MTBF is 50,000 hours. If computed in the more conventional fashion by charging the five failures against the total computer hours (876,000) the result is an MTBF of 175,000 hours. Total clock time is used in this calculation of computer hours since none of the failure modes experienced are accelerated by computer operation.
2. It can be concluded from the material presented that the computer failure rate is independent of whether the computer is operating or not. This conclusion is based on an understanding of the physics of the failure modes experienced to date. It is also a result of a very careful thermal and electrical design which constrains operating conditions of the components to very reasonable limits.
3. A fairly reasonable development period and a reasonably large number of flight computers were necessary in order to shake down the problems and develop confidence in the reliability statistics.
4. Considerable effort was expended to make the various methods of testing and screening used in the computer program as effective as possible. Even so, they were not 100% effective for many of the prevalent failure modes (bonds and contamination) in components being produced.
5. Contamination material in electronic components (flatpacks and relays) has shown a tendency to move around under fairly severe vibration, but has shown no tendency to float freely when at zero gravity.
6. The gold rich bond failure mechanism is a long time dependent failure mode which was detected during screening tests. The defective bond results from the manufacturing processes and is weak initially, however, a crack develops and propagates with time until the open is detected. The ECDU in the guidance system, which also uses the logic gate, has experienced the same failure mechanism thus adding to the total base of information on this part.

SECTION 8.0
SUMMARY

It is possible now to critique many aspects of the computer and the development program. Some items of interest are the design, the contract arrangement, and the methods that were used to obtain a reliable system.

8.1 DESIGN AND DEVELOPMENT

From the development point of view, the program should have allowed more time for design and design evaluation. As can be seen from the milestone charts of Section 3 the time allowed for the design was very short and requirements were changing during the early development. The design and production evaluations were accomplished in parallel with early production, with the identified changes being phased into production. The initial planning did not allow for changes in design therefore, the implementation of design changes became very difficult, requiring retrofits, and resulting in several configuration changes. The initial program plans (mid 1962) did not allow for a concept of Block design change or any significant configuration change within a design. Program planning in the early part of 1963 indicates the second production system, which included AGC7, was to be flown in the Spring of 1965. Also, in the planning for Block II, during the Spring of 1965, the production prototypes were scheduled to start in August 1965 and to be followed by the flight hardware in December 1965. As a result the design efforts and evaluations of both Block I and Block II overlapped. If block changes in design were planned for initially, the change process may have operated more smoothly.

In retrospect part of the problem could have been diverted by the use of additional manpower and placing more emphasis on design analysis and testing in the design phase. A good example of where more testing could have been valuable was the E Memory vibration fatigue problem that developed during production testing. MIT conducted overstress tests on one memory but should have done more extensive testing on several. It was felt by the design engineers that the cost of the memories and test time did not warrant the benefits, especially after the first one survived excessive stress. This also highlights one of the related problems which is the fact that engineering judgement many times under rates or camouflages a possible problem. A requirement for more extensive design reviews, engineering evaluation and preproduction testing might have exposed some of the problems sooner.

The design reviews conducted during the computer development were of three distinct types. First, the formal design reviews which included the critical Design Review by NASA as well as the MIT design reviews as part of the process for release of the NASA drawings and specifications. Second, the engineering tests and analysis that was conducted in parallel with the design, breadboarding, and production. Third, the various reviews conducted more or less independent of the design team. These reviews were initiated at the request of NASA and were usually called to investigate a particular problem area. Many of the reviews of the second and third category were not well documented. In most cases unofficial reports or memorandums were the only output or results. The formal design reviews culminated in the release of the NASA documentation.

The basic function of the MIT design review was to process all documentation for release to production and process all changes to already released documentation. During the development of the computers the magnitude of the task and the type of activity changed from processing initial releases to processing ease of manufacturing changes that were introduced following the production of several flight computers. Likewise the MIT and support contractors team, which performed the review, fluctuated from design engineers to production engineers. The continuity of the activity was provided by the chairman of the review team which was an MIT responsibility. During all phases of the design review activity, close cooperation between the design team and the production team was essential because design evaluation, breadboarding and prototype production was progressing in parallel. Most of the drawings and designs were generated within MIT but some major documentation tasks and design tasks like the Block II DSKY, power supply and oscillator were accomplished at the industrial support contractors facilities from rather informal end item requirements. Some examples of documentation which was turned over to the support contractors were the module and computer procurement specifications. These documents are strongly dependent upon factory test equipment and procedures therefore it was better for the support contractor to prepare the documents with MIT review to insure the adequacy of the test requirements. The design review provided the integration of the design and documentation efforts and was the focal point for the documentation of design information and requirements.

