358 Yes [redacted]

MSC-EB-R-69-9

ACE-SC COMPUTER PROGRAMING SYSTEM

by

Walter T. Murphy
NASA-Manned Spacecraft Center
Houston, Texas

PREFACE:

This report is a description of the acceptance checkout equipment, spacecraft computer programing systems which provide for centralized control using the acceptance checkout equipment, spacecraft hardware system for spacecraft checkout operations and for independent and integrated lunar module and command and service module systems testing. The acceptance checkout equipment, spacecraft computer programing requirements, a functional description of the software implementation, including major additions during the past year, some problem areas which have dictated this particular implementation, and the role of the respective contractors are presented.

In particular, the acceptance checkout equipment spacecraft utilization at the Kennedy Space Center (KSC) launch area is described as being representative for all acceptance checkout equipment spacecraft installations at the various sites; however, specific mission references have been omitted, and spacecraft configuration identified only as that suitable for Apollo lunar landing.

I. INTRODUCTION

With the development of each new manned spacecraft program there comes a growing awareness of the ever-increasing complexity of tasks relating to integrated acceptance testing and preflight checkout of complex spacecraft systems and subsystems.

Many separate and interacting measurements and diagnostic tests must be performed on each spacecraft system before it can be accepted for mating with other systems into a completed spacecraft. Numerous additional measurements must be made, and testing must be performed upon the assembled spacecraft to assure the integrity of the system and to verify the capability to perform the intended mission. The problems relating to the checkout of a spacecraft intended for a manned-flight mission are further compounded by the necessity of achieving maximum system reliability and thereby minimizing the risk to human life.

The magnitude of the Apollo command, service, and lunar module checkout requirements precludes utilization of conventional testing methods. Testing of the Apollo spacecraft and lunar module by manual operation of individual test equipment and devices would be time consuming and costly and would not provide the capability of interrelating various test results. Manual analysis of the many thousands of individual test results and measurements would not only be time consuming, but could also reduce the reliability of the test effort because of possible human error.

The acceptance checkout equipment spacecraft (ACE-SC) computer programing system has been developed to minimize these problems in the checkout of the complex Apollo spacecraft systems and is the subject of this lecture.

FACILITY FORM 602
N70-35830 (ACCESSION NUMBER) (THRU) 1
27 (PAGES) (CODE)
TMX-64460 (NASA CR OR TMX OR AD NUMBER) 08 (CATEGORY)

II. ACE-SC COMPUTER PROGRAM REQUIREMENTS AND DEVELOPMENT

A. ORIGINAL REQUIREMENTS

The requirements placed on ACE-SC for the initial computer programing system were based on a concept for providing computer or manually controlled checkout of the Apollo spacecraft system. Also included was the concept of testing several systems simultaneously from one control station. Therefore, original computer programing requirements provided for the following capabilities.

1. Sending a wide variety of stimuli to the spacecraft from many system consoles independently in real time

2. Response monitoring by the downlink of several hundred spacecraft parameters in real time

3. Processing of these SC parameters in real time for display in various formats and recording of both raw and specially processed data

4. Complete test documentation

1. First ACE-SC Computer Program

These requirements dictated the development of a real-time package using basic symbolic language and providing the following features.

a. Basic monitor and control programs to include all real-time checkout requirements

b. Subroutines designed specifically to meet the spacecraft contractor checkout requirements, generalized for use on any spacecraft

c. Parameter inputs for both command and response table buildup to accommodate mandatory changes (The initial system was capable of real-time processing for display of approximately 450 measurements.)

Inherent in this system was the requirement and the capability for complete verification that the ACE-SC computer programs meet their specified requirements and that they could not be altered during on-line test operations.

B. MAJOR REVISIONS TO INITIAL REQUIREMENTS

Major revisions to the initial design were indicated by the following developments.

1. Increase in SC contractor measurements requirements

2. SC contractor requirements for special subroutines, such as the following:

   a. Closed-loop testing of the Apollo guidance and navigation system

   b. Static firing of service propulsion system (SPS) engines

   c. Controlling of simulated altitude tests

   d. Emergency detanking of the liquid hydrogen and liquid oxygen tanks

   e. Special handling of uplink commands; that is, multiple commands, momentary application of commands, and conditional application of commands

1. Second ACE-SC Computer Program

The following changes to the initial system design ground rules were dictated because of computer memory constraints caused by developments already defined.

a. The Executive Program was modified to handle phasing of test requirements.

b. Control programs and subprograms were rewritten with emphasis upon optimum utilization of memory by integrating subroutines.

c. Parameter formats and tables were changed.

C. REQUIREMENTS TODAY

After experience was gained utilizing this computer programing system, operational problems and the need for more efficient methods dictated the following additional requirements.

1. Closed-loop automatic sequences

2. More flexibility for varied combinations of tests to be run simultaneously

3. Data availability in optimized formats for quicker turnaround from the post-test data-reduction facilities

4. Many special processing requirements for various spacecraft subsystems

5. Automated methods of preparing spacecraft test programs

6. Increased efficiency in the test program verification process

1. Hardware and Software Changes to Meet Additional Requirements

The block II ACE-SC computer programing system, which is the basis for this lecture, refers to the ACE-SC software configuration which has evolved from these requirements including (1) computer programs to provide automatic test sequences; (2) modification to the pulse code modulation (PCM) decommutator which provides direct access to the external memory banks of the ACE-SC computer system; (3) use of magnetic tape as a mass memory device because of memory constraints; and (4) a need for data compression techniques to enhance post-test processing capabilities.

The ACE-SC downlink computer system used prior to the PCM decommutator modification continuously controlled the input of decommutated PCM data, selected and relocated the data to be processed, and then output all display data to the cathode ray tube (CRT) monitors. It should be emphasized that in this system operation the majority of data entry and response interpretation is performed by control room operator personnel.

With the advent of the block II ACE-SC program, many command operations have become automatic, that is, programed, predetermined operations to be executed by the computer. Thus, complete command generation, transmission, and monitoring of the appropriate downlink response can now be performed automatically (for certain specified operations) by the 160G computers. In addition, because of the hardware modification of the decommutator, the downlink ACE-SC computer has been freed from much of the burden of data input, relocation, and processing. The decommutator now possesses the capability to perform certain of these operations and store the result directly in computer memory; subsequently, the basic function of the downlink computer has become the maintenance of a compressed digital-data tape for post-test data reduction and of the special downlink subroutines for special real-time processing and display of spacecraft data.

The compressed-data tape contains a record of all PCM data which are processed by the downlink ACE-SC computer. These data are recorded in digital form in contrast to the analog recording performed by the FR1400 recorders. The compressed data are available immediately after completion of the test since the tape is the output of an on-line real-time operation.

III. FUNCTIONAL DESCRIPTION

The basic ACE-SC software configuration for SC testing is illustrated in figure 1. As shown in the diagram, there are two main control programs which, in turn, are controlled by a systems monitor. The system described is what is called the block II ACE-SC software system. This block II system was built around three modifications: (1) the decommutator direct-access modification; (2) the addition of the capability to automate SC testing in a closed-loop mode (adaptive intercommunication); and (3) the modification of the downlink program to provide for data compression.

(Position of figure 1)

For actual SC testing, the software system consists of the framework shown in figure 2 plus many additional subprograms. All of the programs to be utilized in an SC test are stored on a tape which is referred to as the test file tape (TFT). Portions of this tape, including system monitor, uplink and downlink control, and ADAP control (subprogram) are actually loaded from the tape and remain resident in core; other programs contained on the TFT are loaded by the monitor and executed as needed (normally as a result of an operator call-in of the program).

(Position of figure 2)

This philosophy of test storage and call-in is especially applicable to automatic sequences. The automatic operations desired are stored in groups on magnetic tapes. Each group consists of one or more automatic test sequences; each sequence consists of one or more subsets for the computer to execute. In essence, the sequences are programs which simulate the entry of data through the data entry system. A single entry from the control room can activate a sequence of operations to be performed by the computer. These operations may involve command generation and transmission and subsequent monitoring of downlink responses before the next sequence of operations is executed, or may involve monitoring of downlink responses and command generation on the results of the monitoring. Thus, using this philosophy of testing, an entire subsystem can be checked automatically if adequate test points are available.

A. UPLINK CONTROL PROGRAM

The uplink control program provides the capability for the ACE-SC system to interpret and act upon a command which may be generated manually in the control room or automatically from an activated test sequence. These commands instruct the computer to modify memory or request some action to occur in the spacecraft or associated ground support equipment (GSE). Test commands are initiated by the selection to activate random testing (START) modules located on the various system consoles or by preprogramed sequences which simulate the START modules. In either case, the uplink control program (in conjunction with uplink subroutines) receives the digital word and decodes it to select the desired action (activate a subroutine, generate a relay closure, form a command word to generate an analog signal, and so on). Upon deciding what the desired action was, if it was necessary to require some action to occur in the spacecraft or GSE, a digital command message is formed for transmission to the SC. Each message is transmitted redundantly to one of three command links, depending on where the message is to be sent. The uplink control also provides the logic to receive the verification that each message is properly received by the digital test command system; in turn it notifies the control room equipment of command verification and files on the uplink command file tape pertinent information with respect to the command for a complete test history. In the event that the verification reply indicates a malfunction in the command transmission, the command is retransmitted a predetermined number of times, each time waiting for the verification of proper message delivery with the resulting pertinent information filed. The uplink control program also contains the necessary linkage to the various uplink subprograms required for a particular test, intercommunication with the downlink system, and all possible checks and protections to prevent the generation of erroneous commands to the spacecraft. Included is the capability to recover control room, program, and SC configuration status in the event of ACE-SC hardware failure.

B. DOWNLINK CONTROL PROGRAM

This program (fig. 3), in association with its subprograms, provides the capability to receive from the decommutation equipment the incoming digital PCM data, which represent the real-time status of spacecraft and GSE systems. The downlink programs process selected portions of these data, as defined in the test requirements, for display in engineering units on control room alphanumeric CRT displays. Selected portions of these data are processed for display on the computer-driven event lights, event recorders, and analog meters and recorders. All measurements are output upon significant data change to the digital compressed-data tapes for post-test processing. The downlink program also interprets the preprocessed limit-checking capability of the decommutator for blinking of out-of-limits conditions on the CRT displays. Included is the logic for intercommunication with the uplink for automatic command initiation upon detection of predefined conditions from the spacecraft parameters. The capability is also provided to send special predefined error conditions for critical parameters to uplink to be filed on the command file tape.

