BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D.C. 20024

COVER SHEET FOR TECHNICAL MEMORANDUM

TITLE- Onboard Checkout of a Mid-70's Space Station

TM- 69-1031-3

DATE- June 6, 1969

FILING CASE NO(S)- 730

AUTHOR(S)- J. R. Birkemeier

FILING SUBJECT(S)- (ASSIGNED BY AUTHOR(S)- Inflight Checkout
Onboard Computers
Space Stations

ABSTRACT

The computer system on board a mid-70's space station will be able to handle an extensive amount of inflight checkout.

Principal checkout functions are sampling data from test points, comparing test point data to various limits, predicting failures through trend analysis, and diagnosing failed or failing systems. These functions are required primarily for the safety of the flight crew and the successful conduct of the mission, and secondarily for the gathering of engineering and design data related to the operation of spacecraft systems.

The memory and speed requirements of the onboard computer were evaluated for a wide range of test points. For a variety of checkout program options exercised on sample systems with a total of 1830 test points, memory sizes of 35-93K words, processing rates of 29-85K operations per second, and input-output handling of up to 3200 quantities were found necessary.

The memory and speed requirements are within the capabilities of some aerospace computers already available or under development, which could handle over 2500 test points. Computers of the mid-70's are expected to have sufficient memory and speed to handle as many as 10,000 test points. The large number of inputs and outputs need special interface equipment but should cause no great problem.

Diagnostic testing, particularly that requiring the application of stimuli to spacecraft systems, introduces a number of complications relating to timing and control of the checkout program. Resulting software problems should be resolved early.

SEE REVERSE SIDE FOR DISTRIBUTION LIST

DISTRIBUTION

COMPLETE MEMORANDUM TO

CORRESPONDENCE FILES:

OFFICIAL FILE COPY
plus one white copy for each additional case referenced

TECHNICAL LIBRARY (4)

NASA Headquarters
W.O. Armstrong/MTX
R. F. Bohling/RVA
J. L. East/REI
S. W. Fordyce/MLA
E. W. Hall/MTG
T. A. Keegan/MA-2
R. L. Lohman/MTY
D. R. Lord/MTD
R. F. Lovelett/MTY
E. J. Meyers/MTE
E. P. O'Rourke/MSR-1
A. D. Schnyer/MTV
P. D. Schrock/MLT
G. A. Vacca/REI
M. G. Waugh/MTP
J. W. Wild/MTE

Marshall Space Flight Center
W. B. Chubb/R-ASTR-NGD
H. Kerner/R-COMP-RD

Goddard Space Flight Center
G. H. Ludwig/560

Electronics Research Center
J. W. Poduska
R. M. Snow
D. Van Meter
G. Wang

Manned Spacecraft Center
W. C. Bradford/EB5
G. B. Gibson/EE
C. P. Hicks/FA-4
J. B. MacLeod/FS
M. J. Quinn/FS

COVER SHEET ONLY TO

COMPLETE MEMORANDUM TO

Kennedy Space Center
W. H. Boggs/DE-FSO

Bellcomm
F. G. Allen
G. M. Anderson
J. H. Bredt
A. P. Boysen, Jr.
K. R. Carpenter
C. L. Davis
W. W. Elam
R. Gorman
D. R. Hagner
J. J. Hibbert
B. T. Howard
D. B. James
C. E. Johnson
A. N. Kontaratos
M. Liwshitz
H. S. London
D. Macchia
E. D. Marion
J. Z. Menard
G. T. Orrok
S. L. Penn
I. M. Ross
J. A. Schelke
R. L. Selden
M. H. Skeer
J. W. Timko
W. B. Thompson
A. R. Vernon
J. E. Volonte
R. L. Wagner
All Members Department 1031
Center 10 File
Department 1024 File
Central Files
Library

If the entry module is active during the mission, it could check itself out. If it is normally dormant, the space station would monitor it for safety purposes. In an emergency, the entry module would need to perform some checkout of itself, and may also be required to monitor and test the space station to determine whether to abandon the station or wait until it is again habitable.

BELLCOMM, INC.
955 L'ENFANT PLAZA NORTH, S.W. WASHINGTON, D. C. 20024

SUBJECT: Onboard Checkout of a Mid-70's Space Station - Case 730

DATE: June 6, 1969

FROM: J. R. Birkemeier

TM-69-1031-3

TECHNICAL MEMORANDUM

1.0 INTRODUCTION

Studies have been conducted to determine the level of checkout that can be performed on board a space station, and the degree of automation that can be incorporated into the inflight checkout. For purposes of estimating computer requirements, it was assumed that all mathematical, logical and data manipulation tasks would be handled by the onboard computer system. The primary purpose of the present report is to demonstrate the feasibility of this approach. A subsequent study will consider the tradeoffs between onboard and ground-based checkout functions.

1.1 Significance of Inflight Checkout

Inflight checkout refers to the tasks involved in monitoring the operation of spacecraft systems during a mission, assessing the performance of those systems, and diagnosing them to isolate malfunctions and failures.

The primary purposes of inflight checkout are the safety of the crew and the successful conduct of the mission. An important secondary purpose is the gathering of engineering and design data on spacecraft systems. Space stations are likely to be made only in limited quantities (one or two of a kind), be of new designs, and not be recovered intact at the end of a mission. Although the systems on board the space stations may be made in larger quantities than the stations themselves, these systems may benefit from few if any flight tests before their use on the stations. Therefore, the data gathered from inflight checkout will be especially valuable in designing following generations of spacecraft systems.

Additional uses of inflight checkout data will be mission planning functions, like experiment scheduling and resource management, and the correlation of system performance with experimental data.

1.2 Inflight Maintenance

The topic of inflight maintenance is generally bound to that of inflight checkout. It is interesting to cite what happened in the Apollo program.

In 1963, Bellcomm and RAND performed a study[1] of inflight checkout for the Apollo lunar landing mission and concluded that inflight maintenance would be required for several spacecraft systems on that mission. The checkout scheme adopted was rather simple: Comparators to monitor signal levels against preset limits, and test equipment like a VTVM manually directed to test points. However, as the project evolved, functional redundancy was gradually developed for critical, failure-prone systems. As a result, no explicit requirement for inflight checkout or maintenance exists today in the Apollo program.

Possibly, the development of systems for a space station will follow a course similar to that for Apollo, and inflight maintenance may prove to be dispensible for most systems. However, the Apollo lunar landing mission takes less than two weeks, while the space station must operate for two years or more. Furthermore, weight restrictions prevent the carrying of spares on Apollo, while the space station will be less limited. Therefore, while inflight maintenance may not be as frequent as some designers now foresee, it will almost certainly be both needed and feasible on a two-year mission.

2.0 EXAMPLES OF INFLIGHT CHECKOUT SYSTEMS

Inflight checkout systems have been specified or proposed for all the latest large or complex military and commercial aircraft. Two such systems, whose characteristics are summarized in Table 1, illustrate the different approaches that can be followed in checkout. MADAR (Malfunction Detection, Analysis, and Recording), designed by Lockheed for the C-5A transport, uses a central digital computer to control checkout functions. These functions are passive only, as far as the airborne systems are concerned: Test points are sampled for subsequent processing in the computer, but no stimuli are applied to assist in diagnosing system malfunctions. Signals from test points are multiplexed and routed in analog form, and converted to digital form just before entering the computer. Messages indicating failures or their passing are printed for the flight crew. The computer program includes priority logic to direct the crew when multiple failures occur, and performs a short-term trend analysis to locate incipient failures. Data recorded in flight can be processed on the ground for further study. MADAR can also assist the crew in performing inflight maintenance and repair on systems that are accessible and for which the necessary spares are carried.

In contrast to the centralized approach of MADAR, ILAAS (Integrated Light Attack Avionics System) for the A-7 aircraft relies on built-in test equipment at the subsystem and modular level. The various subsystems apply local stimuli periodically and set indicators when troubles are detected. The status of

these indicators is routed in discrete form to a central evaluation network. Malfunctioning subsystems are identified for the pilot. When a failure makes a desired mode of operation impossible, ILAAS automatically switches to an alternate mode if one is available, even with degraded capability. Malfunction indicators on the modules remain latched to assist the ground crew in isolating and correcting the failure after the flight.

The long mission times typical of space stations make inflight maintenance a more likely requirement for spaceborne than for airborne systems. As a result, spaceborne checkout systems need access to more test points, to allow fault isolation in greater detail. Table 2 summarizes the characteristics of two onboard checkout systems (OCS) for spacecraft. The Martin OCS has been funded in a series of NASA contracts; a laboratory model is now in operation, and a flight-packaged version is under development for MSC. This system, which evolved from the preflight checkout equipment and techniques used at KSC, is intended to provide a general-purpose checkout capability on board a space station. A flexible software package in a central digital computer allows the flight crew to define test procedures in the course of the mission. These procedures may involve repetitive testing of spacecraft systems, application of closely specified stimuli through a centralized generator, and the activation of other systems as a result of tests.

As part of a Grumman effort, RCA has studied the checkout requirements for a large space station. Several features of the RCA OCS are in sharp contrast to those of the Martin system. RCA places heavy reliance on passive monitoring. If stimuli are needed, they should be generated as locally as possible, rather than being routed from a central source in the space station. Although space station autonomy may demand that the crew be able to change the test program, such changes should preferably be developed on the ground and uplinked to the onboard computer.

