BELLCOMM, INC.

TR-65-222-1

STATUS AND SCHEDULES MONITORING OF APOLLO SOFTWARE

May 31, 1965

W. M. Keese
B. H. Liebowitz
W. J. Martin
I. D. Nehama
A. H. Scheinman

Work performed for Manned Space Flight, National Aeronautics and Space Administration, as part of Task 22 under Contract NASw-417.

[redacted]

BELLCOMM, INC.

TABLE OF CONTENTS

BELLCOMM, INC.

ABSTRACT

Effective monitoring by the NASA management of the status and progress in the development of computer software in Apollo requires: (a) A complete and up-to-date identification of the various computer software systems being developed; (b) A determination of the critical milestones to be monitored; (c) The existence of a mechanism for reporting and interpreting status and progress data.

A technique (the Systems Identification and Requirements Review, abbr. SIRR) to satisfy the first requirement has been applied on a portion of the total software effort in Apollo, viz., the real-time computer-based systems for SA-201, with encouraging results. It is recommended that the SIRR be used in the Apollo Program as a tool by the NASA management for control of computer software development.

The other two requirements have been met. Schedule and milestone visibility for computer software have been introduced into the established NASA reporting system (SARP).

BELLCOMM, INC.

STATUS AND SCHEDULES MONITORING OF APOLLO SOFTWARE

INTRODUCTION

An important management function for the purpose of maintaining control and direction of a complex program is the monitoring of its status and critical milestones. The production of computer software (programs) in the Apollo Program is a complex engineering activity, similar in nature to the processes involved in the development of the Apollo hardware. It was recommended in an earlier report* that, because of this similarity, certain techniques of information reporting to management already in use in the program for the control of hardware should be evaluated for possible application in software.

As part of the going effort on Task 22, two such techniques have been examined in detail over the past three months. These are: (a) the Systems Identification and Requirements Review (SIRR), and (b) the Schedules and Resources Procedures (SARP) reporting system.

The purpose of this report is to relate the approach taken in this analysis and to state the conclusions reached.** Detailed results are given in the Appendix. The work being reported on in this paper is concerned primarily with aspects of performance and schedules in software development. The third principal program element, i.e., cost, has been specifically excluded from consideration. However, in the broader effort on Task 22, considerations of costs of computer programs and means for controlling cost are very much included.

TWO TECHNIQUES FOR MONITORING SOFTWARE DEVELOPMENT

A. Systems Identification and Requirements Review (SIRR)

Each mission planned for the Apollo Program is designed to meet a unique set of objectives. Different system configurations

---

*Management Procedures in Computer Programming for Apollo - Interim Report, November 30, 1964.

**One concrete accomplishment of this effort, with the helpful cooperation of NASA personnel at the Centers and Headquarters, was the adoption of the SARP system for reporting software milestones.

(flight, ground, support, etc.) will be involved in different missions. This also means different sets of computer programs to perform the various functions required in the preparation for and conduct of a mission, such as checkout, launch control, guidance, tracking, command and control, etc.

It is characteristic of such complex, "one-time" projects that extremely rarely, if ever, do the planned and actual implementations coincide in all particulars, no mater how "well" the planning is claimed to have been done. There is no reason to demand or expect that the parallel development of a large number of complex computer programs will deviate from this pattern.

Thus, in order to ascertain that the stated mission objectives will be met, it is necessary for management to review periodically the progress of software development. Practical considerations often dictate limiting the scope of a single review to a small number of aspects of the development, leaving the coverage of other aspects to one or more separate reviews.

One type of review that has been used in the Apollo Program with good results is the SIRR*. This type of review focuses principally on:

a) the identification of the end item(s) being developed;

b) the demonstration that the development of the identified end item(s) is

   i) according to well-formulated requirements that derive logically from the stated mission objectives, and

   ii) orderly, i.e., under control;

c) items (a) and (b) for all interfaces between the system(s) under review and other systems, with special emphasis on consistency of requirements.

---

*Systems Identification and Requirements Review, (Memorandum for File), C. A. Bidgood, July 29, 1964.

Assessment of Saturn IB Launch Complex Systems Identification and Requirements Review, (Memorandum for File), L. G. Miller, H. E. Stephens, November 20, 1964.

It hardly needs mentioning that other aspects of program development, such as technical and administrative responsibilities, costs, schedules, etc., may be added to the above list.

The details of setting up a SIRR will not be gone into here, except to emphasize one or two aspects on which the practical "success" of most reviews seems to be particularly sensitive. First, the scope of the SIRR must include only those items or activities which are related strongly in some way or other, e.g., physical interconnections, functional interdependence, simultaneity of operation in accomplishing a single mission function, etc. Second, the frequency of holding SIRR's must be so chosen as to yield the maximum usable information (it is impossible to give a more specific rule). In general, a SIRR with emphasis on software should be held "immediately" following the issue of the SOFTWARE PERFORMANCE SPECIFICATIONS, and "immediately" preceding the activity of ACCEPTANCE TESTING.

<u>In-House SIRR of Software for SA-201</u>

In order to evaluate the effectiveness of a SIRR as a tool in the management of software in Apollo, it was decided to simulate such a review in Bellcomm by performing the following activities:

(a) Select an Apollo mission (or a phase in a mission) with well-established objectives and operational requirements, and focus the SIRR on the information processing systems being developed for that mission.

(b) Prepare a list of the objectives of the review and aspects of software development and production to be covered in the review.

(c) Conduct the review in-house by collecting the information needed to respond to (b) from all available sources, i.e. published documents, and contacts with the personnel at the NASA Centers responsible for the development of the identified systems.

(d) Evaluate the results of the SIRR.

The mission chosen was SA-201; in particular, that portion of SA-201 delimited by the start of preparation for launch at Launch Complex 34 and the end of launch vehicle guidance operations. (For a brief description of the objectives and operational profile of SA-201, see Figure 1, Appendix A).

The following is a list of the on-line software systems required for SA-201:

MSC

1. Real Time Computer Complex
2. ACE-S/C, Checkout

MSFC

1. Saturn Launch Computer Complex
2. Saturn Guidance Computer
   a. Prelaunch
   b. In-Flight

GSFC

1. Network Status Testing (CADFISS)
2. On Site Processing

KSC

1. Central Instrumentation Facility, Real-Time Data Reduction and Display

The review concentrated on the following aspects of development for each system:

A. Identification of Responsibilities

Requirements and Interfaces
Implementation
Operation
Documentation

B. Implementation of the System

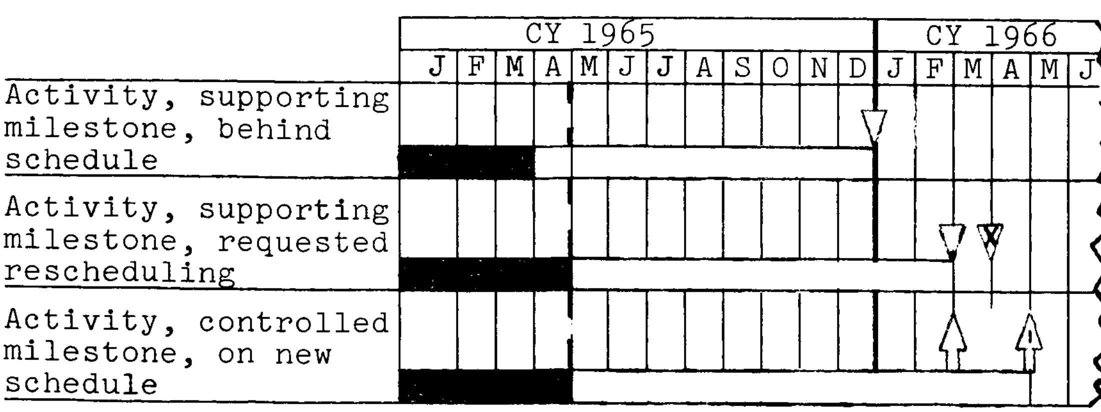
Functional Requirements
System Interfaces
Constraints
Software Performance Specifications
End Items
Schedules and Current Status
Documentation

Detailed results are given in Appendix A.

B. Schedules and Resources Procedures (SARP)

The SARP chart reporting system provides a means for NASA Headquarters to monitor the progress of engineering tasks within the Apollo Project. These charts, see figure below, present in a calendar-like form the schedule status of key milestones and activities within the project. Triangles and arrows are used to show the expected completion date of an event or activity. Each month this estimate is updated and changes in status reflected in the chart. A new estimate is indicated by the addition of a new triangle or arrow with an X in it (see figure below, line 2). Once the Center involved or OMSF has approved the schedule change, the X is replaced by the number of total changes (see figure below, line 3). The addition of triangles and arrows gives an indication of the changing status of the milestones. The status of activities which exist over a significant period of time is shown by a bar in the charts, this bar stretching over a period of time from the beginning to the end of the activity. Monthly progress is shown by shading in the bar (i.e., if the current date is May 1965 and progress on the activity is such that its status is that which was anticipated for May of 1965, the bar will be shaded only to April).

The status of the production of computer software can be monitored via the SARP chart system. A set of milestones appropriate to the wide range of software systems being developed for Apollo has been determined, and is shown in Appendix B. From this set a subset of four milestones are selected as appropriate for reporting by the Field Centers to NASA Headquarters. These



Typical SARP Entry

milestones are:

(1) System Performance Requirements Document Available

The completion of a document generated by the NASA Centers describing to the contractor the job to be done by the computer based systems as it affects computer programming.

(2) Software System Performance and Preliminary Design Specifications Document Complete

The completion of a document by the contractor describing the way in which he plans to do the job specified by the system performance requirements document.

