- Chapter Two -
- Computers On Board The Apollo Spacecraft -

The need for an on-board computer

[28] The Apollo lunar landing program presented a tremendous managerial and technical challenge to NASA. Navigating from the earth to the moon and the need for a certain amount of spacecraft autonomy dictated the use of a computer to assist in solving the navigation, guidance, and flight control problems inherent in such missions. Before President John F. Kennedy publicly committed the United States to a "national goal" of landing a man on the moon, it was necessary to determine the feasibility of guiding a spacecraft to a landing from a quarter of a million miles away. The availability of a capable computer was a key factor in making that determination.

The Instrumentation Laboratory of the Massachusetts Institute of Technology (MIT) had been working on small computers for aerospace use since the late 1950s. Dr. Raymond Alonso designed such a device in 1958-1959[1]. Soon after, Eldon Hall designed a computer for an unmanned mission to photograph Mars and return[2]. That computer could be interfaced with both inertial and optical sensors. In addition, MIT was gaining practical experience as the prime contractor for the guidance system of the Polaris missile. In early 1961, Robert G. Chilton at NASA-Langley Space Center and Milton Trageser at MIT set the basic configuration for the Apollo guidance system[3]. An on-board digital computer was part of the design. The existence of these preliminary studies and the confidence of C. Stark Draper, then director of the Instrumentation Lab that now bears his name, contributed to NASA's belief that the lunar landing program was possible from the guidance standpoint.

The presence of a computer in the Apollo spacecraft was justified for several reasons. Three were given early in the program: (a) to avoid hostile jamming, (b) to prepare for later long-duration (planetary) manned missions, and (c) to prevent saturation of ground stations in the event of multiple missions in space simultaneously[4]. Yet none of these became a primary justification. Rather, it was the reality of physics expressed in the 1.5-second time delay in a signal path from the earth to the moon and back that provided the motivation for a computer in the lunar landing vehicle. With the dangerous landing conditions that were expected, which would require quick decision making and feedback, NASA wanted less reliance on ground-based computing[5]. The choice, later in the program, of the lunar orbit rendezvous method over direct flight to the moon, further justified an on-board computer since the lunar orbit insertion would take place on the far side of the moon, out of contact with the earth[6]. These considerations and the consensus among MIT people that autonomy was desirable ensured the place of a computer in the Apollo vehicle.

Despite the apparent desire for autonomy expressed early in the [29] program, as the mission profile was refined and the realities of building the actual spacecraft and planning for its use became more immediate, the role of the computer changed. The ground computers became the prime determiners of the vehicle's position in three-dimensional space "at all times" (except during maneuvers) in the missions[7]. Planners even decided to calculate the lunar orbit insertion burn on the ground and then transmit the solution to the spacecraft computer, which somewhat negated one of the reasons for having it. Ultimately, the actual Apollo spacecraft was only autonomous in the sense it could return safely to earth without help from the ground[8].

Even with its autonomous role reduced, the Apollo on-board computer system was integrated so fully into the spacecraft that designers called it "the fourth crew member"[9]. Not only did it have navigation functions, but also system management functions governing the guidance and navigation components. It served as the primary source of timing signals for 20 spacecraft systems[10]. The Apollo computer system did not have as long a list of responsibilities as later spacecraft computers, but it still handled a large number of tasks and was the object of constant attention from the crew.

INDEX PAGE

- Chapter Two -
- Computers On Board The Apollo Spacecraft -

MIT chosen as hardware and software contractor

[29] On August 9,1961, NASA contracted with the MIT Instrumentation Lab for the design, development, and construction of the Apollo guidance and navigation system, including software. The project manager for this effort was Milton Trageser, and David Hoag was the technical director[11]. MIT personnel generally agree that they were chosen because their work on Polaris proved that they could handle time, weight, and performance restrictions and because of their previous work in space navigation[12]. In fact, the Polaris team was moved almost intact to Apollo[13]. Despite their experience with aerospace computers, the Apollo project turned out to be a genuine challenge for them. As there were no fixed specifications when the contract was signed, not until late 1962 did MIT have a good idea of Apollo's requirements[14]. One of the MIT people later recalled that

> If the designers had known then [1961] what they learned later, or had a complete set of specifications been available...they would probably have concluded that there was no solution with the technology of the early 1960s[15].

Fortunately, the technology improved, and the concepts of computer Science applied to the problem also advanced as MIT developed the system.