Either Martin's or RCA's OCS could be installed in the space station at the factory and used during some, if not all, phases of preflight checkout and training. This practice could result in less dependence on ground systems before launch, and better correlation of inflight data with prelaunch values.

3.0 CHECKOUT FUNCTIONS

Of the four inflight checkout systems mentioned above, only one, ILAAS, relies heavily on built-in test equipment (BITE); the other three depend on a central digital computer to control their operations. Furthermore, all four systems interact with the crew to some extent, by displaying checkout information or by

receiving instructions or other data. Therefore, it is expedient to look at checkout functions broadly, to determine how they might be performed by the computer or the crew.

3.1 Man-Computer Considerations

Although the specific checkout tasks performed by the space station computer may vary somewhat for different systems, configurations, and missions, some general guidelines can be applied to the division of tasks between the computer and the crew.

Computers are well adapted to performing routine, repetitious tasks. Consequently, system monitoring functions could be highly automated to relieve the crew of monotonous, recurrent chores like observing system parameters and comparing these values to various limits. The computer could also be used to correlate a number of simultaneous measurements, and to analyze successive parameter values for trends indicative of malfunctions.

The ability of the crew to observe and assimilate large amounts of disparate data, particularly in unforeseen circumstances, could be used to advantage in checkout. A typical crew task would be scheduled inspections of the spacecraft and its systems to detect leaks, films and deposits; loose, dented, broken, or worn parts; and changes in color, texture, or sound. These symptoms may indicate troubles that would become more manifest later. However, earlier detection by the crew could result in a less hazardous condition and could allow repair to occur at a more relaxed pace. This checkout function by the crew could be especially important with systems or modules for which it was impractical to provide enough sensors for every possible trouble, or which were not understood well enough, because of their newness, to allow adequate diagnostic software to be written to cover all circumstances. Furthermore, some systems may have such a large variety of failure modes that the cost of finding these modes, devising proper procedures, and developing adequate computer programs would be too expensive in time or money. A similar situation could exist with the space station as a whole; all combinations of systems and their failure modes could not be examined. In these instances, the flexibility of the crew to respond to unforeseen occurrences could promote the successful completion of a mission.

In some instances, the crew and the computer would work together. In typical joint activities, the crew would direct the computer in the execution of diagnostic testing, the presentation of additional or specialized data displays, and the changing of limits or procedures in test programs.

3.2 Suggested Checkout Plan

This section presents a background for the further discussion of checkout requirements. While some aspects of checkout may change with more specific definition of spacecraft systems, the general procedures outlined here should remain largely intact.

3.2.1 System Monitoring

Space station systems should operate satisfactorily most of the time. For example, a Boeing study[2] indicates a mean time between failures (MTBF) of 7.0 days for all systems on a two-year manned spacecraft mission. Even the most failure-prone system, the environmental control and life support system (EC/LSS), has an MTBF of 20 days. The close monitoring of these systems would, therefore, be a very monotonous chore for an astronaut, and should be assigned to the computer as a routine task.

The monitoring of system parameters would be done by reading a test point and comparing its value to a series of limits stored in the computer. Most of the time, every parameter would lie within its designated limits, and no indication of trouble would be generated by the computer. However, when a parameter was found to lie outside a limit, the crew would be alerted by various auditory and visual indicators. Tones could be emitted and panel lights could be turned on to show the crew which system was responsible, and alarms could ring for highly critical malfunctions. Alternatively, a spoken message to attract the crew's attention could also be produced, either by being assembled from prerecorded speech elements or by being synthesized by the computer. In addition, a printer would write the failure messages generated by the computer, so that the crew would have a record of the failure conditions as discovered by the computer.

Not all failures would require immediate attention. The tightest limits could correspond to a caution condition, to which the crew should attend within some arbitrary period like two hours. Broader limits would indicate a greater departure of a parameter from its expected or nominal value. A parameter crossing such a limit would need attention more urgently, say within a half hour, and the crew would be warned accordingly and may also be advised to don space suits. The outermost limits would represent emergency conditions, which may demand immediate attention from the crew or may result in their being instructed to leave the space station, at least temporarily.

3.2.2 Backup Monitors

The computer itself is subject to malfunctions, as are the sensors that measure system parameters. Therefore, some degree of backup would be needed to alert the crew to dangerous conditions when the checkout system was inoperative. Parameters most directly involved in crew safety, like cabin atmosphere pressure, temperature, and composition, would be monitored by redundant sensors that could activate alarms independently of the computer.

3.2.3 Trend Analysis

Trend analysis would be a useful adjunct of system monitoring. Using recent values of a parameter as a basis, the computer could detect high rates of change, abrupt departures from steady-state conditions, and gradual drifts toward out-of-tolerance situations. The occurrence of one of these events does not by itself imply a system malfunction; for example, a sudden increase in the carbon dioxide content of the atmosphere may be detected because the crew is awakening and beginning to move around the cabin. Some correlation of measurements would, therefore, be needed to prevent false alarms from being generated.

3.2.4 Diagnostic Testing

When a parameter was found to be outside a limit, the computer could initiate diagnostic testing to find the cause of the problem. To minimize disturbances to operating spacecraft systems, the first diagnostic testing procedures would be passive if at all possible. If these procedures failed to isolate the trouble, active tests could be called. In general, however, no stimulus should be applied without notification and approval of the crew. This precaution has a twofold purpose: (1) it alerts the crew to the possibility of abnormal system behavior during testing, and (2) it reduces the likelihood that a test will interfere with an experiment in progress or with a tight schedule of astronaut activities.

However, there may also be instances in which departures from this principle would be preferable. BITE, particularly in electronic systems, could often be used without affecting other space station systems. Furthermore, if all crew members are asleep, or if a failure is so critical that immediate diagnosis and remedial action are demanded, stimuli may be applied automatically by the checkout system. The extent to which the crew should direct or interfere with automatic testing is an area that needs more detailed analysis of space station systems before definitive decisions can be made.

The isolation and repair of some malfunctions may require skills beyond those of a typical astronaut, even when the computer is available to assist him. One or more crew members may, therefore, need specialized training as system engineers to handle such situations. Alternatively, the flight crew could depend on assistance from technical specialists on the ground. When the computer detected a malfunction, it would then collect data on the offending system, either for immediate transmission to the ground, or for recording for subsequent playback to the ground. This aspect of checkout will be covered more fully in a later study.

3.2.5 Summary

The suggested checkout plan uses the computer for routine monitoring of space station systems. When abnormal values of system parameters are detected, the crew is alerted, with the means of alerting being dependent on the nature, severity, and criticality of the failure. Passive diagnostic testing could be performed by the computer automatically, but the crew should generally be involved in the application of stimuli. The ground may also assist in the isolation and repair of malfunctions that require specialized technical skills.

The installation of a comprehensive checkout system on the space station before launch opens the way to using this system to perform a considerable amount of preflight checkout. Besides the data consistency mentioned in Section 2.0, the use of onboard checkout equipment for preflight checkout could lead to greatly reduced requirements for checkout equipment, programs, and personnel at the launch site. An additional benefit could be the training of the ground and flight crews in details of checkout that may assist them later in the course of the mission. This subject is another aspect of the space-ground tradeoff question that will receive further attention in a subsequent study.

4.0 COMPUTER FUNCTIONS

Obviously, the computer can play a very important role in inflight checkout operations. An evaluation of the computer requirements is, therefore, an essential step in a study of inflight checkout.

4.1 Memory Requirements

To arrive at a first estimate of the computer memory size and processing speed that might be required, a typical program was devised to perform the functions of system monitoring and trend analysis. Figure 1 shows the major program steps

in block diagram form. Figures 2, 3 and 4 show the various program steps in more detailed, flowchart form. The style found in these flowcharts is very similar to that used in FORTRAN executable statements. This similarity, however, should not be interpreted as a requirement to use FORTRAN, or any other particular programming language, in checkout operations. Names of variables that appear in the flowcharts are explained in the glossary.

Each block in Figures 2, 3 and 4 has three numbers associated with it: (1) the number of instructions stored for that block, (2) the number of variables and constants that could be used by the instructions, and (3) the number of data values that are proper to each system parameter and that could be addressed by the instructions. Each item in the last two categories is counted only once, in the block in which it first occurs.

Although no specific computer type is implied, some assumptions about computer operations were made in arriving at these estimates:

1. The data values for the various parameters would be stored in arrays and the instructions to address them could be indexed.

2. Sampling of parameters would be done under program control, by having the computer put out a word to address the desired test point. This word would activate the proper multiplex circuitry so that the test point measurement would be routed (through an analog-digital converter, if necessary) to a computer input channel, where it could be transferred into memory under program control. Other input schemes are possible, such as automatically stepping a multiplexer through the various test points in a fixed or programmable pattern. However, the method adopted here is relatively simple and direct, and should give meaningful, representative requirements.

3. All parameters would not need to be read on every pass through the program; some may not even need to be monitored but would be used only for diagnostic purposes. Counters that were updated and reset by the program could be used to control sampling rates.

