# PART VI

# GUIDANCE COMPUTER DESIGN

by

Dr. Albert Hopkins

DR. ALBERT L. HOPKINS, JR.

Assistant Director, Instrumentation Laboratory
Massachusetts Institute of Technology

Dr. Albert L. Hopkins, Jr., Assistant Director of Instrumentation Laboratory, Massachusetts Institute of Technology, is a specialist in the development of computers for inertial guidance and navigation systems.

In his present assignment in the Laboratory's Digital Development Group, Dr. Hopkins is responsible for design of memory circuits and logic employed in the guidance system computer for the Project Apollo spacecraft. The computer uses micro-circuitry and represents an advanced state of the art of miniaturization. The Apollo guidance computer would fit into a good size suitcase, but in computational capacity it is the equivalent of M. I. T.'s Whirlwind computer of the late 1940's. Whirlwind circuit racks filled an entire room.

Dr. Hopkins was born in Chicago, Ill., May 6, 1931, and was graduated from Exeter Academy, Exeter, N.H., in 1949. He attended Harvard University where he received the A. B. degree in engineering sciences in 1953, the A. M. degree in applied mathematics in 1954 and the Ph. D. degree in applied mathematics in 1957. He was Instructor in Control Systems Engineering in the Harvard Division of Engineering and Applied Physics from 1957-60 under Professor Howard Aiken.

Dr. Hopkins joined Instrumentation's Digital Development Group in 1960.

Part VI

# GUIDANCE COMPUTER DESIGN

## INTRODUCTION

Some 22 years have elapsed since the first digital computer was completed. For the past several years, digital guidance computers have flown in airplanes, missiles, and rockets. Some of these vehicles are scarcely bigger than an early computer, whose performance is surpassed by the guidance computers they carry.

The first general-purpose computer and the first high speed electronic number processor were each put into use in about 1943. Numbers were stored in relay controlled counter wheels and in vacuum tube ring counters. The next few years brought the high speed general-purpose computer, the acoustic delay line, and the electrostatic storage tube. In the early 1950's higher densities were achieved in logic by the use of semiconductor diodes, and in memory by the introduction of magnetic cores, drums, and tapes. The transistor was developed in 1948, and was employed in some experimental small computers intended as prototype airborne computers in the early 1950's. Transistors began to be employed in large-scale computers in the late 1950's, beginning with computers for military applications. Only now are semiconductor components beginning to match the enormous density which was achieved in magnetic memories ten years ago. In today's guidance computers, we are realizing an overall density thousands of times greater than in computers of fifteen years ago. Part of this difference is due to advances in mechanical design which have been made purely because of the extreme importance of weight and volume in airborne applications. In cases where size is not an important factor, densities are lower by an order of magnitude.

Performance has been increased over the years by advances in logical design as well as in component size and speed. Early contributions of the number bus, binary arithmetic, and common storage have been followed by such improvements as methods of fast arithmetic, indexing and other address modification schemes, multiprogramming, and program interrupt.

Progress in mathematical areas has been a significant factor in our ability to employ digital computers in airborne guidance at this stage of their development. Computer programming developments have given us automatic programming and internal "software" routines such as executive control and interpretive programs. Recent

efforts have produced numerical methods of celestial mechanics which do not overtax the limited resources of today's guidance computers; and sampled-data theory has given rise to methods of stable control of unstable vehicles using digital techniques. Vast amounts of programming and analysis are needed before today's missions become tractable for existing computer performance. It is the purpose of the remainder of this essay to discuss the present state of the art in logical design, hardware, and software, particularly as it is applied in the Apollo Guidance Computer.

The reasons for having an airborne computer are so numerous that the computer engineer is apt to take the parochial viewpoint that the guidance computer is the principal part of the guidance system, and that the other parts, such as inertial, optical, radar, and radio elements are ancillary units for sensing and communication. This picture of a guidance system is somewhat distorted, but not entirely so. In 1962, J. F. Shea, then Deputy Director of Manned Space Flight at NASA, and presently Apollo Project Manager, said, "Although engineers in each discipline tend to regard their particular developments as the most critical, once the propulsion capability has been provided the key to reliable execution of a wide range of complex, long-duration missions is the computational capacity provided aboard the spacecraft."(1)

The Apollo Guidance Computer will be incorporated into the Guidance and Navigation Systems in the Command and Lunar Excursion Modules. Its functions are to aid in operating the inertial and optical subsystems, to provide steering signals where human reaction is too slow, to perform spacecraft attitude control with minimum fuel expenditure, to maintain timing references, to communicate with the Astronauts via display lights and keyboard, to communicate with ground tracking stations via digital data links, and to perform the calculations necessary to deduce position and velocity relative to the earth and the moon from the input data available during all flight phases from boost through lunar landing and rendezvous to final entry and landing.

Chapter VI-1

# CHARACTERISTICS OF GUIDANCE COMPUTERS

## GENERAL

Guidance computers are designed to meet a number of severe constraints. The extent to which the constraints are met depends upon the ingenuity of the designers; but to judge from comparisons among existing computers, it depends far more strongly on the limitations of available hardware. The more significant differences in performance among computers can be traced to the degree to which their designers are willing to commit themselves to advanced technology, and processes previously untried. Today, the complex microcircuit and the multi-layer etched board are areas in which some computer makers are attempting to attain significant advantages, while others wait and watch to see the inevitable problems arise and become solved before venturing.

Requirements for guidance computers are extreme reliability, low weight and power consumption, high performance in terms of mathematical answers per second and inputs and outputs serviced, and flexibility to grow with the scope of the mission. Although immense achievements have been made within the last few years, the present generation of guidance computers rapidly becomes obsolete; for missions already in the planning stages call for much greater achievements in design, programming, production, test, and repair than have so far been realized.

Some feeling for the range of guidance computer characteristics may be obtained from the tabulation in Table VI-1 of published data [1, 2, 3] on computers designed within the past few years. Comparisons are often misleading, especially if one is trying to prove superiority of one computer over another. It is reasonable and often necessary to choose among computers for a specific application, but it is not easy, for subtle differences can be of great importance. It is less difficult and more valid to draw conclusions about the similarities of various computers from a chart of comparative characteristics. Size and weight data have been omitted here because they tend to be particularly misleading in the absence of knowledge of the particular input and output configuration of the computers. This, in turn, is hard to present because of its detailed nature in some cases and its scanty description in others. Sizes range from 0.2 to 2 cubic feet (0.005 to 0.05 cubic meters), and densities are close to that of water.

| | Serial or Parallel | Negative Numbers | Word Length Bits | Number of Operations | Addition Time | Multi-plication Time | Power Consumption | Bits of Memory |
|---|---|---|---|---|---|---|---|---|
| Univac type 1824 | Parallel | 2's complement | 24 | 41 | 8 $\mu$sec | 64 $\mu$sec | 110 watts | $2 \times 10^5$ |
| AC Spark Plug "Magic" | Serial | 2's complement | 24 | 16 (?) | 70 | 258 | 90 | $1 \times 10^5$ |
| Burroughs D-210 | Parallel | 2's complement | 24 | 32 | 30 | 570 | 1-100 | $7 \times 10^4$ |
| Arma Micro Computer | Serial | 2's complement | 22 | 19 | 27 | 135 | 50 | $5 \times 10^4$ |
| IBM Saturn V Computer | Serial | 2's complement | 28 | 18 | 82 | 328 | 131 | $5 \times 10^5$ |
| Autonetics D26C | Parallel | 2's complement | 30 | 100 | 6 | 18 | 192 | $3 \times 10^5$ |
| MIT/IL Apollo Computer | Parallel | 1's complement | 16 | 34 | 24 | 48 | 90 | $6 \times 10^5$ |

Table VI-1 Selected Characteristics of Several Guidance Computers

## LOGICAL DESIGN[4, 5]

Word Length - It is desirable to minimize word length in a guidance computer. Memory sense amplifiers, being high-gain class A amplifiers, are considerably harder to operate with wide margins (of temperature, voltages, input signal) than, for example, circuits made of NOR gates. Memory digit drivers are also critical circuits whose number is equal to the number of bits in a word. Similarly, the time required for carry propagation in a parallel adder or for circulation in a serial machine is proportional to word length, and, moreover, the very size of a computer is dependent on word length.

Factors which discourage the minimization of word length are the numbers of bits required for data words, input and output variables, and instruction words. These numbers are functions of mission requirements and details of logical design. Most guidance computers have word lengths of around 24 bits. The Apollo Guidance Computer is unique among those listed in having 16 bits (of which one is a parity check bit). As explained later, the difference is due largely to a decision to use multiple-precision arithmetic for variables concerned with guidance and navigation. Even the longest word in the list (30 bits) is short by comparison to the large scale computer installations, where size is not of as great concern as are speed and programming ease.

Instruction Repertoire - The implicit requirements for any Von Neumann-type computer demand that facilities exist for:

A. Fetching from memory
B. Storing in memory
C. Negating (complementing)
D. Combining two operands (e.g., addition)
E. Address modification
F. Normal sequencing (specifying the location of the next instruction)
G. Conditional sequence changing.

A single instruction can provide several of these facilities, so that a very limited repertoire is possible,[6] although a large burden is thereby placed on program storage, and speed is limited. For a relatively small additional cost in complexity, a more comfortable repertoire is obtained. An operation set of eight instructions can provide flexibility without sacrificing simplicity. All of the computers listed go beyond this, however, and in general it is done to obtain speed at a cost in hardware. In some instances, the taking of square roots and the conversion of numbers between binary and decimal appear as single instructions. More commonly, the instruction sets contain convenient data handling, branching, and arithmetic operations with from about $2^4$ to $2^5$ codes.

Speed - It is well known that the overall speed of a computer can be enhanced by its logical design, usually at an equipment cost. This may take the form of having separate adders in a parallel machine for indexing and arithmetic; or it may consist of providing circuits to speed up multiplication by processing several multiplier bits at a time. Alternatively, speed can be obtained by providing single instructions which perform complex jobs such as the two mentioned in the preceding paragraph. Speed is important in guidance computers, and logical complexities are employed in order to gain speed in virtually every guidance computer design; but size and reliability restrictions are of sufficient importance to limit the number and extent of such complexities. In data processing computers, where size is less important and where speed is a competitive issue, logic circuits are employed somewhat into the area of diminishing returns. Guidance computers, as a result, are generally slower than their ground based relatives.

Input and Output - Guidance computers, and control computers in general, differ from data processors most significantly in the area of input and output. Modern data processors generally communicate with peripheral equipment which is complex and sophisticated enough to send and receive data over parallel channels without the computer having to spend much time overseeing the process. In some cases the computer sends data to a remote buffer register upon receipt of an indication that the remote unit is ready. In other cases the remote unit interrogates the computer memory as often as necessary, thus eliminating the buffer. In guidance computers, however, the input and output are not generally exchanged with such sophisticated machines. Owing partly to the non-digital nature of such electromechanical machines as inertial measurement units and rocket steering servos and partly to the strong desire to keep interface circuits and cables as small as possible, the guidance computer spends a substantial part of its time (or equipment) budget on maintaining communication with these units.

Another interesting contrast exists between data processors and guidance computers. The former are designed to spread a work load out over a period of time to achieve a good balance between internal computing and input-output activity. One figure of merit of a data processing installation is the degree to which it can keep its various facilities busy by time-sharing them among various independent users. If a large demand occurs for time on a printer, for example, the results to be printed will be buffered on a magnetic tape to be printed later when the facility is available. In a guidance computer, the central processor is time-shared among numerous jobs, but the allowable delays in reacting to a large demand for input-output service are measured in milliseconds rather than minutes; and the logical design of computers and systems must reflect this fact.

Fault Diagnosis - Another area of interesting contrast between data processors and guidance computers is in the area of self-checking, or fault diagnosis. Since time on a large computer is valued at hundreds of dollars per hour, it is economically necessary to locate and correct faults very rapidly. For this reason, modern computers are equipped with circuits whose function is to make fault location nearly automatic.

Guidance computers cannot afford to carry extra hardware for this purpose. It is important, however, to be able to detect that an error has occurred in flight so that the proper course of action may be taken. This action might be to switch to a back-up computer or other means of control, or possibly it may mean that a missile must be destroyed in order that it not stray far off course. The most common means of fault detection is by a programmed self-check which is run at all times when the computer is not otherwise occupied. More refined checking may be done by a limited amount of circuitry. For example, some guidance computers employ a parity test on the contents of memory. Still other types of alarms are included in the Apollo Guidance Computer, including tests for prolonged or insufficient interrupt activity and various sorts of program freezes. The sum total of these checks and alarms reduces to a small value the probability that a malfunction shall go undetected.

## HARDWARE

Memory Devices - The ferrite coincident current core memory is the cornerstone of computer technology, providing fast random access at a few cents per bit in the megabit range. Thin film memories have had a large research investment and have surpassed core memories in speed and bit density by little or none at all. Their cost is relatively high and their capacity is more limited than core. Plated wire promises to be a substantial improvement, but is not yet advanced enough to be producible or reliable in data processing or guidance applications.

Core memory is clearly ahead of thin film in data processing applications. The matter is controversial with respect to guidance computers, where the higher cost and the capacity limitations of film are less important. Film offers somewhat higher speed, where core offers the economy of coincident selection plus a large output signal. Density and reliability are unresolved issues between the two.

High capacity electromechanical memories such as drums and discs are disappearing from guidance computer use. This is so for three reasons: a substantial increase in packaging density of core and film memories, the serial access nature of discs and drums, and the limited time that discs and drums can operate without maintenance. The high bit densities and large capacities attainable with electromechanical memories make them virtually indispensable to large scale data processing installations, however.

Fixed memory is not used in data processing to any large degree except as a means of implementing internal machine logic such as in a program sequence generator. Its broader use in guidance computers stems from its potential for indestructibility and high density. Indestructibility is a two-edged sword. It requires that program and data be determined well in advance of use; moreover it places a limitation on changes in mission plan such as may be required periodically in ballistic missile applications. Wherever these limitations are not overly constraining, a fixed memory offers assurance that the computer program is identical through all phases of testing and in flight. It moreover permits recovery from temporary malfunctions which would alter the contents of an erasable memory.

Some types of memories compromise between reliability and unchangability by having the ability to be electrically alterable. Modifications of film and core memories have this property, although coincident selection is less apt to be possible in the core versions, most of which are relatives of transfluxors. The bit densities of such memories have been well below those which are available in permanent memories.

Logic Devices - In the past few years integrated circuits, or microcircuits, have been adopted nearly universally by guidance computer designers for at least the logic portion of their computer designs. Prior to the advent of microcircuits, magnetic cores were strong contenders as logic elements against all-transistor circuitry. Core circuits were no larger, and in addition were capable of operating on substantially lower power than all-transistor circuits. Although special applications may yet exist which favor the magnetic core, the very small size of microcircuits and their high speed and proven reliability make them preferred in nearly all instances over cores. With each passing year, moreover, the power consumption of new microcircuit logic devices has been substantially reduced. One can now expect to consume less power with microcircuits than with cores at full speed. The latter elements still retain the advantage of reduced power consumption at low speed operation.

Data processing machines are only now beginning to use microcircuit techniques, because of numerous problems which have attended the large scale production of microcircuits. If these problems are solved, we may expect to see the same increase in the ratio of performance to size in the logic area of data processors which was seen in guidance computers a few years ago.

The primary area in which advances need to be made is in interconnection of logic units.[7] A poor job of mechanical design results in unreliable connections or low component density or both; yet it has so far proven quite difficult to arrive at a structure which is satisfactory in all respects. Some of the essential and important requirements are reliability, ease of manufacture, thermal conductance, mechanical soundness, convenient shape, means of inspection and repair, and high density. Some of the

methods which achieve high density are seriously lacking in some of the other attributes listed. The multi-layer etched board appears to be a means whereby the technology of interconnection can be advanced, but there exists some disagreement as to its qualifications in its present state of development. A highly satisfactory, though somewhat less dense method, is the welded wire matrix, which is also a multi-layer device, but not made in an integral unit and not so small as the etched board.

| | |
|---|---|
| Word Length | 15 Bits + 1 Parity |
| Number System | One's Complement |
| Memory Cycle Time | 11.7 μsec |
| Fixed Memory Registers | 36,864 Words |
| Erasable Memory Registers | 2,048 Words |
| Number of Normal Instructions | 34 |
| Number of Involuntary Instructions (Interrupt, Increment, etc.) | 10 |
| Interrupt Options | 10 |
| Addition Time | 23.4 μsec |
| Multiplication Time | 46.8 μsec |
| Double Precision Addition Time | 35.1 μsec |
| Double Precision Multiplication Subroutine Time | 575 μsec |
| Increment Time | 11.7 μsec |
| Number of Counters | 29 |
| Power Consumption | 100 Watts (AGC + DSKY's) |
| Weight | 58 Pounds (Computer Only) |
| Size | 1.0 Cubic Foot (Computer Only) |

Table VI-2 AGC Characteristics

Chapter VI-2

# CHARACTERISTICS OF THE APOLLO GUIDANCE COMPUTER

## LOGICAL DESIGN

General - The AGC has three principal sections. The first is a memory, the fixed (read only) portion of which has 36,864 words, and the erasable portion of which has 2048 words. The next section may be called the central section; it includes an adder, an instruction decoder,(SQ), a memory address decoder, (S), and a number of addressable registers with either special features or special use. The third section is the sequence generator which includes a portion for generating various microprograms and a portion for processing various interrupting requests.

The backbone of the AGC is the set of 16 write buses; these are the means for transferring information between the various registers shown in Fig. VI-1. The arrowheads to and from the various registers show the possible directions of information flow. In Fig. VI-1, the data paths are shown as solid lines; the control paths are shown as broken lines.

The Fixed Memory is made of wired-in "ropes" which are compact and reliable devices. The number of bits so wired is in excess of $5 \times 10^5$. The cycle time is 12 $\mu$sec.

The erasable memory is a coincident current ferrite core system with the same cycle time as the fixed memory. Instructions can address registers in either memory, and can be stored in either memory. The only logical difference between the two memories is the inability to change the contents of the fixed part by program steps.

Each word in memory is 16 bits long (15 data bits and an odd parity bit). Data words are stored as signed 14 bit words using a one's complement convention. Instruction words consist of 3 order code bits and 12 address code bits.

The contents of the address register S do not always determine uniquely the address of the memory word. For example, the 2048 erasable registers are accessed via a 1024 word address field. This is done with a three-bit auxiliary address contained in the "Erasable Bank" register, which is under program control. Part of the address field is one-to-one: addresses between 0 and 1377 (octal, or base eight) always refer to the same registers. Addresses 1400 - 1777 (octal) are ambiguous, and refer to one of 5 sets of 256 words according to the number stored in the Erasable Bank register.