(Position of figure 3)

C. DECOMMUTATOR

The Decommutator Program provides the capability for the decommutator to receive and synchronize on the incoming PCM stream and to route each parameter directly to the computer external memory and directly to the control room display devices. This provides redundancy for critical measurement display. The Decommutator Program now provides the additional capability of checking each parameter for out-of-limits conditions based on prespecified upper and lower limits and of flagging the downlink computer of each of these conditions. Also, the Decommutator Program checks each event measurement for a change of state and checks each analog measurement for a change of 1.57 percent. In each case it notifies the downlink computer of these changes so that downlink can transmit all changed data to the digital compressed-data tapes for post-test processing.

D. REAL-TIME MONITOR

This program is designed to load initially the entire uplink/downlink and Decommutator Program systems and thereafter provide a means of rapidly changing computer load to satisfy the specific test requirements. The monitor is completely resident and operates on a noninterference basis with the control programs. Since the present ACE-SC software system operates under the philosophy of utilizing a digital tape as an extended computer memory, the monitor provides the mechanization necessary to call specific subroutines into a core, complete reloads of (1) uplink (consisting of programs or parameters or both); (2) CRT annotation alternate loads; (3) decommutator alternate loads for Decommutator Programs of different telemetry bit rates; (4) automated test sequences; and (5) downlink (consisting of programs, parameters, and CRT annotation or any combination of the three). Checks are made by the monitor to insure that proper loads can and/or have been achieved.

E. ADAPTIVE INTERCOMMUNICATION CONTROL SUBROUTINE (ADAP)

This feature is a prime example of a change to the ACE-SC software during the last year (fig. 4). The ADAP has not deviated from the concept of command transmission and response monitoring, but has altered the manner in which the operation is performed. In essence, ADAP has added a degree of sophistication and automation to the existing ACE-SC software system. In pre-ADAP operations, commands to spacecraft systems were entered in the ACE-SC system by the data entry equipment in

(Position of figure 4)

the control room. Responses were received, processed, and displayed by downlink of ACE-SC. The majority of data interpretation, evaluation, decisions of actions to be taken, and so on, was performed by control room personnel in compliance with operational checkout procedures. With the evolution of ADAP, many of these operations can now be performed by the ACE-SC computer system operating in a closed-loop fashion. Thus, predetermined commands can be automatically transmitted and the corresponding responses monitored. The results of such monitoring processes determine which of the alternative courses of action are to be taken. Consequently, ADAP allows the automation of portions of an operational checkout procedure (OCP).

The functions of ADAP are stated simply. The ADAP subprogram can simulate START executions (relay and command starts), monitor PCM data, and display messages on the CRT. Various conditions can be specified for the monitoring operation and alternative courses of action to be taken can be established, depending upon the result of the monitoring process. In effect, these alternatives specify the sequence in which the parameters are to be processed. In addition to stipulating the sequence of execution, the parameters can specify fixed-time delays to be observed before proceeding with processing; ADAP can also be commanded to wait until a PCM measurement satisfies a stated condition before continuing with processing.

IV. DEVELOPMENT OF PROGRAMING SYSTEM TO SUPPORT SC TEST REQUIREMENTS

As described, the real-time programing system (all information contained on a TFT) contains all ACE-SC programs utilized for a spacecraft test. This section includes general descriptions of the programs required to generate, verify, modify, and compile SC test programs. The assumption should be made here that the detailed requirements needed for a spacecraft test program have been received. Our goal is to take these requirements; process, compile, and assemble them; and verify that a test program (TFT) is available to meet the detailed SC test requirements. The name TFT is derived from the fact that the programing information is grouped in files on the tapes (control programs, subprograms, alternate loads, online system programs, and so on). An illustration of a TFT format is shown in figure 5. The first file of the tape is the auto-load

(Position of figure 5)

file. It consists of the systems monitor, uplink control program, and downlink calibration table. The loading of this file is initiated by the bootstrap circuitry of the uplink 160G computer; as soon as the file is loaded into core, program control is transferred to the system monitor. The monitor then loads the rest of the TFT, utilizing instructions from operator personnel. These instructions are communicated to the monitor by the communications start (C-START) in the computer room.

The remainder of the files on the tape contain information associated with test cards; there is one file for each test card. The term test card stems from the fact that the TFT is produced using binary cards as the basic source of data. The cards are arranged in files which have test cards as file identifiers; thus, even though the TFT is indeed a binary tape, the card terminology is still utilized when referring to the tape. The test card provides identification which indicates the function that is associated with the particular file. The body of the file consists of a program and/or parameters, overlays, annotations, and so on, or the intercommunication (IC) sequences contained in an ADAP group. Attention should now be focused upon the actual process of preparing a tape.

Detailed formats are defined for each checkout requirement of the SC contractor. The requirements can be categorized into three major areas (fig. 6): (1) the control programs are the basis for any TFT and are used from SC to SC; (2) subprograms are mechanized from subprogram specifications approved by NASA and generated by a contractor. These subprograms handle each unique processing capability defined as an ACE-SC requirement and also are used from SC to SC; and (3) the parameters vary from SC to SC and are supplied as program requirements for each spacecraft to be tested. These parameters are supplied in several decks of cards defined by predetermined formats. The parameters contain such items as measurements to be processed, the subprogram which processes them, display requirements for each measurement or function, commands to be generated, command termination addresses, command execution addresses, filing requirements, measurement limits, calibrations, and so on. Upon receipt of the SC program requirements TFT generation begins. It is here that the support programs play a vital role. Although each TFT could be generated by hand, the constraints on computer time, the number of errors in hand work, and the manpower requirements dictate the requirement for automated TFT generation. The detailed steps required

(Position of figure 6)

to prepare a TFT and station configuration for a specific spacecraft checkout are shown in figure 7.

(Position of figure 7)

The following is a brief description of the programs and operations which go into the generation of the master card deck from which the TFT is made.

Generation of these decks includes the four following interrelated operations.

1. Control room patching and labeling (CORPAL) processing and Decommutator Program generation

2. Downlink parameter processing

3. ADAP parameter processing

4. Uplink parameter processing

Normally, the Downlink Parameter Check Program is run first. The input to this program consists of the SC program requirements cards. The program checks the cards for format errors and so on and provides a listing of these errors. The primary outputs of this program are as follows.

1. Descriptor tape for post-test processing

2. Downlink parameters check tape

3. Downlink equates

These outputs will be utilized as inputs for other support programs.

One usage of the downlink parameters check tape is the running of CORPAL Programs. The tape checks the cards for errors, providing a listing of such, and produces the control room patching and labeling listings. The CORPAL processor uses either programing requirements cards or the check tape as an input (normally the check tape). The output of CORPAL provides control room configuration and patching information and the analog and event routing addresses which will be utilized as an input for the Decommutator Program compiler. It should be noted that these routing addresses are display addresses and are not downlink computer direct-access addresses; downlink computer direct-access addresses are assigned by the check program.

The Decommutator Program compiler will be discussed next. The output of this program provides a Decommutator Program listing and the binary Decommutator Program and locator deck for inclusion in the master binary deck. The Decommutator Program compiler uses the output of CORPAL part IV and of the check tape for inputs. The check tape contains information which includes the prime frame, link, time slot, sample rate, limits, and so on, for the PCM measurements obtained from the programing requirements process specification (PRPS) cards; it also contains the 169G direct-access addresses which were assigned by the Parameter Check Program. The check tape information and display routing addresses provided by CORPAL part IV allow the Decommutator Program compiler to produce the Decommutator Program listing, Decommutator Program binary deck, and locator deck. As mentioned, the binary and locator decks will subsequently be included in the master binary deck.

The check tape is also used as an input to the downlink parameter compiler. The latest version of the compiler requires only the check tape for an input. The compiler provides listings, binary decks, and locator decks for downlink parameters and the downlink scope annotations. The decks also will be incorporated in the master binary deck.

Another output of the Parameter Check Program is the tape containing measurement identification (ID) equates for downlink. These equates specify direct-access 1690 addresses assigned to measurement ID-link contained on the PRPS cards.

The downlink equate tape is used as one of two inputs to the ADAP-Downlink Equate Strip Program. The second input to the strip program is the output of the ADAP Parameter Check Program.

The output of the ADAP check consists of the errors by ADAP check; they are input along with the downlink equate tape to the ADAP-downlink equate strip program. Some ADAP parameter cards specify downlink measurement ID's to be monitored by ADAP. The purpose of the Strip Program is to provide the cross referencing information for ADAP measurement ID's and downlink measurement ID's. This enables ADAP to find the PCM data associated with the measurement ID that is to be monitored.

When the ADAP measurement ID equate cards have been produced by the Strip Program, the ADAP parameter compiler can be run. Two of its inputs are the error-checked ADAP parameter cards and the ADAP measurement ID equate cards; a third input is a basic equate deck which is associated with the compiler itself. The resulting output of the compiler is the ADAP parameters binary deck to be included in the master binary deck. A locator deck is also provided by the compiler.

Another binary deck required for the master deck is the one containing the uplink parameters. Generation of this deck is independent of the ADAP-downlink Decommutator Program operations. These parameters are generated by the uplink parameter compiler using the uplink programing requirements card information as an input. The resulting output of the uplink parameter compiler is the binary cards for the uplink parameters.

As has been described, the master binary deck is assembled, incorporating all the various binary decks for the subprograms, control programs, ADAP groups, locator decks, and so on. This deck is subsequently input to the master tape maker program and the Edit Check Program along with downlink calibration cards. The output of the Edit Check Program provides a listing of calibrations, limit changes, and suppressed data, and, of course, the TFT.

The preceding paragraphs have described the production of test file tapes. There are other nonreal-time operations which are performed utilizing the TFT, which has been produced by the process just described. These operations involve the TFT utility programs (UTL-004) as shown in figure 8.