4. Limit testing would involve three pairs of values, corresponding to high and low bounds for caution, warning, and emergency for each parameter. These limits could also be viewed as moderate, fast, and urgent needs for action. When a parameter was found

to exceed a limit, the computer would put out a word to activate alarms, indicator lights, or other external devices, which would remain active until reset by the crew. For emergency limits, each parameter would have its own unique word for each limit, so that the activation of alarms and other alerting devices could be tailored to the specific needs of each parameter. Since caution and warning limits would not involve such great urgency, the same word for each limit would be used for all parameters within a system.

5. Messages to the crew would be printed when a limit was exceeded, and again when a previously out-of-limit parameter returned within limits. Each message would consist of two parts: One to identify the parameter, and one to designate the limit that was crossed. Each of these parts would contain four words, which would provide 16 characters per part on computers organized around a six-bit byte and a 24-bit word, or an eight-bit byte and a 32-bit word. Other outputs, such as the printing of actual parameter values or the graphic display of data, could be requested by the crew, but were not included in this program.

6. Trend analysis would be done by fitting the current value and the nine previous values of a parameter with a least-squares straight line and extrapolating that line to the nearest future intersection with a limit. If the intersection was less than a fixed number of samples away, a message would be printed for the crew, and the sampling rate for the parameter would be doubled if it was less than the maximum rate. The sampling rate would be restored to its original value when the intersection of the extrapolated line with a limit was again remote enough in the future to be ignored.

Simple, commonly available instructions were used throughout, but register-to-register data transfers and shifts within registers were omitted from consideration. A more powerful instruction repertory could permit some operations to be performed with fewer instructions. However, these savings could be offset by the additional instructions needed between steps of an operation to place data in the proper registers or bit positions to take advantage of these instructions. Therefore, the instruction counts shown in Figures 2, 3 and 4 should not vary appreciably with choice of computer.

The program to perform all the functions shown in Figures 1-4 would require 277 instructions and 77 words of program constants and variables. In addition, each parameter would need 29 words of data. Programs of reduced complexity could be obtained by deleting some functions. For example, if trend analysis were omitted, the program would require 135 instructions and 45 words of program constants and variables, as well as 18 data values per parameter. Since full-word precision would not be needed for most parametric data on some computers, storing two or more values in the same word could lead to additional savings in memory.

4.1.1 Sample Spacecraft System

The overall memory and speed requirements depend heavily on the total number of parameters to be sampled, tested, and stored. A survey of measurement lists and parameter estimates for existing or proposed manned spacecraft indicates that a wide variation in this number can be anticipated. Measurement lists of various Apollo CSM's show roughly 300 analog parameters being telemetered to the ground. For the LM/ATM to be used in AAP missions, about 600 measurements on vehicle systems will be telemetered. The Martin OCS discussed in Section 2.0 considered about 2000 parameters for a cluster station, and the RCA OCS could handle as many as 10,000 test points in a large space station.

Since the configuration of many operational systems on the space station is far from fixed at the present, an accurate measurement list for each system cannot be obtained. However, an Environmental Control and Life Support System (EC/LSS) designed by AiResearch Corporation[3] provided one sample system that could be used for sizing estimates. This system evolved during a preliminary study of a maintainable EC/LSS for a four-man, two-compartment vehicle in Earth orbit for two to five years in the mid-70's. A considerable amount of detailed analysis had been embodied in the design, and the parameter lists presented in Reference 3 were, therefore, accepted with reasonable confidence. These lists indicated a total of 183 measurements that should be made in a ground-test version of the EC/LSS. Of this total, a set of 108 were designated to be monitored in flight.

For an initial sizing estimate, it is better to use the larger of these two numbers, 183. This choice is based on the following reasons:

1. The figure 183 may be viewed as a "worst case"; if it can be handled satisfactorily, less taxing cases should pose even less of a problem.

2. Since the space station may well be one- or two-of-a-kind, the designers would want more data than a "standard" set of inflight parameters would give. This would be especially true if the station were of the integral type, rather than an assembly of smaller modules.

3. The space station assumed for this checkout study would have a crew of up to nine men and would probably be divided into more than two compartments. This station would, therefore, require multiple requirements of certain parameters like cabin temperature and pressure, where single measurements are sufficient for the vehicle used as a guideline in the AiResearch design.

4. A larger number of available parameters would allow checks to be made on the instrumentation.

With the further assumption that instructions and data values would use one word each, the computer memory requirements were estimated for the AiResearch EC/LSS:

1. With trend analysis:

$$\begin{aligned} 29 \times 183 &= 5307 \text{ parameter data words} \\ 277 + 77 &= \underline{354} \text{ program words} \\ &\quad 5661 \text{ total words} \end{aligned}$$

2. Without trend analysis:

$$\begin{aligned} 18 \times 183 &= 3294 \text{ parameter data words} \\ 135 + 45 &= \underline{180} \text{ program words} \\ &\quad 3474 \text{ total words} \end{aligned}$$

4.1.2 Diagnostic Testing

Diagnostic programs must be tailored to the particular systems they are to test. The manner in which a system is partitioned into units and the use of BITE in various units are typical factors that strongly influence testing requirements. No comprehensive diagnostic program was available for the EC/LSS or for any other advanced spacecraft system. To obtain sizing estimates, therefore, it was necessary to base the computer requirements on assumed typical diagnostic functions. Only passive testing was included, since no meaningful estimates of active diagnostic requirements could be made without detailed analysis of specific systems.

When a system parameter is observed to be out of tolerance, three possible causes of the condition can be distinguished:

1. the system unit with which the parameter is associated is defective,

2. the unit is really healthy but is responding to an out-of-tolerance condition elsewhere in the system, or

3. the measurement is in error.

The first task of the diagnostic program would be to determine, through the correlation of data pertaining to various system parameters, which of these causes actually applied to a specific situation.

For diagnostic purposes, a system parameter may be viewed as a measurement at the output of a unit that is affected by other parameters as inputs, and that in turn influences other units in the system. If one or more inputs to a suspected malfunctioning unit are out of tolerance, the unit may be assumed to be operating properly, at least for the first level of diagnosis, and the fault should be sought in other units. Furthermore, if units directly affected by an out-of-tolerance condition do not show the proper response to that condition, the measurement itself should be suspected as being in error. The unit with which the out-of-tolerance parameter is associated should be considered at fault only if the inputs to the unit are within tolerance and if other units are properly affected by the observed condition.

Information to direct the diagnostic program in testing upstream (input) and downstream parameters could be stored in tables. For each parameter, these arrays would identify which other parameters should be tested and would indicate the sense of proper operation. For example, an increase in temperature may result from an increase in the rate of heat generation or from a decrease in the rate of coolant flow. The number of such parameters that needs to be specified depends on the characteristics of the units involved, but three upstream and three downstream parameters would probably be entirely adequate. A program to perform the testing outlined in the preceding paragraph would require about 200 words of memory for instructions and data other than these arrays.

Many if not all parameters may also be involved in specific calculations related to diagnostic testing; for example, the determination of ideal unit outputs based on existing inputs. These calculations in turn could use mathematical library subroutines for such tasks as tabular interpolation, evaluation of

trigonometric and other transcendental functions, matrix manipulations, and the solution of differential equations. Additional tasks could include the assessment of BITE outputs and comparison of redundant or alternative circuits. The memory requirements that these tasks impose can vary considerably for each parameter. For sizing estimates, the following assumptions were made:

1. One-fourth of the parameters would need calculations comparable to evaluating a quadratic function of two variables,

$$y = ax_1^2 + bx_2^2 + cx_1 + dx_2 + e \quad ,$$

which would require 23 words per parameter.

2. One-fourth of the parameters would involve less complicated calculations like evaluating a linear function of three variables,

$$y = ax_1 + bx_2 + cx_3 + d \quad ,$$

which would require 16 words per parameter.

3. The remaining parameters would need only simple calculations or calls to library functions, for which about 6-8 words per parameter would be required.

These figures yield an average of about 13 words per parameter for calculations. This value was combined with those previously given for fault isolation (6 words per parameter and a program of 200 words). A rounded estimate of 20 words per parameter was then assumed. Diagnostic testing of the EC/LSS, with 183 parameters, would, therefore, add 3660 words to the checkout program and would raise the total requirement to about 9300 words with trend analysis and about 7100 words without it.

Since the need for diagnostic testing should arise relatively infrequently, a significant saving of main memory could be realized by placing the diagnostic programs in auxiliary storage and calling for them only when needed. Programs to perform active diagnostic testing could also be placed in auxiliary storage. However, programs associated with highly critical parameters, which would require a fast response by the computer, could be kept in main memory at all times.

4.1.3 Total Spacecraft Parameters

The next question to be answered was: How should the EC/LSS requirements be scaled to arrive at a ballpark estimate of requirements for the entire space station? A survey of existing or proposed manned spacecraft revealed that the EC/LSS accounted for the following portions of the measurements allotted to spacecraft systems:

Apollo CSM	9%
OWS CSM II	14%
Martin OCS	9%
AAP LM-A	10%
Average	∿10%

It was, therefore, assumed that the entire space station would involve ten systems, each with as many parameters as the EC/LSS, for a total of 1830 parameters.