(3) Operational Software Subsystem Programs Coding and Debugging

The period of time during which the operational system is coded and debugged. This activity terminates in the milestone "debugging complete." At this time, integration of the program packages can begin.

(4) Operational Software Subsystem Acceptance Testing Complete

The point in time when the NASA Center has agreed that the acceptance testing of the operational software subsystem has been successfully completed.

CONCLUSIONS AND RECOMMENDATIONS

Two techniques for monitoring by management of the status and critical milestones of computer software development in Apollo were evaluated.

1. The SIRR

The effectiveness of one of these techniques, the Systems Identification and Requirements Review (SIRR), was tested by conducting an in-house review of the on-line software in SA-201. When the necessary allowances are made for the fact that it was a simulated review, the results indicate that the objectives of a SIRR can be realized in practice. For example, partly as a result of this effort, some sixty (60) different software systems have been identified (see Appendix B) to be at various stages of development in Apollo, compared to seventeen (17) before the review was made. This result is only partially explainable by the fact that this was the first time that such a task had been undertaken by NASA Headquarters. The important point is that end items and requirements can be identified via the SIRR. It is by no means clear that a SIRR need be a formal event with management personally participating. Since the SIRR is not intended to be a decision-making mechanism, the necessary activities can be carried out by the specialists in the management-support staff - in the case of computer-based systems, by information processing specialists. It is recommended that a SIRR of the computer-based portions of the ground operations support and other related systems for SA-201 be authorized and conducted prior to the initiation of acceptance testing.

2. The SARP System

As part of the effort to introduce software in the SARP system, visits were made to various NASA Center organizations involved in the production of the identified software systems to determine the appropriateness of the various milestones to their programs. Several problem areas became evident as a result of these discussions. Problems arise from the difficulty of depicting, with a few milestones, the status of a complex task with many parallel interrelated activities. Specific manifestations of this problem are:

a. Difficulty in choosing a milestone representing the end date of what may be a continuing activity. Outputs of certain activities (such as generation of requirements documents) may be available before completion of the activity. Therefore, significant progress within an activity may be difficult to identify using a single milestone. If the task is fragmented sufficiently to identify partial completion, the number of milestones monitored by MSF may increase greatly.

b. Milestone "bunching." The milestones identified for SARP chart reporting may bunch in time leaving gaps for which there are no major milestones.

c. Difficulty of determining relationships between events on the SARP charts thereby obscuring, in many cases, the cause of schedule slippages.

d. Lack of definitive standards indicating the completion of milestone event. In lieu of standards, agreements between MSF and Center personnel as to the meaning of "event completed" are required.

e. Lack of standards for measuring the progress of an activity. The movement of the shaded portion of the bar in the SARP charts provide an indication of progress which may be interpreted differently by different people for different activities.

These problems tend to minimize the usefulness of the SARP charts by maximizing the need for careful evaluation of the information contained therein. This problem is inherent in any system that attempts to depict the status of a complex activity while keeping the number of milestones at a manageable level. The need for well defined measures of programming progress and new methods of reporting are evident. Studies are required in this area. Despite these problem areas it is felt that inclusion of the milestones mentioned in this report will greatly enhance computer programming schedule visibility at MSF, and enhance overall management control in Project Apollo.

W. M. Keese

B. H. Liebowitz

W. J. Martin

I. D. Nehama

for A. H. Scheinman

WMK
BHL
1031-WJM-amh
IDN
AHS

Attached:
Appendix A
Appendix B

APPENDIX A

BACKGROUND

A. Brief Mission Description

Apollo flight mission SA-201 is the first mission on Saturn IB of the L/V-CSM development type. A Block I configuration will be used on the spacecraft with a mission programmer in the CM to generate a pre-programmed set of commands to achieve the flight objectives. There will not be a guidance computer in the CM for this flight. Water landing and CM recovery are planned.

A partial list of flight-specific information follows:

Mission Objectives:

Launch Vehicle development.

S-IVB and IU checkout.

Compatibility and structural integrity of the CSM and S-IVB during boost.

Verification of CSM Systems operation (reaction control system, stabilization control system, service module propulsion system, environmental control system, electric power system, and communications system).

Heat shield verification at approximately 29,000 fps with (a) maximum heat rate and (b) maximum heat load.

Vehicle Configuration:

CSM [Block I]: Air frame 009.

LEM, not included.

LV: 201, consists of S-IB, S-IVB, IU [includes Launch Vehicle Digital Computer (LVDC) for guidance and monitoring function].

Abridged Mission Profile:

Powered flight of launch vehicle (non-orbital supercircular entry "lob-type" trajectory).

CSM/S-IVB separation.

Utilize service module propulsion system to achieve desired entry conditions for maximum heat rate.

Computer-based Systems:

The only computer-based system in the space vehicle is the LVDC in the IU. For SA-201 all other such systems are in various ground support facilities, which include the following "on-line" systems:

i. Launch Complex 34

Launch Vehicle Checkout and Launch Control

ii. MSO Building

Acceptance Checkout Equipment (ACE) for CSM

iii. Central Instrumentation Facility

iv. Mission Control Center

Real Time Computer Complex

v. Goddard Space Flight Center

Manned Space Flight Network
Network On-Site

In Figure 1 an abbreviated operational profile of the major computer-based systems reviewed here is shown superimposed on the mission profile of the SA-201 space vehicle.

APPENDIX A.1

REAL TIME COMPUTER COMPLEX (RTCC)

Identification of Project Responsibility

A. Prime Responsibility

The responsibility for providing a data processing capability to the IMCC for simulation, training and the analysis and display of data during a mission rests with the Manned Spacecraft Center. The implementation of requirements for this Real Time Computer Complex is the responsibility of Mission Analysis and Planning Branch of MSC. The responsibility for integrated testing of the RTCC resides with the Flight Support Division of MSC.

B. Requirements and Interface Responsibility

Specific requirements for a mission come from various divisions at MSC. These include Flight Crew Support Division for spacecraft operations and simulation input, Crew Systems for biomedical and environmental inputs, Flight Operations for Trajectory, abort and LES inputs; in short most if not all operating divisions contribute inputs. These inputs take the form of memorandum specifications, operation handbooks or analyses of particular facets of the mission. The coordination of these inputs and the writing of a set of mission independent requirements is done by the Mission Analysis and Planning Division. These represent general software requirements and may be found in document PHO-TR-120 Vol II, Computer programming requirements. Mission dependent requirements such as the S-IB Programming are also generated by this division.

The interface between the RTCC and the Launch Trajectory Data System (LTDS) is administered by the Mission Analysis Division by memo and coordination meetings between this Division and the Astrionics Division of MSFC (who are responsible for implementation), and IBM. General software requirements may be found in PHO-TR 120-Vol I.

There also exists an interface between Goddard and the RTCC concerning inputs needed to reflect the status of the radar network used for tracking. The Data Operations Branch at Goddard and the Flight Support Division, MSC, have coordinated on this aspect of data inputs. The inputs are in PHO-TR 120. See Figure 2.

C. Identification of the User

The user of the RTCC is the Mission Director and his staff in the Mission Operating Control Room (MOCR), where the data processed by the RTCC is displayed. Other important secondary users are the range tracking stations that are furnished ephemeris data needed for acquisition of the vehicle during flight.

Implementation of the System

A. Functional Requirements

The Real Time Computing Complex receives tracking data from the Ground Operational Support System (GOSS) and processes it on computers to meet the following requirements

(1) Provide the Missions Operation Director and his staff via displays with selected information on the status of the mission and its associated subsystems during the life of the mission.

(2) Provide vehicle acquisition data to sites for proper tracking of spacecraft.

(3) Receive and process telemetry for analysis during and after a mission.

(4) Provide the necessary data processing to generate displays to support simulation, check-out and training.

B. Description of RTCC System

The RTCC receives it telemetry and trajectory information from the Launch Trajectory System (LTDS) at KSC. This information is received by the Communications Processor (CP) which acts as the switching computer for the RTCC. It in turn feeds it to 7094 computers through a system selector unit. Here the information is processed and fed back thru the CP to the Mission Operating Control Room for display or back to the sites in the form of ephemeris data for acquisition purposes (see Fig 3).

The operational programs that perform the processing described above are of two types with a third type called an executive. The first two types are processors and supervisors. A processor is a mission independent program that performs a specific function such as a determination of basic parameters such as apogee, perigee or inclination to the equator. The supervisor is a program that defines which of the many processors are to be used and the order in which they are used. These supervisors are mission dependent since they reflect imputs and processing for a particular mission profile. The third type, the executive is mission independent and handles the inputing and outputing of data to the processors. It is the interaction of the processors under supervisor control that performs all data processing for a mission.

The RTCC computers interface only with the Mission Operating Control Room for transmission of its outputs for display or with the LTDS for receiving inputs or to send ephemeris data to sites.

C. Constraints

The Mission Analysis Division must receive timely mission specific information from all MSC divisions to be served by the RTCC in order to produce the requirements for software supervisors. Any delay in the receipt of such information will result in the corresponding slippage of supervisory program schedules.

D. Software Specifications

The 201 detail software specifications required for mission independent processor programs are generated under the direction of the Mission Analysis Division at MSC by IBM. These are published by IBM as program descriptions and signed off by the Mission Analysis Division.

There are no detailed software specifications for the mission dependent programs since the needed requirements have not been published.

E. End Items, Milestones and Schedules

The end items for 201 processors consist of documentation and the operating programs. The documentation may be found in the IBM Programmer Working Books Vol II, with the operating programs being kept at the RTCC installation.