(Position of figure 8)

This program (UTL-004) contains the following options. The first option provides the capability to list the contents of a TFT by files on the line printer. The program performs limited tape format error checks, primarily checking to insure that the files conform to the standard formats. The checks involved in producing the listing are strictly format checks and in no way verify the results of the Edit Check Program with regard to editing the ADAP, downlink, and decommutator files. The process just described produces an online listing of the TFT. An alternate output of this utility program is a binary coded decimal (BCD) tape for offline listing of the tape. The second option of this utility program provides up to three copies of a TFT with a second pass option of performing a bit-to-bit comparison of the copies against the original. The final option available provides comparison of two test file tapes (normally a newly produced tape against an older tape) with a printout of the differences. Because of the nature of a tape edited by edit check, a seemingly small change in input requirements can cause major differences in test file tapes. For example, a change of calibration can completely alter the addresses in the downlink calibration table. This is reflected in the downlink parameters as a changed location in each downlink parameter set which has calibration address. The program will compare each file on the tapes on a one-for-one basis. If a file has been added or deleted, operator action is required for the adjustment. If two files do compare, the test cards will be printed along with a statement saying the files compare. If the two files do not compare, the utility program will print the test cards and provide a list by card column and 12-bit data word for those locations that do not compare.

That effort and those types of programs required to generate a TFT have been described in general. To summarize, this process requires (1) tightly controlled subprograms; (2) control programs; (3) parameter inputs per predefined formats; and (4) the utility programs to generate, compile, and edit these various items into a tape. Now the total ACE-SC system verification process begins. As shown in figure 9, this generally requires 4 weeks.

(Position of figure 9)

The requirements for program verification are as follows.

1. Verify that each ACE-SC test program and/or SC TFT meets its specified requirements and can be certified for use as an operational program to be incorporated into the official ACE-SC computer program library for SC testing.

2. A software subprogram and program tape verification, certification, and acceptance team shall witness and accept the completed program.

3. Each verification shall be performed in accordance with written operational procedures.

4. Certify that all subprograms which are used to test SC hardware have been adequately verified.

To meet these objectives, verification is broken into two verification phases. Phase I verification consists of five parts.

1. Downlink Verification

Using PCM simulator tapes as an input to the decommutation system verify that (1) all measurements are decommutated and routed to the proper control room displays; (2) all measurements are routed properly to the computer; (3) each downlink subprogram meets its specification requirements using the simulated data; (4) compressed-data tape filing is achieved properly; and (5) CRT display is achieved properly.

2. Uplink Verification

Each spacecraft stimulus is initiated by using the control room START modules or an automated subroutine. Each command generated is recorded for analysis to assure proper sequence, that the correct address and only that address was transmitted, that each analog function was generated properly, and that all pertinent information for each command was filed properly.

3. Verification of Automated Sequences

Using PCM simulator tapes, each preprogramed automated sequence is run to determine that only the correct actions were taken as a result of a measurement data change, proper time delays are used, display of the sequence is proper, and all required filing is achieved.

4. Verification of Control Room Patching

Since a utility program is used to generate the control room patching requirements, each display or control device is exercised to insure that display and control are patched to the proper modules.

Upon completion of Phase I verification we are assured that the test programs do indeed meet their requirement specification. At this time Phase II verification is begun. During this phase of testing, any subprograms which were shown as open items (those subprograms and automated sequences which were not previously verified) must be functionally verified.

Phase II verification utilizes the ACE-SC ground station, the Phase I verified programs, and all associated GSE. This test demonstrates software/hardware compatibility of the ACE-SC system: (1) hardware/software timing compatibility; (2) uplink/downlink computer intercommunication capability; and (3) ACE-SC system to GSE compatibility.

The basic procedures for Phase II verification are as follows.

1. System compatibility is demonstrated, by using the verified test tape, utilizing verification procedures generated by the SC contractor.

2. Verification is demonstrated by initiating commands as required by the appropriate checkout procedures and monitoring uplink response and analysis of command file tape. Downlink verification will be accomplished by monitoring appropriate displays in the control room per OCP.

Satisfactory completion of the system compatibility verification demonstrates ACE-SC system readiness for SC testing.

As was mentioned, our goal is to properly prepare and verify ACE-SC test programs to meet the requirements for SC testing on schedule. To aid in meeting the goal, major emphasis has been placed on the following areas.

1. Make maximum use of offline assemblers and compilers to generate the real-time program.

2. Use all possible previously verified subroutines from the library system.

3. No online compiler-interpreter programs, as all functions must be predefined, preprogramed, and completely verified.

As a result we now have a turnaround time of approximately 6 weeks from required submittal to verified test tape and station configuration (2 weeks of program preparation, 2 weeks of software verification, 2 weeks of hardware/software compatibility verification).

The operation and verification of the test programs has been discussed assuming we have the requirements. The test requirement generation will now be discussed.

V. TEST REQUIREMENTS GENERATION

The SC contractors have the responsibility for generation of test requirements. These are documented originally in the form of process specifications for each subsystem on the SC. These process specifications are then interpreted by the software integration groups at the contractor facilities into functions to be performed by the ACE-SC systems. Basically, these functions are broken down into several areas: (1) measurements to be monitored; (2) display requirements; (3) detailed arithmetic and computational requirements; (4) filing requirements; (5) commands to be generated; (6) actions to be taken (either display or command generation based on out of limits conditions); and (7) diagnostic requirements. These are further broken down into subprogram specifications for each function to be performed (computation, display, tolerance checking, command operations) and the detailed parameters to operate with each of these subprograms (fig. 10).

(Position of figure 10)

The subprogram specifications are given to NASA for review and approval for implementation by a contractor. After subprogram preparation and verification, these subprograms are placed into the ACE-SC library system for use in SC test programs.

The detailed parameter requirements for each individual SC test program and the applicable subprograms are specified in the form of program requirements documents from the SC contractors. At this time, the ACE-SC contractor begins the program preparation for that SC test program.

VI. CONFIGURATION MANAGEMENT

The configuration management for ACE-SC computer programs is based on the Apollo philosophy that tests which will ultimately be performed at KSC are to be exercised using the same equipment and procedure at the contractor and NASA locations. This practice takes on two different aspects: (1) control of test requirements and (2) control of computer programs and test tapes.

1. Since the SC contractors are charged with the responsibility for development of ACE test requirements and many of the same requirements are used at KSC as are used at the factory, the contractor-furnished subprogram specifications are required identically at both locations for several SC. The SC contractors maintain the control of their specifications at all facilities.

2. The control programs used on ACE-SC are used at all sites and are maintained in the official library system. The subprograms and control program, once verified and accepted by NASA, are entered into the ACE-SC library system and also distributed to all applicable sites. The ACE-SC contractor maintains its software management and its software configuration management centrally at Houston and, through the use of software change notices, a software change control board (SCCB), and the ACE-SC library system, insures that any required change to a control program or subprogram is made in the ACE-SC library tapes and that it is expedited to all sites.

3. Test program tapes are verified at the location where they are to be used and are impounded by the responsible quality assurance agency at that site. All changes in test programs require a formal approval cycle prior to incorporation and formal verification of the change prior to use in SC testing.

VII. CONCLUSIONS

The acceptance checkout equipment spacecraft computer programing systems have the following advantages.

A. REDUCTION IN TEST TIME

1. The systems are capable of monitoring all critical data simultaneously.

2. The monitoring of data by computers minimizes human observation time by calling attention to deviation in tolerance and provides automated controlling of critical functions.

3. The use of standarized programs at factory and launch sites with only parameter changes from test to test reduces turnaround time for recycling tests and for reprograming from SC to SC.

4. Reduces postcheck and data analysis time by making available the complete test data tape for near real-time quick-look analysis by test personnel or complete reconstruction of a test.

B. AVAILABILITY

1. Software is modular in construction; that is, standarized control programs are used at all sites with associated subprograms.

2. The use of compilers to take parametric input generates new test tapes with a minimum turnaround time.

3. Any program can be run on any ACE-SC station.

C. REDUCTION IN COST

1. Centralization of programing effort using any manpower at any site to solve a peak load at another site allows minimum manpower levels.

2. Use of standarized programs from site to site insures that minimum effort is required to develop programs for each site.

3. Configuration management allows a change required at one site to be incorporated at all sites.

D. CAPABILITY TO MEET FUTURE NEEDS

1. The use of modular software permits easy addition of programs to meet future requirements.

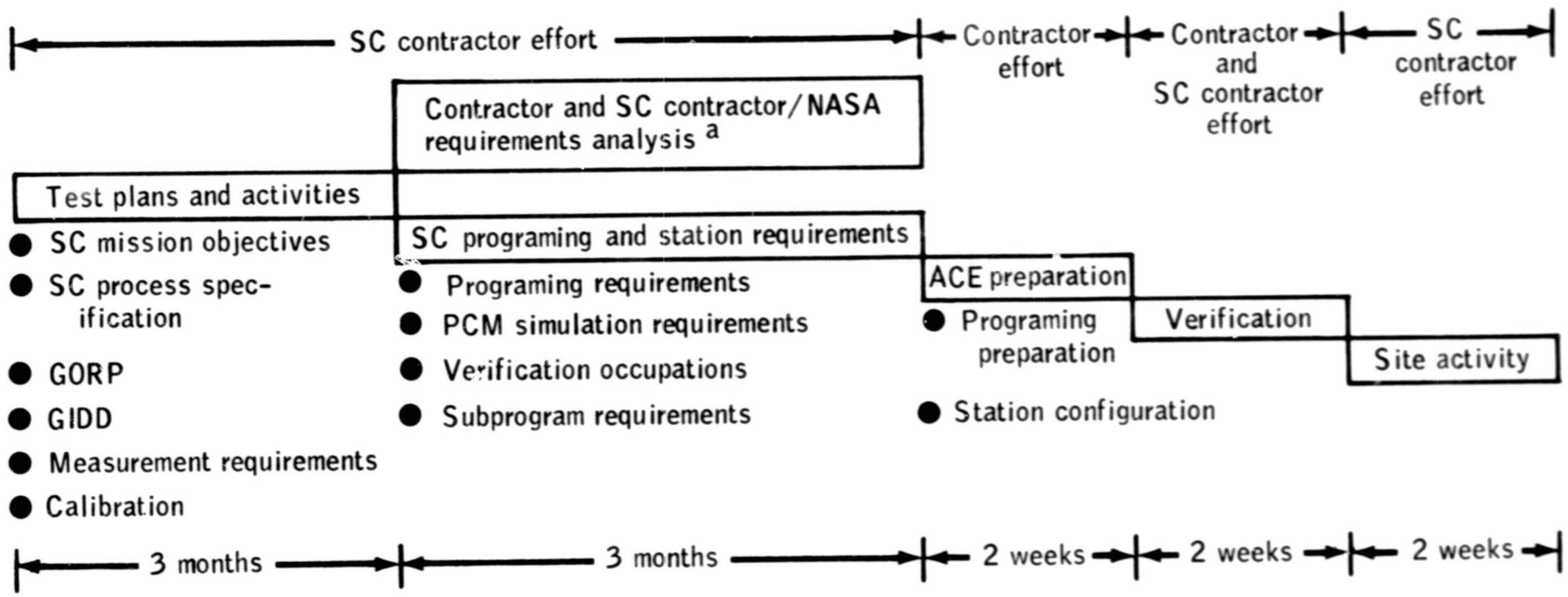
2. The concept of test phasing (rapid call-in/call-out) of specific sequences allows much growth potential.