A further assumption was made regarding the checkout program, to the effect that each of the ten space station systems would have its own program in the computer. Although the same program, as outlined in Figures 1-4, could be used for all systems, separate programs allow variations peculiar to a system to be handled more readily. Furthermore, the checkout program for a particular system can then be treated as a more distinct entity, to be modified or replaced as needed without disturbing programs for other systems.

With these assumptions as a basis, the memory requirements for checking out the entire space station can then be estimated as 35K words for monitoring functions. Trend analysis would require 22K additional words, for a total of 57K words. Diagnostics would add 37K words, for a total of 71K without trend analysis, or 93K with trend analysis. (These totals are more accurate than those that would be obtained by simply adding the rounded figures given in this paragraph, namely 72K and 94K, respectively.)

Figure 5 shows how the total memory requirements vary as a function of the number of system parameters. It is assumed here that the checkout program will handle ten systems, regardless of the total number of parameters involved. For a checkout program containing both trend analysis and diagnostic testing, memory requirements may vary from 24K words (for 500 parameters) to 490K words (for 10,000 parameters). The significance of these requirements in light of present and future computer capabilities is considered in Section 4.4 below.

4.2 Speed Requirements

Another important computer requirement is processing speed. An upper bound can be established by assuming that all steps in the program shown in Figures 1-4 (a total of 227 instructions) would be executed for all parameters (183 for the EC/LSS) for all systems in the space station (10 times the EC/LSS requirement). These assumptions lead to an overall total of about 427,000 instructions.

However, these conditions would not normally apply in practice. All parameters that were sampled would pass through the program without incident most of the time; i.e., parameters would generally be within limits and not cause any messages to be generated. Each such parameter would require the execution of 110 instructions if trend analysis were included, or 26 instructions if it were not. Parameters that were not read would use only 13 instructions each.

The need to print messages for the crew could introduce some complications into speed estimates. The instruction counts given in Figures 2-4 assume that print instructions would be inserted in the program as needed. However, an operating program would be more likely to use computer system software for printing, and to communicate with the software through a calling sequence. Since the print messages contain only alphanumeric information stored in the program, several time-consuming processing tasks like binary-to-BCD conversion, alignment of numerical fields, and character editing could be omitted. For a streamlined alphanumeric print routine, less than 100 operations may be needed to assemble the characters of a message and print a line. However, the need to print should arise only infrequently, and should not cause any noticeable effect in computer requirements.

The discussion so far has centered on the number of operations in the checkout program; a time factor remains to be introduced. Discussions with AiResearch engineers revealed that about 1/4 of the parameters for the EC/LSS would need "continuous" monitoring which, again for the EC/LSS, means about once per second. Other parameters would be sampled less frequently, at rates from once every five or ten seconds to once every 20 minutes. Based on these figures, the following assumptions were made regarding sample rates:

1. The checkout loop would be executed once per second.

2. Each time through the loop, 1/3 of the total system parameters would be sampled.

3. Diagnostic testing involving the execution of 20 instructions per parameter would be required for one system each time the checkout loop was completed; that is, once per second. This is a very high level of diagnostic activity but could be representative of a system that is performing quite erratically.

These assumptions, applied to a space station with 1830 parameters, lead to a requirement of 29K operations per second for a simple monitoring program. The addition of trend analysis would raise this figure to 80K. Diagnostic testing would increase the required speed to 33K operations per second without trend analysis, or 85K with trend analysis.

Figure 6 shows how speed requirements vary as a function of the number of parameters in the checkout loop. A checkout program that included trend analysis and diagnostic testing could range from 23K operations per second (for 500 system parameters) to 462K (for 10,000 parameters). Further discussion of these results appears in Section 4.4 below.

4.3 Input-Output Requirements

The details of input-output operations vary somewhat with computer structure. The statements made in this section should therefore be taken in a general sense, with the knowledge that some adjustments in requirements may be necessary or desirable when a specific computer configuration is being considered.

The inputs to the checkout computer can be grouped into four categories:

1. Parameter measurements received either from a digital sensor or through an analog-to-digital converter;

2. Discrete signals from spacecraft systems and test equipment;

3. Data and commands from crew input stations;

4. Uplink data and commands, and data from other computers.

Corresponding categories of outputs can also be designated:

1. Commands to initiate the sampling of parameters; (These commands would not be needed if the sampling of parameters were handled at fixed rates controlled from outside the computer. However, the commands would still be useful for diagnostic testing.)

2. Discrete signals to control spacecraft systems or test equipment;

3. Data displayed to the crew or printed for their use;

4. Downlink (telemetry) data, and data to other computers.

4.3.1 Parameter Data

Although inflight checkout may require access to about 2000 parameters, it is neither feasible nor necessary to provide each parameter with a separate channel to the computer. Rather, through suitable multiplexing, a single output channel could be used to handle commands for sampling, and a single input channel could manage the measurements coming to the computer. Since a channel on most computers can be used for either input or output (but not both simultaneously), a single channel could handle parameter data operations.

Allocating 11 bits in the output word to multiplexer control would allow 2048 parameters to be addressed. A few additional bits may be needed for parity checks and control purposes; these would raise the output word length to about 16 bits.

Measurement data read into the computer would probably have 8-12 bits for most parameters; parity and control bits may increase these values to 12-16 bits and, if the parameter address is to be included as another check item, at least 11 more bits would be needed in the input word.

4.3.2 Discrete Signals

The number of discrete input and output signals varies greatly from one spacecraft system to another. In the Apollo CSM, the ratio of discrete signals to analog measurements averages about 70%. If the same ratio is applied to the space station, about 1300 discrete inputs to the computer may be involved. Providing a unique channel for each of these signals would probably not be feasible, and some level of multiplexing would be expected. With suitable multiplexing and a sufficient number of buffer registers outside the computer, one channel for both input and output could suffice. In fact, with appropriate coding of the data identification, the same channels could be used for discrete signals, both input and output, as for parameter sample commands and measurements.

4.3.3 Crew Data

Computer outputs to the crew from the checkout loop would typically take one of two general forms: alphanumeric information for a printer or other display, and discrete signals

to indicator devices like panel lights and alarms. As mentioned in Section 3.2, the alarm could take the form of an assembled or synthesized spoken message. In addition, a cathode ray tube or similar graphic display device could be used to present data in specialized formats upon request of the crew. Computer inputs from the crew would consist of alphanumeric information from a keyboard, and discrete signals from buttons, switches, and other binary-coded devices.

Data transfer between the computer and crew-oriented devices could often involve delays arising from mechanical timing, synchronization, and acceptance-response requirements. Therefore, it would be desirable to have a separate channel to handle this transfer, so that other input-output operations would not be hampered. A single channel for input and output, with a few special lines for emergency purposes, should suffice for the more routine crew data. However, speech devices and graphic displays may require long strings of data obtained from the computer memory at rates determined by these devices. A separate channel may be necessary or at least desirable to satisfy these requirements.

4.3.4 Link Data

Data transferred between the computer and the ground would normally flow through a telemetry and command system on board the spacecraft. The computer's role would require electronic communication with this system. One input and one output channel would probably suffice. However, very high data transfer rates may require that additional channels operating in parallel be employed to handle the peak loads. Transfers between computers, such as would occur when one computer tested another, may also involve high data rates.

4.3.5 Summary

Detailed specification of input-output requirements depends strongly on the design and organization of a particular computer. A sophisticated channel, for example, may be able to control the simultaneous operation of several input and output devices, while a simple channel may be able to handle only one device at a time. Although the concept of "channel" is left admittedly vague, at least three channels, each capable of handling input and output operations, should be provided: One for checkout communications with spacecraft systems, one for communications with the crew, and one for communications with the ground. However, additional channels may be needed to handle occasional peak loads.

4.4 Computer Resources Available

A survey[4] of aerospace computers in existance or under development shows that requirements of up to 131K words of memory and about 250,000 operations per second could be met today. Furthermore, aerospace computers for the mid-70's are expected to have memory capacity in excess of one million words and be able to execute well over one million operations per second. Even the heaviest checkout requirements derived above, 93K words of memory, and 85K operations per second, would use only 71% of the memory and 34% of the speed capability of present-day computers, or about 9% of the memory and 8% of the speed capability of a mid-70's computer.

The memory and speed capabilities of present and future computers are shown by horizontal lines on Figures 5 and 6, and can be seen to give a wide margin for larger checkout programs or for more test points than were used to obtain estimates in this study. With a moderate allowance for an executive program, present-day computers could handle over 2500 test points, while a mid-70's computer could handle as many as 10,000 test points. Therefore, the memory and speed requirements of checkout functions by themselves should pose no special problem for the onboard computer systems. However, it must be pointed out here that the checkout functions have been considered independently of other computer tasks in this report; some revision of estimates may be needed when other tasks are included in the overall requirements.

The input-output requirements of aerospace computers are generally satisfied through the design of special interface equipment. Because of the large amount of multiplexing needed, a similar policy should be expected for the space station. Existing computers are usually multichannel, and future designs will probably have even more channels and more lines for special signals and interrupts. Therefore, as far as the computer itself is concerned, input-output requirements arising from checkout functions should not cause any significant problems. However, some consideration should be given to the weight, volume, and power of multiplexers and wiring for a large number of test points. With today's integrated-circuit technology, a multiplexer to handle the number of test points used for estimating in this study may weigh as much as 50 lbs, occupy one cubic foot, and consume 25 watts of power. Hopefully, these figures could be reduced by factors of 2-10 through the use of LSI.