Schedules and Milestones are not available due to the non-availability of the mission specific requirements.

F. Status and Schedules

A realistic schedule giving dates for completion of 201 supervisors is attached. (See Figure 4) It is based on a date two months prior to when the RTCC will be needed for 201.

APPENDIX A.2

ACCEPTANCE CHECKOUT EQUIPMENT (ACE)

Identification of Project Responsibility

A. Prime Responsibility

The Manned Spacecraft Center has the responsibility for the advanced checkout system that provides for the assurance of systems using computers and test equipment.

The Apollo Space Project Office (ASPO) at MSC has been designated as the MSC unit with overall responsibility for the Project.

"It (responsibility) includes ground computer, all ground displays -- processing and transmitting checkout information." This statement may be found in NASA document, Checkout Policy and Directives for ACE-S/C Implementation dated March, 1964.

B. Requirements and Interfaces

The inputs for the programming requirements come from a series of documents that have been generated for MSC-ASPO by North American Aviation. They include the following types:

1. Checkout Process Specification which establishes basic configuration requirements, sequences, stimuli measurements and limits.

2. Program Requirement Process Specifications that specify general data processing requirements.

The inputs for these NAA documents come from MIT for G&N information or from Florida Operations concerned with GSE. (See Figure 5) The result is a general software requirements document entitled, "Apollo Spacecraft Programming Requirements Definition", Technical Document No. 5C, MSC Florida Operations, March 3, 1964.

C. Identification of the User

The ultimate user of ACE station is the test conductor and the test engineers engaged in testing out the spacecraft for a mission.

Implementation of the System

A. Functional Requirements

The Acceptance Checkout System is the computer checkout of S/C systems through control of S/C servicing equipment to meet the following requirements:

1. Provide for the generation of command and stimuli by computer to the S/C and GSE for test purposes.

2. Provide for the processing by computer of the measurement and response from the S/C and GSE.

3. Provide the necessary associated equipment at a Ground Station under computer control for display of the processed S/C generated data.

B. Description of the ACE System

An ACE station consists of two Control Data 160-G computers with associated ancillary gear, called the uplink and downlink computers. The uplink gives command and stimuli to test a S/C subsystem. The down-link receives the resulting measurement and/or response from the subsystem, processes it, to drive test engineer CRT displays or record it for further reduction.

The two computers share a common memory that acquires and transmits data through bi-directional Data Transmission and Verification Units (DTVC) and Data Acquisition and Decommunication Equipment (DADE). Provisions are made for storing on tape drives acquired data from the down-link computer.

The uplink computer controls the sending of commands that can activate equipment or send stimuli to operating equipment. It sequences the tests as required by the computer or as initiated by the test engineer at a control console through the Communications Unit Executor (CUE) to the computer. These computer requests are processed utilizing information from the test tape and become test sequences that determine which subsystem receives stimuli. These sequences are formatted, verified and transmitted by the DTVC that relays the command via hard line or RF Telemetry to the subsystem being tested.

The downlink computer receives telemetry from the tested spacecraft subsystem or spacecraft-vicinity test equipment through hardline or RF receivers. These data flow through a video distribution unit and are then decommutated (by the DADE) and checked prior to sending to the computer memory. Once processed the downlink computer takes the acquired information etc., from memory and transmits it to the DTVC where it is verified for transmission to the Digital Communications and Control Unit (DCCU) for formatting to the symbol generator. The symbol generator creates the inputs for CRT displays that are used by the test engineer in determining status of the subsystem under test. (See Figure 6)

The ACE operational software is broken down into three general classes of programs: Uplink, for processing commands and stimuli; downlink, for processing the measurement or response from a S/C system being tested; and test diagnostics, used for hardware and line assurance. All these programs are mission independent. In addition test requirements are needed which are wholly mission dependent. This is data which defines the particular system, its inputs and outputs, tolerances, type and location of test, etc. This information is prepared and assembled for computer use. The combination of the operational programs plus the test requirements is the software needed for the ACE computers.

C. Constraints

A mission specific program is assembled by properly sequencing the operational programs. It is absolutely essential to receive timely test requirements from the appropriate contractors to enable the necessary sequencing of the operational programs and to provide needed test data as inputs to the operational programs. Failure to receive this information on time will result in slippage of the scheduled software completion date.

D. Software Specifications

The responsibility for software was delegated to MSC-F/O. The implementation of the software for all ACE sites NAA, AMR, etc., was contracted to the Apollo Support Group of G.E.

The G.E. group has produced several documents relating to software specifications. The more general and complete is the "Computer Programming Plan ACE-S/C Ground Stations," contract NASw-410, March 23, 1964. This gives flow charts and descriptions of the various types of programs along with its relation to other ACE equipment.

A more detailed specification and description has been produced for the ACE #1 station at NAA. Although hardware oriented, the flow of data within ACE is better described.

E. End Items, Milestones and Schedules

In defining the end items it must be pointed out that the first ACE station is operating at NAA. Hence, the end items for this installation are identifiable. They consist of the running programs plus documentation. A list of the programs may be found in MSC document; PG/0-1062-64, dated November 2, 1964. Milestones and schedules for this installation have been detailed in the Computer Programming Plan.

As far as ACE ARM #1 no end item, milestone or schedule information on software is available.

F. Status and Schedules

The status of ACE programs can be summed up as follows for 201:

If the ACE programs currently being used at ACE #1 NAA are used at ACE AMR #1, then it can be said that the AMR software is finished. There is no indication that this will not be the case.

There are no schedules for ACE AMR #1 available. Hence, no judgment can be made for this phase of the project.

APPENDIX A.3

SATURN LAUNCH COMPUTER COMPLEX

Identification of Project Responsibilities

A. Prime Responsibilities

The task of checkout and countdown of the launch vehicle is done by the Astrionics Laboratory at MSFC. This organization is responsible for the planning, implementation and testing of these programs.

B. Requirements and Interfaces

The general task simply evolved and is not set forth in any particular document.

The general requirements are generated in various working committees or groups which are made up on a very mixed basis; some members will be from LVO, some from other organizations in MSFC, etc. At least one member of each such group will be a systems engineer from Astrionics. The output from these groups reaches the rest of Astrionics through various notes and memoranda, not being collected in any single document.

The only inter-system interface is with ACE, and the Apollo Launch Vehicle Interface Panel is in charge of this interface. Minutes of panel meetings in fact describe the interface which will eventually exist between SLCC and ACE. However, the 201 interface is described only in an internal note of Astrionics which is not yet titled or published.

C. Identification of Operator and User

Activation of these programs takes place at KSC, where people from Astrionics and LVO will jointly undertake the activation.

The programs will be operated by Launch Vehicle Operations. Maintenance will be done partly by LVO people and partly by Astrionics people in accordance with an agreement between KSC and MSFC described in the KSC memo called "Master Tape Changes at the Launch Site".

The user of these programs (and of their operators) is the Launch Control Director.

Implementation of the System

A. Functional Requirements

This complex conducts a checkout and countdown of the launch vehicle including the Instrument Unit. It gives program tests and displays results.

The configuration is shown in Figure 7. The computer sends discretes to the launch vehicle through the ESE equipment and sends data to the Saturn guidance computer through a data line into that computer. It receives data back through this same line, discretes back from the ESC equipment and can scan any data from the ground digital data acquisition system. Using these facilities it must administer tests as directed by the ESC control switchers and the keyboard units associated with the display consoles and display the results (and any other information requested) on display units.

These requirements are described in the Astrionics document "Preliminary Specifications for the Saturn IB Operating System Programs".

B. System Interfaces

The specifications for the interface with ACE are generally given in minutes of meetings of the Apollo Launch Interface Panel. For mission 201 flights these are summarized in an untitled Astrionics internal memo by Ludie Richard.

There are no other inter-system interfaces.

C. Constraints

Requirements for the ACE interface are now in hand. The organization of the program is such that these did not materially hold up any part of the program.

While all other requirements are essentially specified in-house, this specification has required inputs from various other organizations. The programming organization has tended, insofar as possible, to localize these dependencies to the plugging in of parameters. Insofar as inputs have been required on things such as order of test sequence, which cannot be plugged in quite so simply, the information is on hand at this time and appears in the specifications mentioned under E.

D. Mission Dependence

The programming effort has been designed to localize the mission dependencies insofar as possible. More than half of the effort will have been put in on an operating system (although this will comprise less than 1/3 of the instructions written) which is essentially mission independent. The remainder of the effort will be devoted to test programs which are launch vehicle dependent, although most of them may span several vehicles.

These programs are all critical in the sense that the mission could not proceed, or would suffer a compromise of objectives, without them. They are (mainly) all used for the first time in mission 201.

E. Software Requirements Specifications

These have been split into two parts in this effort: requirements for the operating system, and those for the test programs. Requirements for the operating system were roughly set in the Astrionics document "Preliminary Specifications for the Saturn IB Operating System Programs". Exact specification will occur in an IBM document: "Specifications for the Operating System Programs for the Saturn IB Launch Computer Complex" which must then receive Astrionics approval. This document does exist, and has been approved in the main but sent back for some clarifications. A document describing requirements for the test programs will be submitted by IBM for Astrionics approval in January 1964.

F. Identification of End Items, Milestones and Schedules

The set of end items has been defined and schedules of milestones have been set. (See Figure 8) However, these do not appear in any official document. There is no formal milestone either of IBM turnover to Astrionics or Astrionics turnover to LVO.