The second form of design review was conducted on an informal bases and was conducted by and more or less at the discretion of the design engineers. Many design analyses were made on critical areas of the design and in several cases engineering tests were performed to verify the design. The thermal design analysis is an example of one successful analysis which was accomplished early enough in the design to be influential. The thermal tests which were conducted on thermal

models and prototype equipments verified the analysis and design. The E-Memory and Block II DSKY electroluminescent panels are examples where engineering testing or analysis was not adequate to verify the design. In both cases neither the engineering tests nor the qualification tests revealed the marginal condition that existed in the design. Their failure modes were dependent upon the strength of materials and workmanship defects but the probability of occurrence was low enough to make detection by testing one or two samples unlikely. One might say, the major deficiency in the engineering analysis and test part of the design review is that it was not properly staffed and organized in such a way that design deficiencies could be exposed before they became visible in the production phase. One may only hypothesize that a more active design analysis may have eliminated some of the design problems earlier in the program.

The third category of design review was conducted for the most part by groups outside the design team. The designers would participate in the review in order to provide information but in general did not contribute actively in the review. The most notable example of this type of review was documented in an informal publication by AC Electronics in May 1966. This report summarizes the findings of a special review team which was organized at NASA request and spent considerable time at MIT, the support contractors, and spacecraft contractors. The activities of this review team was somewhat limited since most of the Purchase Specifications, Qualification Test Specification, Interface Control Documents, etc. had not been completed and released. In spite of the incomplete documentation the review team performed a very worthwhile function and identified some marginal areas in the design.

Another area that would have benefited by more attention early in the design was the equipment and methods used for the final test of the computer. The computer design had fault detection circuits and selfcheck software but these were designed primarily to indicate faults during the mission phase and not for fault isolation and verification during acceptance testing. When it became known how difficult it was to verify an indication of an anomily with the computer and computer test equipment as designed, additional factory test equipment was developed to aid analyses. It is obvious that more engineering effort devoted to the test problem should have been factored into the design of both the flight hardware and its test equipment. The deficiency in the test cooperative features made both hardware and software checkout time consuming.

8.2 CONTRACTUAL ARRANGEMENTS

The contractual arrangement was difficult to manage with MIT/IL the prime contractor responsible for design and technical direction, and AC Electronics the prime contractor responsible for the production of the hardware with the computer subcontracted through AC to Raytheon Company. In addition the fact that the GN&C was Government furnished equipment to the spacecraft contractors added another dimension to the communication problems during design and development. The contractors responsible for production are naturally interested in producing a good product with a reasonable profit for the company. MIT being responsible for design and having technical direction emphasized the quality of the product. NASA provided the judgement to maintain balance between the extremes of an excellent product and a product manufactured at a reasonable cost. In addition, probing by NASA provided a catalyst which prevented the design and production activities from relaxing the vigilance in searching for solutions to problem areas.

The most difficult part of the contractual arrangement was related to the processing of design changes which had cost or schedule impact. The process was extremely slow in response and in many cases open loop in that the contractual details were processed through a different channel then the documentation for the design changes. This made it difficult for MIT to monitor both the introduction of design changes and the documentation which specified the changes. In many cases the processing through contractual channels would cause changes to some of the technical details which were undesirable. The beneficical aspect of this process is the checks and balances which resulted.

8.3 APPROACHES TO RELIABILITY

In general three approaches to reliability that were used in the APOLLO computer; first, emphasis on the reliability constraint; during design, second, qualify all components, materials, and assemblies; third, rely on process control during the manufacturing in order to maintain uniform production with known quality. The procedures for test and inspection at each level of manufacturing and assembly were designed to provide monitoring for the production process. Some unique methods of testing were implemented such as the Flight Processing Specification (FPS) for qualifying components on a lot by lot basis and the use of acceptance testing procedures with environmental conditions which exceeded the mission environments and in some case exceeded qualification environments. These extremes in test conditions were implemented to screen critical failure modes by temperature cycling for detection of defective interconnections and by vibration for detection of

contamination in of components as well as defective interconnection. The result of the methods used are demonstrated by the success of the computer as indicated in Section 7.5 but there are several other points that can be made.