4.5 Software Considerations

A closer study of the techniques for carrying out diagnostic testing has revealed some potential problems in the software that would be used to implement inflight checkout. The two most serious problems are the bridge between monitoring and diagnosis, and the time-sharing of the computer between these functions.

4.5.1 Preliminaries to Diagnostic Testing

The need to perform diagnostic testing will normally be made known when one or more parameters are found to be out of tolerance during system monitoring. The computer program can respond to this condition in two basically different ways: (1) branch to a diagnostic program immediately, or (2) note which parameters are at fault and delay diagnosis until all remaining parameters in the monitor loop have been tested.

The first approach has the advantage of starting trouble-shooting procedures as soon as possible. However, the monitor loop may be interrupted to do this. As a result, the handling of more serious problems may be unacceptably delayed, particularly if the diagnostic testing became lengthy and involved. Furthermore, knowing all the parameters that are out of tolerance may facilitate diagnosis. Therefore, the second approach, namely noting which parameters are at fault and handling diagnostics later, is preferable. This notation process could be as simple as storing in a table the index that identifies the parameter, along with data to designate which limit the parameter was found to exceed, and whether the problem is an actual one based on current values or an anticipated one based on trend analysis.

When the checkout program was finally at the end of the monitor loop, the first task of the diagnostic program would be to evaluate the failure conditions that were found. Another sampling of parameters may be called for, to determine whether the out-of-tolerance value was repeatable or merely a transitory phenomenon. Even a transitory phenomenon would be noted: It may indicate problem areas like noise sensitivity, faulty sensors, or instantaneous overload, which, although not generally a cause for alarm, should still be corrected. If the program determined that a failure had indeed occurred, the seriousness of the failure would also be assessed, and a warning of dangerous conditions would be given to the crew if necessary. When multiple failures were present, another task for the diagnostic program would be the assignment of priorities for handling them.

Using the identification of an out-of-tolerance parameter as a key, the diagnostic program would determine which test routine should be called, and would initiate loading that routine from secondary memory if it were not already in the main computer memory. The diagnostic program would also determine whether that routine required additional parameters and, if so, would sample and process them. Next, the monitor loop would be adjusted to inhibit the sampling and testing of out-of-tolerance parameters in the loop until the cause of the problem was found.

The transmission or storage of checkout data would also be initiated. Even if the data were stored for possible later transmission, information describing the trouble could be sent to the ground.

4.5.2 Initiation of Diagnostic Testing

As mentioned in Section 3.2, diagnostic testing would start with passive tests and would proceed to active tests only if the passive tests failed to isolate the trouble. The first tests, whether active or passive, may be arranged to give a quick look at the system so that the astronauts could be advised to terminate the mission, switch to a backup mode, or wait for further diagnostic testing before deciding what course of action to follow. Ground notification may also be involved in some instances.

Besides the delays in execution that are implicit in real time communication with the crew or the ground, an active diagnostic program would experience delays in carrying out its test procedures. When a stimulus is applied, the computer would normally wait a specified length of time before taking measurements in a system. With mechanical systems like the EC/LSS, such delays may be a number of seconds or even minutes in duration. It would be impractical, if not operationally impossible, to suspend computer activities during these periods of delay. Therefore, the diagnostic routines must be designed to time-share their functions with other computer tasks. Since monitoring of spacecraft systems would normally be required on a rather continuous basis, the monitor program would be given highest priority, and diagnostic tasks would be interleaved between monitoring operations. However, a highly critical fault could cause these priorities to be reversed.

The monitor loop could be started periodically through the action of an interrupt that arose either from an internal timer or from some external timing device. Other timing signals activating the proper interrupts could indicate the end of a delay period, such as might be desired following the application of a stimulus. Input-output processing with interrupts has been in use for many years already, and could be expected to be available on the spaceborne computer. Emergency conditions, signaled by monitoring devices that operate independently of the computer, could also be handled through interrupts.

4.5.3 Summary

Developing the software for inflight checkout may be a large problem. To date, multiprogram systems have posed a number of peculiar difficulties, and the checkout program is not likely to be any better. The situation may be further complicated by the large amount of sharing that the checkout program entails: Input-output devices, data, and program segments.

Additional difficulties may arise when fuller specifications for spacecraft systems allow a better definition of the details of diagnosing a system, such as:

1. Which parameters should be tested, in what sequence, and how often?

2. What happens when some parameters are within tolerance while others are not? How should different parameters within a system be correlated? How should conflicts in data be resolved?

3. How do other spacecraft systems relate to the observed conditions? How can data from various systems be correlated to determine whether an out-of-tolerance condition is caused by a malfunction or failure, or is the result of some unusual but acceptable occurrence?

4. What should the checkout program do if it cannot isolate the cause of the failure?

Decisions made in these problem areas could significantly affect the software design, and should, therefore, be resolved at an early stage of software development.

5.0 EARTH ENTRY MODULE

The entry module that the crew will use to return to Earth may present some special checkout problems. Although the module would be unoccupied most of the time, an estimated 50-100 parameters would need to be monitored to insure that no unsafe conditions went unnoticed. If the module is normally kept active, it could monitor and diagnose its own systems, in essentially the same manner as the space station. However, to conserve power and prolong the systems' lifetimes, the entry module may normally be kept in a dormant condition until just before it was to return to Earth. During this quiescent period, typically two to four months in duration, data from sensors on board the module would be routed to the space station where, for checkout purposes, the module could be treated as another operating system.

A normally dormant module would require some special tests in addition to monitoring. Periodically, one or more crew members would enter the module and conduct power-on checks of at least some systems. Just before use for entry, an extensive checkout would be done. The space station computer could assist in these checkout tasks. However, using the entry module computer has the advantage of keeping the checkout program with the vehicle, thereby reducing interface problems between the module and the space station.

Additional complications in checkout result from the possible use of the entry module as an escape vehicle, to be boarded by the crew members if they must leave the space station under emergency conditions. The module may remain docked to the station, but an apparently dangerous situation could force the crew to undock and move off to a safe distance.

In an emergency, the first checkout function would be an assessment of the status of the space station, to determine whether to abandon it and return to Earth, or to wait until it can be made habitable again. Unless the station experiences a catastrophic failure as a result of fire, explosion, or heavy meteoroid damage, the decision to abandon the station or not may require some degree of checkout directed by the crew from the entry module. If the checkout system on board the station were functioning, it could be directed to test the station. Alternatively, a number of key parameters related to the habitability of the station would be sent to the entry module, either through a direct connection or, if undocked, by means of a telemetry link, for processing by the entry module computer.

If the decision is made to abandon the station, the entry module computer should conduct some checkout of the module, at least to assess its performance capabilities so that the crew will be able to select an appropriate target landing site.

The preceding paragraphs show the many checkout contingencies that could arise with regard to the entry module. To deal effectively with these various situations, the computers and associated equipment on both the entry module and the space station must give attention to such factors as which systems are in operation, where control is originating, how data is being routed, what processing should be done, and what should be done with the results. Providing the required capability may raise some serious problems to be solved in the design of both hardware and software.

6.0 CONCLUDING REMARKS

The studies reported here show that, although large numbers of test points may be involved, the inflight checkout requirements of a mid-70's space station are well within the memory and speed capabilities of present onboard computer systems. Over 2500 test points could be handled with present computers, and as many as 10,000 could be considered feasible for future computers. These large numbers of test points require extensive interface equipment to handle the computer inputs and outputs, but no serious problems are foreseen in providing such capability.

Routine monitoring of operating systems can be highly automated, but the crew should conduct periodic inspections for abnormal conditions. Details of diagnostic testing can also be handled automaticlly, with the approval and under the overall supervision of the crew.

However, the need to perform diagnostic testing, particularly that requiring the application of stimuli to failed or failing systems, can greatly complicate the checkout program and may introduce special problems in software design, like time delays, multiprogram operation, and multilevel interrupt processing. It is important that these problems be anticipated and resolved at an early stage in the development of the checkout program.

A dormant entry module should be monitored by the space station for safety purposes, but a normally active module could monitor itself. In either case, if an emergency arises, the entry module must provide a capability for conducting some testing of the space station to assess the seriousness of the situation, and must also be able to perform some checkout of itself to determine its operational capabilities.

J. R. Birkemeier

1031-JRB-gdn

Attachments

REFERENCES

1. "In-Flight Checkout: Project Apollo", Bellcomm, Inc. and The RAND Corp., June 21, 1963 (Confidential).

2. "Reliability and Maintainability Analysis of a Two Year Manned Spacecraft Mission", Hugh A. Jennings, The Boeing Company, (presented at AIAA 5th Annual Meeting and Technical Display, October 21-24, 1968).

3. "Design Study and Implementation Program, A Space Station Life Support System for Use in an Altitude Chamber", Final Report Vol. 1, Part 2; AiResearch Manufacturing Company (Garrett Corp.), February 6, 1968.