3. Hardware is modular in concept so that additional memory or other components in system would be compatible with present computer programs.

4. The use of offline compilers to handle parametric input to generalized subprograms provides the capability to add parameters which complete the reprograming.

This information reflects the current status of ACE-SC computer programs. However, continuing changes in spacecraft hardware and test procedures and therefore in ACE-SC operations will cause eventual modifications. One of these presently foreseen is the addition of a mass memory device, which would be used to store more of the automated sequences and programs for rapid call-in of many routines to facilitate changing test loads. This mass memory device could also be used for enhancing the automation of the ACE-SC operation and for rapid access to data for near real-time processing.

NASA-S-67-5623



[a]Continuous analysis of new requirements to determine compatibility with present requirements and ACE concept.

Figure 9. - Test preparation timeline.

NASA-S-67-5624

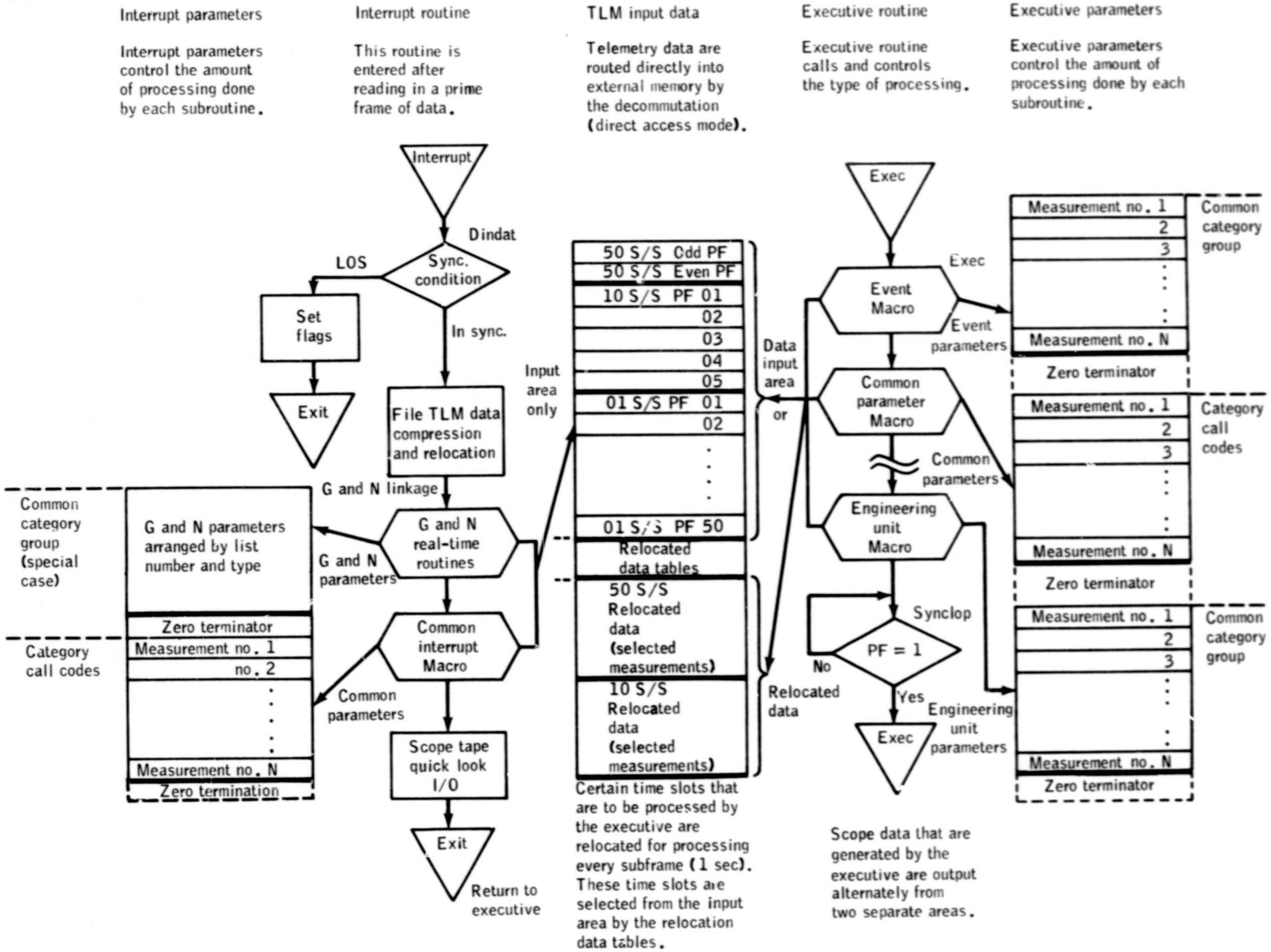


Figure 3. - Basic downlink theory.

NASA-S-67-5620

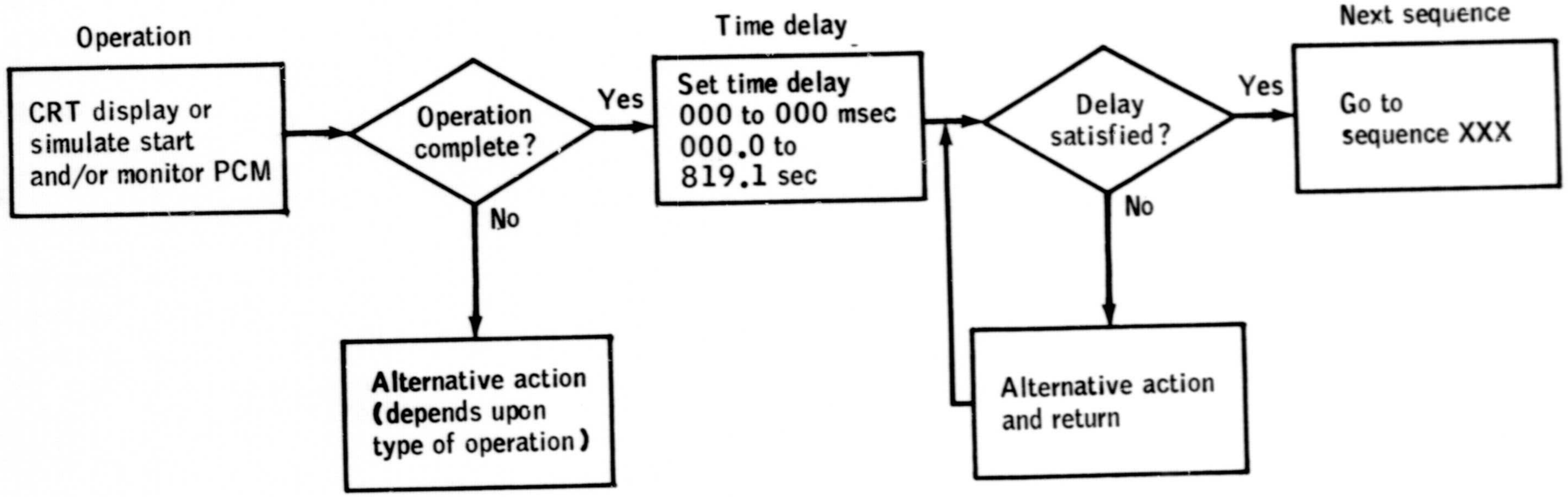


Figure 4.- Processing of ADAP sequence.