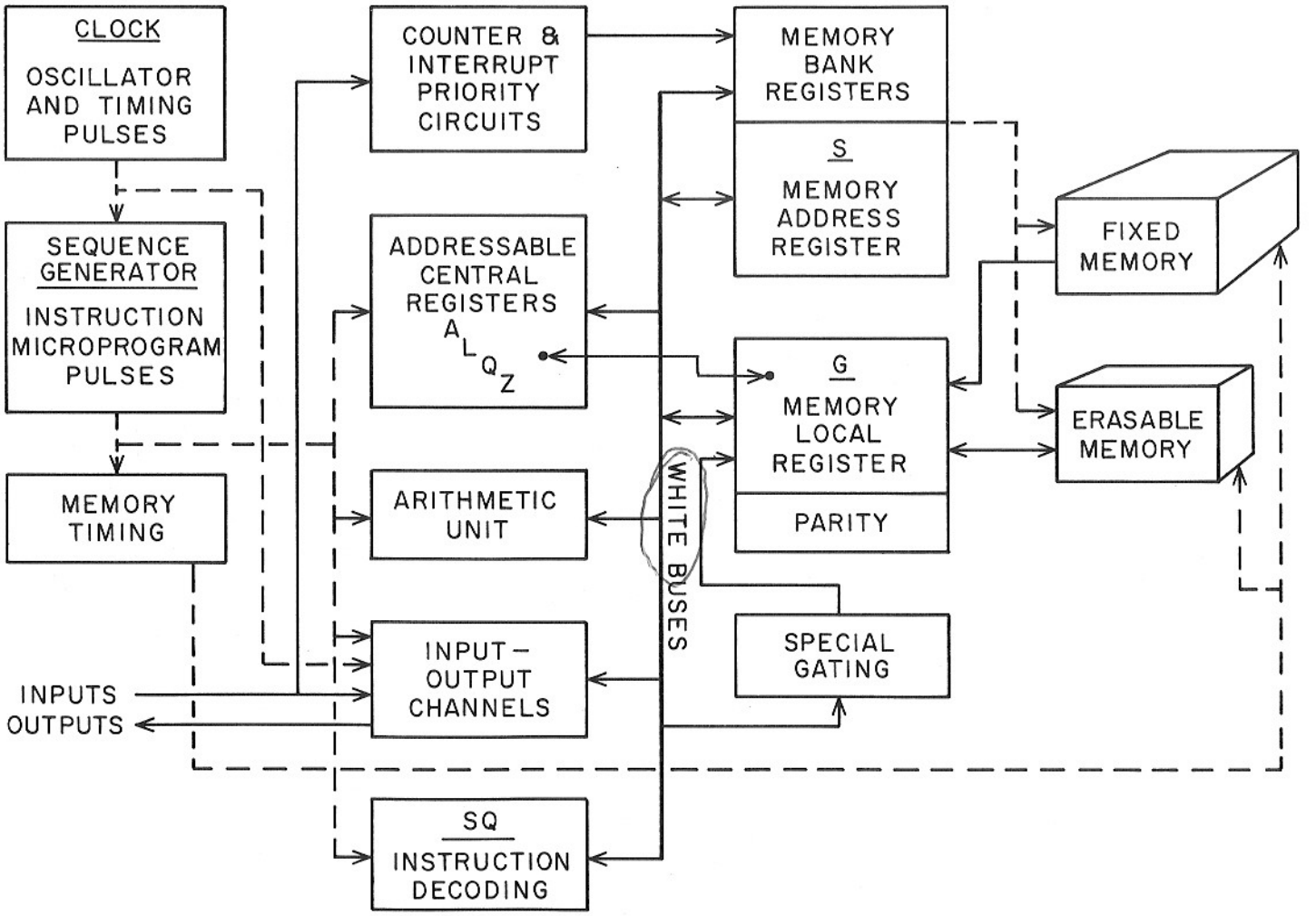



Fig. VI-1 AGC Block Diagram

The 3072 word fixed-memory address field encompasses 36,864 words by means of a 5-bit "Fixed Bank" register and a one-bit "Fixed Extension" channel. Addresses between 2000 and 3777 (octal) are ambiguous, and refer to one of 34 banks of 1024 words each according to the number in the Fixed Bank register. If this number exceeds 30 (octal) then the bank selection further depends on the Fixed Extension bit. The Bank registers and the Extension channel are addressable, and are all in the non-ambiguous portions of the erasable memory and channel fields.

Transfers in and out of memory are made by way of a memory local register G. For certain specific addresses, the word being transferred into G is not sent directly, but is modified by a special gating network. The transformations on the word sent to G are right shift, right cycle, left cycle, and 7-position right shift for editing interpretive instruction words.

The middle part of Fig. VI-1 shows the central section in block form. It contains the address register S and the memory bank registers which were mentioned above. There is also a block of addressable registers called "central and special registers," which will be discussed later, an arithmetic unit, and an instruction decoder register, SQ. The arithmetic unit is an adder with shifting gates and control logic. The SQ register bears the same relation to instructions as the S register bears to memory locations; neither S nor SQ are explicitly addressable. The central and special registers are A, L, Q, Z, and a set of input and output channels. Their properties are shown in Table VI-3.

The sequence generator provides the basic memory timing and the sequences of control pulses (microprograms) which constitute instructions. It also contains the priority interrupt circuitry and a scaling network which provides various pulse frequencies used by the computer and the rest of the navigation system.

Instructions are arranged so as to last an integral number of memory cycles. The list of instructions is treated in detail later. In addition to these there are a number of "involuntary" sequences, not under normal program control, which may break into the normal sequence of instructions. These are triggered either by external events, or by certain overflows within the AGC, and may be divided into two categories: counter incrementing and program interruption.

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

| 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. |



Table VI-3 Addressable Special and Central Registers

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.

Word Length - The AGC is a "common storage" machine, which 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 is to have equal word lengths. 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 are:

A. Precision desired in the representation of navigational variables;
B. Range of the input variables which are entered serially or incrementally;
C. Instruction word format. Division of instruction words into two fields, one for operation code and one for address.

As a start, the word length (15 bits) for two previous machines in this series[4] was kept in mind as a satisfactory word length from the point of view of mechanization; i.e., the number of sense amplifiers and inhibit drivers, and the carry propagation time, etc., were all considered satisfactory. The influence of these principal factors will be taken up in turn.

The data words used in the AGC may be divided roughly into two classes: data words used in elaborate navigational computations, and 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 $0(10^{8\pm1})$. 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 word length is 29 bits (including sign), rather than 30, for a basic word length of 15 bits.

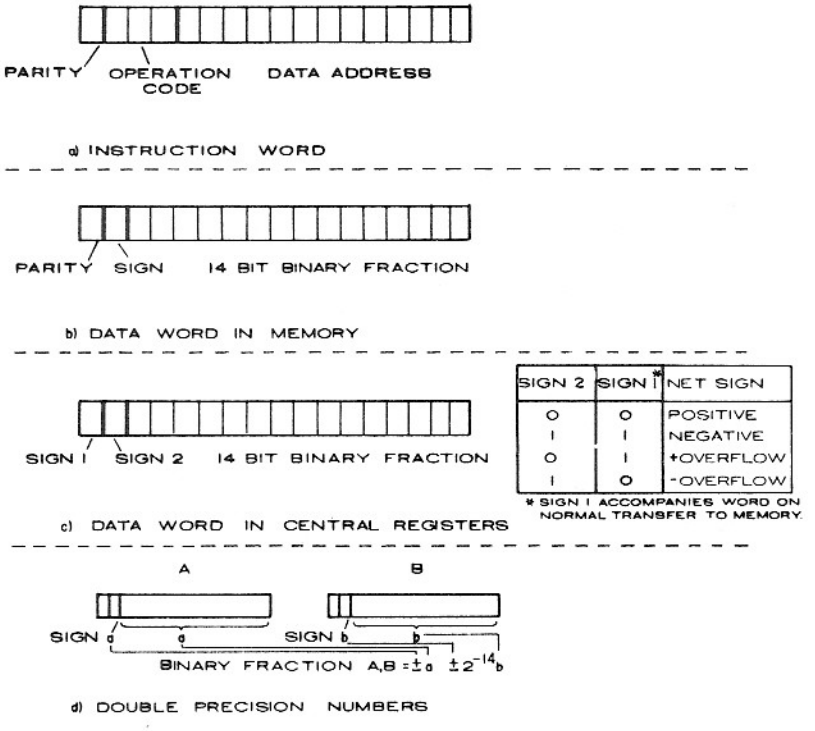



Fig. VI-2 Word Formats

The initial estimates for the proportion of 15-bit vs. 29-bit quantities to be stored in both fixed and erasable memories indicated the overwhelming preponderance of the former. It was also estimated that a significant portion of the computing had to do with 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 of which are of angles. Two principal forms of conversion exist: one renders a whole number, the other produces a train of pulses which must be counted to yield the desired number. The latter type of conversion is employed by the AGC, using the counter incrementing feature. When the number of bits of precision required is greater than the computer's word length, the effective length of the counter must be extended into a second register, either by programmed scanning of the counter register, or by using a second counter register to receive the overflows of the first. 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.

An initial decision was made that instructions would consist of an operation code and a single address. The straightforward choices of packing one or two such instructions per word were the only ones seriously considered, although other schemes, such as packing one and a half instructions per word, are possible.[1] The two previous computers had a 3-bit field for operation codes and a 12-bit field for addresses, to accommodate their 8 instruction order codes and 4096 words of memory. In the initial core-transistor version of the AGC, the 8 instruction order codes were in reality augmented by the various special registers provided, 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 the order code. 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. Furthermore, as will be seen, it is possible to expand the number of order codes.

The address field of 12 bits presented a different problem. At the time of the design of the previous computers, it was estimated that 4000 words would satisfy the the storage requirements. By the time of redesign it was clear that the requirement

was for $10^4$ words, or more, and 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 and special registers, which amounts to increasing the size of the non-memory portion of the AGC.

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 or 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.

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 AGC 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; and "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 may use both. The one's complement notation has the advantage of having easy sign reversal, which is 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 addition is required in the process.

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

| | | STANDARD | | | | | MODIFIED | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $S_1$ | 4 | 3 | 2 | 1 | $S_2$ | $S_1$ | 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. VI-3 Illustrative example of properties of modified one's complement system.

adopted in the AGC has the advantage of being a one's complement system with the additional feature of having a static indication of overflow. The implementation of the method depends on the AGC's 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 Fig. VI-3). 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 $S_2$ bit is stored, which is the standard one's complement sign except in the event of overflow, in which case it is the sign 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.

Multiple-Precision Arithmetic - A short word computer can be effective only if the multiple-precision routines are efficient corresponding 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 in which 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 subtraction, and the identical sign representation is costly because sign reconciliation is required after every operation. For example, (+6, +4) + (-4, -6) = (+2, -2), a mixed sign representation of (+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 operands. 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. Apollo programs do not require greater than double-precision multiplication and division, however. The algorithm for double-precision

multiplication is directly applicable to numbers in the independent sign notation. The treatment of 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.

Instruction Set - The major goals in the AGC 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 multiply 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 AGC 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 AGC the size of the instruction set is doubled by an Extend operation, which 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.

Partial codes are instruction codes which encroach upon the address field. This technique capitalizes upon the essential difference between fixed and erasable memory. More specifically, a wider variety of instructions are applicable to erasable than to fixed memory; for example, all instructions which modify the operand register are not fully applicable for fixed memory. Since the fixed memory address field in the AGC is 3 times as big as the erasable memory field, it is possible to pack 3 extra erasable memory instructions into that portion of the entire address field. 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 modulo 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 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 VI-4 lists the normal AGC instructions. These include facility 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, which 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 AGC instructions are involuntary or address dependent, and are listed in Table VI-5. The last four 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.

## HARDWARE

Logic - The design of the Apollo Guidance Computer began at a time when microcircuits were first being produced. Microcircuits held great promise, but were not well enough proven for the design of this computer to be based on them; magnetic core and transistor logic had been used in its immediate ancestry and was scheduled to be used here. Nevertheless, during the first year of design, microcircuits were evaluated for possible use in the AGC. When it became clear that microcircuits could be reliably produced with rigid specifications, the decision was made to substitute them for the core-transistor logic. In the course of this change, the power consumption increased by a factor of three, but size and weight were reduced by half, and performance and speed were doubled. Moreover, though it could not be known at the time, the reliability of the logic hardware was greatly increased.

One of the important decisions made at that time was to confine the use of logic microcircuits to a single type to avoid having to develop successfully a number of

| | | | |
|---|---|---|---|
| A. | Sequence Changing | | |
| | 1. Transfer control, set return address | 1 MCT | All Memory |
| | 2. Transfer control only | 1 MCT | Fixed only |
| | 3. Four way skip and diminish by one | 2 MCT | Erasable |
| | *4. Branch on zero | 1 or 2 | Fixed only |
| | *5. Branch on zero or minus | 1 or 2 | Fixed only |
| B. | Fetching and Storing | | |
| | 1. Clear and add to Accumulator, A | 2 MCT | All |
| | 2. Clear and subtract from Accumulator, A | 2 MCT | All |
| | *3. Double clear and add to A and Lower Accumulator, L | 3 MCT | All |
| | *4. Double clear and subtract from A and L | 3 MCT | All |
| | 5. Transfer to storage | 2 MCT | Erasable |
| | 6. Exchange A with storage | 2 MCT | Erasable |
| | 7. Double exchange A and L with storage | 3 MCT | Erasable |
| | 8. Exchange L with storage | 2 MCT | Erasable |
| | *9. Exchange Q with storage | 2 MCT | Erasable |
| C. | Instruction Modification | | |
| | 1. Index (add to next instruction) | 2 MCT | Erasable |
| | *2. Index and extend | 2 MCT | All |

* Requires Extend instruction
MCT = Memory Cycle Time

Table VI-4 Normal Instructions (Part 1)

D. Arithmetic and Logic

| | | | |
|---|---|---|---|
| 1. | Add to A | 2 MCT | All |
| *2. | Subtract from A | 2 MCT | Erasable |
| 3. | Add to Storage and A | 2 MCT | Erasable |
| *4. | Modular subtract from A (mixed number system) | 2 MCT | Erasable |
| 5. | Add 1 to storage (Increment) | 2 MCT | Erasable |
| *6. | Increase absolute value of storage by 1 (Augment) | 2 MCT | Erasable |
| *7. | Decrease absolute value of storage by 1 (Diminish) | 2 MCT | Erasable |
| 8. | Double add A and L to storage | 3 MCT | Erasable |
| 9. | Logical product to A | 2 MCT | All |
| *10. | Multiply; product to A and L | 3 MCT | All |
| *11. | Divide A and L by storage; quotient to A | 6 MCT | Erasable |

E. Input Output

| | | | |
|---|---|---|---|
| *1. | Transfer channel to A | 2 MCT | Channels |
| *2. | Transfer A to channel | 2 MCT | Channels |
| *3. | Logical product (of A and channel) to A | 2 MCT | Channels |
| *4. | Logical product to channel and A | 2 MCT | Channels |
| *5. | Logical sum to A | 2 MCT | Channels |
| *6. | Logical sum to channel and A | 2 MCT | Channels |
| *7. | Exclusive or to A | 2 MCT | Channels |

* Requires Extend instruction

MCT = Memory Cycle Time

Table VI-4 Normal Instructions (Part 2)

| | | | |
|---|---|---|---|
| A. | Involuntary | | |
| | 1. Transfer to interrupt program, store c(Z) and c(B) | 3 MCT | Limited |
| | 2. Increment by 1 | 1 MCT | Counters |
| | 3. Increment by -1 | 1 MCT | Counters |
| | 4. Diminish absolute value by 1 | 1 MCT | Counters |
| | 5. Shift left | 1 MCT | Counters |
| | 6. Shift left and add 1 | 1 MCT | Counters |
| B. | Address Dependent | | |
| | 1. Resume interrupted program = Index 0017 (octal) | 2 MCT | |
| | 2. Extend = Transfer control 0006 | 1 MCT | |
| | 3. Inhibit interrupt = Transfer control 0004 | 1 MCT | |
| | 4. Permit interrupt = Transfer control 0003 | 1 MCT | |
| | 5. Cycle right each access | Address 20 (octal) | |
| | 6. Shift right each access | Address 21 (octal) | |
| | 7. Cycle left each access | Address 22 (octal) | |
| | 8. Shift right seven places each access | Address 23 (octal) | |

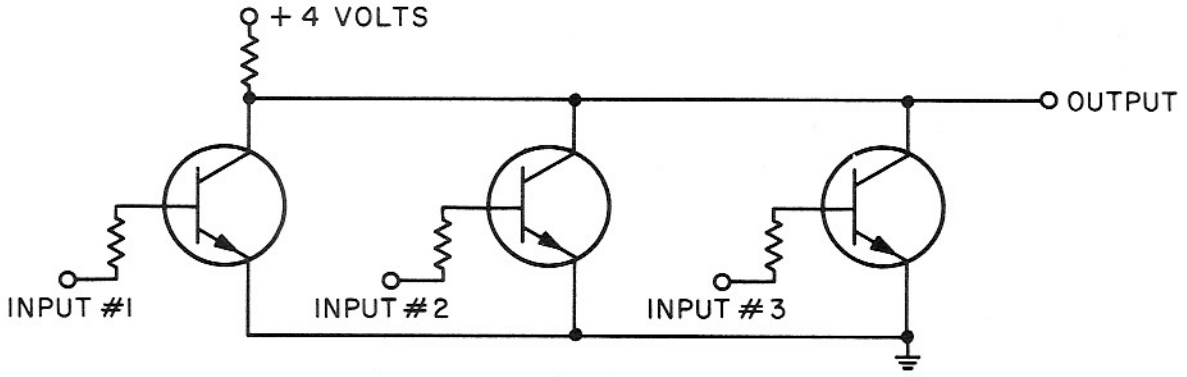
MCT = Memory Cycle Time

Table VI-5 Special Instructions

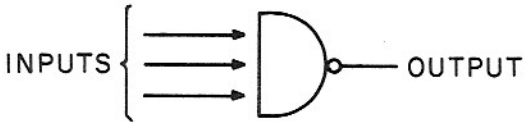
different devices. A logic circuit was required whose operational function was capable of synthesizing all switching functions, and which was simple enough to be controllable, testable, and producible. The circuit chosen was a NOR gate which employs a configuration known as modified direct-coupled transistor 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 twelve milliwatts of power. A recent modification of design has resulted in a new unit with approximately the same specifications except for a power dissipation of 5 milliwatts instead of 12. These gates are designed to operate over the temperature range from 0 to 70°C.

The importance of using a single circuit should not be underestimated. Thousands of logic gates are employed in each computer and barring the use of redundancy techniques, every one may be considered critical. Indeed, most redundancy techniques depend on randomness of failures; and in general new components and assembly methods introduce failure modes which make erroneous the basic assumptions on which the redundancy is based. High reliability is essential for every gate. It can best be attained by standardization, and can only be demonstrated by the evaluation of large samples.[8] Had a second type of logic microcircuit been employed in the AGC, 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, for neither of the two circuits would have accumulated the high mean time to failure and high confidence level that the one NOR-circuit has.

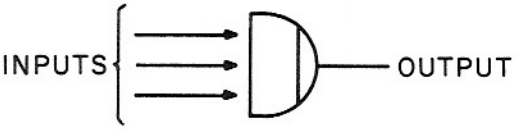
Logic equations expressed in the familiar AND, OR, NOT notation may readily be realized with NOR operators. A two-level and-or expression is realizable in a two-level NOR-circuit. The NOR function of three variables is as follows: $N(x, y, z) = \overline{x}\,\overline{y}\,\overline{z} = \overline{x + y + z}$. An AND function is $A(x, y, z) = x\,y\,z$, and an OR function is $0(x, y, z) = x + y + z$. By comparison, $N(x, y, z) = A(\overline{x}, \overline{y}, \overline{z}) = \overline{0(x, y, z)}$. The NOT operation, or complementation, is the NOR function of one variable; i.e., $\overline{x} = N(x)$. Complex Boolean expressions ordinarily arise only in connection with non-sequential, or combinational aspects of the computer logic. Sequential operations require storage; and the basic logic storage element is the flip-flop. Two NOR gates 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 one, the flip-flop is forced into one state, whereas if a free input on the opposite gate is made equal to one, the other state is obtained. Most frequently, the condition for setting a flip-flop to a particular state is that two or more other signals simultaneously take on prescribed values. Detection of such coincidence requires a NOR operation separate from the flip-flop plus any NOR operation required to invert (complement) the inputs.




a) EQUIVALENT CIRCUIT OF NOR GATE




b) DIAGRAM NOTATION FOR NOR GATE




c) DIAGRAM NOTATION FOR UNPOWERED NOR GATE (+4 VOLTS DISCONNECTED)

Fig. VI-4 The NOR Gate

It is frequently necessary to implement NOR functions of more than three variables, and also to be able 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 (fan-in) capacity, or the output (fan-out) capacity, 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 be able to saturate the transistors, whose current gain is limited. 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 employment of NOR logic in the AGC is provided by the Central register flip-flops. Digits are transferred from one register to another via a common set of wires called write buses. The sending and receiving flip-flops are selected by read and write pulses, respectively, applied to gates which either set or interrogate the flip-flop of the corresponding register. Figure VI-5 shows a hypothetical set of three flip-flops similar to those in one column of the AGC central register section. Dashed lines imply the existence of other registers than the three shown. Diagonal lines, or slashes, after signal names denote inverse polarity. Thus WRITE BUS/ is normally in the one state, and changes to zero while transferring a one. Suppose REG 1 contains a one, i. e., the top gate of its flip-flop has an output of zero. At the time that the READ 1/ signal goes to zero from its normally one state, the output of the read gate, CONTENT 1, becomes a one. This propagates through a read bus fan-in and an inverter and fan-out amplifier to make WRITE BUS/ become zero. Suppose that WRITE 2/ is made zero concurrently with READ 1/. Then the coincidence of zero's at the write gate of REG 2 generates a one at the input to the upper gate of its flip-flop, thus setting the bit to one.

If REG 1 had contained a zero, the write bus would have remained at one, and no setting input would have appeared at the upper gate of REG 2. The CLEAR 2 pulse, which always occurs during the first half of WRITE 2, would have forced the flip-flop to the zero state, where it would remain; whereas when a one is transferred, the SET 2 signal persists after the CLEAR 2, and thus forces the register back to the one state. Thus the simultaneous occurrence of READ 1/, WRITE 2/, and the short CLEAR 2 pulses transfer 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 READ BUS/. An increase in fan-out is achieved by the two gates connected in parallel to form the signal WRITE BUS/.