4. "Aerospace Digital Computer Design Trends - Case 103", Bellcomm Memorandum for File, D. O. Baechler, January 4, 1969.

TABLE 1

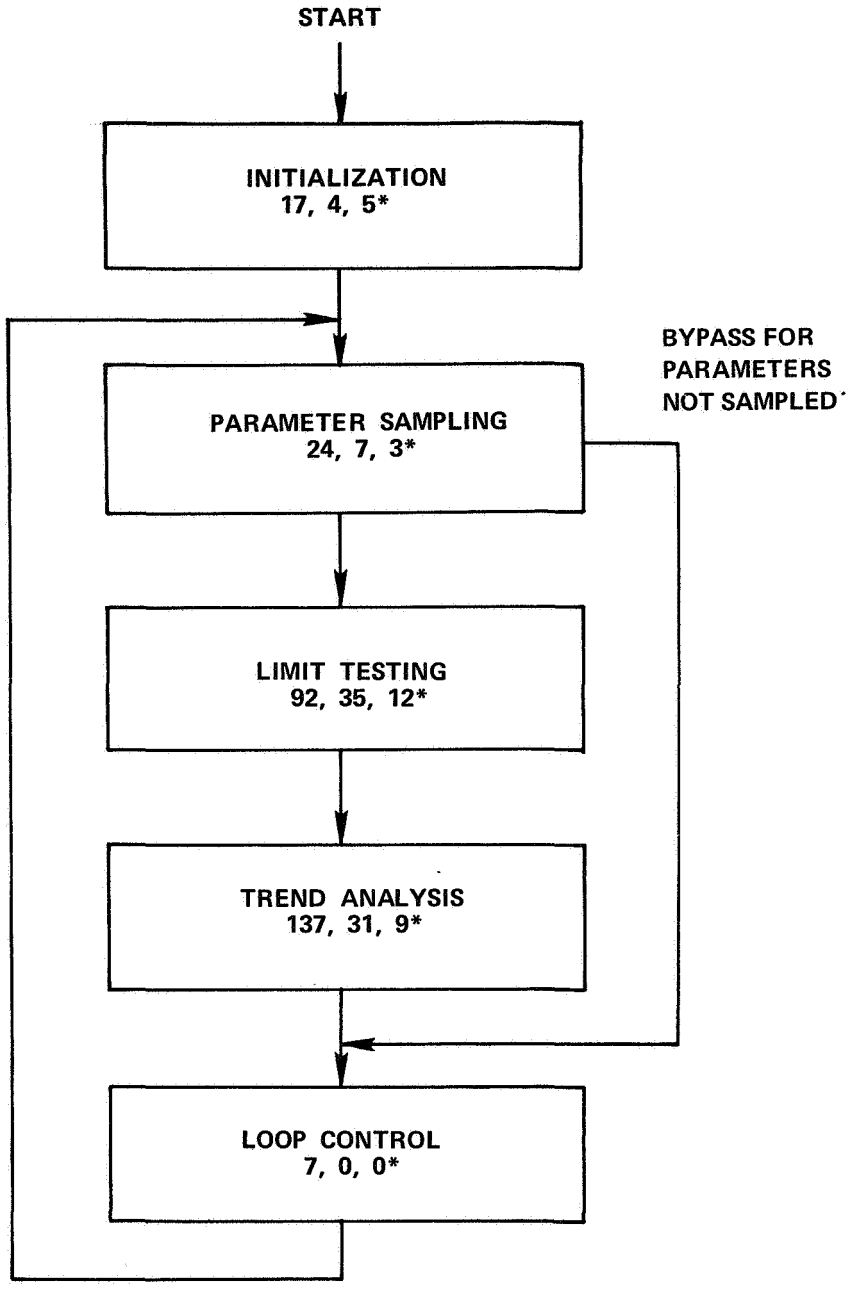
AIRCRAFT CHECKOUT SYSTEMS

MADAR FOR C-5A	ILAAS FOR A-7
Lockheed Design	Sperry-Rand Prime
Central Computer Control	Built-In Tests at Modular Level
Passive Only, No Stimuli	Periodic Local Stimuli
Analog Multiplexing and Routing	Discrete Status Data Routing
Priority Logic to Direct Crew in Multiple Failures	Automatic Operation with Degraded Capability
Printed Messages for Flight Crew	Status Displays for Flight Crew, Latching Malfunction Indicators for Ground Crew
Short-Term Trend Analysis for Failure Prediction	No Trend Analysis

TABLE 2

SPACECRAFT CHECKOUT SYSTEMS

MARTIN OCS (MSC)	RCA OCS (GRUMMAN)
Lab Model Developed, Flight Packaging Now	Study Program
General-Purpose Checkout Capability for OWS	Checkout Requirements for Large Space Station
Active, Passive, Flexible Control	Heavy Reliance on Passive Monitoring
Centralized Stimulus Generation	Local Stimuli Wherever Practical
Programmable by Flight Crew	Program Changes via Data Uplink



*THESE THREE VALUES INDICATE RESPECTIVELY:

(1) THE NUMBER OF INSTRUCTIONS STORED
(2) THE NUMBER OF PROGRAM VARIABLES AND CONSTANTS
(3) THE NUMBER OF ARRAYS OF PARAMETER DATA