NASA-S-67-5619

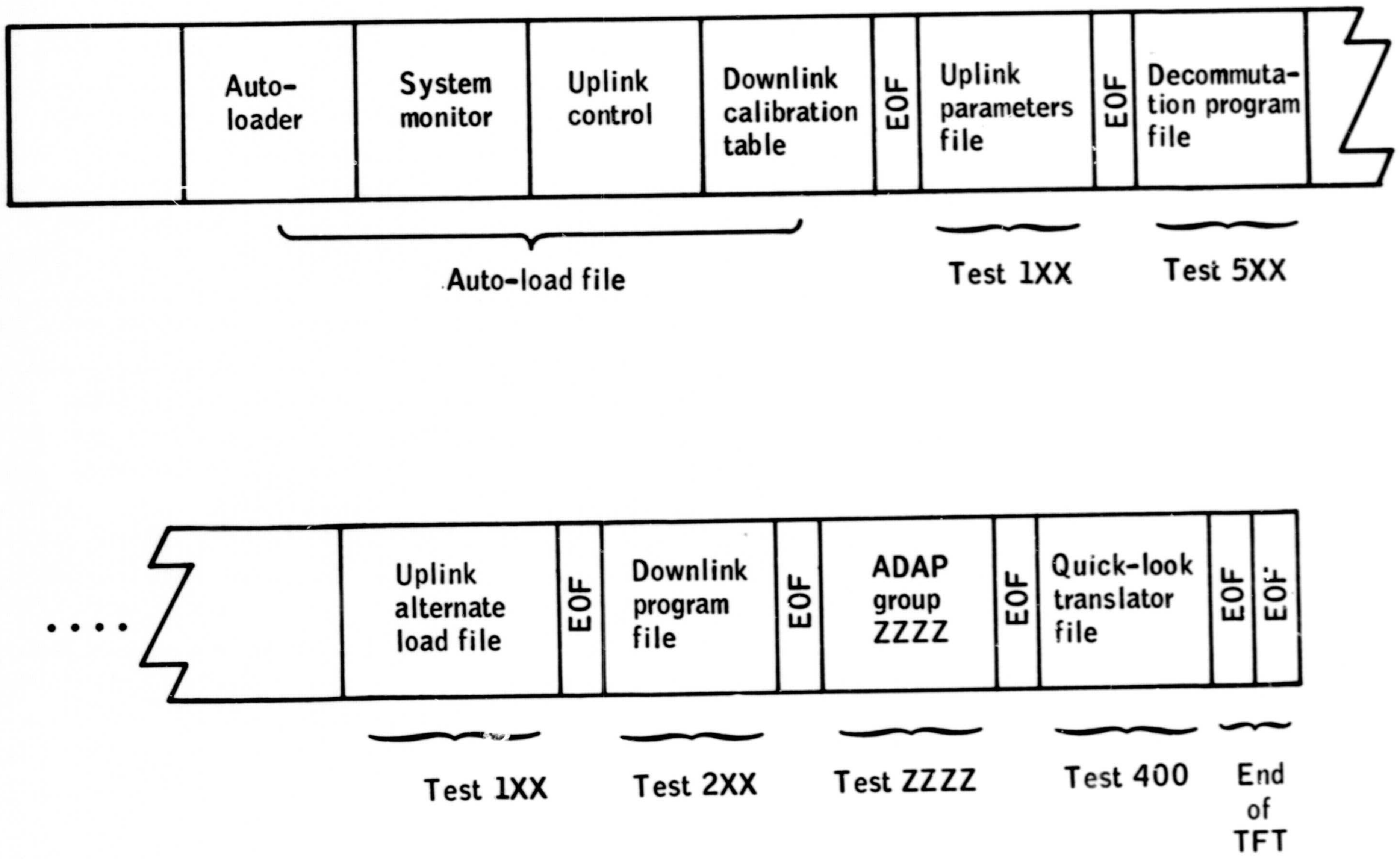


Figure 5.- Simplified TFT format.

NASA-S-67-5622

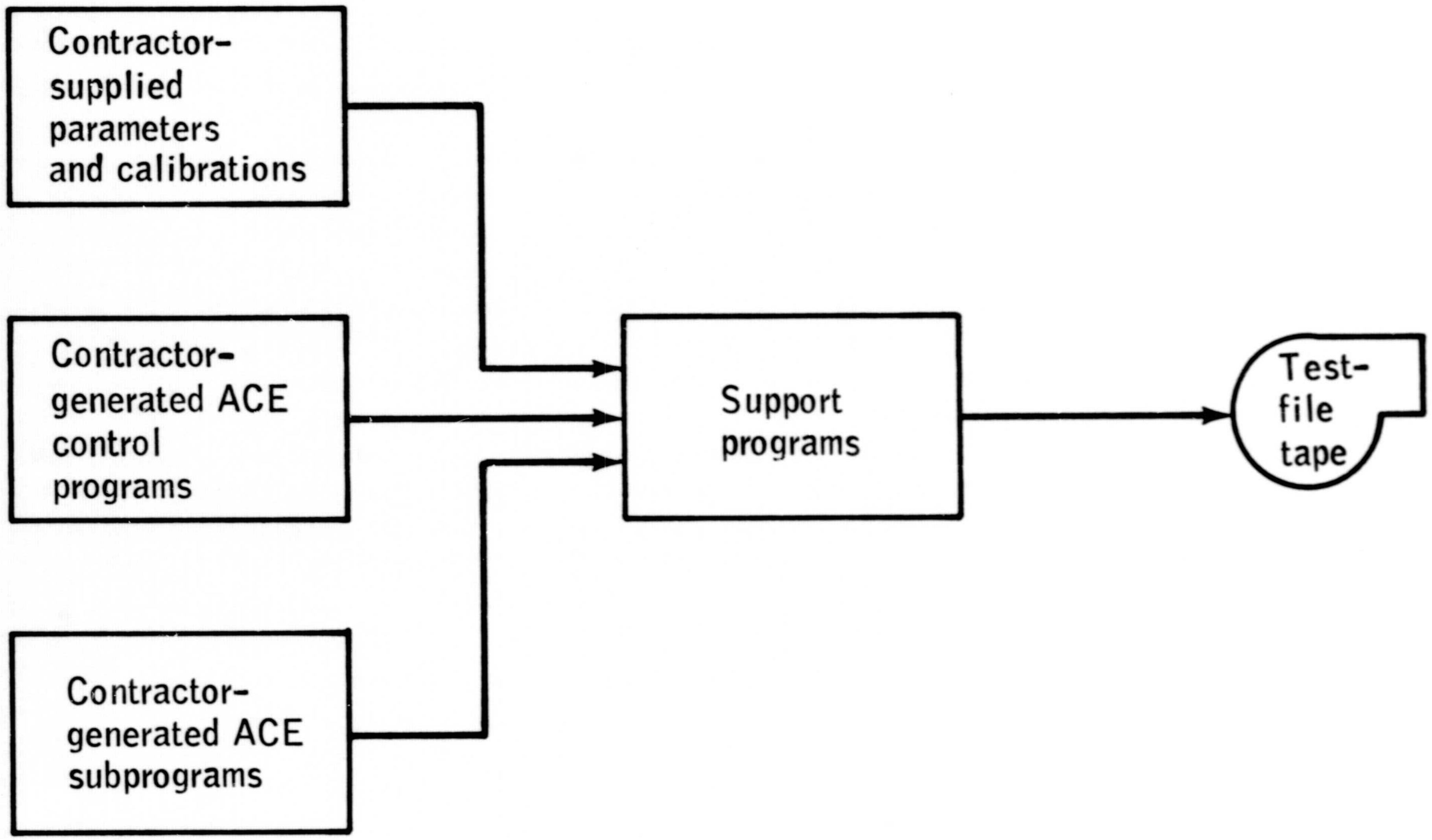


Figure 6.- Simplified test-file tape preparation.

NASA-S-67-5615

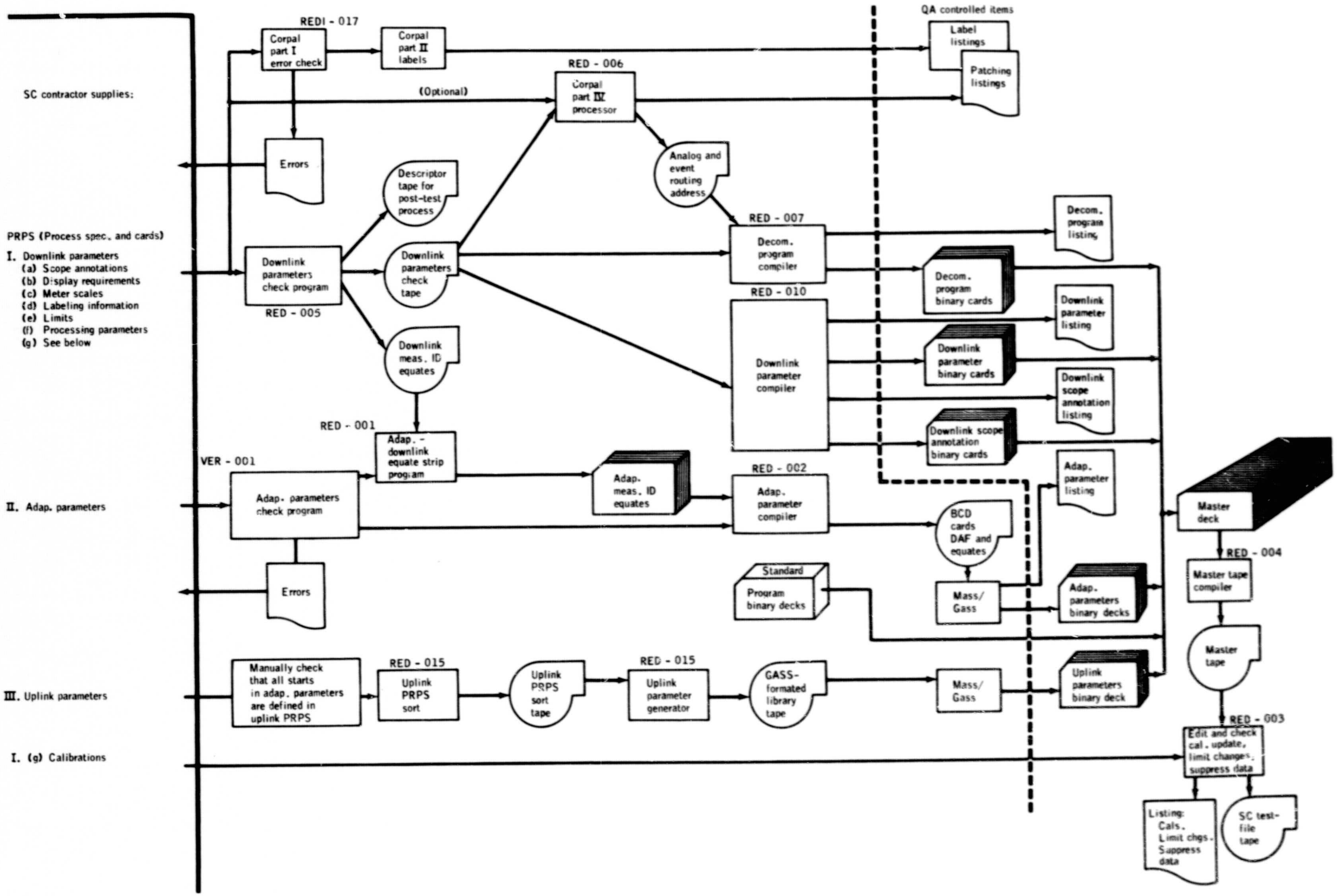


Figure 7. - Test file tape generation for the SC; ACE block II system, April 1, 1967.