1. Qualification testing should be designed to determine the failure limits and failure modes rather than just proving the assembly or part can pass the maximum stress levels assumed for the operating system. A single statistical accident, that is, a subsystem passing a level of test, may conceal a serious design deficiency. A system of step stress tests if properly designed should provide better failure limits which can be used to determine the margin of safety between subsystem environment and failure point. In addition, for a lengthy program involving relatively low level production, like APOLLO, reliance on initial qualification and process controls is not sufficient to prevent changes in manufacturing techniques, material suppliers, and manufacturing personnel from impacting the quality of parts and assemblies. As a result, 100% test and inspection techniques like those used on the computer are more important than the qualification program.
2. The effectiveness of the FPS depends heavily on accurate failure analysis, rapid data reporting and motivation of the people. In general the experience with this technique of controlling procurement and component screening was good in that problems were emphasized and corrected at a low level of assembly. This resulted in higher quality parts being used in the fabrication of flight computers than would have been used if only screen and burn-in procedures were followed. It is the visibility into failure modes provided by the analysis and the rejection of lots that brought the problems to the attention of management and provided the motivation for corrective action. Motivation and good management are still essential to realizing success. The FPS procedures attempts to put a measurable control on quality that project management and component vendors can depend upon. Even though the procedures were not 100% effective in screening all failure modes they did make all of them visible early in the production, so that corrective action could be taken.
3. Project management should insure an effective reliability program by continuous support and attention to every detail. Better tools for failure reporting and failure analysis need to be developed in order to make the feedback to management more sensitive and effective. Tight documentation control, material traceability, and process controls are important yet must be flexible enough to allow for changes due to reliability or design type problems. For low level production quantities it might be easier to have a separate documentation package for each system thus providing the flexibility for changes yet providing complete and detailed documentation of each unit. The main

purpose of these documents is to provide complete information to management but the documents and information must be in a form for easy assimilation of the information.

8.4 DESIGN FEATURES

Some of the approaches used in the APOLLO Computer were, and still are, quite controversial. For example, read-only memories, short word length, interrupts to accomplish real time control, etc. were used in the design in order to provide the computer with as much computing capacity as possible within constraints of power, weight, volume, and reliability. Frequently the question is asked whether the designers would take the same approach if they had the opportunity to repeat the design process. The question is difficult since in many cases hindsight does not consider all the constraints under which the original designers were operating. A typical example of this type of problem surrounds the question of whether the word length could have been increased when redesigning for the Block II computer. Using AGC type technology, it is still debatable whether there is enough space within the computer to add another two bits, therefore during the design it would have been necessary to justify the need for more volume in addition to justifying the requirement for the longer word length.

8.4.1 READ-ONLY MEMORIES - ROPES

There are many reasons why the use of read-only memories are desirable. These are both hardware and software related. For example:

1. Much higher density as compared to coincident current core memory (ratio of 5 to 1). One study indicated that if constrained to the AGC envelope and technology, the maximum coincident current memory would have been approximately 9,000 words.
2. Higher reliability as compared with coincident current core memories. In addition, the integrity of the data contents during power sequencing or other transient conditions is guaranteed.
3. If 38,000 words of erasable memory had been used, the total cost to the program for hardware would have been much higher. This point was very difficult to establish during the design phase, but actual cost comparison between the two approaches has varified the difference. The result favors the rope memories even if additional power, size and weight required for the erasable memory are neglected as a cost factor.
4. The inability to change the program without rebuilding one or more modules provides an effective management tool for the control of software changes. It also provides another incentive to make the software error free.

In contrast there are many factors that make the read-only memories undesirable. Probably the two most important are the time lag required by the manufacturing cycle and the necessity of maintaining a manufacturing capability until all mission operations are completely specified and modules manufactured.

8.4.2 THE 16 BIT WORD LENGTH

Early in the development, studies indicated that 16 bits with double precision instructions were sufficient for the computations necessary. The constraints that the word length would place on the memory addressing, instruction codes, and computational speed were not considered critical when compared with an increase in volume, weight and power. An increase in size would probably have been required to guaranty the proposed design could be built within the allocated volume. A larger instruction set and more flexible memory addressing would obviously make the software coding easier but would the difference be adequate justification for the increased hardware? There is no straightforward answer even yet, but for the APOLLO program it seems that minimum hardware was the correct decision.

8.4.3 INTERFACE DESIGN

The most controversial feature of the interface design is the DC isolation for signals and power return. These interfaces violated the requirements of the EMI specifications but again, minimum hardware was considered more important. The use of an interface configuration which utilized high amplitude, low power and low response time signals made it possible to design an interface tolerant to noise that met the intent of the EMI specification with very simple circuits and one wire transmission. To meet the specification verbatim would require DC isolated power supply and interface signal isolation, such that, the signal and signal return current are balanced and transmitted over a twisted pair. It is interesting to note that the only problem with noise in the interfaces was on the transformer coupled signals which were transmitted over shielded twisted pairs (see Section 6.4.1). The reason is, this type of interface was designed for fast response and thus was more susceptible to high frequency noise. While this circuit was more desirable from the EMI specification point of view, the configuration was not sufficient to protect the signal from high frequency, high amplitude induced noise.