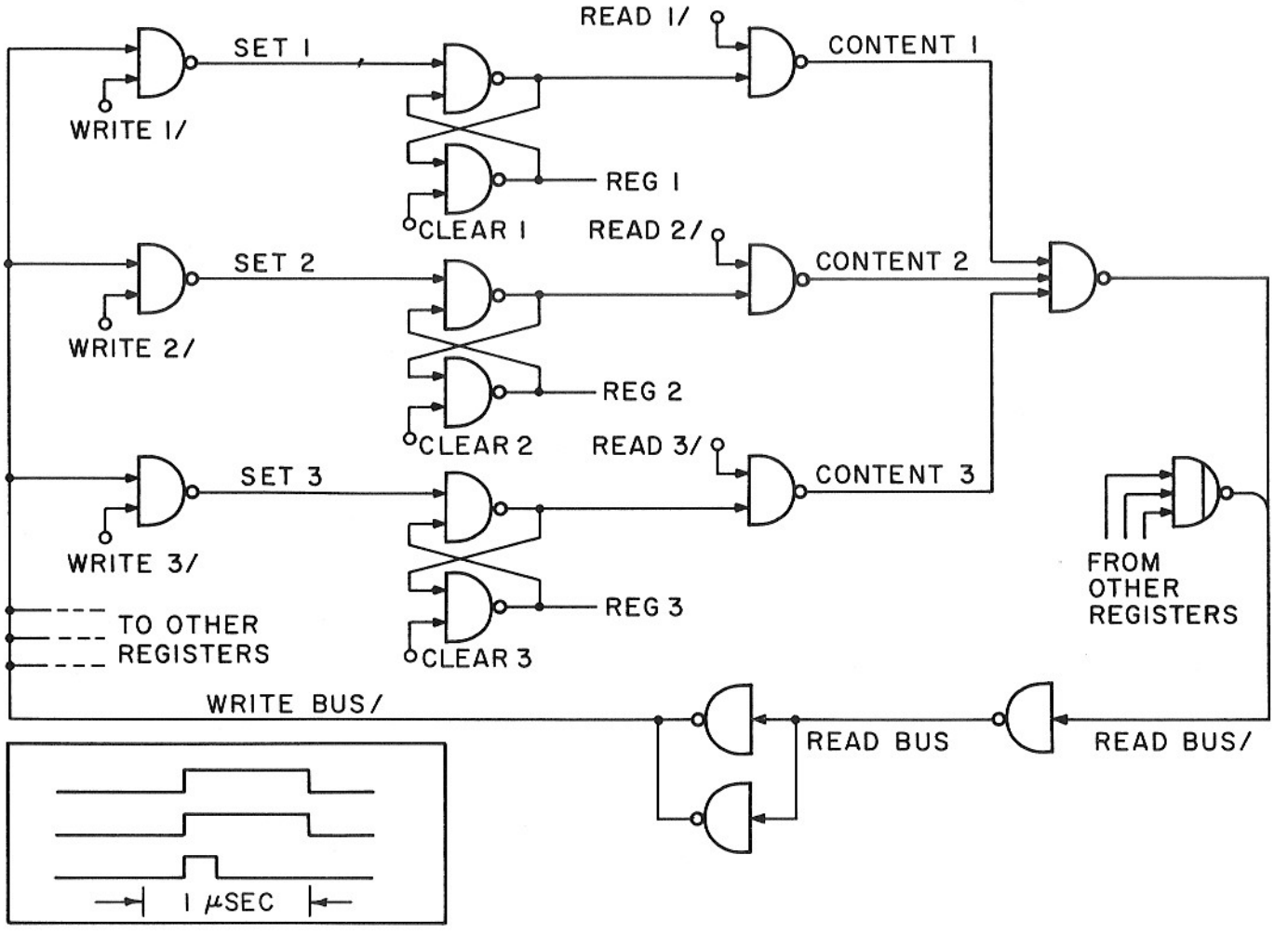



Fig. VI-5 NOR Gate Circuit Resembling One Column of AGC Central Registers

The main problem of mechanical design in guidance computer logic is the creation of signal interconnections; indeed, approximately 3/4 of the volume of the AGC is used for this purpose. Interconnections are primarily of two types: between modules by wrapped wire, and within a module by welded matrix.

The carrier into which all modules are inserted is called a "tray." The AGC comprises two trays: one for logic, power supply and interface modules, and the other for memory and ancillary circuit modules. The 15,000 jacks on the tray into which signal pins are inserted pass through the tray and extend out the other side in the form of posts with square cross sections. Interconnections between pins are made by wires whose ends are tightly wrapped around the posts without the use of any further contact mechanism such as solder or welds. This method has several advantages: it is executed by a machine, which requires only a few seconds per wire, it is controlled by a punched card input, it can easily be altered if a change is desired, wires can be run point to point if desired, and the reliability of the connection is extremely high since there is no single point where bending stress is applied. Moreover, it is compatible with hand wiring, which is required wherever wires are twisted together to protect low level signals, or where heavy gauge wire is needed in order to accommodate high currents.

In the AGC, one of the basic goals has been to make the electronic circuits in small pieces which are easily installed and removed, for the sake of producibility, testing, easy diagnosis and economical maintenance. This can only be realized insofar as it does not excessively degrade the overall packaging density of the computer; for volume is, of course, critically limited in the spacecraft. It was found expedient to make 24 modules each containing 120 microcircuit units, separated into two independent groups of sixty. The twelve-milliwatt gates are packaged one to a unit; sixty gates are connected together into a circuit with 72 pins to bring signals in and out. The more recent five-milliwatt gates are packaged two to a unit, because of their less severe heat transfer requirement. These are organized into sub-groups of thirty, such that sixty gates are again connected together; and 72 pins are again available to each sixty gates. The modules are the same size, so that the low-power units are packaged with double the density of the high power units. Accordingly, the density of pins and interconnections has been doubled along with that of the gates.

The main method of connection internal to the module is by matrix. Gates are disposed in a single row within each sixty-gate sub-group. An array of vertical wires (at right angles to the row of gates) access every connection to the gates. Horizontal conductors (parallel to the row of gates) carry signals from gate to gate and from gate to pin. Connections between horizontal and vertical matrix members as well as between matrix members and gate connection pins are made by a spot welding process. The process was developed in an earlier guidance computer project in order to eliminate the problems of cold solder joints.

Fig. VI-6 Photograph of Wire-wrapped Tray Connections

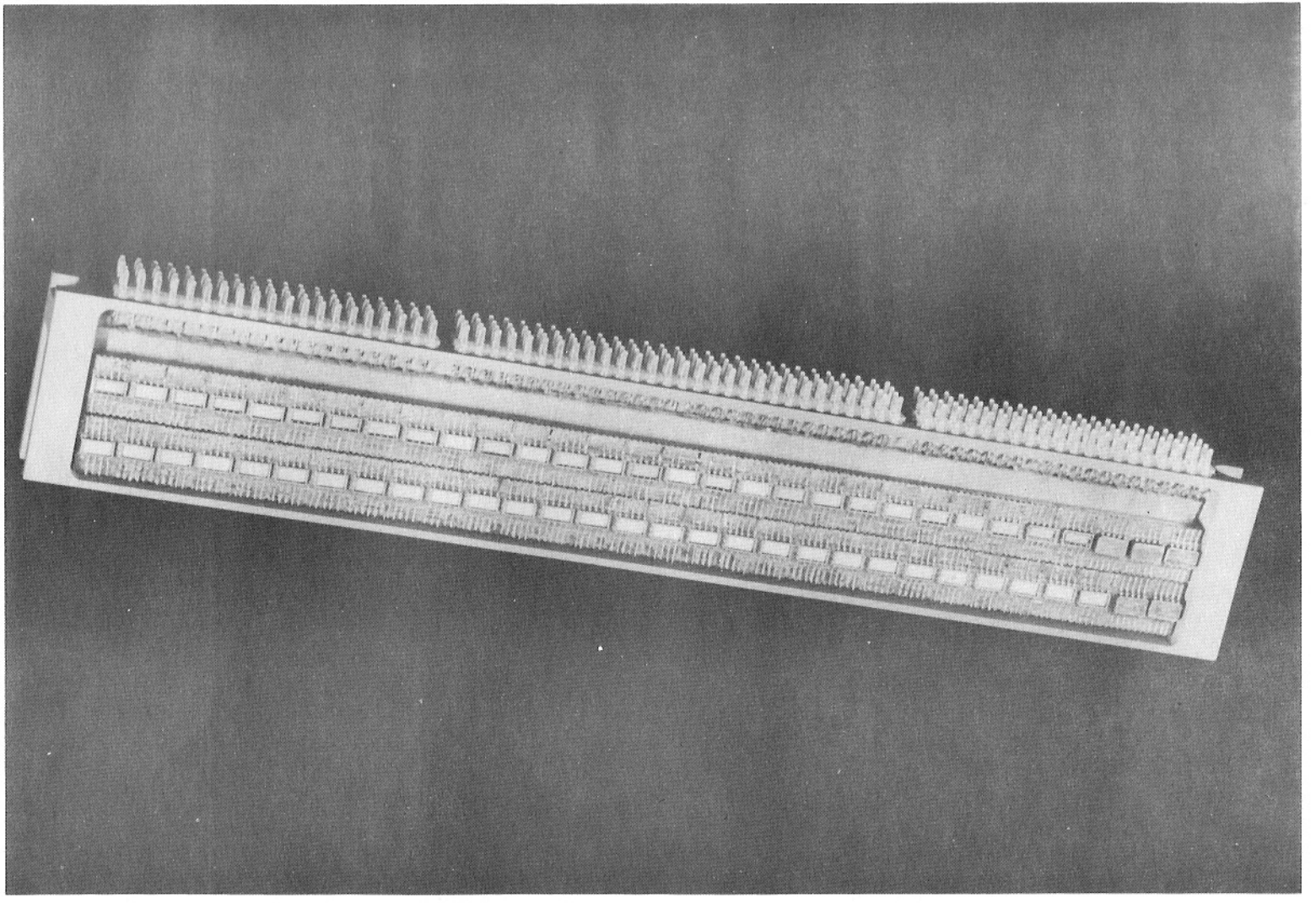


Fig. VI-7 Logic Module Containing 120 Microcircuit Units

Power distribution is a special problem in this computer. The current drawn by the gates is about 6 Amperes for low power gates and about 13 Amperes for high power. For the sake of efficiency, these large currents must be distributed from the power supply to the logic modules with very little d.c. voltage drop. Moreover, the current return, or zero-volt distribution must not sustain any a.c. voltages of such a frequency or amplitude as to turn on or turn off a gate inadvertently. This is all accomplished by building an interlaced gridwork of heavy conductors upon the terminal posts of the tray. Each group of sixty gates shares a ground plane in a module which is brought out at three equally spaced places to connect to this gridwork, which provides multiple paths for return current much the same as a ground plane. The other power line, the positive voltage, is distributed by a gridwork circuit to two points on the power bus shared in a module by each sixty-gate group.

The main tray structure of the AGC is an aluminum alloy frame into which the modules are affixed by jacking screws, providing a good thermal path between modules and tray. The tray in turn is screwed to a cold plate where heat is removed. A model of the AGC is shown in Figs. VI-8 and VI-9. The front is shown in the first figure, with the outlines of the six fixed memory modules just visible. One of the six is partially extracted. The second figure shows the rear and the connector deck with system and test connector covers. The ruler is calibrated in inches (one inch = 2.54 cm.).

Memory - The erasable memory of the AGC was inherited from its core-transistor logic ancestor.[9] 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 which drive this memory vary their outputs with temperature in such a way as to match the requirements of the cores over a wide range, from $0^{\circ}C$ to $70^{\circ}C$. 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 AGC, 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. Bi-directional currents are generated in each selection wire by a double-ended transistor switching network. The selection of one wire in 32 is made by twelve switch circuits in an 8 x 4 array; the selection of one wire in 64 is made in an 8 x 8 array. The operation of the switching network is illustrated in Fig. VI-10. The transistors are driven by magnetic cores, which offer two advantages over all-transistor circuits: 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 which operate the switch cores is based on the duration of the write current in the memory array, which is

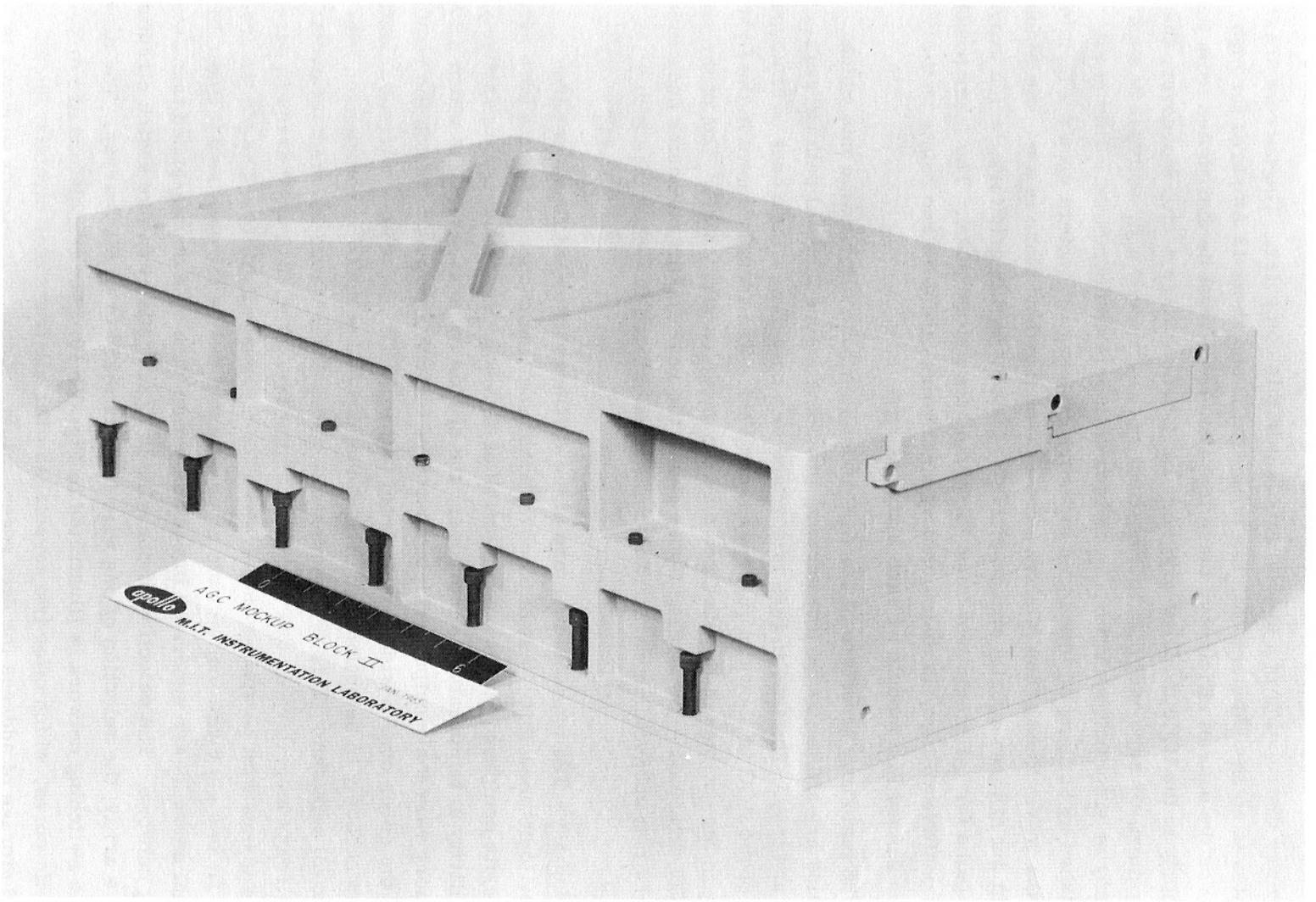



Fig. VI-8 AGC Mockup - Front View

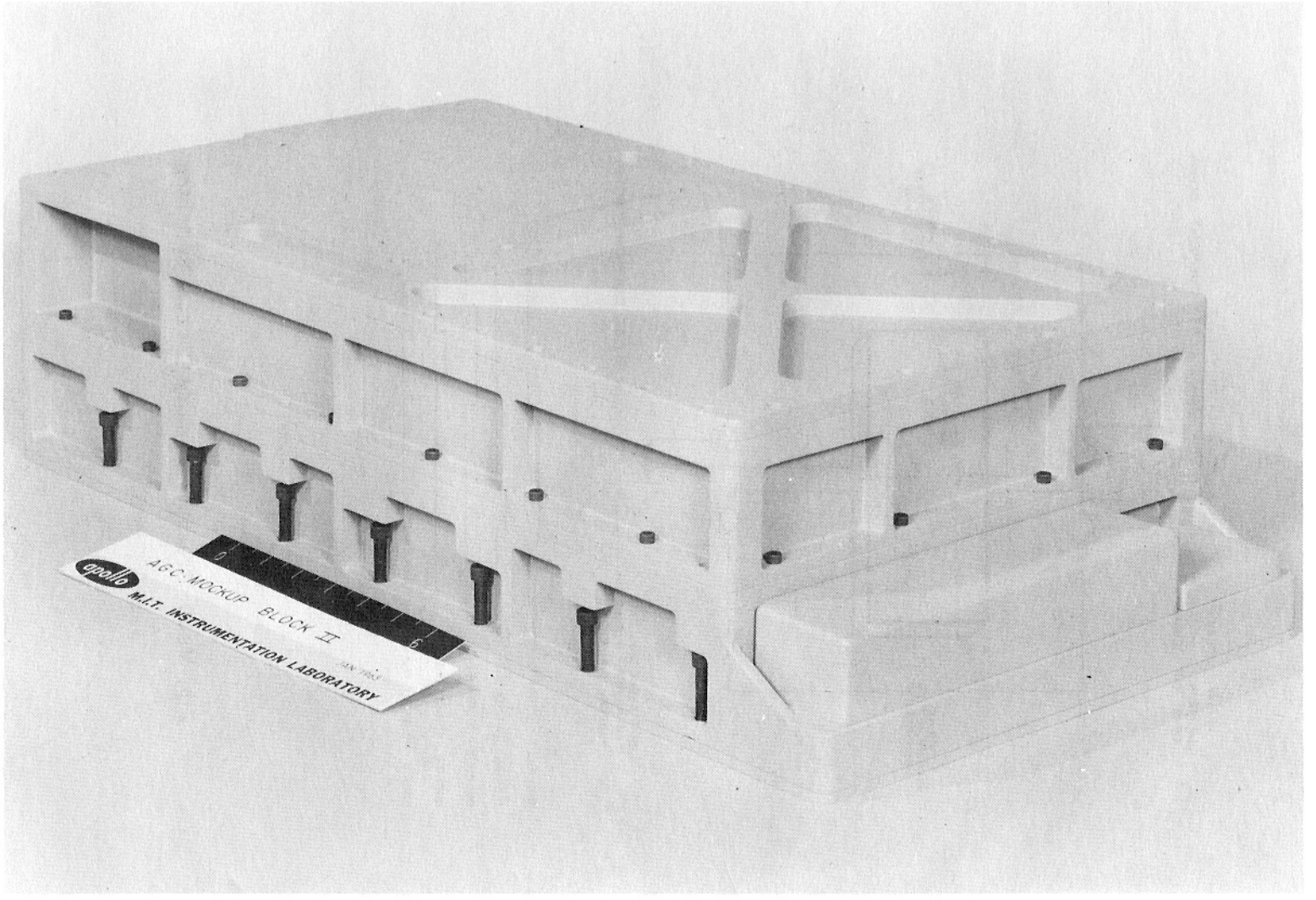



Fig. VI-9 AGC Mockup - Rear View

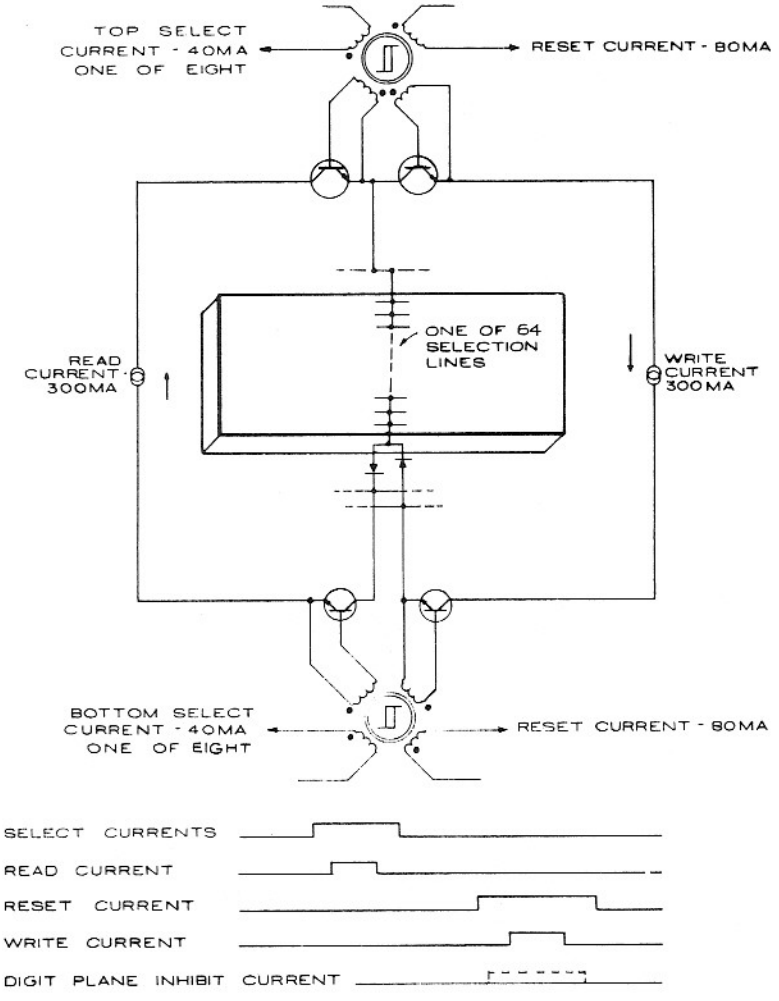



Fig. VI-10 Erasable Memory Current Switching

2 microseconds. 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 which control the writing phase of the memory cycle. Current amplitude is governed by the forward voltage drop across a silicon junction, so that temperature compensation is achieved without any circuit complications for changes in coercive force.