NASA-S-67-5621

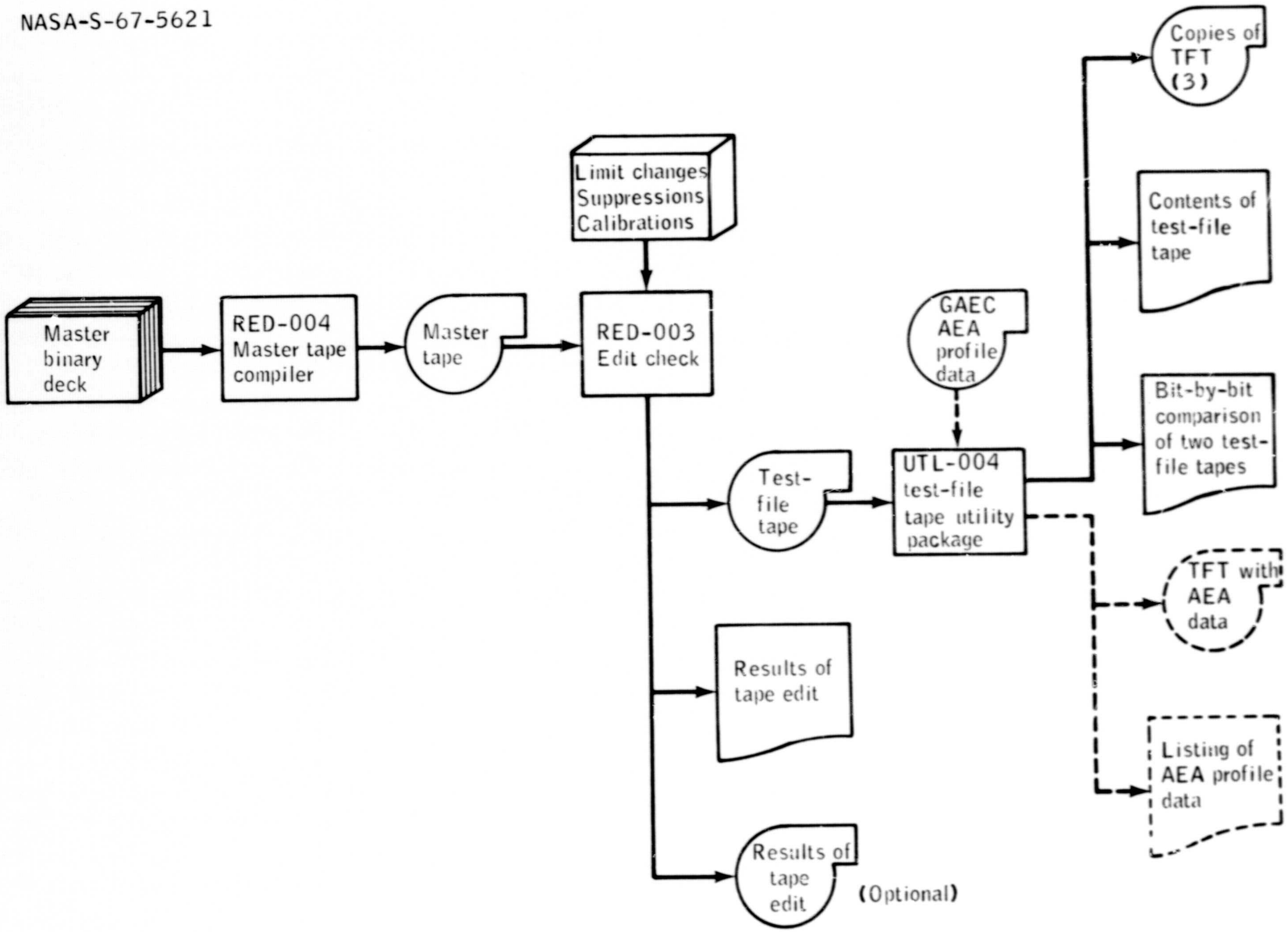


Figure 8.- Block II tape-making system.

NASA-S-67-5618

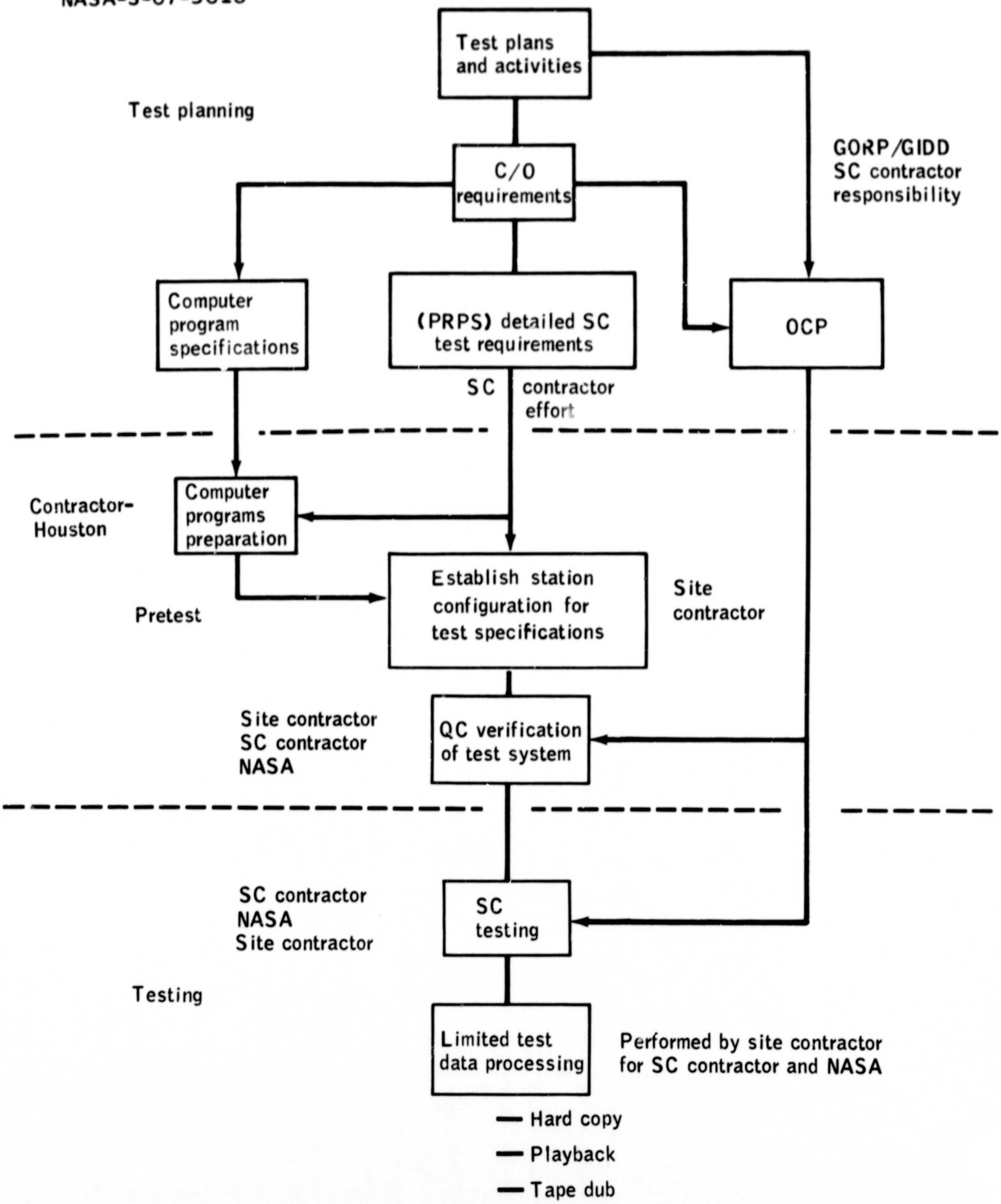


Figure 10.- Apollo ACE test cycle.

NASA-S-67-5618

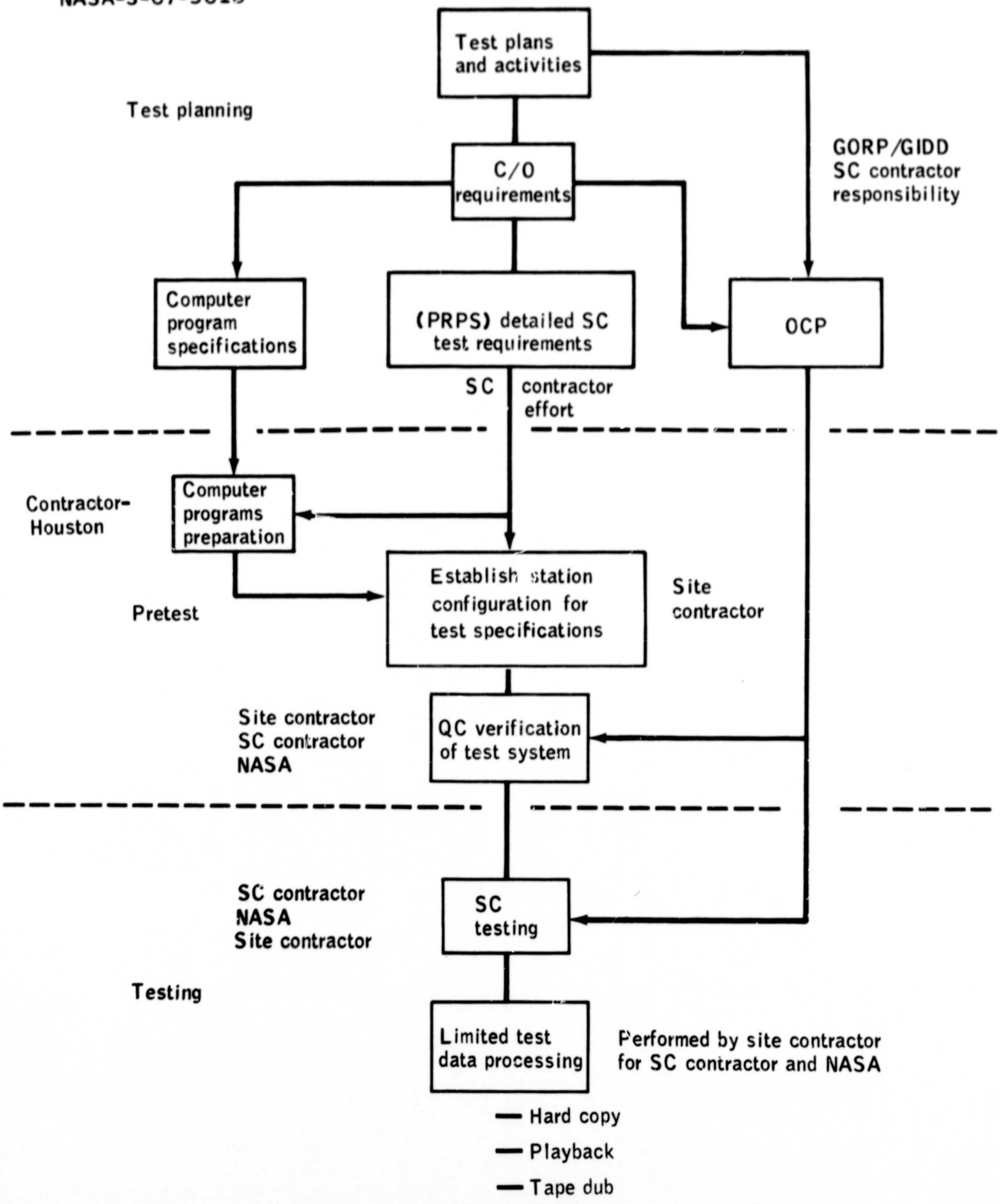


Figure 10.- Apollo ACE test cycle.