8.4.4 GENERAL CAPABILITY

In summarizing the capabilities of the AGC some of the very important features are not obvious. It is easy to state the number of words of memory, number of

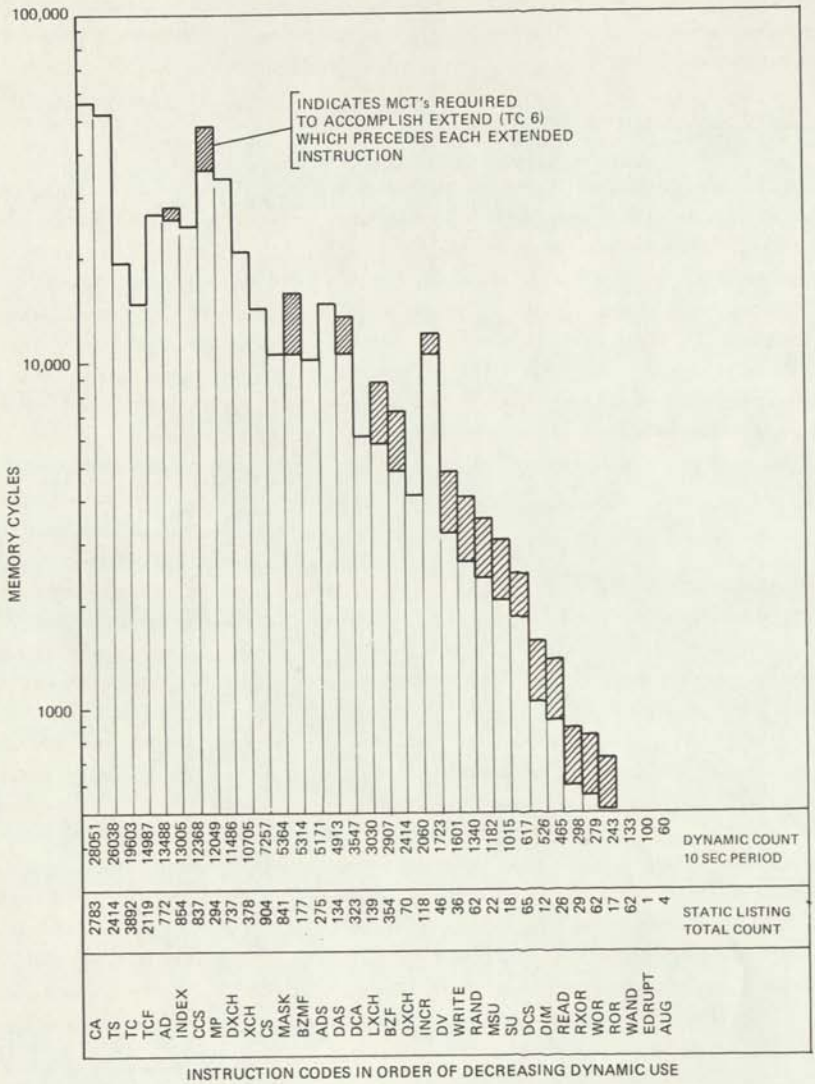


Fig. 8-1 Voluntary Instruction Code Statistics
Program LUMINARY 103

instructions, number of interfaces, volume, weight, etc. More difficult are statements or facts that can be used to evaluate the functional capability which is influenced by the type of instructions, processing speed, memory addressing, interrupts, electronics for restarting, etc. The most significant approach to evaluation of the latter can be accomplished by indicating the usefulness of some of these functions.

The instruction codes both voluntary and involuntary were designed to provide flexibility for the types of computation and control functions needed in the APOLLO mission. These were augmented by an elaborate set of software interpretive instructions which were considerably slower but provided much more powerful arithmetic operations. The voluntary set of instructions although more limited than desirable from the software point of view were chosen with consideration for the computations required of the computer. One way of evaluating the instruction set is to analyze the usage of each instruction during a segment of a mission. Figure 8-1 tabulates the instruction usage statistics for a lunar landing program. A dynamic count is obtained from an all digital simulation of ten seconds of powered descent and the static count is obtained from the assembly listing. Manual inputs (DSKY and hand controller) and uplink were not active during this simulation. Interpretive codes (1700 were processed during this ten second period) are not broken down into the machine instructions therefore the instruction codes resulting from interpretive operations are ignored. The involuntary instructions were designed to process interface signals and to assist programming real time control functions. Table 8-1 lists the I/O and interrupt activity during the same ten second period. It is interesting to note that the time required to process counters consumed about 1% of the total time available for computation.