FIGURE 1 - TYPICAL CHECKOUT PROGRAM-OVERALL PROGRAM FLOW

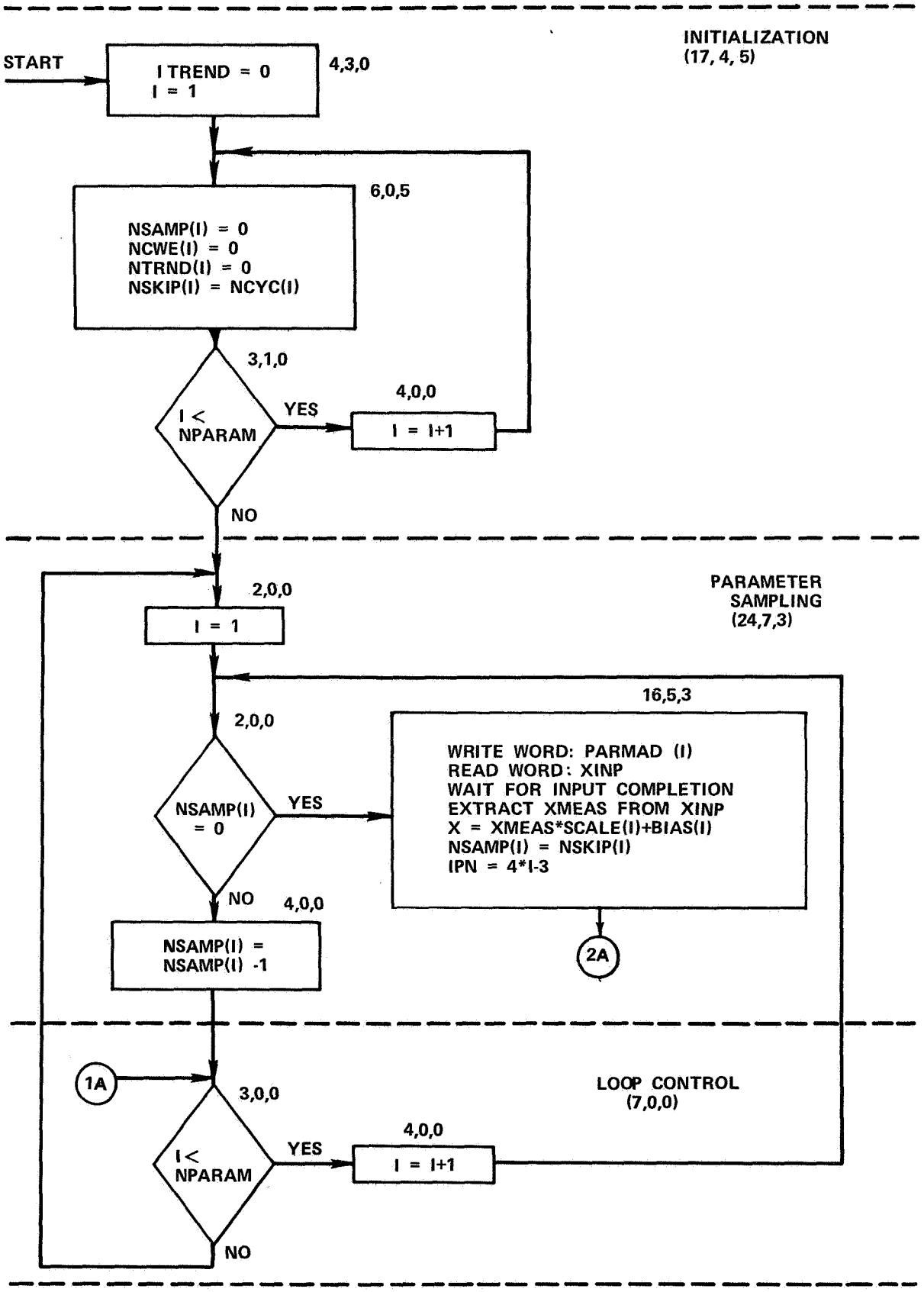


FIGURE 2 - TYPICAL CHECKOUT PROGRAM-INITIALIZATION, PARAMETER SAMPLING, AND LOOP CONTROL

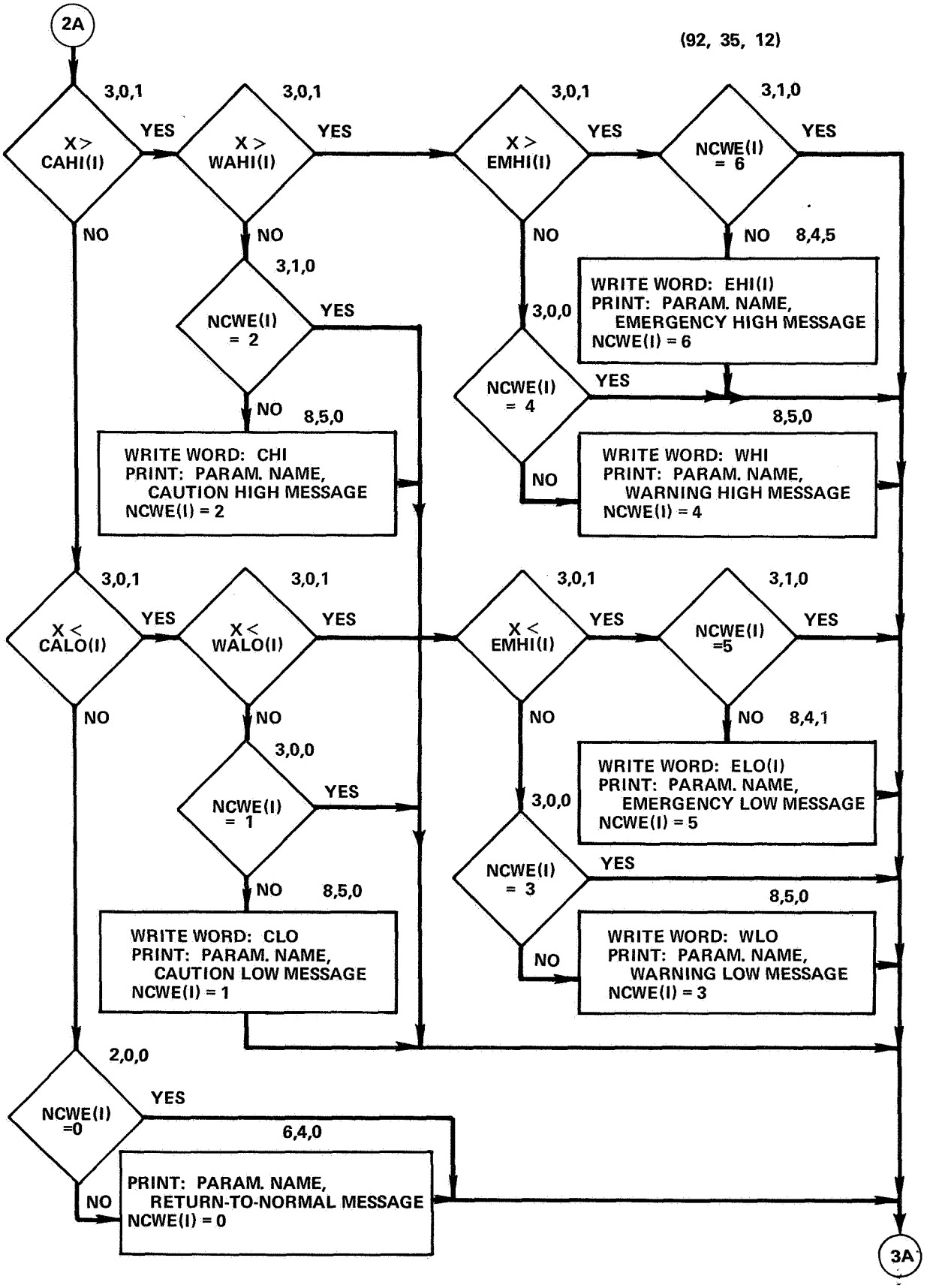


FIGURE 3 - TYPICAL COMPUTER PROGRAM-LIMIT TESTING

(137,31,9)