[30] NASA's relationship with MIT also proved to be educational. The Apollo computer system was one of NASA's first real-time, large scale software application contracts[16]. Managing such a project was completely outside the NASA experience. A short time after making the Apollo guidance contract, NASA became involved in developing the on-board software for Gemini (a much smaller and more controllable enterprise) and the software for the Integrated Mission Control Center. Different teams that started within the Space Task Group, later as part of the Manned Spacecraft Center in Houston, managed these projects with little interaction until the mid-1960s, when the two Gemini systems approached successful completion and serious problems remained with the Apollo software. Designers borrowed some concepts to assist the Apollo project. In general, NASA personnel involved with developing the Apollo software were in the same virgin territory as were MIT designers. They were to learn together the principles of software engineering as applied to real-time problems.

INDEX PAGE

- Chapter Two -
- Computers On Board The Apollo Spacecraft -

The Apollo computer systems

[30] The mission profile used in sending a man to the moon went through several iterations in the early 1960s. For a number of reasons, planners rejected the direct flight method of launching from the earth, flying straight to the moon, and landing directly on the surface. Besides the need for an extremely large booster, it would require flawless guidance to land in the selected spot on a moving target a quarter of a million miles away. A spacecraft with a separate lander would segment the guidance problem into manageable portions. First, the entire translunar spacecraft would be placed in earth orbit for a revolution or two to properly prepare to enter an intercept orbit with the moon. Upon arriving near the moon, the spacecraft would enter a lunar orbit. It was easier to target a lunar orbit window than a point on the surface. The lander would then detach and descend to the surface, needing only to guide itself for a relatively short time. Alter completion of the lunar exploration, a part of the lander would return to the spacecraft still in orbit and transfer crew and surface samples, after which the command module (CM) would leave for earth.

With a lunar orbit rendezvous mission, more than one computer would be required, since both the CM and the lunar excursion module (LEM) needed on-board computers for the guidance and navigation function. The CM's computer would handle the translunar and transearth navigation and the LEM's would provide for autonomous landing, ascent, and rendezvous guidance.

NASA referred to this system with its two computers, identical in design but with different software, as the Primary Guidance, [31] Navigation, and Control System (PGNCS pronounced "pings"). The LEM had an additional computer as part of the Abort Guidance System (AGS), according to the NASA requirement that a first failure should not jeopardize the crew. Ground systems backed up the CM computer and its associated guidance system so that if the CM system failed, the spacecraft could be guided manually based on data transmitted from the ground. If contact with the ground were lost, the CM system had autonomous return capability. Since the lunar landing did not allow the ground to act as an effective backup, the LEM had the AGS to provide backup ascent and rendezvous guidance. If the PGNCS failed during descent, the AGS would abort to lunar orbit and assist in rendezvous with the CM. It would not be capable of providing landing assistance except to monitor the performance of the PGNCS. Therefore the computer systems on the Apollo spacecraft consisted of three processors, two as part of the PGNCS and one as part of the AGS.

INDEX PAGE

Computers in Spaceflight: The NASA Experience

- Chapter Two -
- Computers On Board The Apollo Spacecraft -

Evolution of the hardware: Old technology versus new block I and Block I designs

[31] The computer envisioned by MIT's preliminary design team in 1961 was a shadow of what actually flew to the moon in 1969. There always seem to be enough deficiencies in a final product that the designers wish they had a second chance. In some ways the Apollo guidance computer was a second chance for the MIT team since most worked on the Polaris computer. That was MIT's most ambitious attempt at an "embedded computer system," a computer that is intrinsic to a larger component, such as a guidance system. Although the Apollo computer started out to be quite similar to Polaris, it evolved into something very different. The Apollo guidance computer had two flight versions: Block I and Block II. Block I was basically the same technology as the Polaris system. Block II incorporated new technology within the original architecture.

Several factors led from the Block I design to Block II. NASA's challenges to the MIT contract and the decision to use the rendezvous method instead of a direct ascent to the moon were decisive. A third factor related to reliability. Finally, the benefits of the new technology influenced the decision to make Block II.