G. Status and Schedules

These are summarized in Figure 8. The only critical item at this time is the scheduled delivery of the first computer and its associated utility routines in January 1965. This is necessary for the beginning of a rather tight debugging schedule. Any appreciable slippage here (and the debugging of such an operating system could prove very difficult) would slip the beginning of LVO operation in July 1965.

APPENDIX A.4

LAUNCH VEHICLE DIGITAL COMPUTER PRE-LAUNCH PROGRAMS

Identification of Project Responsibilities

A. Prime Responsibilities

Checkout of the launch vehicle Instrument Unit Guidance and Control system is assigned to Astrionics Laboratory at MSFC. This particular task is split off from the rest of the testing of the launch vehicle (done by the SLCC) in that most of these tests are done by the Launch Vehicle Digital Computer itself, rather than by the RCA-110As used in SLCC. While the same organization is responsible for planning, implementation, and testing of both efforts, the personnel directly involved in the two are different.

B. Requirements and Interfaces

Responsibility for setting requirements rests with Astrionics Laboratory although IBM (Owego) plays a large part in their actual generation. No document exists defining requirements at the general level.

General interface requirements are defined in-house in Astrionics, and set forth in the functional requirements document*. The only computer system with which this interfaces is with the SLCC, an in-house interface. No one in particular is in charge of the interface.

C. Identification of Operator and User

These programs are activated and maintained by Astrionics and will be operated by LVO at KSC. This operation is indirect, however, for they are actually operated by the SLCC without direct human intervention.

The user is the launch director, for the results of these programs ultimately contribute to the SLCC displays which indicate the condition of the launch vehicle.

---

*See Functional Requirements (next page).

Implementation of the System

A. Functional Requirements

These programs test and exercise the guidance and control system, including the LVDC itself, upon instruction from the SLCC, returning results to the SLCC. Input from the SLCC and IU and all output are through a special hardware unit. See figure 9.

In particular, "In conjunction with the launch computer, these routines shall provide the following capabilities:

1. Operational checkout of the LVDC;
2. Operational checkout of the LVDA;
3. Operational checkout of the switch selector - LVDA interface;
4. Operational checkout of the CIU-LVDA interface;
5. Operational checkout of the accelerometer and gimbal angle processing feature;
6. Operational checkout of the vehicle discretes;
7. Operational checkout of the LVDA-PCM interface."*

*Taken from "Specification for SA-201 Launch Vehicle Digital Computer Pre-launch Programs", Astrionics, R-ASTR-NDF.

B. System Interfaces

These interfaces are all handled by hardware, so that no programming specifications would be relevant, save that one routine must be able to interpret instructions received from the SLCC. Particular specifications are not given in the mission requirements document, but are expected in the forthcoming exact specification document (see E). They are generated by the LVDC programmers for their own convenience.

C. Constraints

No particular restraints exist on these programs because of information interfaces. However, the fact that they sit in memory along with the in-flight routines means that they must be tested almost as carefully as these. This means that any change after the programs are checked out would result in another long checkout period. It is unlikely, however, that any nominal change in the mission would reflect itself in any change in these programs.

D. Mission Dependence

These programs are quite independent of the mission. In general, only a small portion of them could be affected by any change in the mission which did not affect the hardware of the guidance and control system. Even this small portion would not be affected unless there were a major change in the structure of the orbital programs.

All of the programs in this effort are new, and all are critical to the success of the mission.

E. Requirements Specifications

Detailed specifications do exist in an IBM working document called "Detailed Flow & Descriptions, SA-201 LVDC Pre-launch Programs". This document should be issued as official shortly. No publication date can yet be obtained.

F. Identification of End Items, Milestones, & Schedules

End items are fully identified in the requirements specification. Milestones and schedules exist. See Figure 9.

The coding, debugging and checkout of these programs require support programs of size and complexity which dwarf those of the operational programs. These consist of the assembler and open loop simulator. These are already done, however. They are essentially mission independent and will not need to be considered for subsequent missions.

G. Status and Schedules

These are summarized in Figure 10. The effort seems to be in good shape.

This effort is characterized by the fact that the operating programs are quite small (about 5,000 instructions) but quite critical. This is reflected in a checkout period which is very long in relation to the coding period, a situation which could absorb some slippage.

APPENDIX A.5

LAUNCH VEHICLE DIGITAL COMPUTER IN-FLIGHT PROGRAMS

Identification of Project Responsibilities

A. Prime Responsibilities

The guidance and control programs in the Launch Vehicle Digital Computer (LVDC) are done under the control of Astrionics Laboratory at MSFC. This laboratory is responsible for the planning, implementation, and testing of the programs.

B. Requirements and Interfaces

Astrionics is responsible for setting the requirements, but they require inputs from the Aero-Astrodynamics and Propulsion and Vehicle Engineering Laboratories to do this in proper detail. No general requirements document exists.

There were no official documents which spelled out who would provide them with what information, or when.

This system of programs does not interface with any other software system, save that it will be initiated by the prelaunch system and must receive requests and send answers through telemetry. No interface documents exist.

C. Identification of User

Astrionics activates and maintains these programs. They are 'operated' by LVO, but this 'operation' consists only of turning them on: an irreversible process.

Implementation of the System

A. Functional Requirements

These programs guide and control the launch vehicle. They steer the vehicle and watch for a fixed velocity. When this velocity is attained, they issue a discrete which causes SIVB cutoff. A final correction in attitude is made, and a discrete is issued to the spacecraft, notifying it that it can uncage its gyros and begin its own timing processes. While in orbit, the system must also answer any requests from the digital command system. All input-output is through a special hardware unit. See Figure 9.

A mission definition document was completed only in October 1964. This was done jointly by IBM and Astrionics, but Astrionics bears the responsibility. It is an informal collection of papers called the "Mission Definition Document."

B. System Interfaces

These programs have none. The only interface is with the instrument unit and telemetry, and this is by a special hardware unit. No document exists on any software ramifications.

C. Constraints

There are no constraints caused by system interfaces. However, there is an extreme timing constraint on information input: the extreme criticality of these programs requires that they undergo a checkout period of approximately six months. Even slight changes (e.g., the value of a constant) might make it necessary to go through an appreciable part of this again.

D. Mission Dependence

These programs are written specifically for mission 201. However, it is likely that a large part of them may be adapted for other missions.

E. Requirements Specifications

Astrionics expects to complete "SA-201 Equation Defining Document" about December 15, 1964. This is to be a complete requirements specification, despite its title.

F. Identification of End Items, Milestones, and Schedules

These are all done. See Figure 11. They are not collected in any official document.

The coding, debugging and checkout of these programs require support programs of size and complexity which dwarf those of the operational programs. These consist of an assembler and both open and closed loop simulators. The first two of these are done, and there is no reason to doubt the schedule on the third. These support programs are essentially mission independent, and are very similar to programs which IBM has done in the past. They will be usable for missions subsequent to 201.

G. Status and Schedules

These are shown in Figure 11. They appear to be in good shape. This effort is characterized by an extremely long debug time, as compared to coding. The criticality of the programs demands that they be tested during all the time available for it, but this actually allows a great deal of flex in schedule, which they apparently will not need.

APPENDIX A.6

COMPUTATION AND DATA FLOW INTEGRATED SUBSYSTEM (CADFISS)

Identification of Project Responsibilities

A. Prime Responsibility

Goddard Space Flight Center is responsible for the testing of the tracking network equipment to determine its status and operational readiness to support a mission. This responsibility is delegated in the "Management Plan for the Manned Space Flight Network" of February 5, 1963, forwarded by Dr. Seamans to Dr. Goett on March 11, 1963. The Responsibility for the implementation of the computer programs required lies with the Manned Flight Operations Division, specifically under the direction of the CADFISS Test Director, Mr. W. I. Adams.

B. Requirements and Interfaces

1. Requirements

It is the responsibility of the NASA Centers to generate appropriate requirements documents for GSFC to perform its function of implementing the instrumentation for the Manned Space Flight Network. These documents also provide the broad requirements for CADFISS testing of the network. CADFISS requirements are generated from a variety of sources in addition to the NASA Center requirements documentation. The general procedure for generation of requirements and GSFC response thereto is discussed below followed by specific remarks pertinent to the computer programs for CADFISS.

MSC generates an Apollo/Saturn Program Instrumentation Requirements Document (PIRD). PIRD defines the instrumentation requirements for the Ground Operational Support System (GOSS) for Project Apollo. It includes all of the Engineering requirements on the Manned Space Flight Network which is a subset of GOSS. The requirements in a PIRD are oriented towards an entire vehicle program (such as the Saturn IB program, for example).

PIRD must be approved by the Associate Director of MSC and forwarded to OMSF for validation and finally to GSFC via OTDA for action. A PIRD for Saturn IB (dated April 10, 1964) has been issued and approved by MSC. This document was validated by OMSF and forwarded to OTDA in September 1964. MSC will also furnish an Apollo/Saturn Data Acquisition Plan (DAP). DAP defines the requirements for real-time data for a specific flight and is distributed to all elements of GOSS. A DAP is issued approximately one to two months prior to the scheduled mission launch date, hence none is available for flight 201 at the present time.

MSFC generates an Engineering Instrumentation Requirements Document (EIRD). EIRD defines the instrumentation requirements for engineering evaluation of launch vehicle R&D flights. An EIRD is written for a launch vehicle program or for a segment of a program if it has a unique set of requirements (for example, Saturn I/Block II).