The output signal from the memory cores has an amplitude of about 50 millivolts. Transformer coupling to the sense amplifiers provides common mode noise rejection and a voltage gain of two. The sense amplifiers have a differential first stage operated in the linear, or 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 sense amplifiers are implemented as integrated circuits in order to realize a number of advantages inherent in single-chip semiconductor circuits. Differential amplifiers pose a special problem in component matching both internal to a single amplifier in achieving balance, and among a group of amplifiers in achieving uniform behavior for common reference voltages. In discrete-component amplifiers a great deal of time and effort go to 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 exposed to the same chemical processes. A similar situation holds for uniformity from one amplifier to another. Within the same batch, amplifiers tend to be very much alike, their differences being easily compensated by external trimming resistors. The efforts expended in specifying the integrated sense amplifier and in a program of reliability testing are little if any more costly than for discrete component amplifiers. Indeed, for a given degree of matching, the cost may be expected to be lower for the integrated circuit. Considerably more complex than the integrated NOR gate used in the AGC, the sense amplifier is used in far less number (32 per computer), so that is is feasible to test and screen each amplifier more comprehensively than a NOR gate, of which there are over 5,000 in each AGC. Performance of the sense amplifiers has been superior, with no spontaneous failures recorded in about a million operational device-hours as of this writing. Their small size is advantageous in obtaining temperature tracking, 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.

The AGC fixed memory is of the transformer type and was developed at MIT.[5, 10] It is designated a "core rope" memory owing to the physical resemblance