NASA-S-67-5617

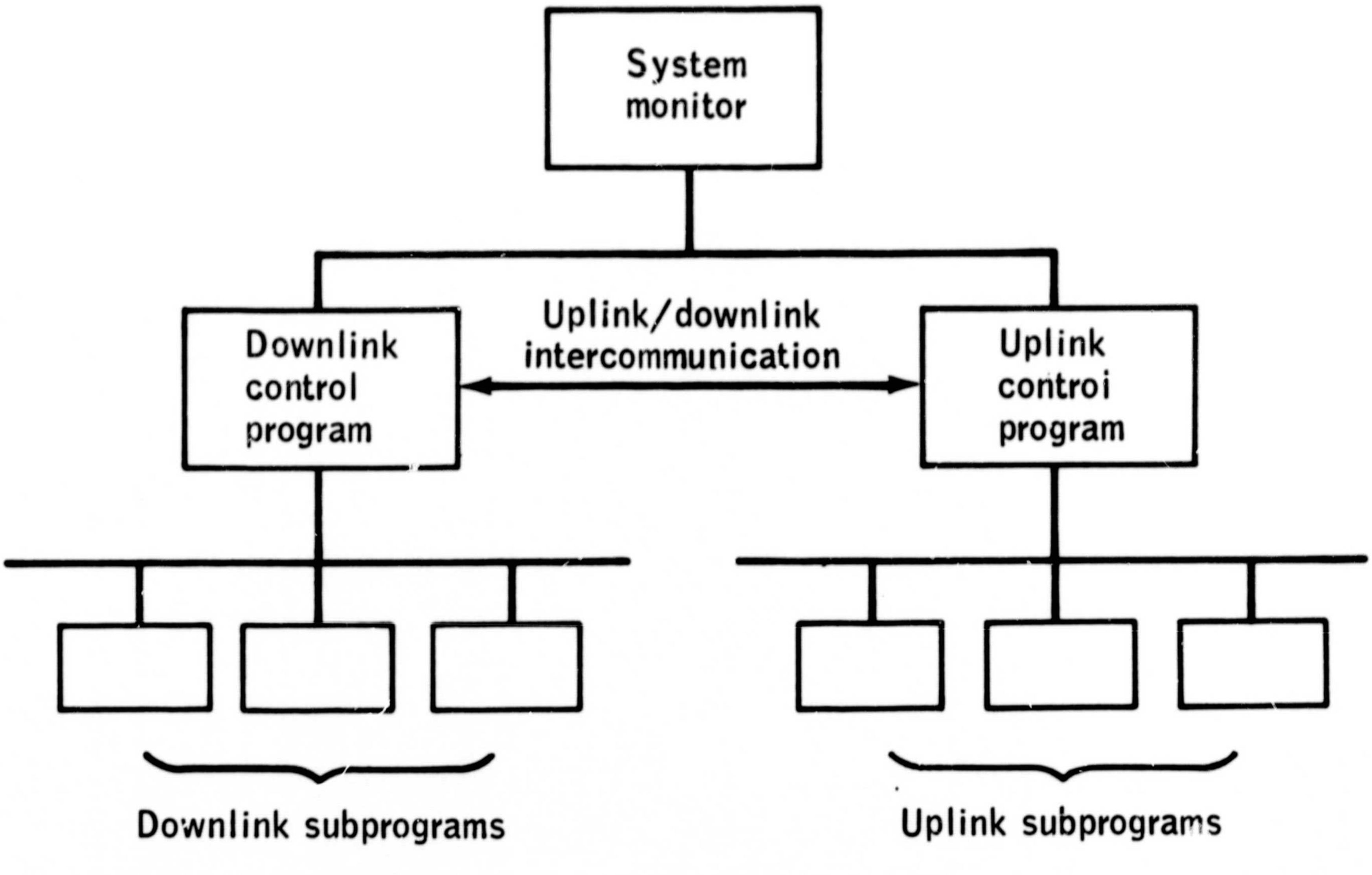


Figure 1.- ACE system operational program organization.

NASA-S-67-5617

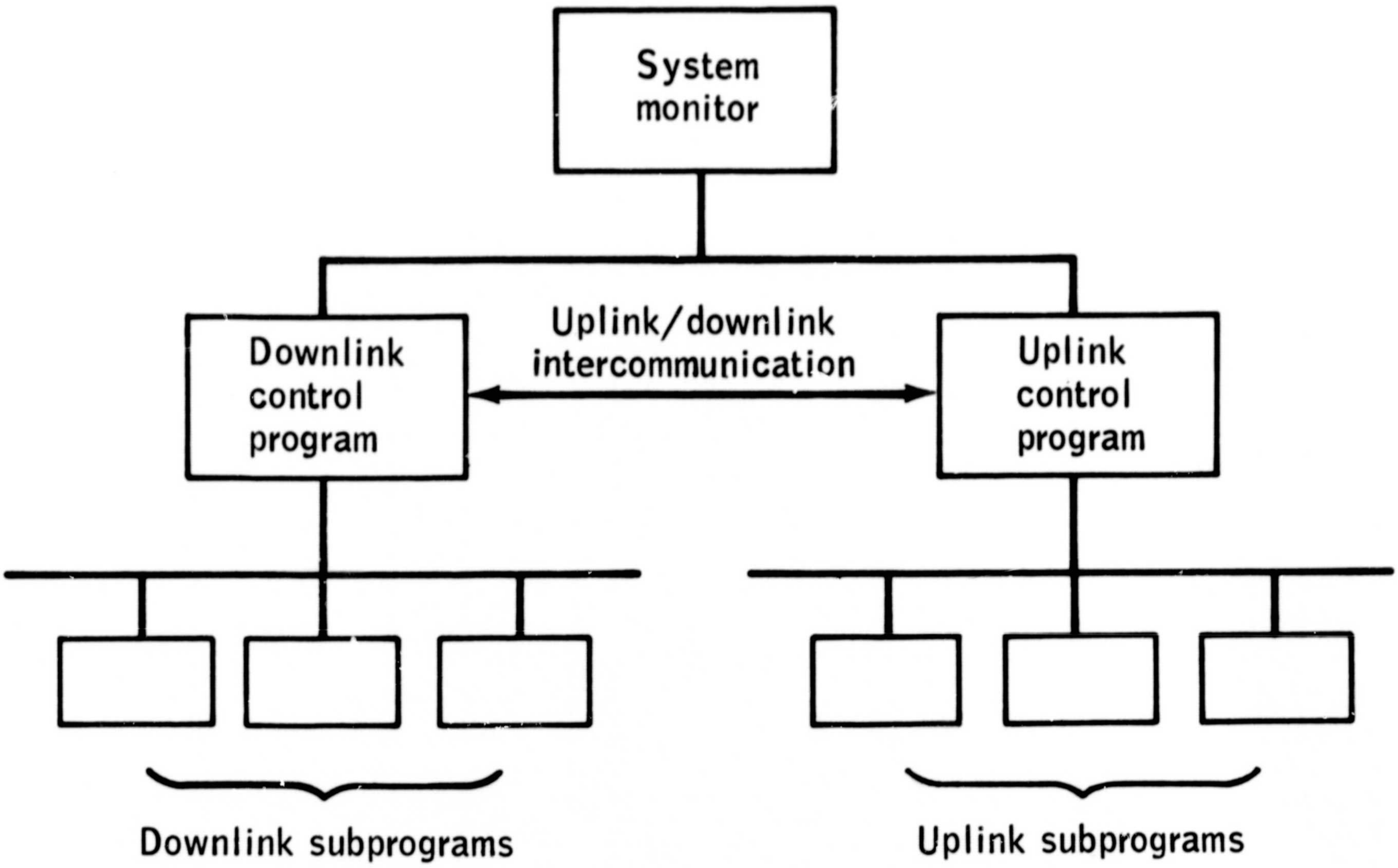


Figure 1.- ACE system operational program organization.