EIRD must be approved by the Director, Saturn Systems Office of MSFC and forwarded to OMSF for validation before being forwarded to GSFC via OTDA for action. An initial issue of the Saturn IB EIRD has been published and forwarded to OMSF (July 1964). It was forwarded officially to OTDA on September 1, 1964. MSFC also furnishes a Data Acquisition Requirements Document (DARD). A DARD is issued for each individual flight after the detailed flight mission has been established. GSFC requirements are included in a separate section. The document includes detailed flight specific instrumentation requirements not covered in the EIRD. A DARD is issued approximately two months prior to the scheduled mission launch date, therefore none is available for flight 201 at the present time.

GSFC responds to the general requirements (PIRD and EIRD) with a Data Systems Development Plan which is forwarded to OTDA. It is more than a specific reply to the PIRD and EIRD and contains much unrelated information (fiscal, for example). OTDA has the responsibility for generating a Ground Instrumentation Support Plan (GISP) to meet the requirements of the NASA Centers. These documents have not yet been issued for Saturn IB.

GSFC requirements for CADFISS computer programming are not called out specifically in the MSC-MSFC requirements documents since these are implementation requirements and are directed more to the efforts of the Network Engineering branch and Network Operations branch at GSFC. It is the responsibility of the CADFISS Test Director to derive computer program requirements from all of the above documents. Computer programming requirements also originate within GSFC. For example, requirements may be originated by the Network Engineering branch or Network Operations branch. Since CADFISS is equipment oriented, it is often necessary to obtain considerable information from contractors from which programming requirements are derived.

In all cases, requirements must be coordinated by the CADFISS Test Director (W. I. Adams). These requirements are documented in a CADFISS Engineering Request (CER). A CER is written for each new test to be implemented or when tests are to be modified.

Shortly before launch date for a mission, usually two-three weeks, a Network Operations Plan is published by the Network Operations group at GSFC. This document specifies the equipment to be used for support of the mission. At this time a mission specific CADFISS test program can be assembled by incorporating appropriate tests for the equipment specified.

2. <u>Interfaces</u>

The interface for PIRD between GSFC and MSC is administered by the Flight Control Division of MSC. The Division Chief is J. D. Hodge. All corrections and/or changes to the requirements document must be coordinated by this division.

The interface for EIRD between GSFC and MSFC is administered by R-AERO-FP, MSFC. Changes and/or corrections to this document are coordinated by G. R. Emanuel and D. R. Haynes of R-AERO-FP, MSFC.

a. Identification of the User

The ultimate user of the output of the CADFISS test procedure is the Network Controller at IMCC.

Implementation of the System

A. Functional Requirements

The CADFISS system provides a measure of how well the remote site subsystems are functioning and a determination of whether the sites are properly integrated to the central computer. Since the tests are run at GSFC, it is necessary that IMCC check the integrity of the data flow to the IMCC computers prior to launch. It is necessary for the CADFISS tests to be conducted as close to launch as possible because of the tendency for electronic equipment to drift off calibration after being peaked for the support of a mission.

The system must provide the capability for testing individual items of equipment or group of equipment forming subsystems at the site for diagnostic testing when required. It is therefore necessary for the computer to identify the test to be made. A specific time is allowed for each test to be completed. Some of the functions that must be performed are 1) verification of the data flow paths between the sites and the central computer, 2) verification of radar orientation, alignment and calibration, 3) verification of acquisition aid orientation, alignment and calibration, 4) tests of telemetry subsystems, 5) tests of impact predictor, ground guidance, and remote site computers.

B. System Description

The GSFC real-time computer complex within which the CADFISS programs are designed to operate is shown in Figure 12. The complex consists of a triplex arrangement of IBM 7094 computers each of which has associated with it the equipment indicated in the figure and a data communications channel to provide the required timing and data subchannels. The tests will be run on two of the computers. One of the computers will be "on-line" and powered by commercial power, while the other will function as a back-up and will be powered by a diesel generating system. The system is made up of individual tests under the control of a master program (CADMON).

The computer sends a cue message via teletype (see Figure 13) to each of the sites participating in the test. This cue message functions as a "start-of-test" indicator and also serves to inform the site personnel of the test being executed. The site responds by transmitting the required data to the computer at GSFC via teletype. The computer compares the data received with expected values prestored in memory.

In order to support a specific mission, it is necessary to know the equipment to be used and then to select the appropriate tests and assemble them on a magnetic tape. The equipment required to support the mission is specified in a Network Operations Plan generated by GSFC usually two-three weeks before the mission. The assembly process requires a minimal time for preparation, one day being sufficient if all required test programs are available. There apparently is no testing procedure for the final assembled program since all of the individual test modules of which it is composed have previously been checked. Thus, the final assembled program as an entity may be placed in operation untested.

The total number of words written for the CADFISS Test system is approximately 65,000. The actual program required to support a mission will be a subset of this program and is dependent on the equipment to be used on site for that mission.

Test selection, test sequence, and test parameters are specified by the CADFISS Test Director. These are inserted into the input generator program (CADGEN) in the form of punched cards. CADGEN converts, expands, and organizes these test requirements to the form required by the monitor and analysis program (CADMON) and prepares a test module magnetic tape and a listing of the tests with all pertinent information. The CADMON program is loaded into core by CADLOD and then CADMON starts the test procedure using the test module magnetic tape for selection and sequencing information. The CADOUT program generates periodic on-line printed reports which are used by the CADFISS test director to monitor test progress.

C. System Interfaces

1. Equipment

GSFC has complete control and authority over the equipment interfaces. Equipment interfaces include on-site equipment such as C-Band and S-Band radar, DCS, PCM, and the data processors, and in addition the launch monitor subsystem.

The individual tests which make up CADFISS are equipment oriented and are written to satisfy the interface requirements.

2. GSFC-MSC Test Interface

GSFC is responsible for the CADFISS tests to determine the operational status of all network supporting systems. This includes equipment located at the remote sites and at the launch area. When the tests are completed and operational integrity assured, the network control is then released to IMCC. A data flow test is then run from the IMCC computer complex to check the integrity of the data flow from the launch and remote sites to the operational computers at the IMCC. The Network Controller at IMCC can release individual malfunctioning stations to the CADFISS test director at GSFC as required. If this should happen, the CADFISS test director will perform the required tests and when again operational return the station to the Network Controller who will then direct that a data flow test from IMCC to that station be made.

D. Constraints

The entire CADFISS programming effort from the planning to implementation is a GSFC responsibility. The only constraints therefore, are the need to have a detailed knowledge of equipment to be used in a mission in sufficient time before the mission to enable writing test programs. For flight 201 since it is expected that no new on-site equipment will be introduced all required test programs are assumed to be complete.

E. Software Requirements

Software requirements documentation consists of CADFISS Engineering Requests (CER). A CER is written for each new test to be implemented or when tests are to be modified.

F. End Items and Schedules

The end items for a specific mission consist of a Test Module Magnetic Tape and a listing of the tests with all pertinent information. These are in addition to the operational programs which are not mission oriented.

These items are not specifically scheduled. They are prepared as soon as the necessary information is available. This information is provided by the Network Operations Plan which is published two-three weeks before launch.

G. Current Status of Project

The CADFISS test system is now operational. In order to support a particular mission, it is necessary to know what equipment will be used and to assemble a flight specific CADFISS test program to test that equipment, as described above. There has at this time been no consideration of flight 201 per se, but in a discussion (via telephone) of this flight and the sites involved with the CADFISS test director (W. I. Adams) it appears that all required test programs have been written. A minimal amount of time is required to assemble the CADFISS program. Mr. Adams indicated that this can be done in as little as a few days before launch.

APPENDIX A.7

NETWORK ON-SITE COMPUTER PROGRAMMING

Identification of Project Responsibilities

A. Prime Responsibility

Goddard Space Flight Center is responsible for all Network On-Site Computer Programming tasks.

The responsibility for the implementation of the computer programs required lies with the Data Systems Division under the direction of Mr. S. S. Lechter at the present time. However, the responsibility is being transferred to the Manned Flight Operations Division under the direction of Mr. W. H. Wood. The transfer is being effected at the present time. The Manned Flight Operations Division will be responsible for implementing the programs for missions subsequent to GT 3.

B. Requirements and Interfaces

1. Requirements

The general framework for requirements generation discussed above Appendix A.6 applies to the requirements for on-site computer programming. Specific requirements for this task are furnished in an Appendix to the PIRD. The PIRD of April 1964 indicated that the Appendix for Remote Site Computer requirements will be provided in a later revision. Mr. D. Owen of the Flight Control Division, MSC has informally indicated that this Appendix will be included in the January 1965 revision of the PIRD.

There is extensive communication on a relatively informal basis with Flight Control Division, MSC for corrections, changes or additional information as required. Requirements obtained in this manner are formalized by memorandum from the Flight Control Division, MSC and forwarded to GSFC via the IMCC program office, Mr. H. E. Clements, Manager. About three weeks to two months before the launch date MSC forwards a Data Acquisition Plan (DAP) to GSFC via OTDA. The DAP includes specific data as to parameters to be displayed on the various meters and event lights at the computer associated consoles. This enables the generation of a specific paper tape for each site having a computer.

2. Interface

Requirements interface between GSFC and MSC is administered by the Flight Control Division, MSC. The Division Chief is Mr. J. D. Hodge. All corrections and/or changes to the requirements document must be coordinated by this division.

C. Identification of the User

The ultimate user of the computer output is the mission Flight Director.

Implementation of the System

A. Functional Requirements

The system functions to receive telemetry data and to select, convert, format, and transmit this data in near real-time in teletype form. This transmission must be compatible with the existing communications circuits at the site which may be two kilobit/second data circuits or teletype (60 and 100 wpm). The system must automatically generate and transmit messages upon request from the flight control personnel. These messages consist of data from the PCM ground station. The messages shall include the necessary directing codes, message identification, station identification, a time tag, and the encoded data. These messages are transmitted to the RTCC at the IMCC.