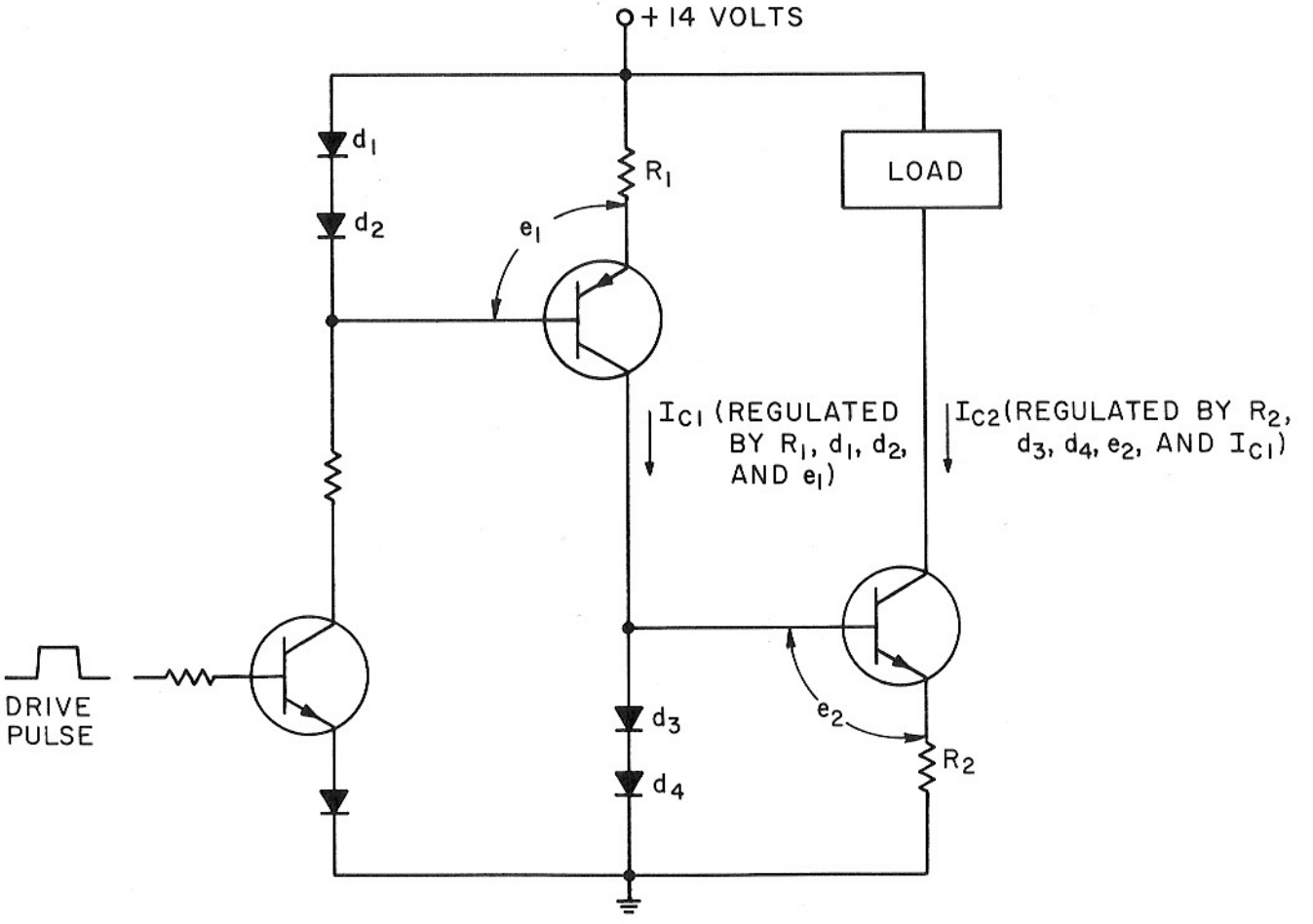



Fig. VI-11 Regulated Pulsed Current Driver Circuit

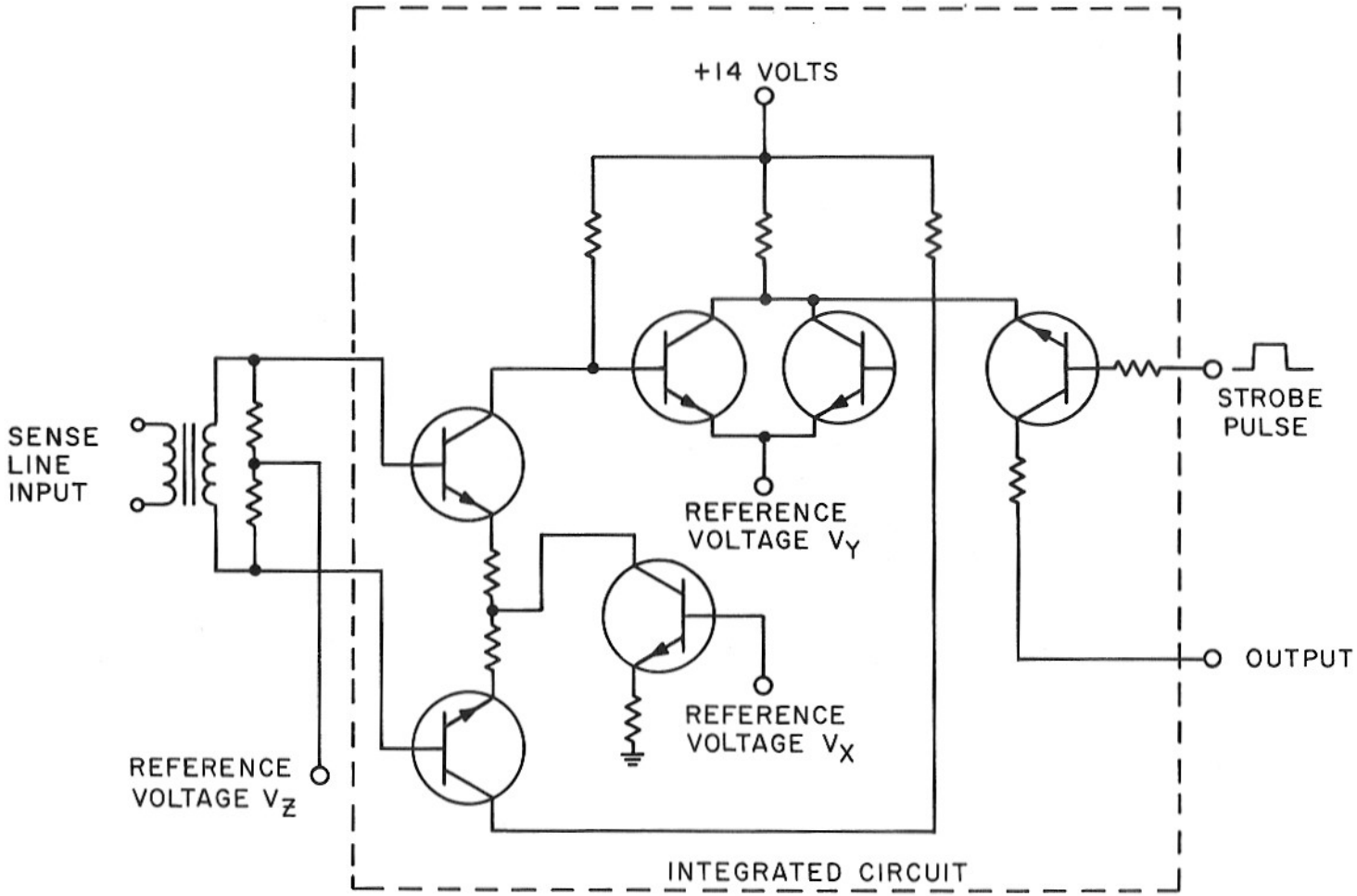



Fig. VI-12 Sense Amplifier Circuit

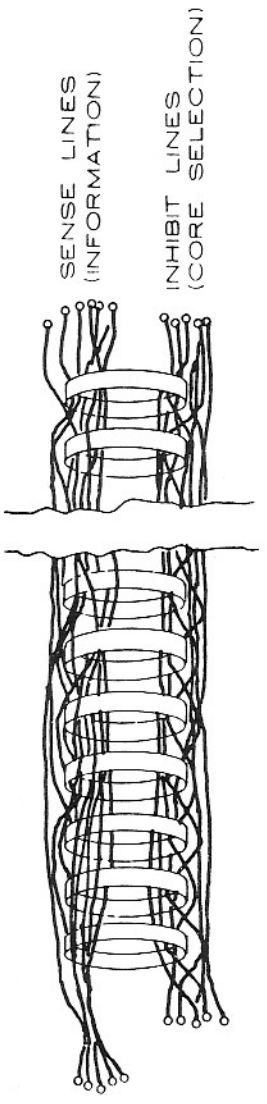



Fig VI-13 Inhibit and Sense Lines through a Rope - Conceptual Drawing

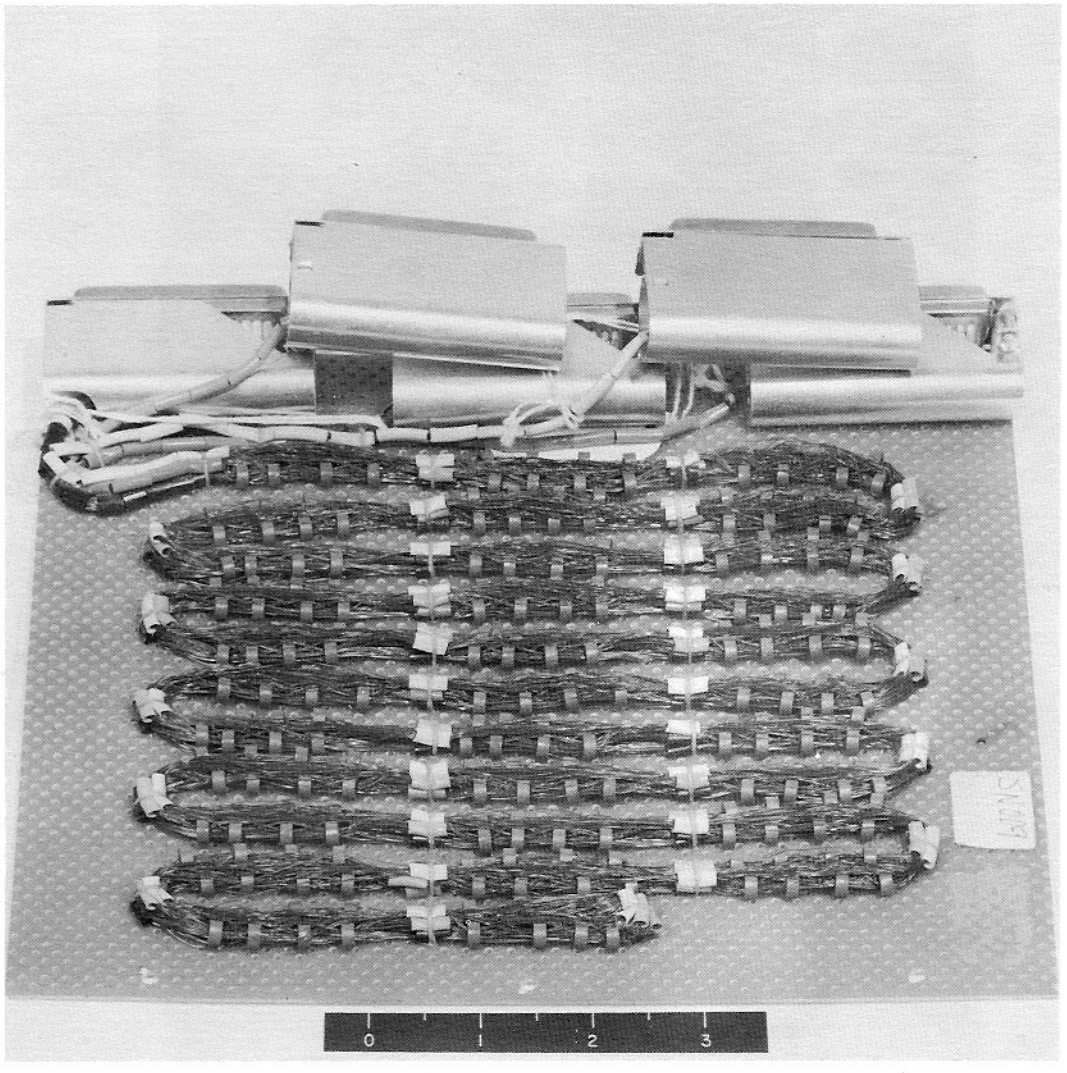



Fig. VI-14 Early Model of Core Rope

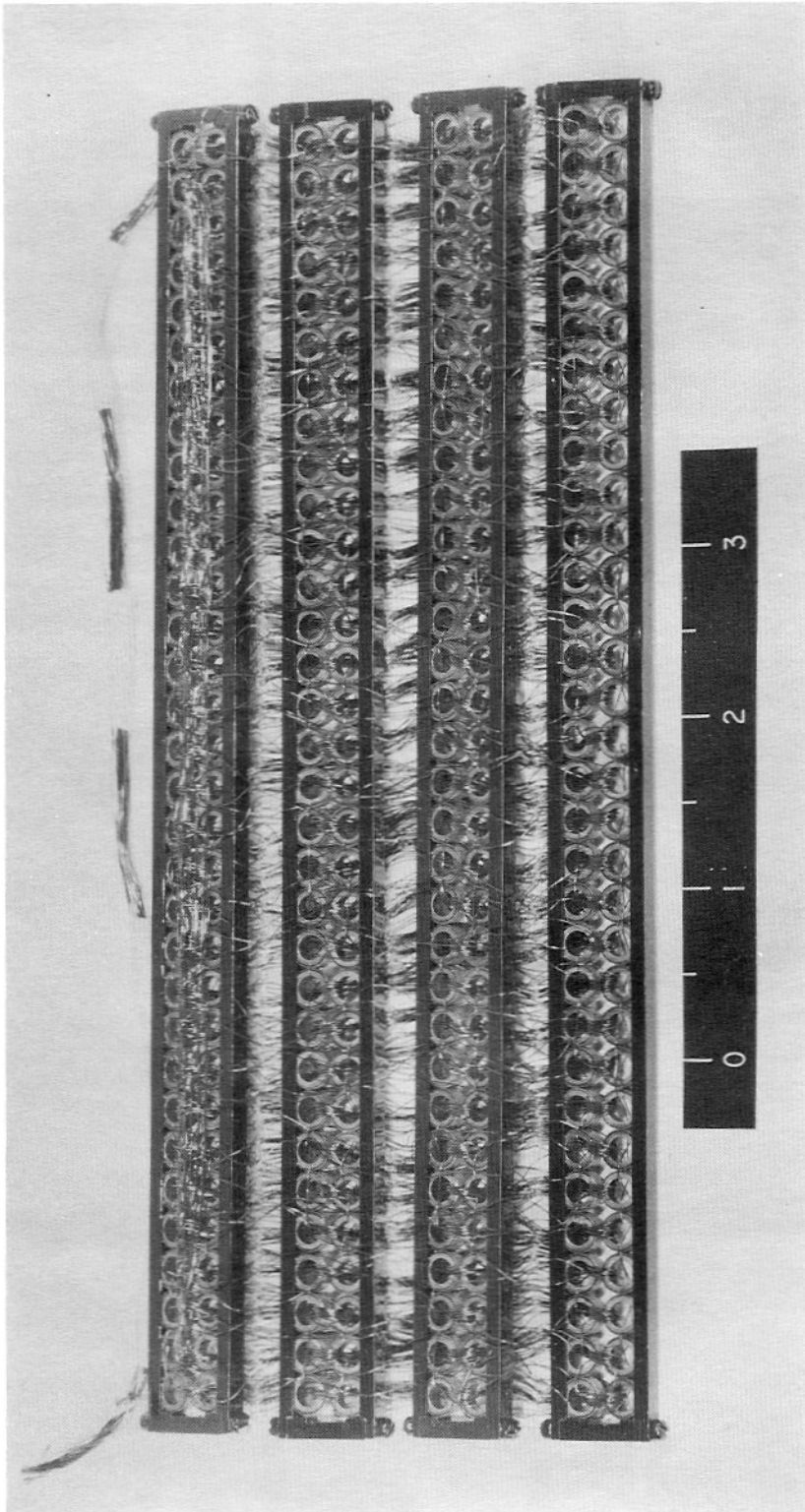



Fig. VI-15 Prototype Core Rope, 1963

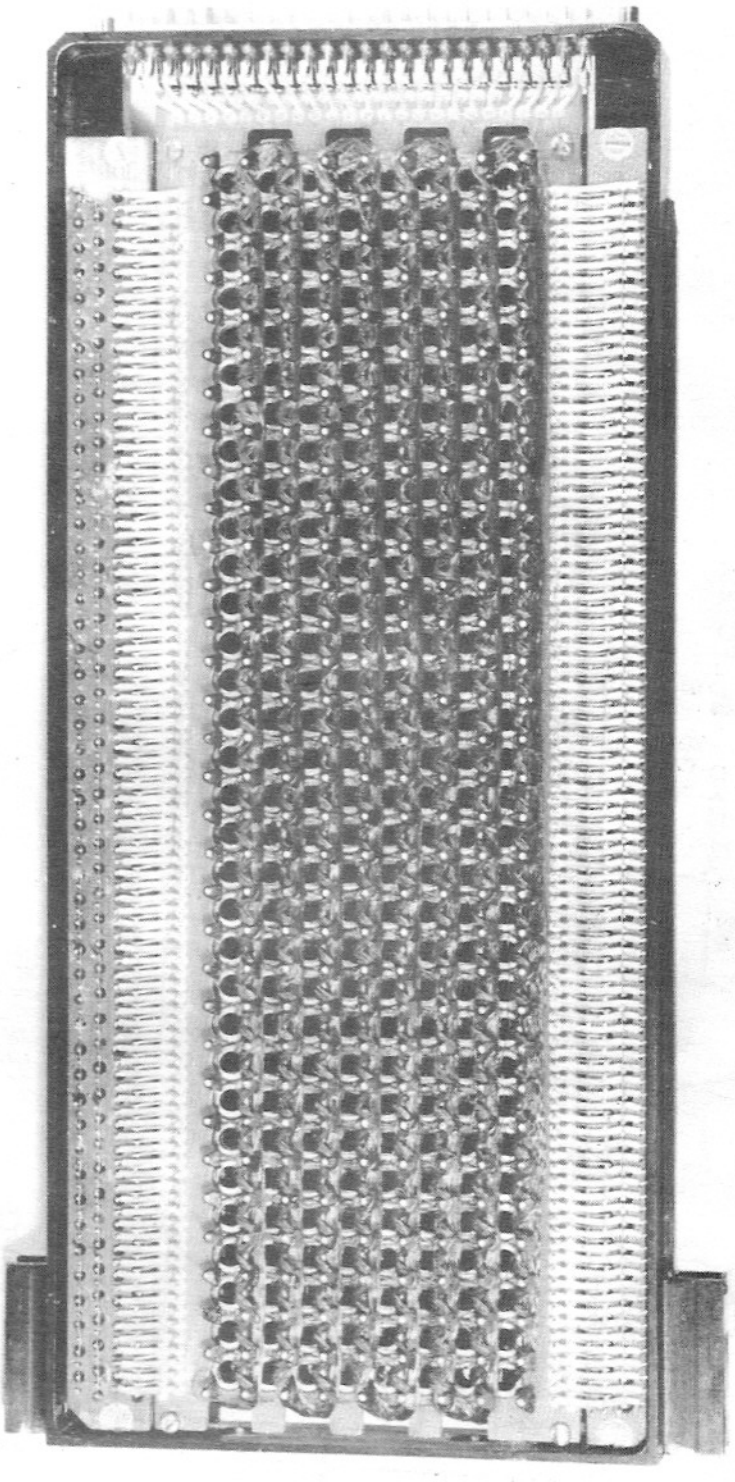


Fig. VI-16 AGC Core Rope Module

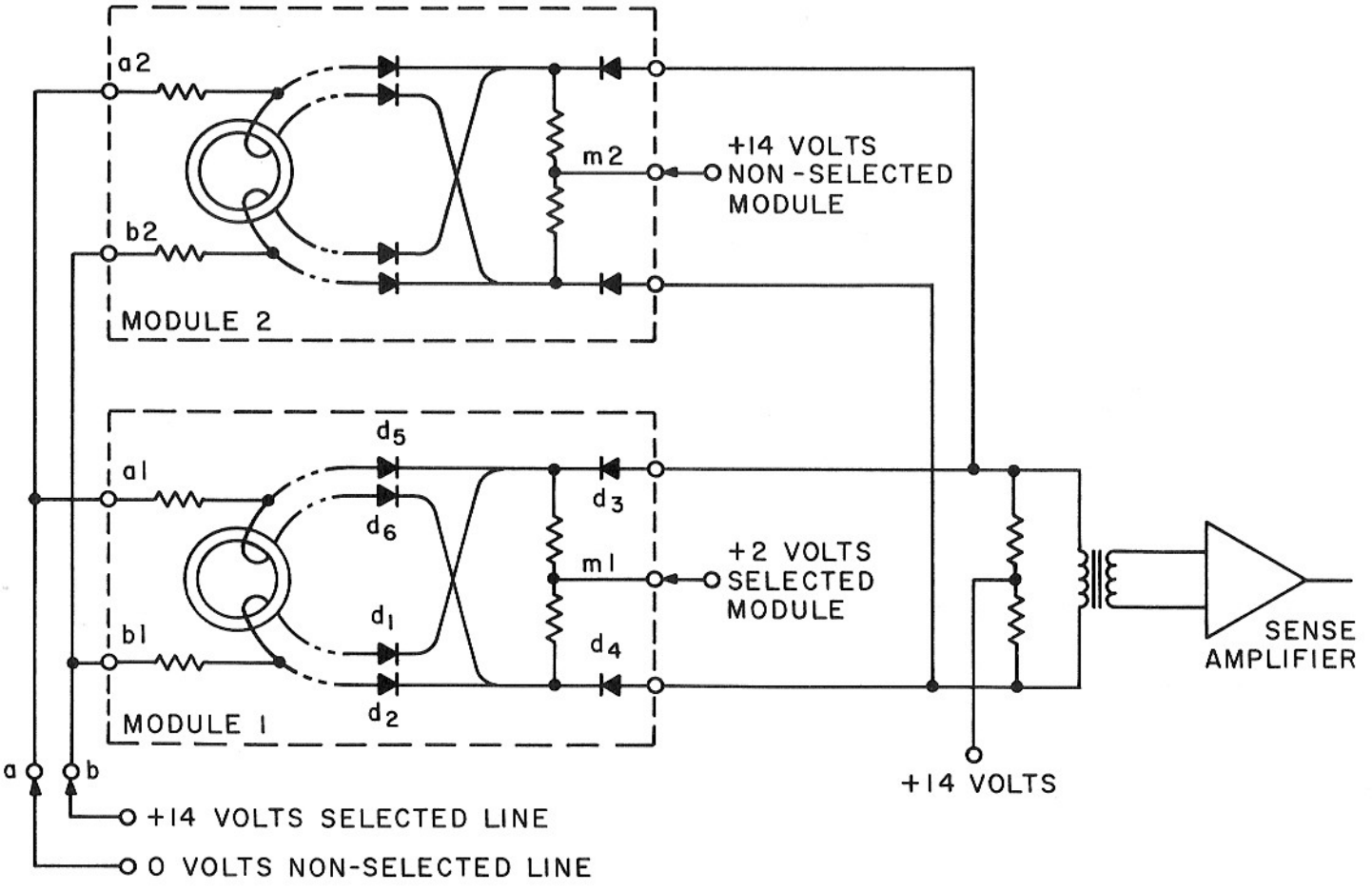



Fig. VI-17 Simplified Rope Sense Line Switching Diagram

of early models to lengths of rope. Incorporated into its wiring structure is an address decoding property, because of which its cycle time 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, (or about 100 bits per cubic centimeter) including all driving and sensing electronics, interconnections, and packaging hardware. This high density of storage is achieved by "storing" a large number of bits in each magnetic core. A stored bit is a one whenever a sense wire threads a core, and is a zero whenever it fails to thread a core. The total number of bits is the number of cores multiplied by the number of sense lines having a chance to thread the cores. The AGC's memory is composed of six modules. Each module contains 512 cores and 192 sense lines and hence contains 192 x 512 = 98,304 bits of information. This information is permanently wired; once the module has been manufactured, not a single bit can be changed, either intentionally or unintentionally, except by physical destruction or by failure of one or more of a number of semiconductor diodes whose functions are described below.

In the operation of the rope memory, a core is switched in a module, thus inducing a voltage drop in every sense line which 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 which have voltage drops and hence store one's. Thus it is that each core stores 12 words; and within each module a switching network is included in order to transmit no more than one of the 12 to the module's output terminals. The principle of the switching network is illustrated in Fig. VI-17. 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 line 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 six modules' outputs to be transmitted to the sense amplifiers. Only the selected line in the selected module is transmitted; all others are blocked by one or two sets of reverse-biased diodes. All of 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 which is external to the rope; the balance is internal. The means by which a single core in a module is caused to switch is as follows: First, a switching current is applied, which attempts to set 128 cores. Four such current lines serve a 512 core module. Second, an inhibit current is simultaneously applied to either the first half or the second half of each group of 128 cores. Two inhibit lines exist for this purpose. Third, another inhibit current is simultaneously applied to either the first or the second half of each half-group. Two more inhibit lines exist for this purpose. Fourth, 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 8 pairs in all to select among $2^7$ cores, of which 7 pairs correspond to the 7 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, a reset current is passed through all cores. Only the core which 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, which is the time it takes to read the memory after the address is available, is longer that way. Both ways have been used in the AGC; the newer design uses the longer access time and produces the address earlier to make up for it.

## INTERFACE METHODS

General - Information transfer between the AGC and its environment occupies a substantial fraction of the computer's hardware and its time budget. An attempt has been made to minimize the number of different types of circuits involved. This minimizes engineering effort and makes the computer more easily produced and tested. The impact of this design philosophy on the system has been substantial, and it came about only by active cooperation between subsystem design and system integration groups. The nature of information handled through the interfaces is varied. In some cases computer words are transferred bodily into and out of the computer. Prelaunch and in-flight radio links are maintained between the computer and ground control. Owing to the great difference in data rates between up and down directions, the mechanizations differ considerably. The down link operates at a relatively high rate (50 AGC words or 800 bits per second) and is made so as to occupy a minimum of the computer's time budget. The circuit is relatively expensive. It serializes a word stored in parallel in a flip-flop register and, upon command, sends the bits in a burst to the central timing system of the spacecraft. The up link uses memory cycles to effect a serial to parallel conversion of data. Each bit received requires a memory cycle; a maximum of 160 bits per second are allowed. The cost in equipment is small. The same procedure used in the up link for serial to parallel conversion is used for whole word transfers out of the computer to digital spacecraft display units. It is also used to accept data from the radar measurement subsystem and can be used if desired to communicate between the two AGC's in the command module and the LEM.

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, and where the concept of positional

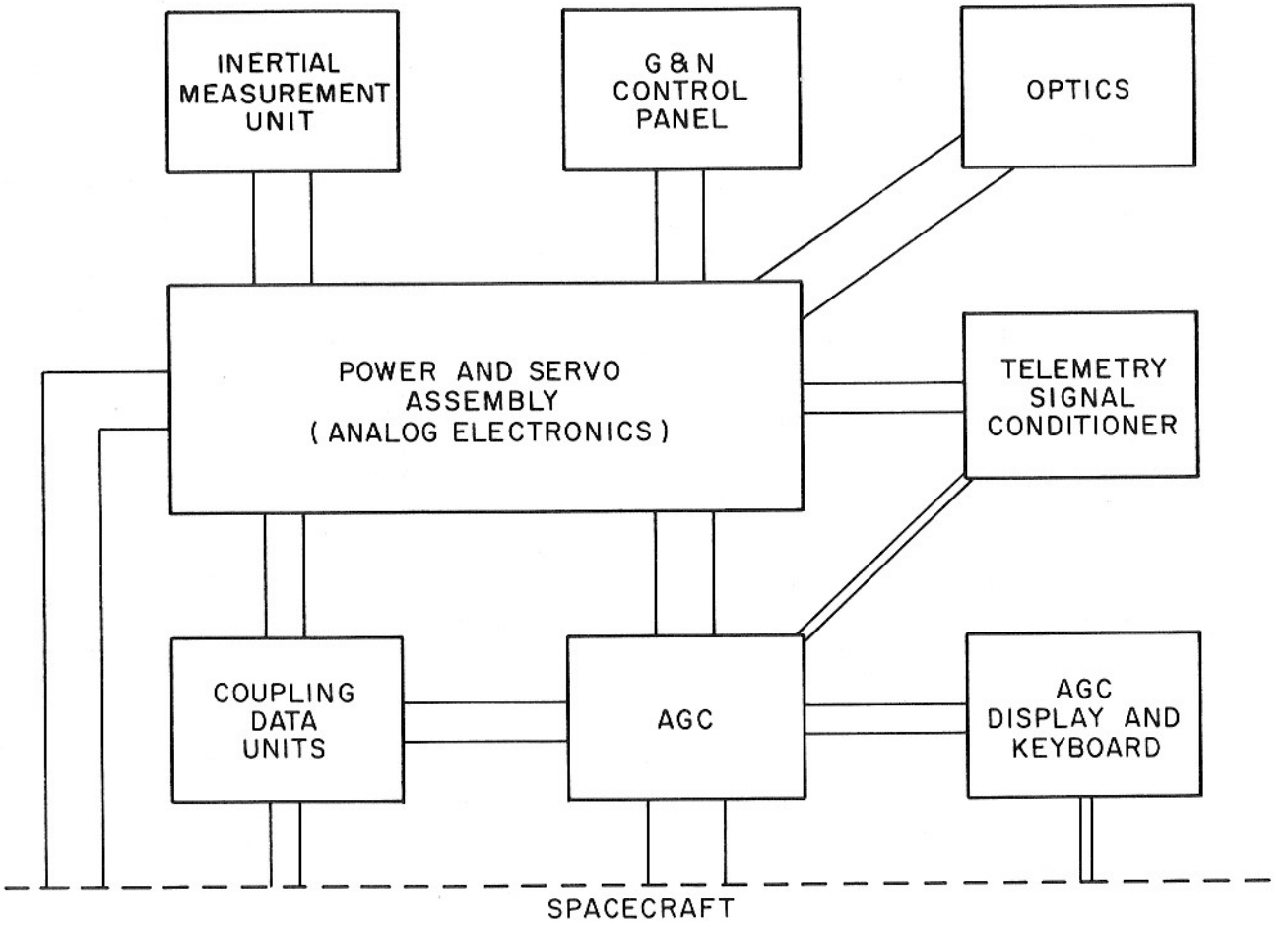



Fig. VI-18 Simplified G&N Interconnection Diagram

notation is employed. An incremental receiver counts pulses to form a word, where a serial receiver shifts pulses to form a word. Incremental information transfer was adopted as a means of analog data transmission in order to maintain high precision and standardization. In the conversion of gimbal angles and optics angles, an intermediate transformation to incremental form is made in the Coupling Data Units (CDU's), the inputs to which are electrical resolvers. The alternative to having this extra conversion is to measure the time difference between zero crossings of resolver outputs, to do which may involve equally "expensive" hardware. The Apollo accelerometers are incremental by nature, producing a pulse output to the computer for each unit change in velocity. Incremental transfer is also used for angle commands from the computer to the gyros and the CDU's, and for thrust control and certain display functions in the spacecraft. Pulses are sent in groups or "bursts" at a fixed rate. Pulse rate multipliers would be required in order to send smooth, continuous pulse trains, and these are more costly in equipment.

Discrete signals are individual or small groups of binary digits which give commands or feedback for discrete actions, such as switch closures, mission phase changes, jet firings, display initiations and many other similar controlled events. The display portion of the computer communicates with the computer proper by discrete signals in groups which carry encoded information. Serial transmission might be suitable for this communication, but would be costlier owing to the small number of bits involved.

The computer is the primary source of timing signals for all spacecraft systems; and within the guidance and navigation system it furnishes in the neighborhood of twenty time pulse signals to various subsystems.

The development of the Apollo Guidance system has followed a number of principles which reflect experience gained in previous missile-borne control systems: electrical isolation and asynchronism. Electrical isolation is an important point which has both electrical and logical implications. The computer is connected to the power supply return at a single place, thus avoiding "ground loops." Isolation of interface signals is accomplished by transformer coupling, by switch closure (relays), or by a high resistance d.c. current signal. Input and output circuits are designed so that no damage can be caused by improper connections at the interface, such as short circuits to ground. The ability to accept asynchronous inputs, i.e., those not related to computer timing signals, is desirable because it affords a design with a minimum of reference to signals outside the computer, and reduces the number of signals across the interface. This principle is particularly important for signals whose functions are to interrupt normal program sequences.

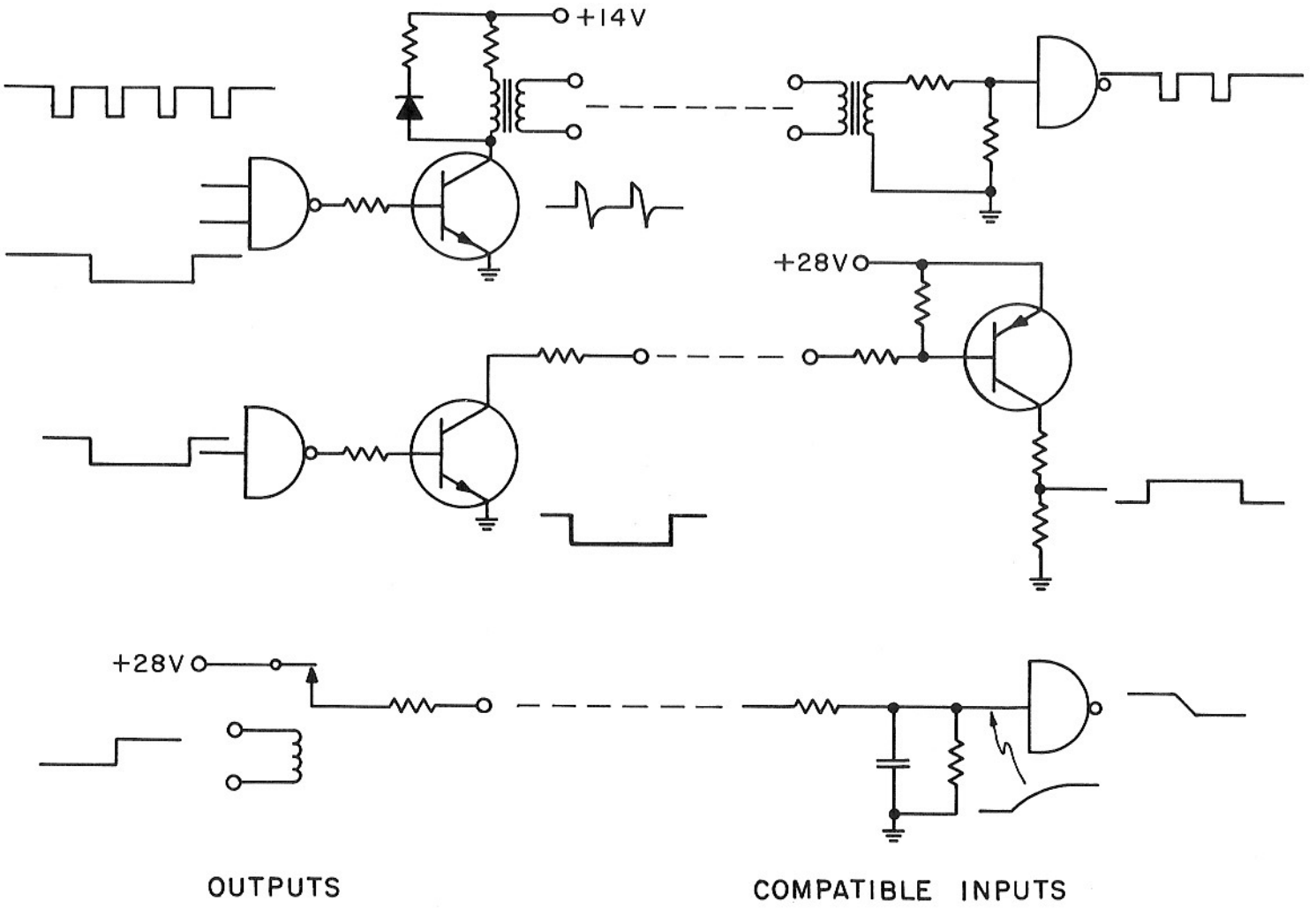



Fig. VI-19 Examples of AGC Interface Circuits

| | |
|---|---|
| Number of Discrete Inputs | 73 |
| Number of Input Pulses (Serial and Incremental) | 33 |
| Number of DC Output Discretes | 68 |
| Number of Variable Pulsed Outputs (Serial, Incremental, and Discrete) | 43 |
| Number of Fixed Pulsed Outputs | 10 |
| Number of Connector Wires | 365 |

Table VI-6 AGC Interface Summary

Inputs - Incremental and serial inputs are received in special erasable memory cells called "counter registers." They are made special by the fact that pulses received by the computer cause short interruptions of the program sequence during which one of these registers is interrogated and modified. Just which counter register is involved and what the modification to its contents is are determined by the particular input being responded to. Since there are 29 of these registers, some provision must be made to accommodate simultaneous requests for servicing several counter registers. The circuit which accomplishes this and in addition satisfies the principle of asynchronism is known as the "Counter Priority" circuit. This circuit stores incoming pulses until they can be processed. If more than one request is pending, preference is given to the one for the counter register whose address is lowest. When the "Counter Increment" cycle begins, the priority circuit delivers to the S register the address of the counter register having the outstanding request of highest priority. At about the same time, the choice is made as to how the counter's contents will be modified when they are obtained. This choice is based upon the source of the request. The counter word is shifted if it is one of the serial data receiving counters. If the "one" input received the request, a low order one is added after shifting; if the "zero" input received the request nothing is added. When the register is full, a program interrupt is requested. For a counter which is an incremental receiver, a low order plus one or minus one is added, depending upon whether the positive or negative requesting input was received. Since counter words are in the erasable memory, they are always readily accessible by any program. Each increment or shift requires a memory cycle for its execution, so the aggregate counting rate has to be limited in order to avoid an excessive use of the computer's time budget. In some instances, this requires having a logic circuit between the interface and the priority circuit which prevents the input pulse rate from exceeding a chosen level.