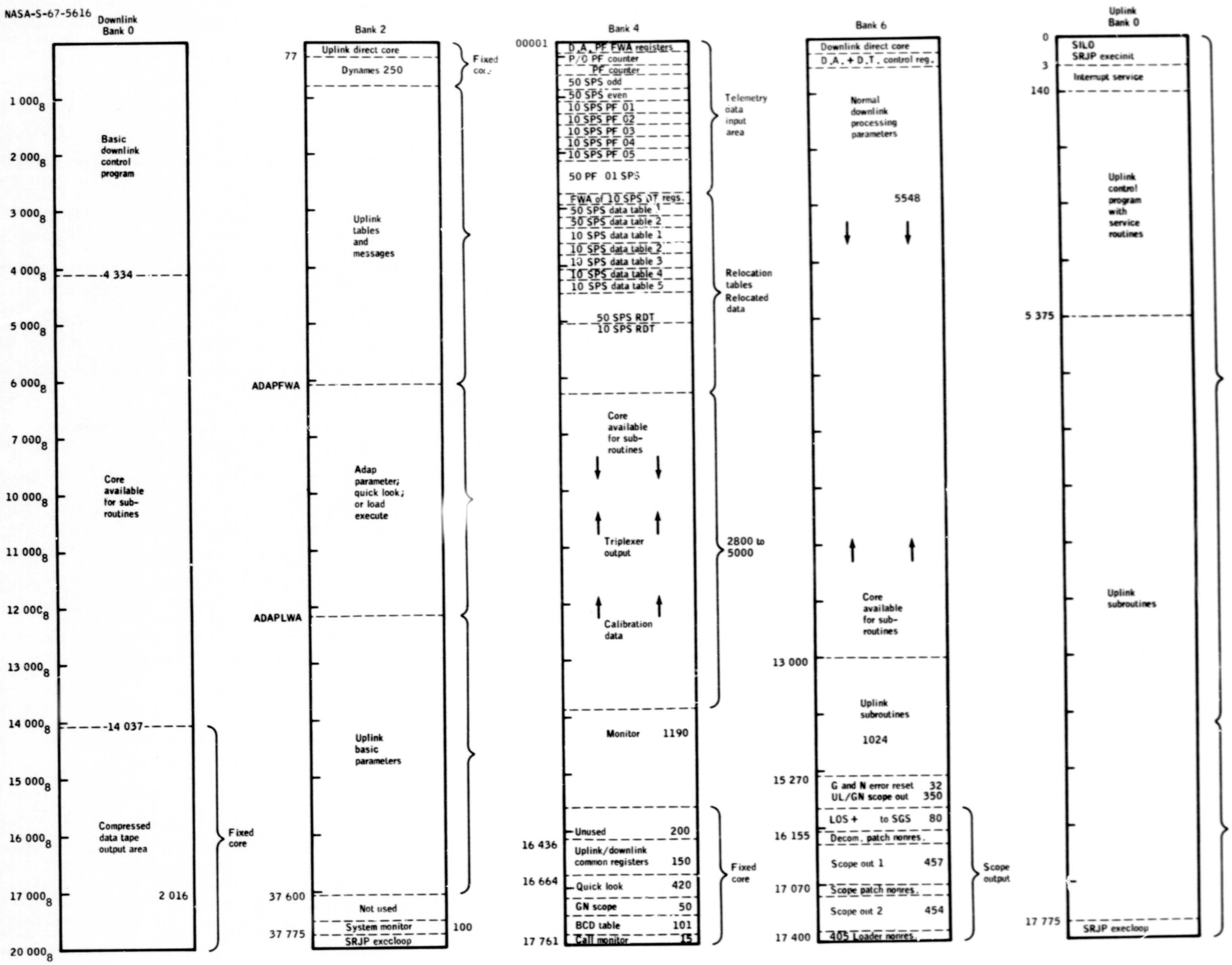


Figure 2.- Memory map for the ACE-SC programing system.

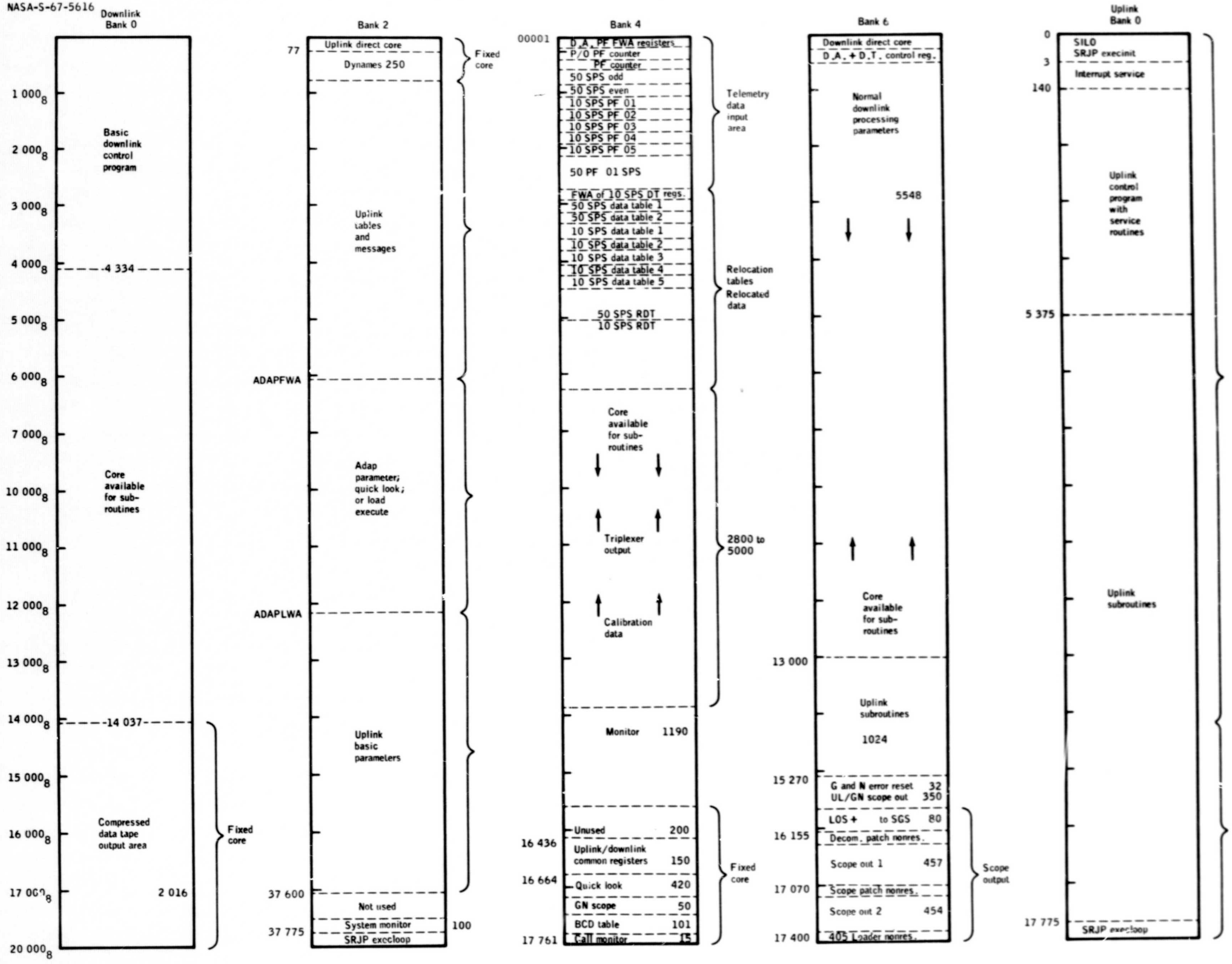


Figure 2.- Memory map for the ACE-SC programing system.

NASA-S-67-5617

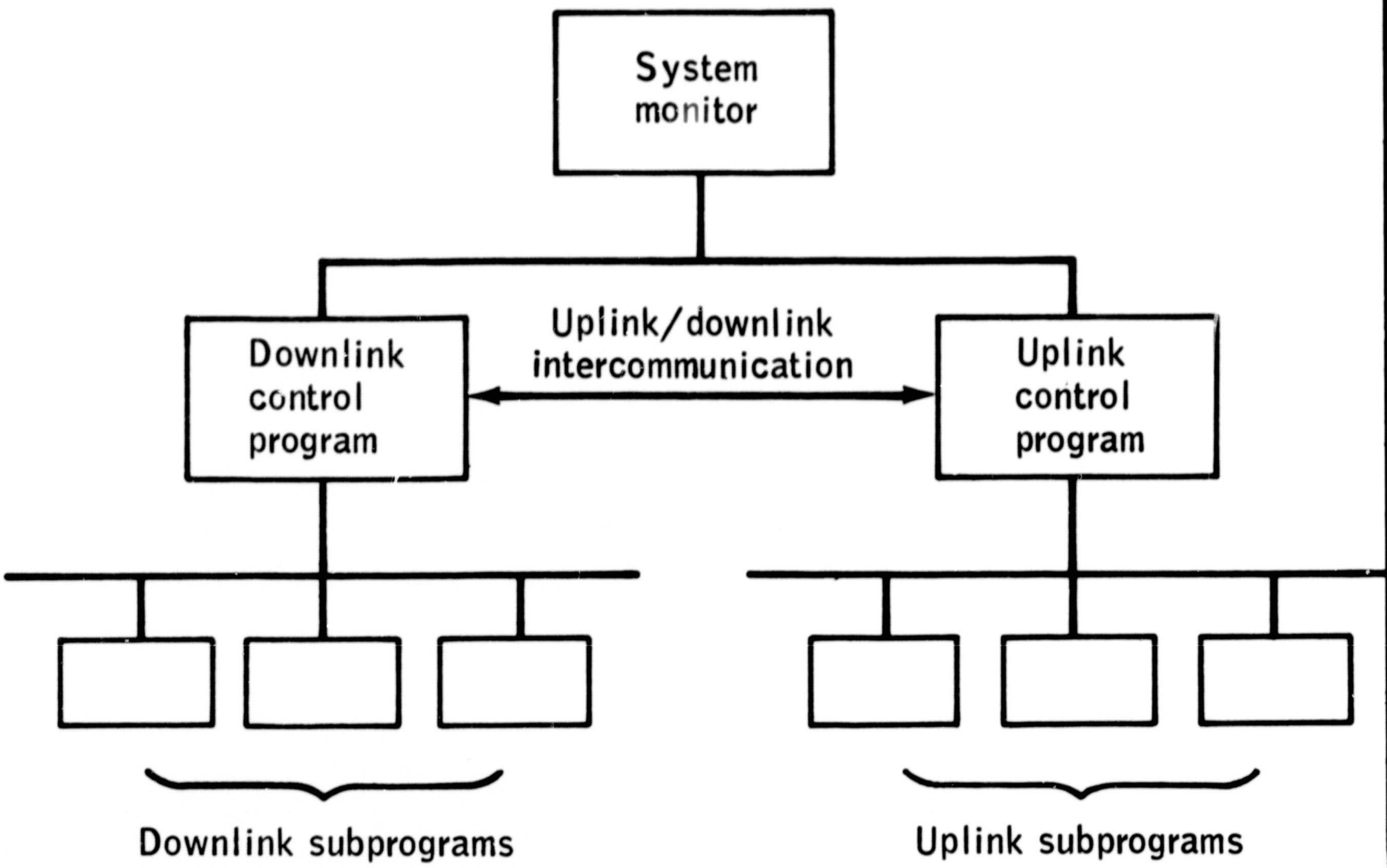


Figure 1.- ACE system operational program organization.