The system must be capable of driving displays at associated consoles. These displays consist of meters and event lights.

B. System Interfaces

The computer has the following equipment interfaces:

1. PCM ground equipment

2. Teletype transmission equipment

3. Four control consoles.

Additional interfaces may be identified in the next PIRD revision, January 1965.

C. Constraints

The programs for the on-site computers required for support of Gemini are operational. These programs completely fill the 16K core memory of the UN1218. Therefore, any additional requirements for the support of flight 201 will require a larger core memory and complete program rewrite.

D. Software-Requirements Specifications

There are no official software requirements specifications generated. The programming group is small [about eight people] and specifications are generated on a very informal "in-house" basis.

E. Identification of End Items and Schedules

A paper tape is generated for each remote site having a computer. The machine is loaded by a paper tape reader under control of an Input-Output console.

This tape must be generated and available on-site a minimum of about two weeks prior to launch.

F. Current Status of Project

There has been no computer programming effort directed to the Saturn IB program up to the present time.

APPENDIX A.8

CENTRAL INSTRUMENTATION FACILITY - REAL TIME DATA DISPLAY SYSTEM

Identification of Project Responsibilities

A. Prime Responsibility

The KSC Assistant Director for Instrumentation, K. Sendler, has primary responsibility for the Real Time Display System as part of his overall responsibility to provide general support for launch operations.

The specific responsibility for the implementation of the computer related items lies with the Data Acquisition and Systems Analysis Division, R. H. Bruns, Chief.

B. Requirements and Interfaces

The C.I.F. is a facility of many and diverse functions. Within the scope of computer functions the C.I.F. provides computational support for all NASA Centers and NASA contractors while at KSC, and a real-time data reduction and display capability. In these areas the C.I.F. apparently proceeds in a loose manner from requirements generated as a result of inter-center agreements. There are no official requirements documents generated. In general any requirements for KSC must be submitted to and approved by the project office headed by Col. R. A. Petrone, Assistant Director for Program Management.

C. Identification of the User

Launch Vehicle Test Conductor

Implementation of the System

A. Functional Requirements

The Real-Time Data, Reduction and Display System Figure 14 formats and displays computer processed telemetry information, radar information, weather information, and TV information from onboard and ground cameras.

The GE 635 computer, which functions as part of this system, must monitor up to 3000 telemetry measurements which it receives from the launch vehicle and spacecraft via the Data core system. The computer then formats and stores these data on the system magnetic tapes, and stores the data for the last 1000 seconds on 3 magnetic drums which are the system mass storage files.

Simultaneously with the data input and storage, the computer must accept requests for display, retrieve the desired data, convert data to appropriate engineering units, format data for display, and output the data to the buffer distributor in the data display system.

The buffer/distributor channels the data to the proper display. The data is then converted by a data formator and converter circuit into a video format. The converter refreshes the data at a 30 frame per second rate for display on monitors located in the launch control center.

B. System Interfaces

The major interface is with the telemetry systems of the vehicle. It is the function of the Datacore system to satisfy the interface requirements by accepting the telemetry and converting it to a form acceptable to the computer system. The word must contain the identity of the measurement and its magnitude. The maximum rate of transmission from Datacore is 250,000 words/second.

The interface requirements between the computer and the Data Display System are satisfied by the buffer circuit and the data formator and converter circuit.

C. Constraints

The system is constrained by the 3000 measurement capability of the Datacore system which transmits data to the computer. The computer must be able to format, block and store 1000 seconds of data received at an average rate of 432,000 bits/second. Information must be stored with the data to enable maintenance of an accurate time correlation. This data must be displayed on request.

D. Software Requirements Specification

The Request For Quote of April 1964 entitled "Computer Complex for the C.I.F., MILA, Cape Kennedy, Florida", serves as the software requirements document. There appears to be no other software requirements documentation.

E. End Items and Schedules

The end items for the real-time programs are appropriate documentation and an acceptable operating program. The end items apparently are not delineated in a more specific manner. There are no formal acceptance plans.

The contractual delivery date for the computer and the required programs is March 15, 1965. Mr. Griffin of the Data Acquisition and Analysis Division indicated that the real-time telemetry monitoring system of the C.I.F. will be operational in June 1965. He does not have any requirements as yet for flight 201 per se but pointed out that this system is not mission oriented and the C.I.F. will be ready to support flight 201 as long as the C.I.F. measurement capability is not exceeded.

F. Current Status

The effort of GE is closely monitored by cognizant C.I.F. personnel through informal monthly meetings. It is their opinion that the GE effort is proceeding satisfactorily and there is no indication at present that they will not meet their contractually scheduled delivery date. General Electric has a well defined but informal implementation plan, and coding has begun.