Two types of discrete inputs to the computer are distinguished -- interrupting and non-interrupting, the latter class being much larger than the former. Non-interrupting discrete inputs are signals which can be interrogated by input-output channel instructions. They are mechanized either as d.c. inputs through a filter to a logic gate, or as a.c. signals, transformer coupled to a flip-flop which is reset after interrogation. Interrupting inputs, in addition to appearing where they can be interrogated, announce their presence to the computer's sequence generator by initiating a program interrupt. A priority circuit similar to the Counter Priority circuit buffers the asynchronism of the inputs and resolves ambiguities caused by simultaneous interrupt requests. At the earliest permissible time, the Interrupt Priority circuit forces a transfer of control to a particular address, where the computer program interrogates the appropriate inputs and initiates any necessary action. The original program is then resumed at the point where it was interrupted. Interrupt programs never exceed a few milliseconds in running time.

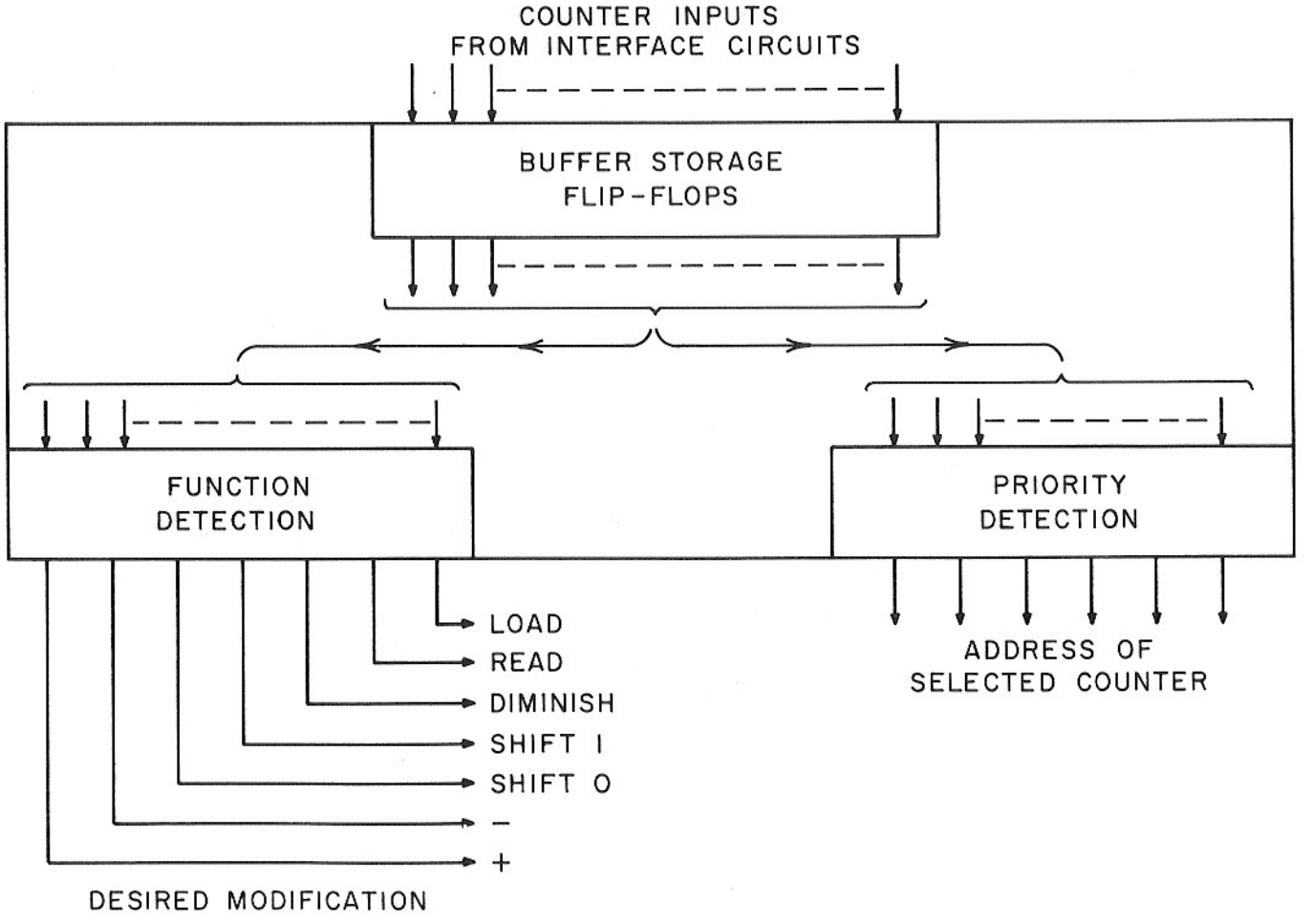



Fig. VI-20 Structure of Counter Priority Circuit

Outputs - With one major exception, the AGC uses its counter register processing facility to make all conversions from parallel words to serial and incremental pulse trains. The exception is the down data link circuit mentioned previously, where a rather costly circuit is used so that the high bit rate will not detract from the time budget. Serial outputs originate from counter registers which contain the word to be transmitted. Fifteen successive shifting requests are applied to the Counter Priority circuit; each time an overflow occurs during the shifting process, a one pulse is sent to a transformer output circuit. If no overflow occurs, a pulse representing a zero is sent to another output. This is a self-timing form of serial transmission, and is fully compatible with the serial input counter circuits.

Incremental transmission is made by placing a number in an output counter register and activating the Counter Priority circuit at a fixed rate of 3200 times per second. Each time the counter is processed, the number in the counter register is diminished by a low order one of such a sign that the register's contents approach zero. An output pulse is generated concurrently each time on one of two lines, depending on the sign of the number in the counter register. When the number has reached zero, the periodic activation of the Counter Priority circuit ceases, and the pulse burst terminates. Bursts of anywhere from one to 16,384 pulses can be generated this way. Two forms of digital-to-incremental-to-analog conversion are used in the Apollo Guidance Equipment. The simpler of the two is used for gyro torquing. During an output pulse burst, a precision current source is gated on, so that an amount of charge proportional to the desired angle change is forced through the torquing element. A single precision element is used for this form of conversion, but external storage is required for the result. In this case the storage is in the mechanical gyro angle. The second form uses a counting flip-flop register and a resistance ladder. The number in the register controls the switching of a set of precision resistors in an operational amplifier network such that the amplifier output is proportional to that number. These ladder networks are located physically in the Coupling Data Units. An incremental form of information transfer from the computer to the Coupling Data Units is used in order to minimize the interface. Several precision elements are needed for this kind of conversion, but no external storage is needed. Thus these analog signals are available as voltages for driving such equipment as attitude displays and steering gimbals for a rocket engine.

Discrete outputs are controlled either directly or indirectly by program. Typically, a discrete output is turned on by placing a one in the proper bit position of an output channel, which sets a flip-flop. If the output is transformer coupled, the flip-flop signal drives a transistor output circuit. For higher power levels, the transistor output circuit drives a relay located in the Display and Keyboard unit, and the relay's contacts are connected to the interface.

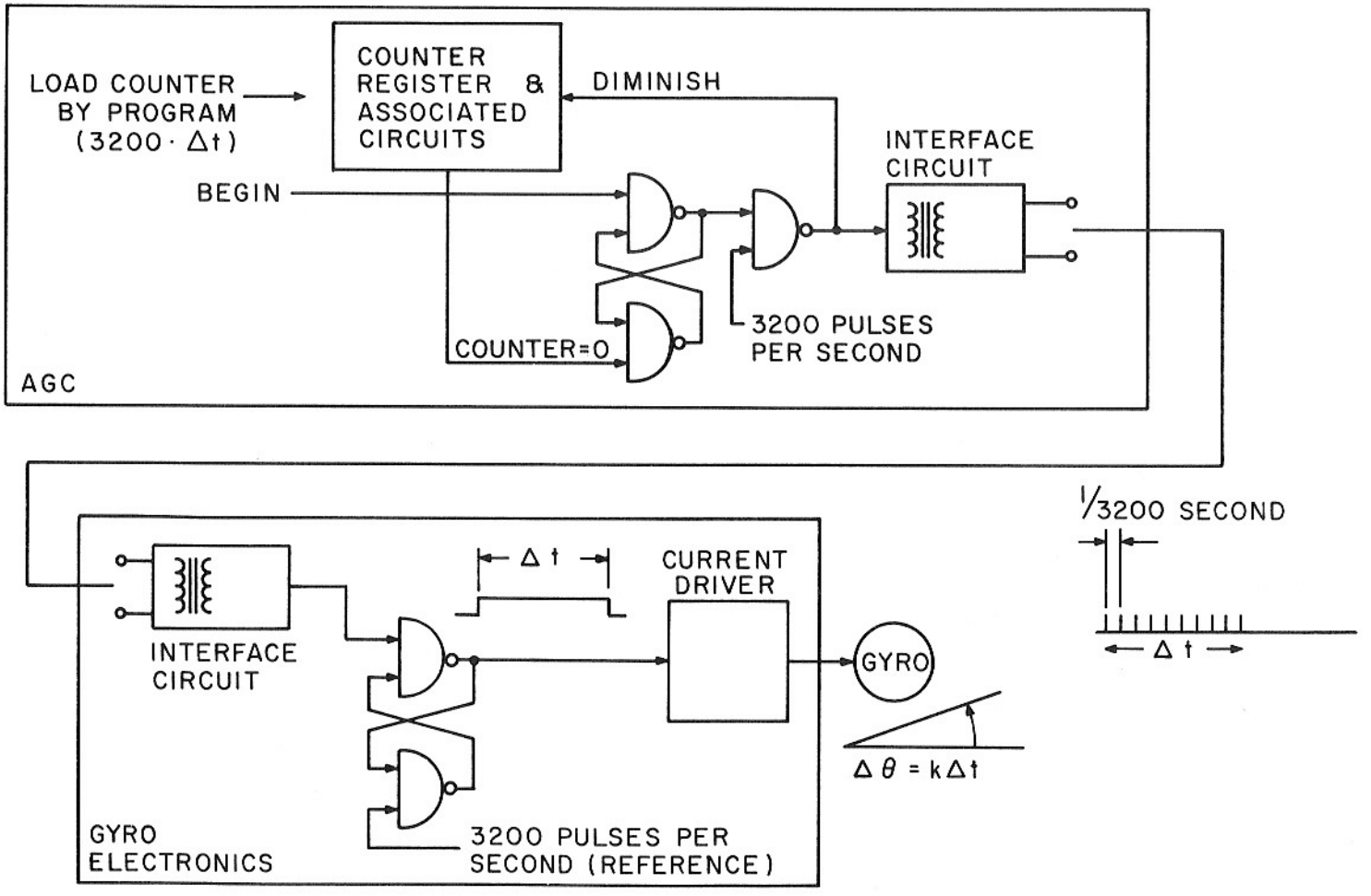



Fig. VI-21 Digital-to-Analog Conversion by TimeDuration of Precision Current

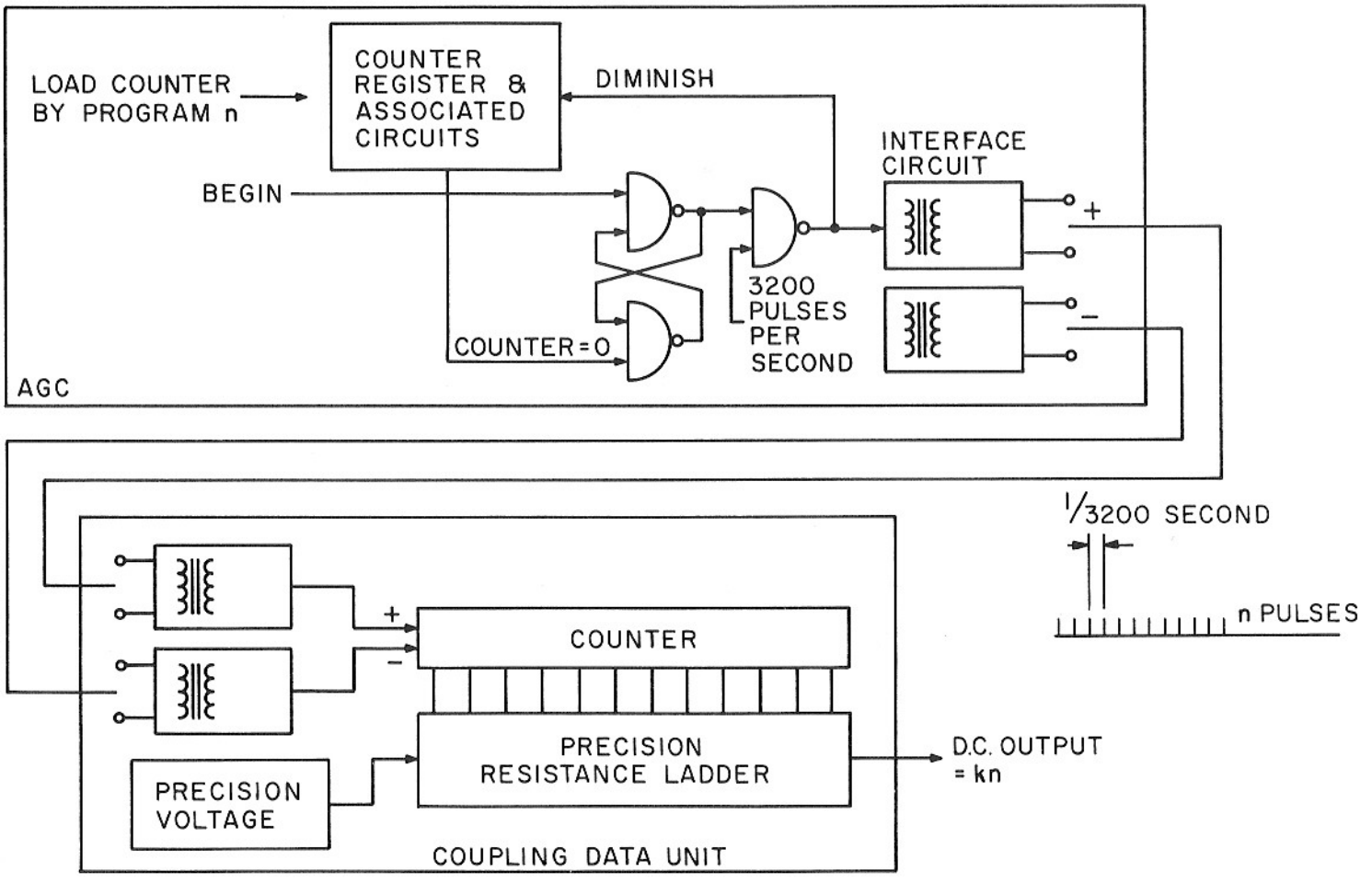



Fig. VI-22 Digital-to-Analog Conversion by Resistance Ladder

| Purpose | Serial or Incremental | Nature of Signal | Maximum Rate | Implementation |
|---|---|---|---|---|
| Digital down data link | S | 50 bursts of 40 pulses each | 51.2 KC | Special circuit |
| Digital up data link | S | 10 or fewer bursts of 16 pulses each | 1.0 KC | Counter increment |
| Digital cross links | S | 10 or fewer bursts of 16 pulses each | 3.2 KC | Counter increment |
| Altitude display | S | 2 or fewer bursts of 16 pulses each | 3.2 KC | Counter increment |
| Radar data link | S | 10 or fewer bursts of 16 pulses each | 3.2 KC | Counter increment |
| Gimbal and optics angles | I | All rates to maximum | 6.4 KC | Counter increment |
| PIPA velocity increments | I | All rates 0 to maximum | 3.2 KC | Counter increment |
| Gyro torquing | I | Occasional bursts of 0 to $2^{14}$ pulses | 3.2 KC | Counter increment |
| Engine thrust control | I | Occasional bursts of 0 to $2^{11}$ pulses | 3.2 KC | Counter increment |
| Engine steering | I | Occasional bursts of 0 to $2^{11}$ pulses | 3.2 KC | Counter increment |

Table VI-7 Partial List of Serial and Incremental Interface Characteristics

Fixed outputs are steady pulse trains which are used to synchronize other equipment. Nearly all of these are transformer coupled, and are generated simply by driving the transformer circuit with the appropriate pulse signal.

Display and Keyboard - The Display and Keyboard unit (abbreviated DSKY) is in some respects like an integral computer part, yet it is operated with the same interface circuits used for connection to other subsystems and systems. Since it serves as the channel for human communication with the computer, it needs a rather high peak data rate without either being very large in itself or having overly many wires between itself and the computer located a few meters distant along a cable.

The principal part of the display is the set of three light registers, each containing five decimal digits composed of electroluminescent segmented numerical lights. Five digits are used so that an AGC word of 15 bits can be displayed in one light register by five octal digits. No fewer than three registers are used because of the frequent need to display the three components of a vector. No more than three are used because the extra space and weight penalty cannot be justified. In addition to the numerical lights, a sign position is included in each light register. The convention is used that when a sign appears, the number is to be interpreted as decimal. Otherwise it is taken to be octal. Electroluminescent lights are small and easy to read, and require relatively little power. They are driven by an 800 cycle alternating voltage supply, switched by miniature latching relays. These have a substantial power advantage over the equivalent electronic circuitry. The contacts, well suited to the high a.c. voltage, are used for decoding, while the latching action provides a storage function. Both latching and non-latching relays are used for interfaces with other subsystems and systems, and are located in the DSKY's where they share driving circuits with the light register relays. A double-ended selection matrix is used for actuating the relays. This is organized so that one of fourteen groups of eleven relays is set at a time. Eleven signals are required from the computer to govern the configuration of the eleven relays in a group, and four more bits are used to select one group out of fourteen, making fifteen bits. The fact that this is the size of an AGC word is not entirely coincidental. This arrangement allows one word in the DSKY output channel to control enough relays to light two numbers in a light register and one stroke of a sign.

Digits are entered into the computer from a keyboard of nineteen push buttons including the ten decimal digits, plus and minus, 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. In order 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, which also makes a request to the computer's executive program to process the character at the earliest opportunity. A


Fig. VI-23 AGC Display and Keyboard

"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 present only when all keys are released. Problems from key contact bounce are avoided by having flip-flops at the input channel to receive the key code signals.

In addition to the three light registers, the display has other digit displays labeled verb, noun, and program. The keyboard has keys labeled verb, noun, enter, and clear as well as three others. These are used to enable concise yet flexible communication between man and 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 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 63 actions, or "verbs" and 63 objects, or "nouns." These verbs and nouns are, of course, displayed by number rather than by written word; so it is necessary to learn them or else to have a reference document at hand. 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 enter 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 which 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 IMU gimbals. The operator keys in the verb and noun numbers for "load 3 components, IMU gimbal angles." When the enter 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 one. When all five digits have been keyed, the enter 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.