TABLE 8-1

INVOLUNTARY INSTRUCTION CODE STATISTICS
PROGRAM LUMINARY 103

I/O Channel and Special Register Activity

CDU X	CDU Y	CDU Z	CDU T	CDU S	PIPA X	PIPA Y	PIPA Z	UPLINK
340	356	365	0	0	50	50	50	0
RNRAD	GYROCMD	CDUXCMD	CDUYCMD	CDUZCMD	CDUTCMD	CDUSCMD	THRUST	ALTM
90	24	50	50	50	40	40	5	40
CHAN 4	CHAN 5	CHAN 6	CHAN 7	CHAN 10	CHAN 11	CHAN 12	CHAN 13	CHAN 14
118	270	100	478	131	34	195	656	240
CHAN 15	CHAN 16	CHAN 30	CHAN 31	CHAN 32	CHAN 33	CHAN 34	CHAN 35	
0	0	232	563	115	92	500	500	

Interrupt Activity

T6RUPT	T5RUPT	T3RUPT	T4RUPT	KEYRUPT1	KEYRUPT2	UPRUPT	DOWNRUPT	RADAR	HANDCNTRL
13	100	110	131	0	0	0	500	30	0

Analysis of this data indicates the instruction set is fairly well designed. Instructions like AUG, DIM and EDRUPT are probably less valuable than originally estimated as can be seen from both the dynamic and static usage. Some computation time could be saved if MP was not an extended instruction. Interchanging MASK with MP would be a trade-off between speed and memory size since MASK appears more often in the static listing. Interchanging INCR with MP would limit the address field available to MP. In the light of this analysis, a more ambitious shuffling of instructions would provide only minor improvements in either speed or memory usage.

The built-in failure detection circuits with the software capability to restart has made the difference between success or possible failure of some APOLLO missions. The APOLLO 11 Lunar landing being the prime example (see Section 6.4.5). In the case of the APOLLO 12 lightning a few seconds after launch, the computer experienced a series of V-Fails which resulted in a successful restart. If this capability had not been available, there would have been no mission impact at that point but initialization in orbit could have been difficult. These and other mission experiences has verified the value of the restart capability and has also verified an early assumption that transient failures in digital systems occur more frequently than hard failures.

If it were possible to compare this computer with other aerospace computers available now, the AGC still stands out as being very competitive when size, power, reliability and computational capacity are considered even though the AGC design is 8 to 10 years old.

REFERENCES

1. MIT/CSDL Report E-1877, "Show-Banksum and Final Block I AGC Selfcheck", by E. Smally, November 1965.

2. MIT/CSDL Report E-2065, "Block II AGC Self-Check and Show Banksum", December 1966.

3. MIT/CSDL Report R-395, "Apollo G&N System Reliability Apportionments and Initial Analysis", by G. Mayo and G. Kruszewski, Part IV, Appendix A, Abstracts Section 9, March 1963 (CONFIDENTIAL).

4. MIT/CSDL Report R-457, "Quarterly Reliability and Quality Assurance Progress Report (1 April 1964-30 June 1964), by Group 51 Staff, (CONFIDENTIAL).

5. MIT/CSDL Report E-1679, "Progress Report on Attainable Reliability of Integrated Circuits for Systems Application", by J. Partridge, L.D. Hanley, and E.C. Hall, Part IV, Appendix A, Abstracts Section 9, July 1965.

6. MIT/CSDL Report E-1838, "The Application of Failure Analysis in Procuring and Screening of Integrated Circuits", by L.D. Hanley, J. Partridge, and E.C. Hall, Part IV, Appendix A, Abstracts Section 9, October 1965.

7. MIT/CSDL Report E-1944, "The Impact of the Flight Specifications on Semiconductor Failure Rates", by J. Partridge and L.D. Hanley, Part IV, Appendix A, Abstracts, Section 9, June 1967.

8. MIT/CSDL Report E-2218, "Lead Failure Study for the Motorola 1-MIL Wedge-Bonded 1006323 Transistor", by W. Day and J. Partridge, January 1968.

9. Autonetics Report X5-1399/37, Wescon Session 16B Papers, July 1965.

10. MIT/CSDL Report E-2478, "Apollo Erasable Memory Analyses", by G. Ives and S. Benichasa (to be published by Draper Laboratory).

11. MIT/CSDL Report E-2129, "Keyboard and Display Program and Operation", by A. Green and R. Filene, June 1967.

12. MIT/CSDL Report R-700, Volume V, "The Software Effort", by M.S. Johnson, July 1971.