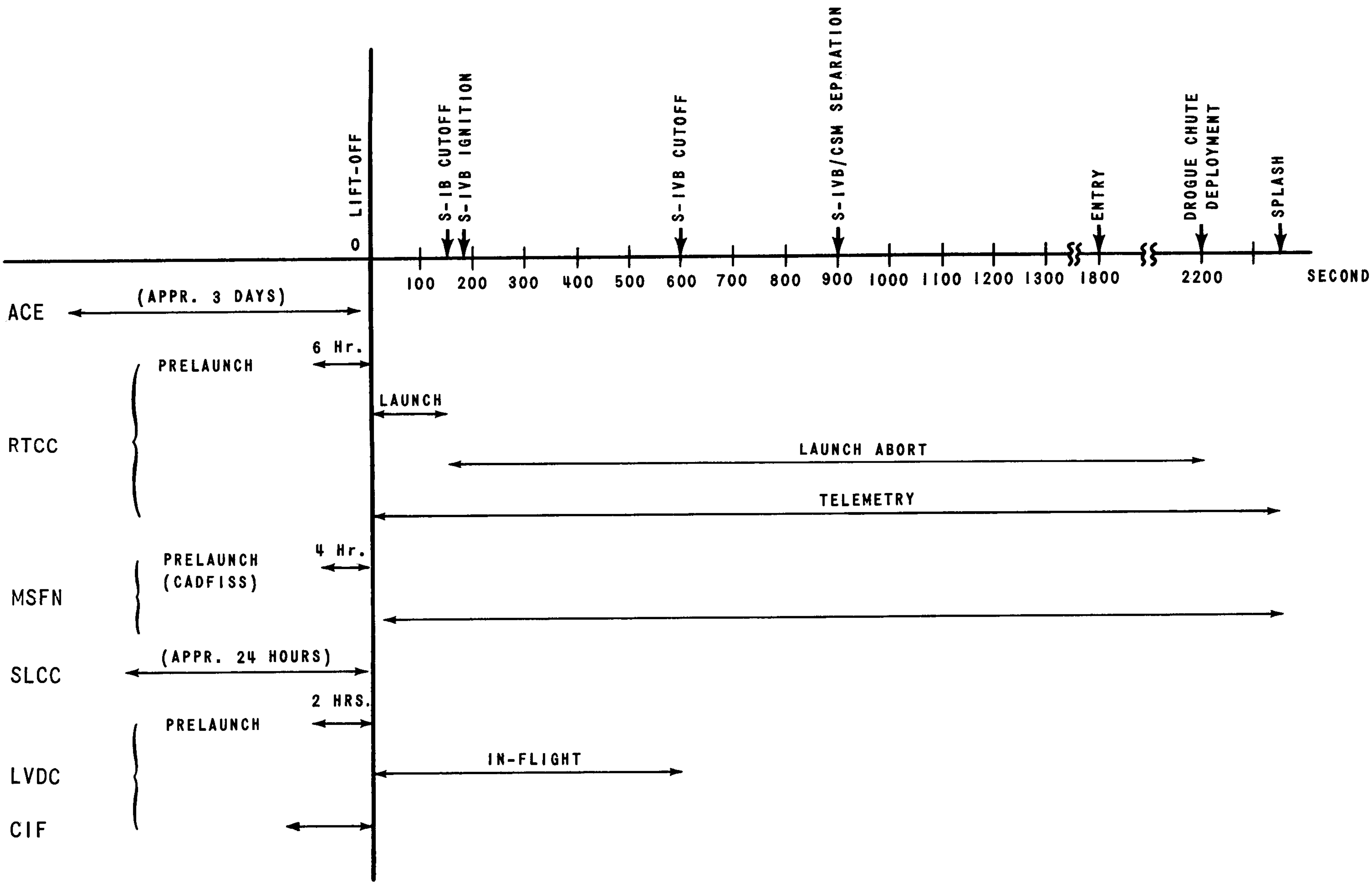


FIGURE 1 - COMPUTER PROGRAM PROFILE (PRELIMINARY) FOR SA-201

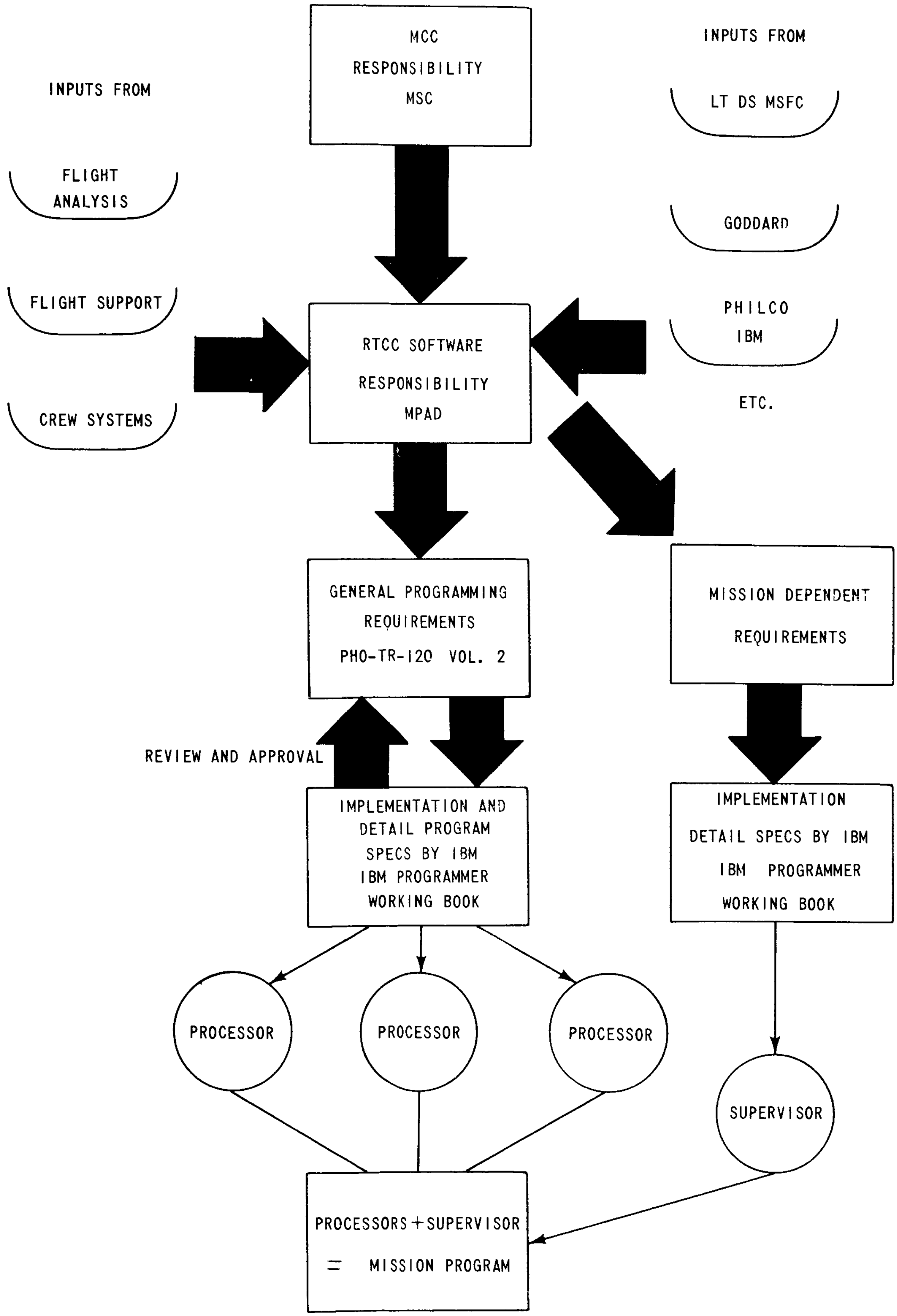


FIGURE 2 FLOW OF SOFTWARE ACTIVITY REQUIREMENTS TO END ITEMS FOR RTCC

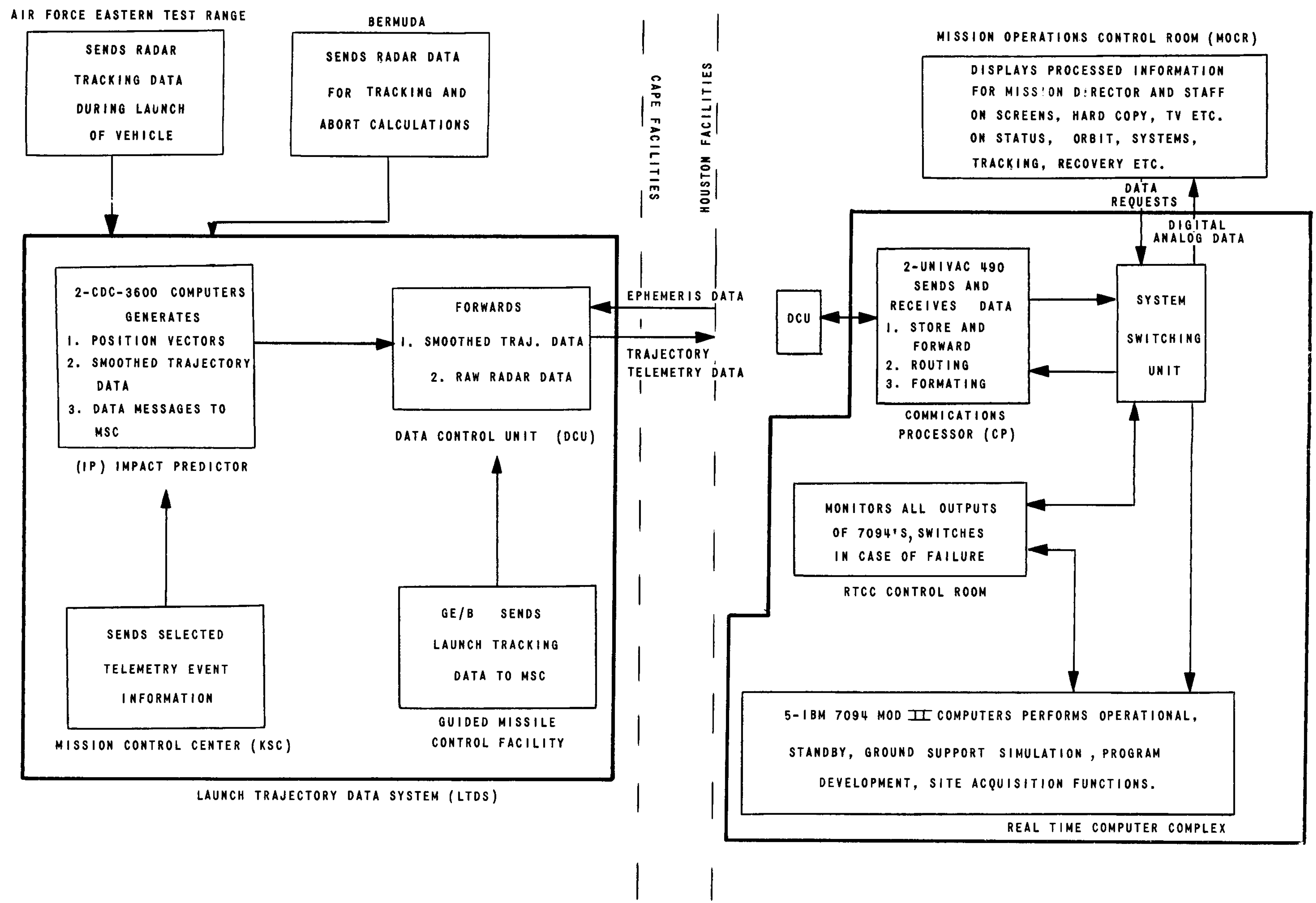


FIGURE 3

FUNCTIONAL DIAGRAM OF RTCC DATA FLOW

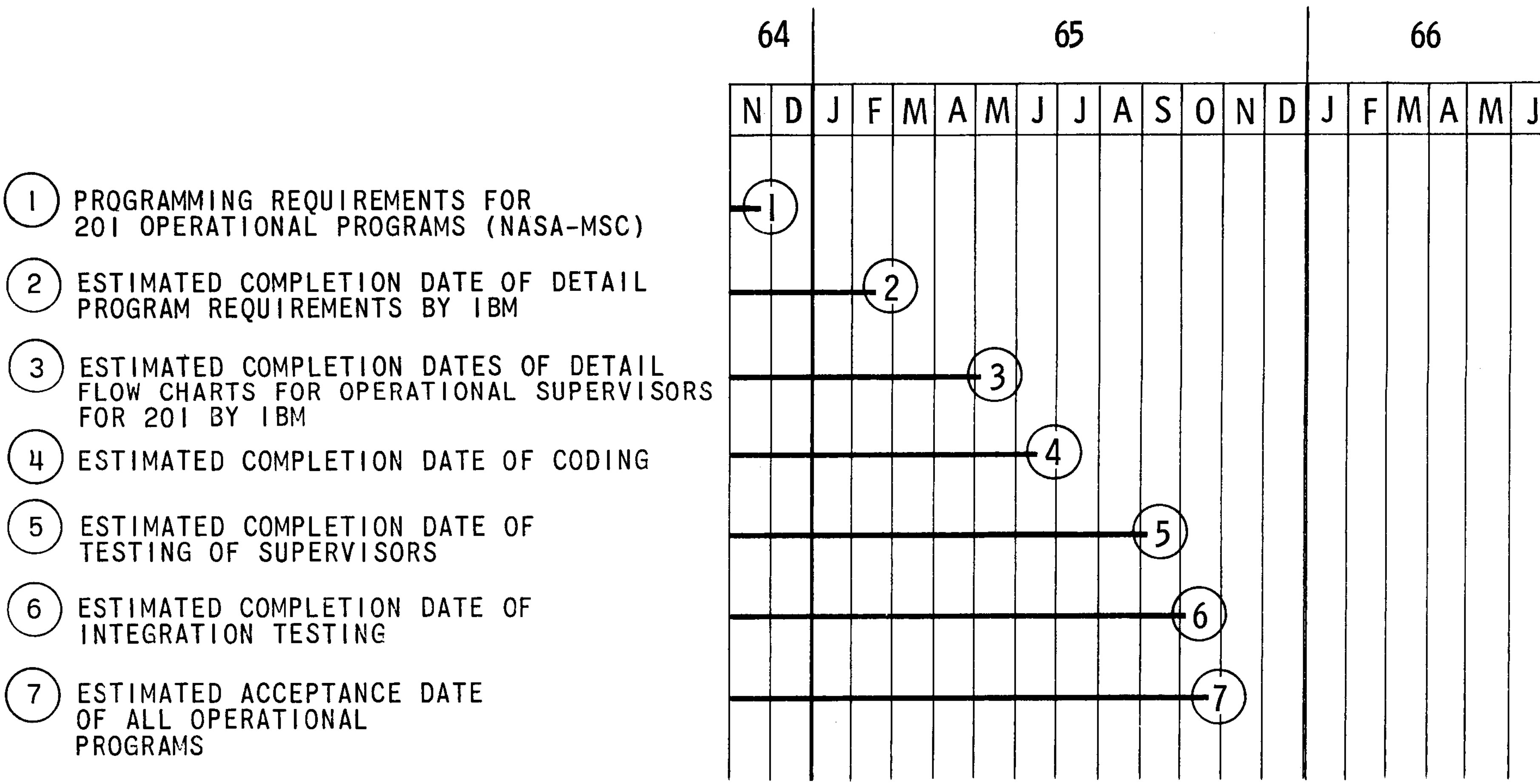


FIGURE 4 ESTIMATED COMPLETION DATES FOR RTCC MISSION OPERATIONAL SUPERVISORS