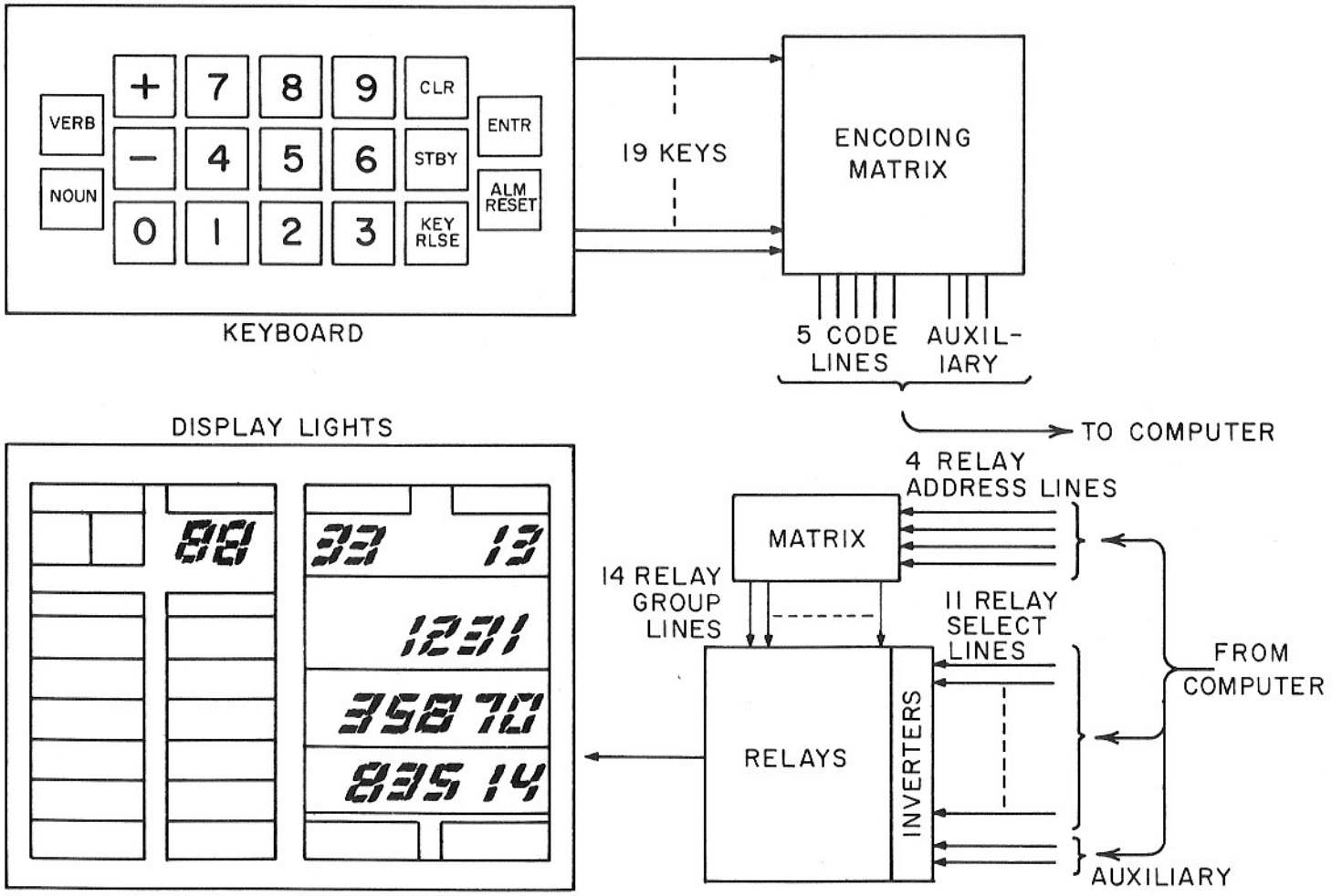



Fig. VI-24 Block Diagram of AGC Display and Keyboard Unit

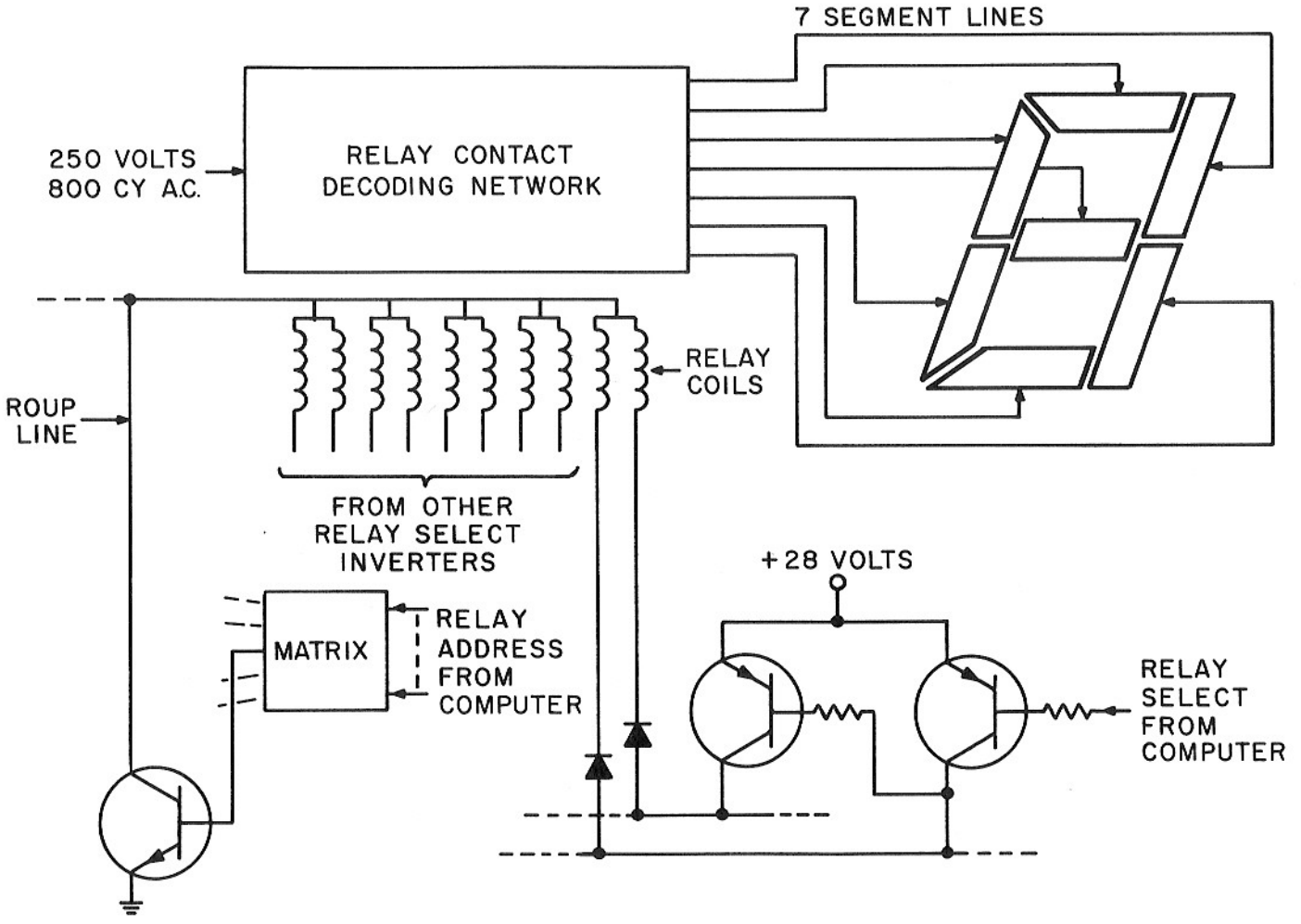



Fig. VI-25 Display Circuit Scheme

Program lights give the operator an indication showing what major programs the computer is running. Additional features of the DSKY are discrete alarm and condition lamps, a condition light reset key, and a key with which the operator relinquishes his use of the display lights to the computer. The last named function is useful because it is not always known a priori whether the operator's command has a higher priority than the computer's request. This is resolved by having the operator make the decision. If a keyboard entry sequence is in progress at a time when the computer program has a request or a result to display, a condition lamp is turned on to notify the operator of the fact. When he is ready to have the computer use the display, he need only depress this release key.

## UTILITY PROGRAMS [11, 12, 13]

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 AGC as a list of data words. An 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 VI-8.

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 which 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 and 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.

Executive - All AGC programs operate under control of the Executive routine except those which are executed in the interrupt mode. Executive controlled programs are called "jobs" as distinct from so-called "tasks," which are controlled by the Waitlist routine and completed during interrupt time. The functions of the Executive are to control priority of jobs, to permit time sharing of erasable storage, and to maintain a display discrete signal denoting "Computer Activity."

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 which gives

DP = double precision

| 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 |

Table VI-8 Partial List of Interpretive Operators

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 through the period of inactivity. When the anticipated event occurs the job is "awakened" by transfer of control to an address which 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 which 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.

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 which 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, the interrupt occurs which calls the Waitlist program. Before the interrupting program resumes 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.

Display and Keyboard - The programs associated with operation of the two Display and Keyboard units are basic to the employment of the AGC 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 interrupt to a program which 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 Keyboard and Display programs are highly sophisticated routines to which a certain amount of computer hardware is expressly devoted 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.

Chapter VI-3

# MECHANIZED AIDS TO DESIGN AND PRODUCTION

## MANUFACTURING

A number of automated processes are employed in the production of the Apollo Guidance Computer hardware. This has been done largely for the sake of minimizing human error and thus minimizing the consequent problems of reworking parts which were improperly built.

The case of the computer consists of metal pieces processed on a numerically controlled milling machine.

Signal matrices which interconnect microcircuit logic packages within a 60-package module are made semiautomatically, using punched paper tape to control a punching die which forms a matrix layer from a thin sheet of metal. Layers are insulated and stacked by hand before being hand-welded to the logic packs.

Another tape-controlled semiautomatic process is used for threading sense wires in Core Rope memory modules. Information on paper tape is used to position a rope fixture for an operator to pass a wire bobbin through core holes in which one's are to be stored, bypassing those where zeros belong. The bobbin is passed as many times as there are one's on the particular sense wire; and following each pass the tape is advanced so as to cause the fixture to be properly positioned for the next pass. When all 192 wires have been fully threaded and terminated, the module is tested on a rope memory tester, which operates the module as it will be operated in the computer. A punched tape input to the tester is compared against the information content of the module.

Interconnections between modules are made by wire which is terminated by tightly wrapping it about a rectangular post. The wrapping process may be done manually or automatically. In the AGC most of the wiring is done automatically by a machine whose information input is in the form of punched cards. The machine positions the wire over the pins with as many as 2 right angle ($90^{o}$) bends in it, cuts it, strips the insulation at the ends, and wraps both ends

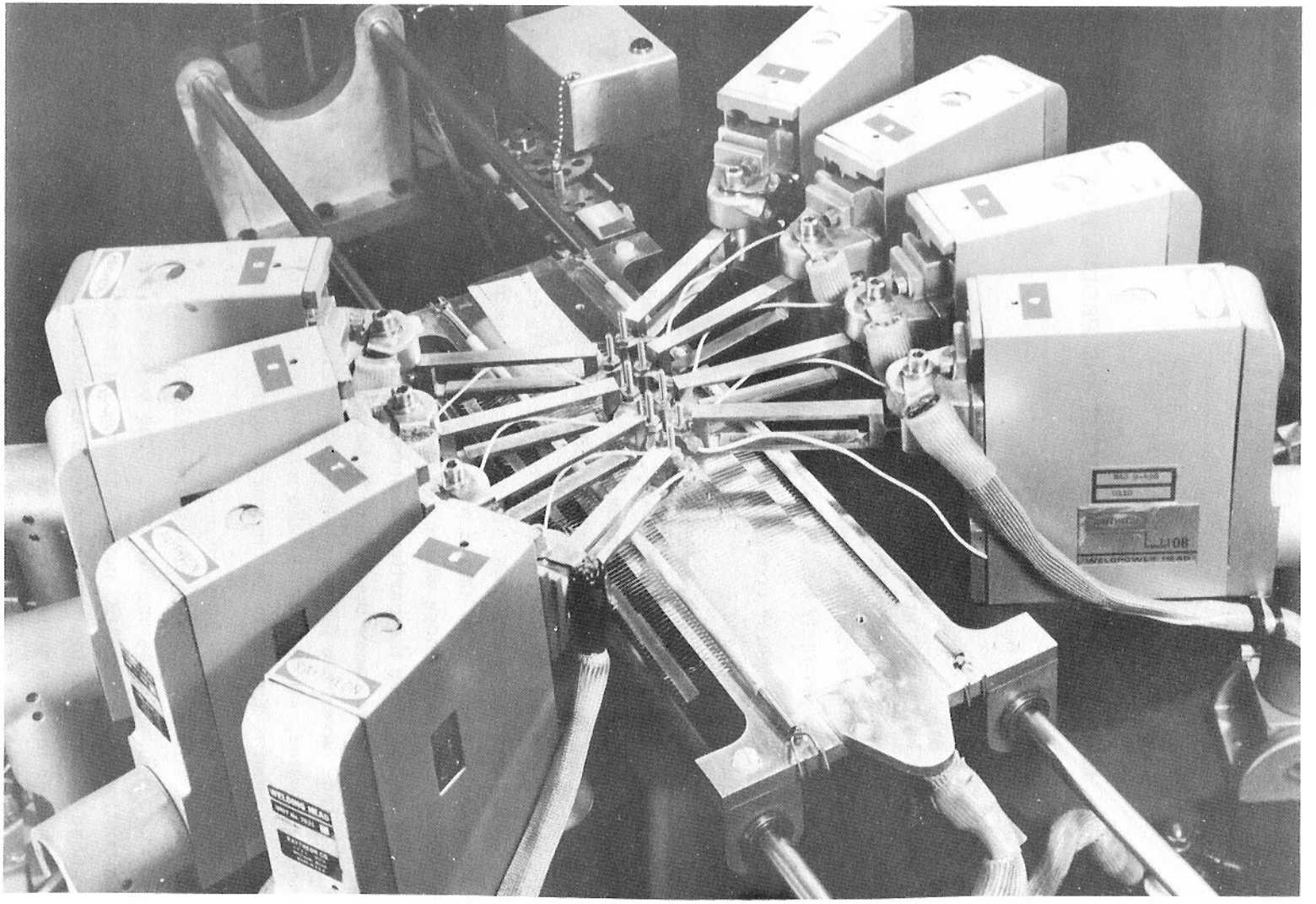


Fig. VI-26 Automatic Fabrication of Welded Matrix

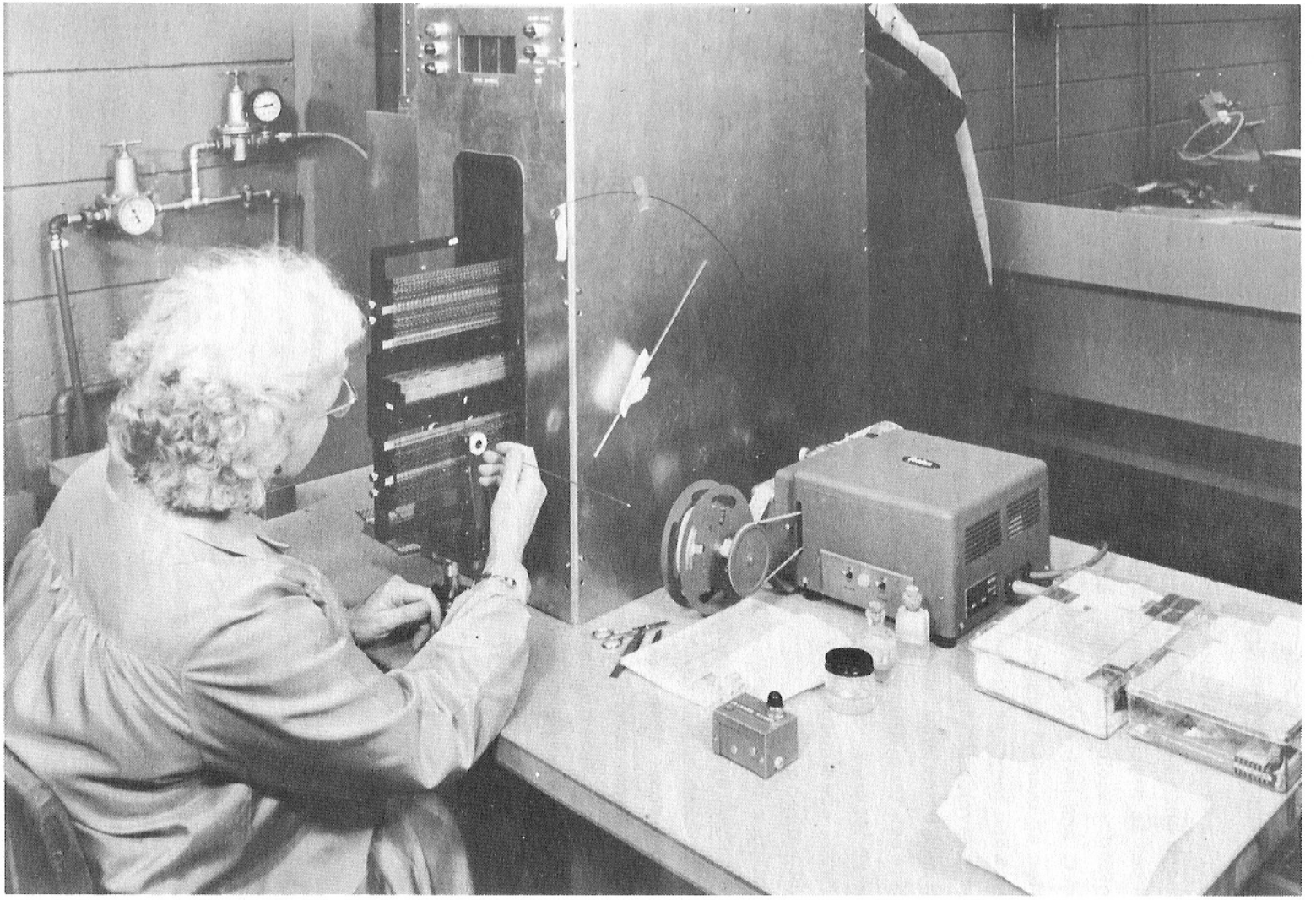


Fig. VI-27 Semiautomatic Core Rope Fabrication

## INTERCONNECTION WIRING

The raw data for interconnection of modules is necessarily originated by hand. There have been several instances of computer makers using automated logical design procedures in which the manual input was in the form of Boolean expressions to be mechanized. Such procedures, attractive as they sound, are difficult to prepare and check out, and owing to their necessary inflexibility are not as efficient in hardware utilization as manual design methods. In the AGC, logic circuits are assigned to modules when they are drawn, and terminal assignments are made at the same time. A name is given each terminal signal using the rule that all terminals bearing the same signal name shall ultimately be connected together, and no connections shall be made between terminals bearing different names. A punched card is prepared for every used terminal of every module, whether it be a logic module or any other kind. These cards are accepted by a so-called "Wirelist" program, which sorts the inputs by signal name to show the terminal groups. On each card pertaining to a logic module, a number is added stating the loading or generating nature of the circuits connected to the terminal within the module. The Wirelist program processes these numbers showing for each signal name (group) whether the load exceeds the drive or vice versa, and by how much. This is a useful feature, for whereas it is not difficult to analyze loading of a signal entirely contained in a module and hence drawn on a single sheet of paper, it is perplexing to analyze loading on a signal which goes several places and appears on several different drawings.

The Wirelist program prepares a printed document showing the terminal groups listed alphabetically, and also showing the signal names for each module terminal listed in numerical order. The information file, obtained from the cards and stored on tape, constitutes the input to the program which prepares the card deck for the wire-wrap process.

The preparation of the card deck for wire wrapping would be a tedious task without the aid of automatic data processing. Starting from a magnetic tape file listing all of the interconnections in the AGC, a computer program assigns a path to each wire, punches the card deck, and prints a listing of what it has done. The program examines in order each group of terminals which are to be wired together. Since there is no pre-specified order in which the terminals in a group are to be joined to one another, the program tries to minimize the wire length by a sequential trial and error process. Not all possible configurations can be tried owing to the excessive time required; so an algorithm is used which can find an optimal or nearly optimal solution in a short time. The program thus arrives at a definition of the connections within a group, i.e., a list of terminal pairs.