Before NASA let the contract to MIT, but after it was known that the Instrumentation Laboratory would be accorded "sole source" status, several NASA individuals began studying ways to consolidate flight computer development. In June 1961, Harry J. Goett of Goddard Space Flight Center recommended that the computers needed for the Orbiting Astronomical Observatory (OAO), Apollo, and the Saturn launch vehicle be the same. He cited an IBM proposal for $5 [32] million to do just that[17]. On the same day Goett's recommendation, RCA proposed the use of a 420-cubic-inch computer with only an 80- watt power consumption and 24-bit word size as a general-purpose spaceborne computer[18] . This proposal got nowhere and NASA's Robert G. Chilton challenged Goett's idea, showing that the expected savings would not materialize. Even though the projected cost of the Apollo computer would decrease to $8 million from $10 million, the OAO development costs would rise from $1.5 million to $5 million[19]. Ironically, in the same month, Ramon Alonso from MIT met with Marshall Space Flight Center personnel about the use of the Apollo computer in the Saturn[20]. Although MIT got the Apollo contract and IBM got the contract for the Saturn computer, the idea of a duplicate system did not die. Two years later, when the deficiencies of the Polaris-based system were obvious and the solutions offered by the new technology of the Block II version still unproved, David W. Gilbert, NASA manager for Apollo guidance and control, proposed replacing the MIT machine with the one IBM was building for Saturn[21]. It did not occur because Gilbert wanted NASA to accept the reprogramming costs, and the existing configuration of the IBM computer would not fit in the space allotted for it in the CM. Nevertheless, MIT would still have to deal with NASA misgivings about the hardware design as late as May 1964, when Maj. Gen. Samuel C. Phillips, deputy director of the Apollo Program, reported on a meeting to discuss the use of the triple modular redundant Saturn launch vehicle computer in Apollo [22].

The decision to have a separate CM and the LEM influenced the transition to Block II by providing a convenient dividing point in the Apollo program. The early Apollo development flights were to use the CM only. Later flights would include the LEM. Since Block I design and production had already proceeded, planners decided to use the existing Block I in the unmanned and early manned development flights (all relatively simple earth-orbital missions) and to switch to Block II for the more complex combined CM-LEM missions [23].

Reliability was another force behind Block II. Daring early planning for the guidance system, redundancy was considered a solution to the basic reliability problem. Designers thought that two computers would be needed to provide the necessary backup; however, they dropped this scheme for two reasons. The ground had primary responsibility for determining the slate vector (the position of the craft in threedimensional space) in translunar, lunar orbit, and transearth flight[24]. Moreover, none of the variations of the two-computer or other redundancy schemes could meet the power, weight, and size requirements[25]. One way to provide some measure of protection is to make the computer repairable in flight. The Block I design, due to its modularity, could be fixed during a mission that carried appropriate spares. At any rate, its predicted mean time between failures (MTBF) [33] was 4,200 hours, about 20 times longer than the longest projected mission [26]. But Block I's repair capability became a negative factor when sealing the computer began to be considered more important to reliability than the ability to repair it[27]. Aside from packaging, overall malfunction detection was improved in the Block II design, further increasing reliability[28].

The most important reason for going to Block II was the availability of new technology. The Block I design used core transistor logic. It had several disadvantages:

- It could not be complemented, a very important basic operation in computer arithmetic that changes a one to a zero or vice versa.
- It had the characteristic of "destructive readout," in which a datum read from a flip-flop using core transistor logic loses the datum; that forces the inclusion of a circuit to rewrite the datum if it is to be retained after the read cycle.
- Memory cycle time could not be fixed: in Block I it was an average of 19.5 milliseconds, which was quite slow for computers at the time, and the varying cycle caused timing problems within the machine[29].

These disadvantages led MIT to begin studying, as early as 1962, the possible use of integrated circuits (ICs) to replace core transistor circuits. ICs, so ubiquitous today, were only 3 years old then and thus had little reliability history. It was therefore difficult to consider their use in a manned spacecraft without convincing NASA that the advantages far outweighed the risks.

To accomplish this, the MIT team chose a direct-coupled transistor logic (DCTL) NOR gate with a three-input element,[30] consisting of three transistors and four resistors. NOR logic inverts the results of applying a Boolean OR operation to the three inputs. It took nearly 5,000 of these simple circuits to build an Apollo computer. Using a variety of circuits would have simplified the design since the component count would have been reduced, but by using the NOR alone, overall simplicity and reliability increased[31]. Also, the time it took the machine to cycle became fixed at 11.7 milliseconds, a double bonus in that speed increased and cycle time was consistent[32].