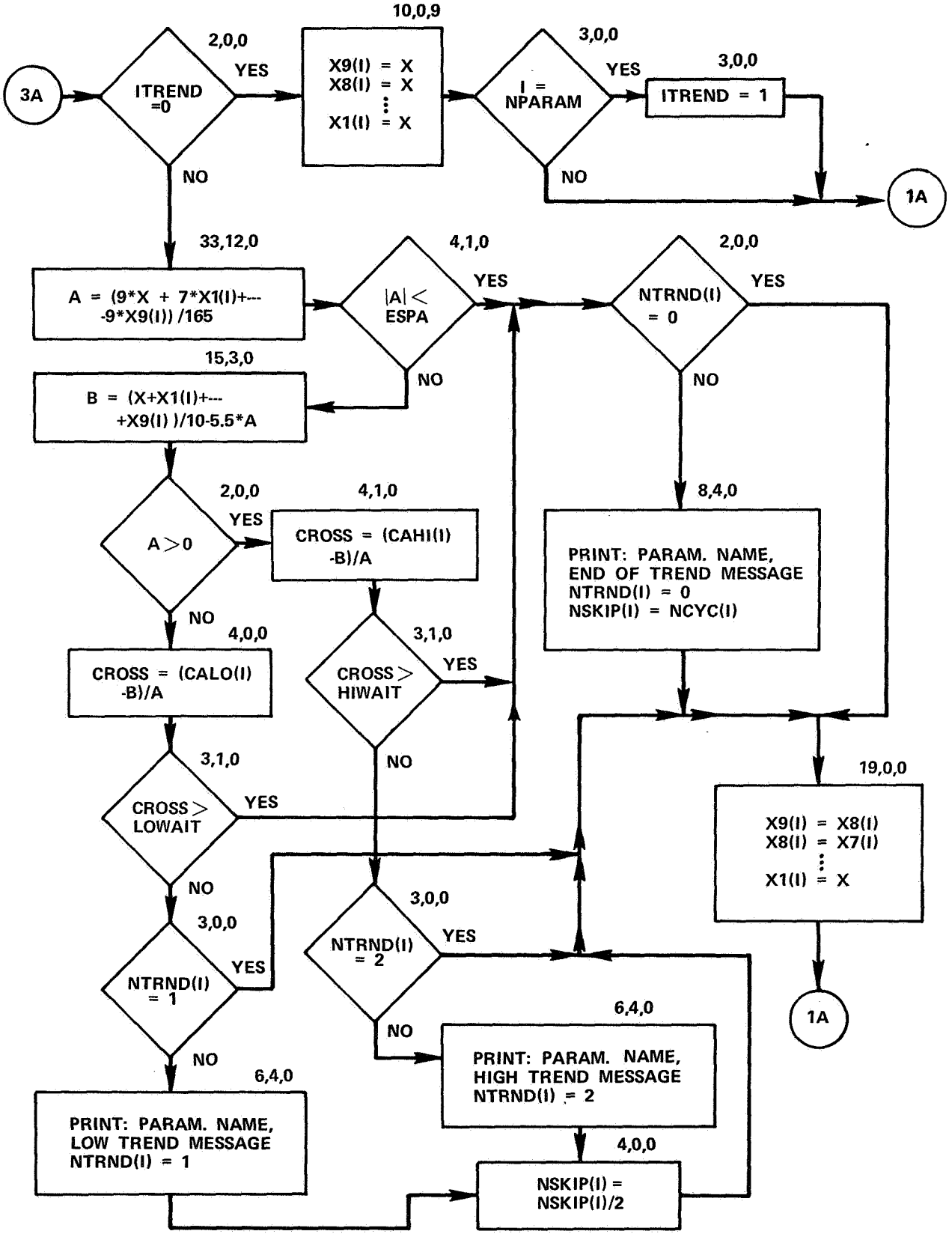


FIGURE 4 - TYPICAL COMPUTER PROGRAM – TREND ANALYSIS

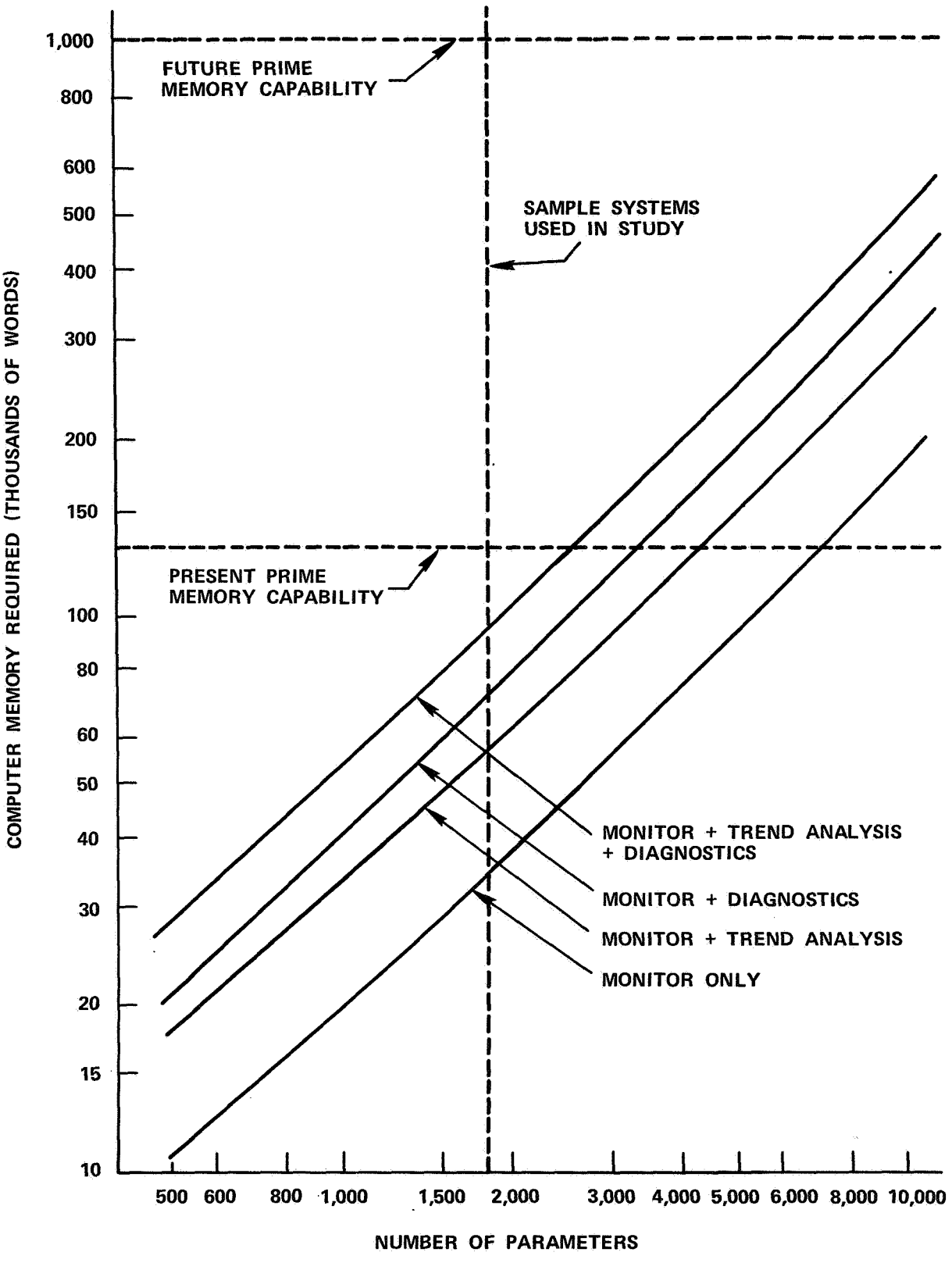


FIGURE 5 - ESTIMATED MEMORY REQUIREMENTS

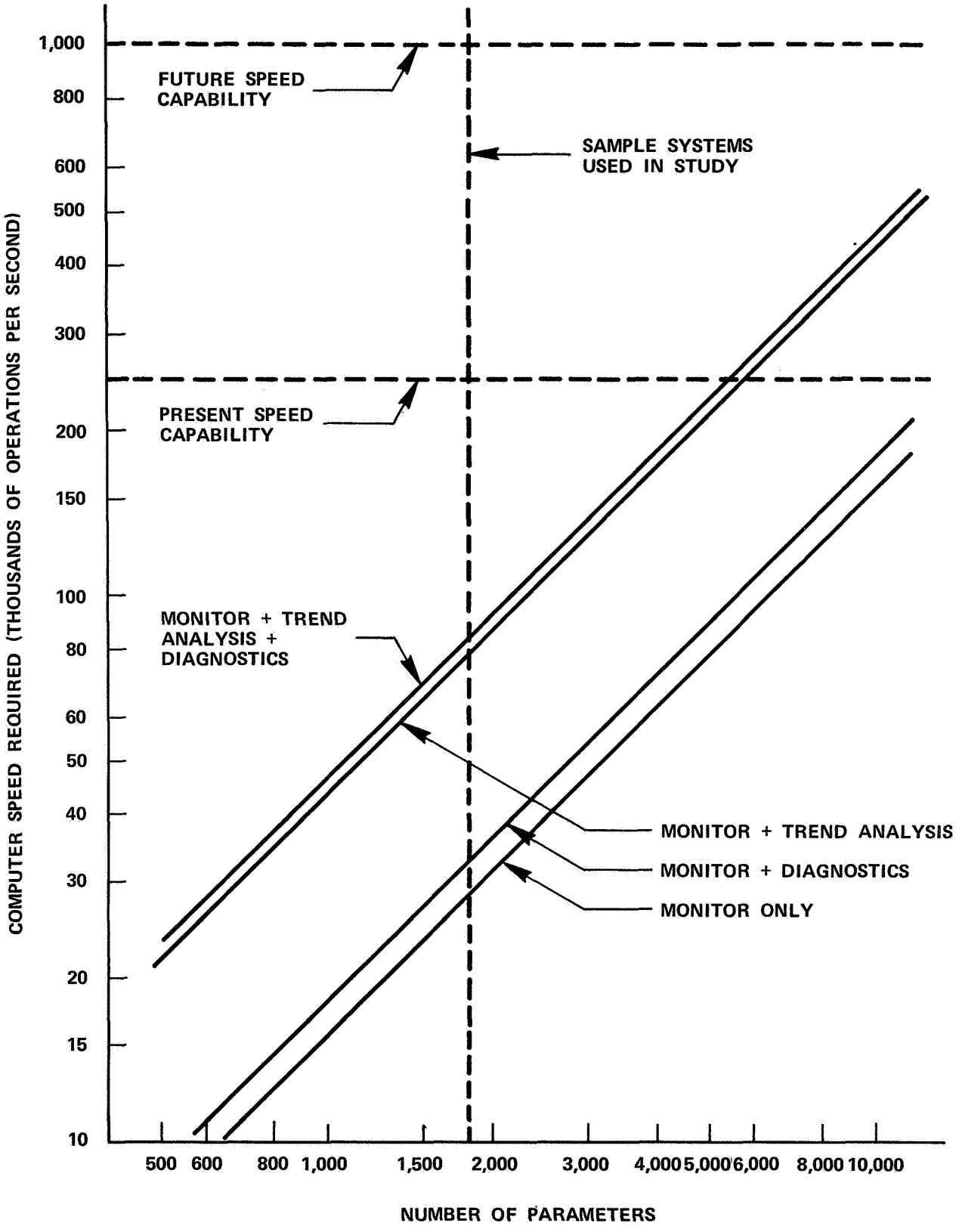


FIGURE 6 - ESTIMATED SPEED REQUIREMENTS

GLOSSARY OF COMPUTER PROGRAM VARIABLES

ITREND	-	Control for trend analysis: 0 means first pass through loop is not complete. 1 means first pass is complete.
I	-	Index for parameter data arrays ($1 \leq I \leq 183$ for EC/LSS)
NSAMP(I)	-	Counter specifying the number of loops to bypass before sampling the Ith parameter. ($0 \leq$ NSAMP(I) $\leq$ NSKIP(I))
NCWE(I)	-	Number indicating a specific out-of-tolerance condition for the Ith parameter: 0 means no trouble. 1 means caution low. 2 means caution high. 3 means warning low. 4 means warning high. 5 means emergency low. 6 means emergency high.
NTRND(I)	-	Number indicating a specific trend condition for the Ith parameter: 0 means no trend conditions of concern. 1 means low trend. 2 means high trend.
NSKIP(I)	-	Initializing value for NSAMP(I)
NCYC(I)	-	Initializing value for NSKIP(I), used to restore NSKIP(I) after a trend condition passes.
NPARAM	-	Number of parameters (183 for EC/LSS)
PARMAD(I)	-	Parameter multiplexer address word to initiate sampling of the Ith parameter
XINP	-	Input word received in sampling
XMEAS	-	Parameter (measurement) portion of XINP.
SCALE(I)	-	Scale factor for converting XMEAS to the desired units of the Ith parameter in the program.
BIAS(I)	-	Shift applied to converted measurement data for the Ith parameter.

IPN - Index for four-word arrays containing parameter names

CAHI(I) - Caution high threshold for the Ith parameter

WAHI(I) - Warning high threshold for the Ith parameter

EMHI(I) - Emergency high threshold for the Ith parameter

EHI(I) - Emergency high output word for the Ith parameter in the system

WHI - Warning high output word for the system

CHI - Caution high output word for the system (similarly for CALO(I), WALO(I), etc., for low conditions)

X1(I)···X9(I) - Nine consecutive previous values of the Ith parameter (X1(I) is most recent).

A - Slope of least-squares straight line.

B - Intercept of least-squares straight line.

EPSA - Threshold of A, such that (-EPSA < A < EPSA) is considered "no trend condition of concern".

CROSS - Point at which the fitted line intersects the caution low threshold (if A < 0) or the caution high threshold (if A > 0)

LOWAIT - The value of CROSS, for A < 0, above which "no trend condition of concern" is assumed

HIWAIT - The value of CROSS, for A > 0, above which "no trend condition of concern" is assumed