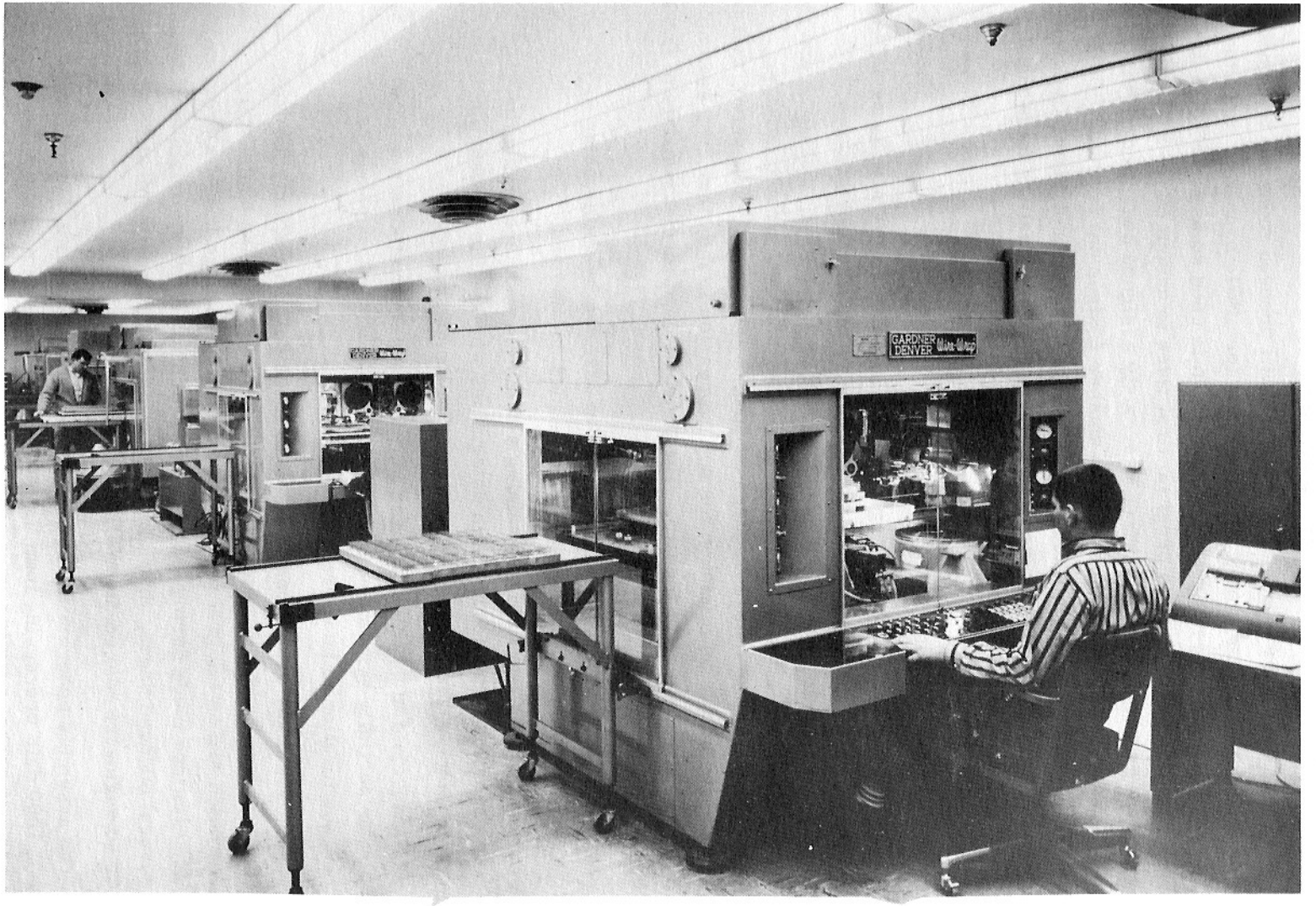



Fig. VI-28 Wire Wrapping Machine ( Photo Courtesy Gardner Denver Co. )

It is next necessary to decide how a wire will proceed from one terminal to the other, a straight line is only possible if the terminals happen to lie in the same row or column. A rectangular layout pattern is used so as to avoid choking the gaps between terminals. The program first tries to assign either a straight wire or one with a single bend. If wires previously assigned have blocked the gaps through which this would have to pass, then another assignment is tried. If all possible assignments have been tried without success, no control card is punched for this wire. Rather an entry is made on the printed listing calling for a manual insertion of the wire after the automatic wrapping is complete.

## PROGRAM PREPARATION

Assembler - The standard programming language for the AGC is an assembly language, in which each machine word of program is represented by a symbolic expression on an 80-column punched card. In fact, it can be said that there are two programming languages, basic and interpretive, and that any sizable program contains large amounts of both. As these languages are mutually exclusive, that is, no expression in one can be mistaken for any expression in the other, the assembler readily handles any mixture of the two. The punched-card format consists of three main fields.

A. Location field, which may be used to assign a symbolic name to the location of the machine word defined on the card;

B. Operation Code field, containing a symbolic code specifying the operation to be done, which determines whether the expression on the card is in basic or interpretive language; and

C. Address field, which in basic language usually contains a wholly or partially symbolic expression that specifies the address of the location to be operated upon.

There are many exceptions in detail to those definitions; only the most common use of each is given. Ample space is provided also for explanatory remarks, which are an integral part of the assembly-language file of a program. The printed listings made during assemblies, which for complete mission programs run to several hundred pages, thus constitute a medium of communication among the 50 or so engineers who do most of the programming.

The assembly program itself runs on the Instrumentation Laboratory's Honeywell 1800 data processing installation and performs in parallel its two major tasks, assembly and updating.

The assembly process translates programs in assembly language into absolute binary form for simulation and manufacturing, prints a listing in which the symbolic and absolute forms of each word of the program are displayed side by side, and prints diagnostic information about syntactical errors. The assembly process also allocates memory space to the program and to its variables and constants. The updating process maintains magnetic tape files of current programs in both assembly-language and absolute form, greatly reducing the need to handle large numbers of punched cards. For example, a program may be revised by presenting to the assembler just enough cards to specify the changes.

Assembly and updating take from less than 30 seconds to several minutes, depending on the size of the program being assembled and the amount of information on the file tape. The absolute binary files generated by assembly and maintained by updating are the input to the AGC simulator program and to program manufacturing activities that are described in later sections.

It may be instructive to trace briefly the history of part of a mission program -- re-entry guidance, for example -- from the conceptual stage to actual readiness for flight.

A. The mathematical ideas are blocked out roughly, tolerances guessed, variables and effects judged to be significant or negligible, and some such decisions are left to be settled by trial and error.

B. A procedure employing the concepts is worked out, using one mathematical model for the spacecraft with its guidance and navigation system, and another for the environment. This procedure is then employed in a data processor program for testing.

C. The program is compiled, tested, revised, and retested, until the mathematical properties of the procedure are satisfactory. Because these programs retain 2 to 4 more digits of precision than double-precision AGC programs, the variables at this stage may be considered free of truncation or round-off error. It is desirable to do as much pinning down of the procedure as possible in steps B and C, since these programs run a good deal faster than real time.

D. An AGC program is written by translating the relevant parts of the program into AGC assembly language; the more mathematical parts, such as position and velocity updating, into interpretive language, and the more logical, such as turning reaction control jets on and off, into basic.

E. In combination with the utility programs described previously, the AGC program is assembled. After two or three revisions, when the grammatical errors that can be detected by the assembler are eliminated, the program checkout is begun by use of the AGC simulator, another data processor program, which is described later. In advanced stages of checkout, this simulation incorporates the part of the program of steps B and C that models the environment. AGC simulation discovers the numerical properties of the procedure, since all effects of scaling, truncation, and round-off are present. Steps D and E are repeated until the program fulfills the goals determined in step A. Severe problems may send the engineers back to step B, or even to step A.

F. Up to this point, everything has been done on the initiative of the 3- or 4-man "working group" whose specialty is the particular phase of the mission. Now, however, this AGC program must be integrated with the rest of the mission program. Using the updating facilities of the assembler, the working group transfers its own coding to the mission program and, in cooperation with the group in charge of program integration, checks out not only its operation but its ability to "get along with" the other parts of the mission program, e.g., staying within its part of the time budget. Here again, the AGC simulator is the primary tool.

G. At this point, or sometimes before step F, it is necessary to run an AGC attached to guidance and navigation and ground support equipment. This is the last procedure devoted entirely to the checkout of an AGC program.

H. When all of steps A through G for a whole mission program have been completed, the assembly listing of the program is given the status of an engineering drawing. Only now are rope memory modules wired; and further testing is for the benefit of other subsystems as much as of the program.

It should be explained that by "mission" is meant not only a flying mission but lesser responsibilities as well. Early mission programs, shipped with computers to other contractors to aid them in testing their systems, consisted mostly of utility programs.

Simulator - The assembler is designed to detect programmer errors of the nature of inconsistencies, but it is not capable of checking program validity in general. The hazards are numerous: faulty analysis of a problem, incorrect scaling, wrong use of instructions, interference with other programs, wrong timing, endless loops, and many other pitfalls familiar to programmers. Strictly speaking, the AGC program can never be fully tested before it is operated in a system in flight. Short of this, however, it is possible to prove out a program to a high degree of confidence by simulations

of the program operating in an environment. Several possible approaches have been taken to the simulation study. In one, a computer and other parts of the guidance and navigation system are operated in conjunction with a real-time hybrid analog and digital simulation of the environment.

The other approach to simulation is an all digital simulation program which is run on the large scale data processing installation at the Instrumentation Laboratory. This is an important tool in the development of system oriented programs. Since it does not run in real time, it is possible to halt for the purpose of recording information relevant to program progress, such as periodic values of control constants or guidance and navigation variables, traces of interpretive instructions, environmental data and so forth. Initial conditions are easily set, and there is no limit, in principle, to the extent to which one can reproduce anticipated operational environments. The penalty is having running times from 2 to 40 times as long as real time (10 times is typical).

The Simulator comprises 3 major sections. The first simulates the AGC, and even operates an AGC Display and Keyboard (DSKY) connected on-line to the data processor. The second simulates the environment, and is constantly being added to and improved as operating experience is gained. These two sections run largely independent of one another, as their functions are basically incompatible. One is a function of elapsed time, and the other is a function of AGC program execution status. The third section exists for the purpose of communication between the other two. It is capable of representing the environment to the computer over short periods of time while the two operate in isolation. From time to time the simulation halts while the AGC and environment sections reconcile with one another. The communicator section is re-initialized; and the simulation proceeds, with the communicator extrapolating the recent past history of the environment.

Information which is recorded out is subsequently edited in such a way that system analysts can easily process it with their own programs in order to make error analyses, study correlation of events, or perform any other mathematical or editing operation.

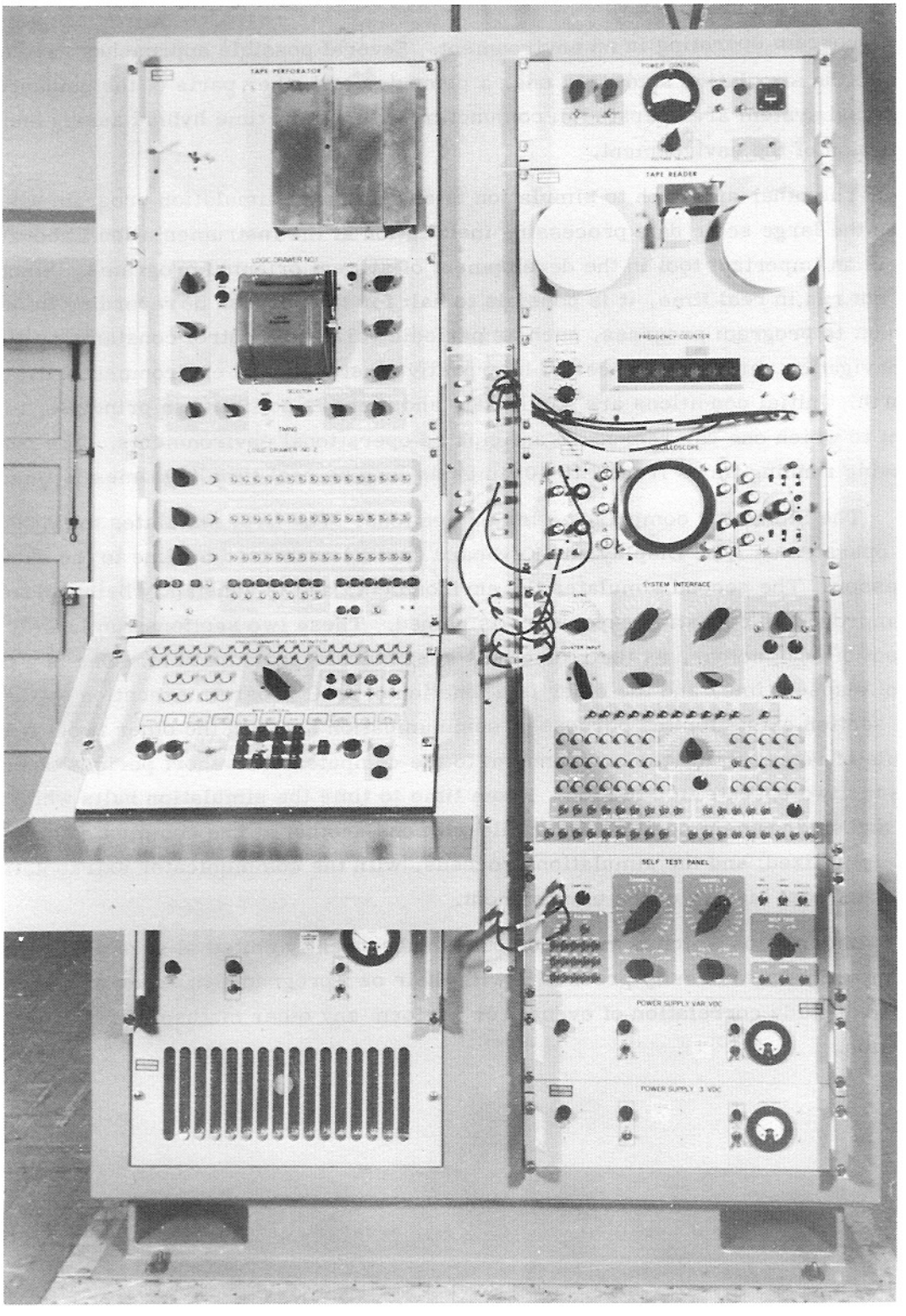



Fig. VI-29 Computer Test Set

Chapter VI-4

# GROUND SUPPORT EQUIPMENT

## TEST SET

Because of volume and weight restrictions, guidance computers are built without the extensive maintenance features found in data processing computers. During all stages of design, assembly, and field testing it is important to have facility for manual intervention into the computer's operation, as well as a means for certifying thall all circuits are properly operating. This facility is provided by a separate unit which connects to the conputer through what is usually a separate interface.

The unit which does this job with the AGC is called the AGC Test Set. It contains a number of flip-flop registers which can be made to serve various roles by selecting among numerous modes of operation. Over 100 signals are exchanged between the AGC and Test Set through an interface called the test connector. The Test Set has access to the AGC's write buses. It can sense them and also force signals upon them. This feature, together with the AGC's timing pulses and central register control pulses, permits the Test Set to follow the progress of AGC programs.

In its monitor mode the Test Set uses flip-flop registers to duplicate, or mimic, the contents of AGC central registers. Any of these registers may be displayed in lights on the control panel. Since in particular the S, or address, register and the G, or memory local, register are displayed, it is possible to see what is stored in memory. The successive contents of any erasable register can be displayed by commanding the Test Set to sample the G register and display its contents each time the address is the same as the address specified by hand set switches on the panel.

Facility is also available for causing the computer to halt when it interrogates a specified address, or when the content of a specified register reaches a specified value, or at the end of each instruction, memory cycle, or alarm. The computer may be made to proceed from where it is, or it may be started at any desired instruction. Any register may be read to the lights whether or not it is accessed by program, and any erasable location may be loaded from the Test Set with any desired value.

The Test Set contains circuits for exercising and testing all of the AGC's interface circuits. A separate cable connects the Test Set with the AGC's system connector and a switch panel causes an oscilloscope to be connected to an AGC output, or else a signal generator of appropriate characteristics to an AGC input. This feature is used to make a detailed test of output signals, including their rise time, amplitude, duration, and any other important characteristics. A faster and less complete check of the interfaces can be made by a special connector which exercises pulse inputs by connecting them to pulse outputs, and exercises DC inputs by connecting them to DC outputs and supplying a dummy load. The check can then be made by an AGC program read into the erasable memory.

AGC MONITOR

Before an AGC program is wired into fixed memory modules, it must be exercised on an AGC. For this purpose, there exists a form of ground support equipment incorporating some features of the Test Set together with an erasable type of memory whose contents are sent to the AGC when it interrogates a fixed memory address. This machine is the AGC Monitor, so called because it encompasses the monitor feature of the Test Set. The memory control circuits of the Monitor are arranged so that only a part of the AGC fixed memory is read from the Monitor, while the remainder is read from the AGC. This is done, moreover, without having to remove fixed memory modules from the AGC. Rope simulations are done in 1024 word segments. When the AGC Fixed Bank register and S register are observed to be set to access a simulated bank, the sensing of the AGC fixed memory is inhibited by a signal from the monitor while the simulated word is transferred into the G, or memory local, register of the AGC.

The memory used in the Monitor is a set of 9 erasable memory units, each containing a 4096 word coincident-current ferrite core stack with driving electronics similar to those in the AGC. The logic in the Monitor and in the Test Set is made from microcircuit NOR gates. One reason this is done is to put to use those gates which are not found qualified for flight hardware, but are still satisfactory for less stringent environments.

Chapter VI-5

# CONCLUSION

Guidance computer engineering is a simultaneous effort in mechanical, electrical, and logical design disciplines.

In order to obtain high efficiency in terms of performance, volume, and power consumption, guidance computers are designed to work in a single specific system, unlike most commercial computers. They commonly have fewer and less flexible instructions, shorter word length, and less complex arithmetic units. Compactness and short term reliability predominate over considerations of programming ease, maintainability and manufacturing cost. Whereas the commercial computer designer strives to maximize answers per month, the guidance computer designer seeks the capability of handling high peak loads, and is concerned with answers per second.

The guidance computer receives data from various sources and delivers answers to various destinations over tens or hundreds of signal paths, each requiring appropriate conditioning circuitry at origin and at destination. One of the challenging design problems is to minimize the number of these interface signals, and moreover to minimize the number of different circuits used.

The AGC is quite representative of the state of the art in guidance computers as contrasted with the rest of computer technology. Most conspicuous of the attributes common in guidance computers are the extensive use of microcircuits and high density interconnections, a dense fixed memory of about half a million bits and a small erasable memory, and a short word length.

The computer was designed to employ certain utility programs. The Interpreter program allows efficient expression of double precision matrix programs for navigation, attitude control, and steering. The Executive program allots computer time among various jobs according to a priority schedule. The Waitlist program provides interrupted entry to other programs at specified intervals of real time.

Guidance computers are often supported by commercial computers for automatic programming and in various areas of mechanical and electrical design. A large scale computer is used in connection with the Apollo Guidance Computer in several respects. It assembles and makes simulation runs on programs; it generates the input

card decks for automatic wire wrapping machinery used in computer manufacture; and it also prepares punched tape for use in fixed memory fabrication, and information input to ground support equipment.

The constraints on the guidance computer designer are severe. In addition to the requirements of size, performance, and reliability is the urgency for early delivery which stems from trying to get the best equipment possible without introducing unnecessary delay in flight schedules. For this reason, we may expect that guidance computer engineering will continue to be a highly productive, competive discipline for years to come.

ACKNOWLEDGEMENT

The Apollo Guidance Computer has been used here to exemplify the state of the art in broad scope, from hardware to software. The AGC owes its existence to the pioneering work of Ramon L. Alonso, Eldon C. Hall, and J. Halcombe Laning, Jr. Its development has encompassed the efforts of many workers, some of whose contributions are referenced below. Others, whose contributions are substantial but unwritten include D. J. Bowler, E. J. Duggan, J. S. Miller, R. F. Morse, and H. A. Thaler. To them and to the Raytheon Company, manufacturers of the Apollo Guidance Computer, the author is grateful for advice and assistance in preparing this description of their achievements.

BIBLIOGRAPHY


1. Proceedings of the Conference on Spaceborne Computer Engineering, I. R. E., P. G. E. C., Anaheim, California, October 30 - 31, 1962.

2. M. M. Dickinson, J. B. Jackson, and G. C. Randa, "Saturn V Launch Vehicle Digital Computer and Data Adapter," Proceedings 1964 Fall Joint Computer Conference, Spartan Books, Inc., Baltimore, Maryland, 1964.

3. H. Wyle, S. Z. Rubenstein, L. Miller, and S. E. Githens, "The Logical Design of the D26C: A Microminiature Real Time Processing System," National Electronics Conference, May, 1964.

4. R. L. Alonso, H. Blair-Smith, and A. L. Hopkins, "Some Aspects of the Logical Design of a Control Computer: A Case Study," IEEE Transactions on Electronic Computers, Vol. EC-12, No. 5, December, 1963.

5. R. L. Alonso and J. H. Laning, Jr., "Design Principles for a General Control Computer," Institute of Aeronautical Sciences, New York, New York, S. M. Fairchild Publication Fund Paper FF-29, April, 1960.

6. R. L. Alonso, A. I. Green, H. E. Maurer, and R. E. Oleksiak, "A Digital Control Computer; Developmental Model 1B," Report No. R-358, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, April, 1962.

7. T. C. Taylor, "A Proposed Circuit Structure for Computer Logic Based on Semiconductor Flat Packs Interconnected by Multilayer Circuit Boards," Report No. E-1699, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, December, 1964.

8. J. Partridge, L. D. Hanley, and E. C. Hall, "Progress Report on Attainable Reliability of Integrated Circuits for Systems Application," Report No. E-1679, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, November, 1964.

9. D. Shansky, "Erasable Store Mod 3C," Report No. E-1158, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, July, 1962.

10. R. L. Alonso and A. L. Hopkins, "The Apollo Guidance Computer," Report No. R-416, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, August, 1963.

11. C. A. Muntz, "User's Guide to the Block II AGC/LGC Interpreter," Report No. R-489, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, April, 1965.

12. A. I. Green and J. J. Rocchio, "Keyboard and Display System Program for AGC," Report No. E-1574, Instrumentation Laboratory, MIT, Cambridge, Massachusetts, August, 1964.

13. T. J. Lawton and C. A. Muntz, "Organization of Computation and Control in the Apollo Guidance Computer," Instrumentation Laboratory, MIT, Cambridge, Massachusetts, April, 1965.