Aside from these advantages, MIT believed that the lead time to the first flight would permit reliability to be established and the cost of the ICs to come down[33]. At the time, the production of such circuits low, and they were more expensive than building core transistor circuits. To place the production rate in perspective, MIT chose the NOR ICs in the fall of 1962 and by the summer of 1963, 60% of the [34] total U.S. output of microcircuits was being used in Apollo prototype construction[34]. This is one of the few cases in which NASA's requirements acted as a direct spur to the computer industry. When MIT switched to ICs, it kept the Apollo computer as "state of the art" at least during its design stage. It would be hopelessly outdated technologically by the time of the lunar landing 7 years later, but in 1962, using the new microcircuits seemed to be a risk. This view is contested by one member of the MIT team, who later said that the decision "wasn't bold; it was just the easy thing to do to get the size and power and other requirements"[35].

With the ICs fully incorporated in the Apollo computer and the transition from Block I to Block II complete, NASA possessed a machine that was more up to date technologically. It had double the memory of the largest Block I, more I/O capability, was smaller, and required less power[36]. Besides, it was also more reliable, which was, as always, the major consideration.

INDEX PAGE

Computers in Spaceflight: The NASA Experience

- Chapter Two -
- Computers On Board The Apollo Spacecraft -

The Apollo guidance computer: Hardware

[34] The Apollo Guidance Computer was fairly compact for a computer of its time. The CM housed the computer in a lower equipment bay, near the navigator's station. Block II measured 24 by 12.5 by 6 inches, weighed 70.1 pounds, and required 70 watts at 28 volts DC. The machine in the lunar module was identical.

Crew members could communicate with either computer using display and keyboard units (DSKY, pronounced "disky"). Two DSKYs were in the CM, one on the main control panel and one near the optical instruments at the navigator's station. In addition, a "mark" button was at the navigator's station to signal the computer when a star fix was being taken. A single DSKY was in the lunar module. The DSKYs were 8 by 8 by 7 inches and weighed 17.5 pounds. As well as the DSKYs, the computer directly hooked to the inertial measurement unit and, in the CM, to the optical units.

The choice of a 1 6-bit word size was a careful one. Many scientific computers of the time used 24-bit or longer word lengths and, in general, the longer the word the better the precision of the calculations. MIT considered the following factors in deciding the word length: (a) precision desired for navigation variables, (b) range of input variables, and (c) the instruction word format[37]. Advantages of a shorter word are simpler circuits and higher speeds, and greater precision could be obtained by using multiple words[38]. A single precision word of data consisted of 14 bits, with the other 2 bits as a sign bit (with a one indicating negative) and a parity bit (odd parity). Two [35] adjacent words yielded "double precision" and three adjacent, "triple precision." To store a three-dimensional vector required three double precision words[39] . Data storage was as fractions (all numbers were less than one)[40]. An instruction word used bits 15-13 (they were numbered descending left to right) as an octal operation code. The address used bits 12-1. Direct addressing was limited, so a "bank register" scheme (discussed below) existed to make it possible to address the entire memory [41].

The Apollo computer had a simple packaging system. The computer circuits were in two trays consisting of 24 modules. Each module had two groups of 60 flat packs with 72-pin connectors. The flatpacks each held two logic gates[42]. Tray A held the logic circuits, interfaces, and the power supply, and tray B had the memory, memory electronics, analog alarm devices, and the clock, which had a speed of one megahertz[43]. All units of the computer were hermetically sealed[44]. The memory in Block II consisted of a segment of erasable core and six modules of core rope fixed memory. Both types are discussed fully below.

The Apollo computer used few flip-flop registers due to size and weight considerations[45], but seven key registers in the computer did use flip-flops:

- The accumulator, register 00000, referenced as "A".
- The lower accumulator, 000001, L".
- The return address register, 000002, "Q".
- The erasable bank register, 000003, "EB".
- The fixed bank register, 000004, "FB".
- The next address, 000005, "Z".
- The both bank register, 000006, "BB" (data stored in EB and FB were automatically together here)[46].

The use of bank registers enabled all of the machine's memory to be addressed. The largest number that can be contained in 12 bits is 8,192. The fixed memory of the Apollo computer contained over four times that many locations. Therefore, the memory divided into "banks" of core, and the addressing could be handled by first indicating which bank and then the address within the bank. For example, taking the metaphor "address" literally, there are probably hundreds of "100 Main Street" addresses in any state, but by putting the appropriate city on an envelope, a letter can be delivered to the intended 100 Main Street without difficulty.

The computer banks were like the cities of the analogy. The erasable bank register held just 3 bits that were used to extend the direct [36] addressing of the erasable memory to its "upper" region, and the fixed bank register held 5 bits to indicate which core rope bank to address. In addition, for the addresses needing a total of 16 bits, a "super bank bit" could be stored and concatenated to the fixed bank data and the address bits in the instruction word[47]. This scheme made it possible to handle the addressing using a 1 6-bit word, but it placed a greater burden on the programmers, who, in an environment short of adequate tools, had to attend to setting various bit codes in the instructions to indicate the use of the erasable bank, fixed bank, or super bank bit. Although this simplified the hardware, it increased the complexity of the software, an indication that the importance of the software was not fully recognized by the designers.

To further reduce size and weight, the Apollo computer was designed with a single adder circuit, which the computer used to update incremental inputs, advance the next address register, modify specified addresses, and do all the arithmetic[48]. The adder and the 16 I/O channels were probably the busiest circuits in the machine.

Memory

The story of memory in the Apollo computer is a story of increasing size as mission requirements developed. In designing or purchasing a computer system for a specific application, the requirements for memory are among the most difficult to estimate. NASA and its computer contractors have been consistently unable to make adequate judgments in this area. Apollo's computer had both permanent and erasable memory, which grew rapidly over initial projections.

Apollo's computer used erasable merry cells to store intermediate results of calculations, data such as the location of the spacecraft, or as registers for logic operations. In Apollo, they also contained the data and routines needed to ready the computer for use when it was first turned on. Fixed memory contained programs that did not need to be changed during the course of a mission. The cycle times of the computer's memories were equal for simplicity of operation[49].

MIT's original design called for just 4K words of fixed memory and 256 words of erasable (at the time, two computers for redundancy were still under consideration)[50]. By June 1963, the figures had grown to 10K of fixed and 1K of erasable[51]. The next jump was to 12K of fixed, with MIT still insisting that the memory requirement for an autonomous lunar mission could be kept under 16K [52]! Fixed memory leapt to 24K and then finally to 36K words, and erasable memory had a final configuration of 2K words.

Lack of memory caused constant and considerable software [37] development problems, despite the increase of fixed memory 18 times over original estimates and erasable memory 16 times. Part of the software difficulties stemmed from functions and features that had to be dropped because of program size considerations, and part because of the already described addressing difficulties. If the original designers had known that so much memory would be needed, they might not have chosen the short word size, as a 24-bit word could easily directly address a 36K bank, with enough room for a healthy list of instruction codes.

One reason the designers underestimated the memory requirements was that NASA did not provide them with detailed specifications as to the function of the computer. NASA had established a need for the machine and had determined its general tasks, and MIT received a contract based on only a short, very general requirements statement in the request for band. The requirements started changing immediately and continued to change throughout the program. Software was not considered a driving factor in the hardware design, and the hardware requirements were, at any rate, insufficient.

The actual composition of the memory was fairly standard in its erasable component but somewhat unique in its fixed component. The erasable memory consisted of coincident-current ferrite cores similar to those on the Gemini computer, and the fixed memory consisted of core rope, a high-density read-only memory using cores of similar material composition as the erasable memory but of completely different design. MIT adopted the use of core rope in the original Mars probe computer design and carried it over to the Apollo[53]. Chief advantage of the core rope was that it could put more information in less space, with the attendant disadvantages that it was difficult to manufacture and the data stored in it were unchangeable once it left the factory (see Box 2-1).

---

[38] Box 2-1: Core Rope: A Unique Data Storage Device

Each core in an erasable memory could store one bit of information, and each core in the core rope fixed memory could store four words of information. In the erasable memory, cores are magnetized either clockwise or counterclockwise, thus indicating the storage of either a one or a zero. In fixed memory, each core functions as a miniature transformer, and up to 64 wires (four sets of 1 6-bit words) could he connected to each core. If a wire passed through a particular core, a one would be read. If a particular wire bypassed the core, a zero would he read. For example, to store the data word 10010001000011 1 1 in a core, the first, fourth, eighth, and thirteenth through sixteenth wires would pass through that core, the rest would bypass it. A 2-bit select code would identify which of the four words on a core was being read, and the indicated 16 bits would be sent to the appropriate register[54]. In this way. up to 2~000 bits could he stored in a cubic inch[55].

The computer contained core rope arranged in six modules, and each module contained 6,144 1 6-bit words[56]. The modules further divided into "banks" of 1,024 words The first two banks were called the "fixed-fixed memory" and could he directly addressed by 12 bits in a instruction word. The remaining 34 were addressable as described in the text, using the 5-bit contents of the fixed bank register and the 10 bits in a instruction word[57].

The use of core rope constrained NASA's software developers. Software to be stored on core rope had to be delivered months before a scheduled mission so that the rope could be properly manufactured and tested. Once manufactured, it could not be altered easily since each sealed module required rewiring to change bits. The software not only had to be finished long in advance, but it had to be perfect.

Even though common sense indicates that it is advantageous to complete something as complex and important as software long before a mission so that it can be used in simulators and tested in various other ways, software is rarely either on time or perfect. Fortunately for the Apollo program, the nature of core rope put a substantial amount of pressure on MIT's programmers to do it right the first time. Unfortunately, the concept of "bug"-free software was alien to most programmers of that era. Programming was a fully iterative process of removing errors. Even so, many "bugs" would carry over into a delivered product due to unsophisticated testing techniques. Errors found before a particular system of rope was complete could be fixed at the factory[58], but most others had to be endured. Raytheon, the subcontractor that built the ropes, could eliminate hardwiring errors introduced during manufacture by testing the rope modules against the [39] delivery tape of the programs. The company built a device to do this[59].

(MISSING FIGURE)
Figure 2-1. This diagram shows the principle behind core rope. Suppose that the data shown above the cores in the drawing is to be stored in the specific core. Thus 1000 is stored in the first core on the left by attaching the top wire from the select circuit to the core and bypassing it with the next three wires. When that core is selected for reading, the wire attached to the core will indicate a "one" because all cores in a rope are permanently charged as ones; the wires bypassing the core will indicate zeroes.

Production Problems and Testing

Development and production of the Apollo guidance, navigation, and control system reflected the overall speed of the Apollo program. Design of the system began in the second quarter of 1961, and NASA installed a Block I version in a spacecraft on September 22, 1965. Release of the original software (named CORONA) was in January 1966, with the first flight on August 25, 1966 [60]. Less than 3 years after that, designers achieved the final program objective. Even though fewer than two dozen spacecraft flew, NASA authorized the building of 75 computers and 138 DSKYs. Fifty-seven of the computers and 102 of the crew interfaces were of the Block II design [61]. This represents a considerable production for a special-purpose computer of the type used in Apollo. The need to quickly build high-quality, high-reliability computers taxed the abilities of Raytheon.

Through AC Electronic Circuits (contractor for the entire guidance system), Raytheon was chosen to build the computers MIT had designed largely because of its Polaris experience, but it had [40] never built a computer as complex as the one for Apollo. The Polaris machine was much simpler. Despite the use of experienced Polaris personnel, Raytheon's production division for the Apollo computer went from 800 to 2,000 employees in a year's time in order to handle the increased difficulties and speed of production[62].

Rapid growth, underestimation of production requirements, and reliability problems dogged Raytheon throughout the program. Changes in design made by MIT in late 1962 caused the company its initial trouble. The original request for proposal had featured Polaris techniques, so Raytheon bid low, expecting to use the same tools and production line for the Apollo machine. The changes in component types and memory size caused cost estimates to nearly double, resulting in considerable friction with NASA[63]. NASA was also worried when two computers and fully 50% of the Block I DSKYs failed vibration tests[64]. These failures turned out to be largely caused by contaminated flat packs and DSKY relays. Particles would shake loose during vibration testing[65]. The Block II computers would not work at first due to excessive signal propagation time in the micrologic interconnection matrix. The solution was to switch from nickel ribbon connectors to a circuit board, causing an increase of $500,000 in production costs[66].

These sorts of problems caused the Manned Spacecraft Center to authorize a complete design review of the AGC in February 1966. The lack of adequate support documentation was found to be the most significant fault of the Block II computer[67]. This sort of problem is usually the result of speeding up development to the point at which changes are not adequately documented.

Continuous and careful attention to reliability led to the discovery of problems. Builders flight-screened components lot by lot[68]. Post-production hardware tests included vibration, shock, acceleration, temperature, vacuum, humidity, salt fog, and electronic noise.[69] As D.C. Fraser, an engineer on the project, later remarked, "reliability of the Apollo computer was bought with money" [70].