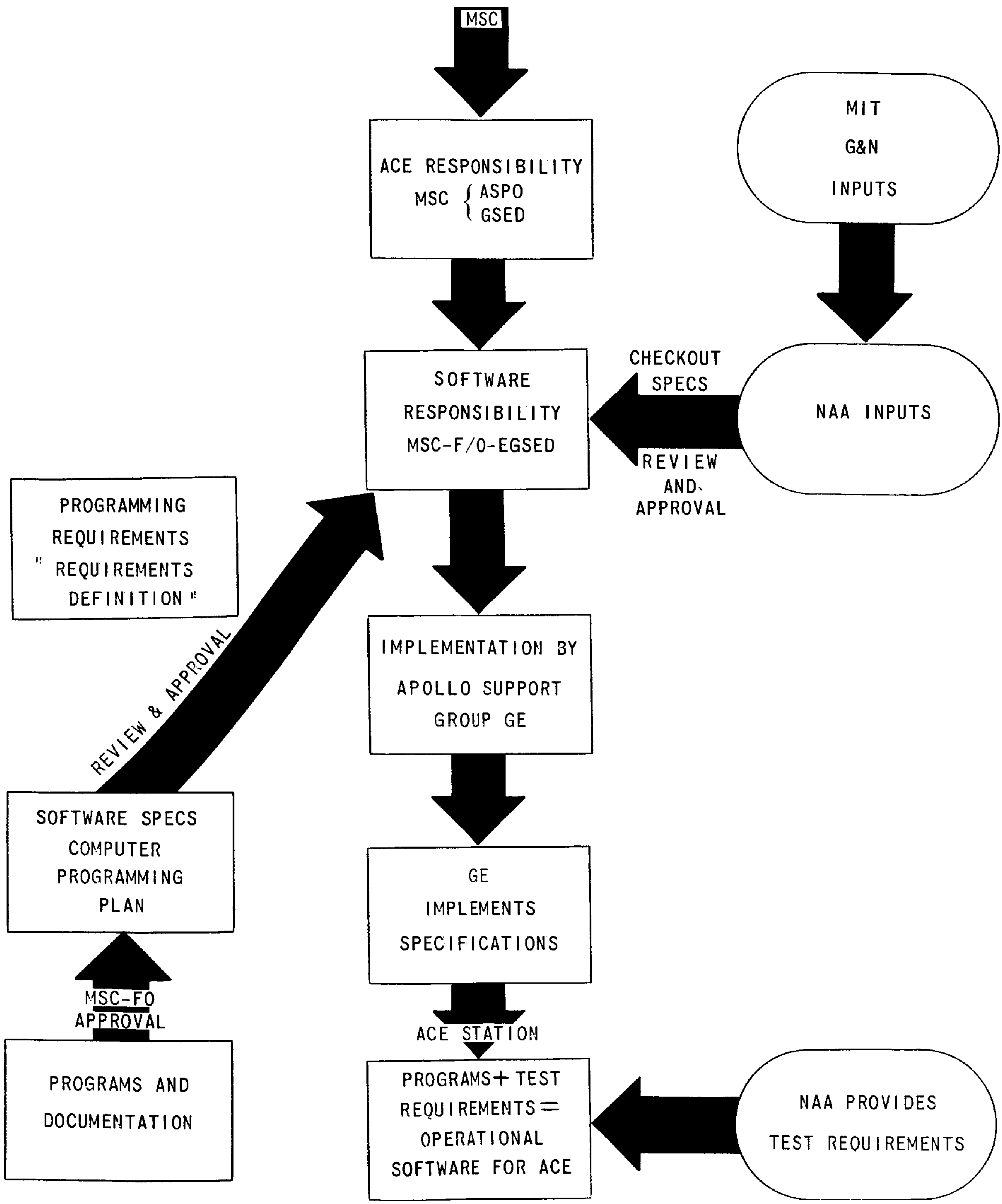


FIGURE 5 FLOW OF SOFTWARE ACTIVITY-REQUIREMENTS TO END ITEMS FOR ACE

DADE--Digital Acquisition Decommutation Equipment
CUE--Communications Unit Execute
SGS--Symbol Generator System
DTVC--Digital Transmission and Verification Converter
DCCU-- Digital Communication and Control Unit

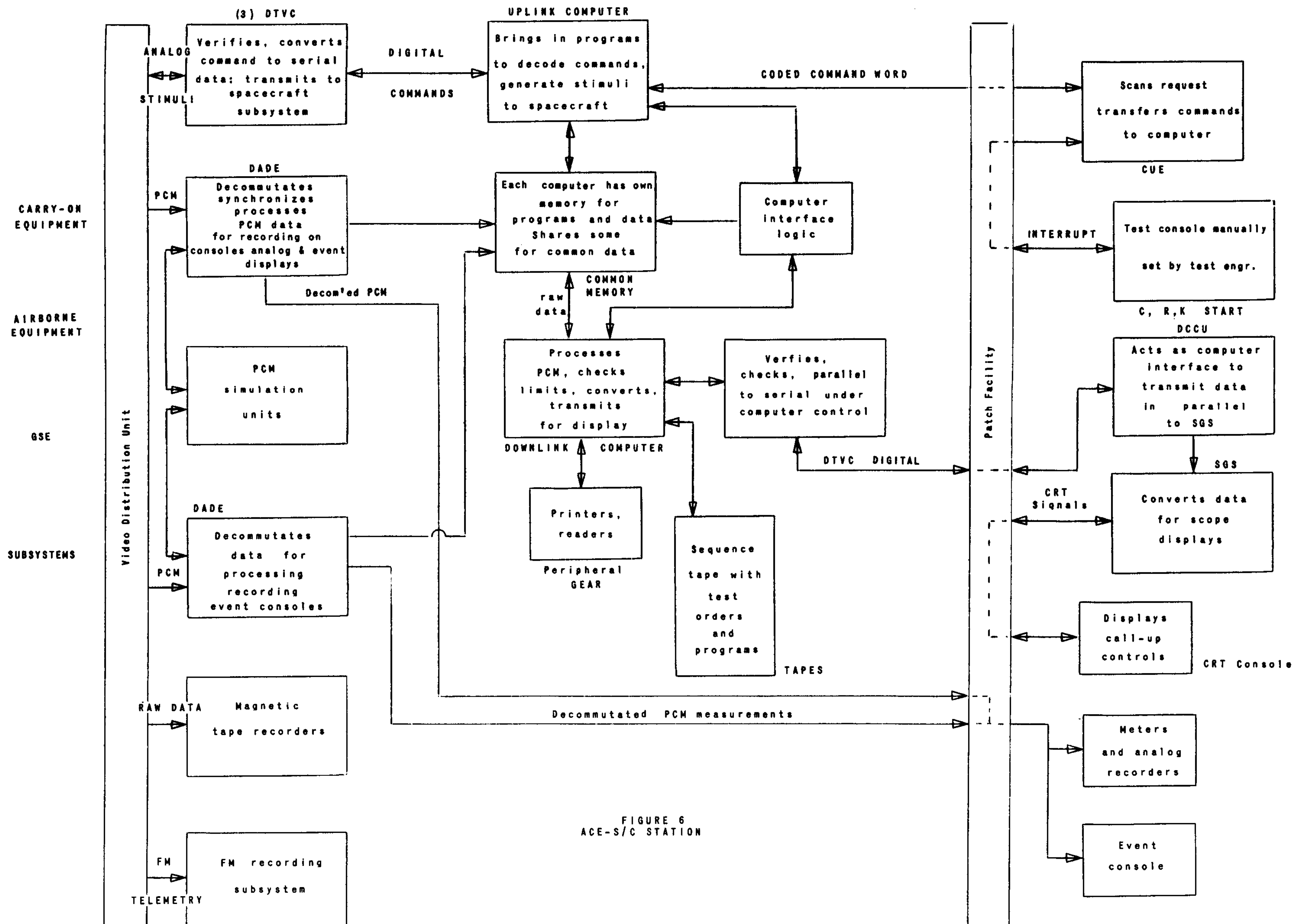
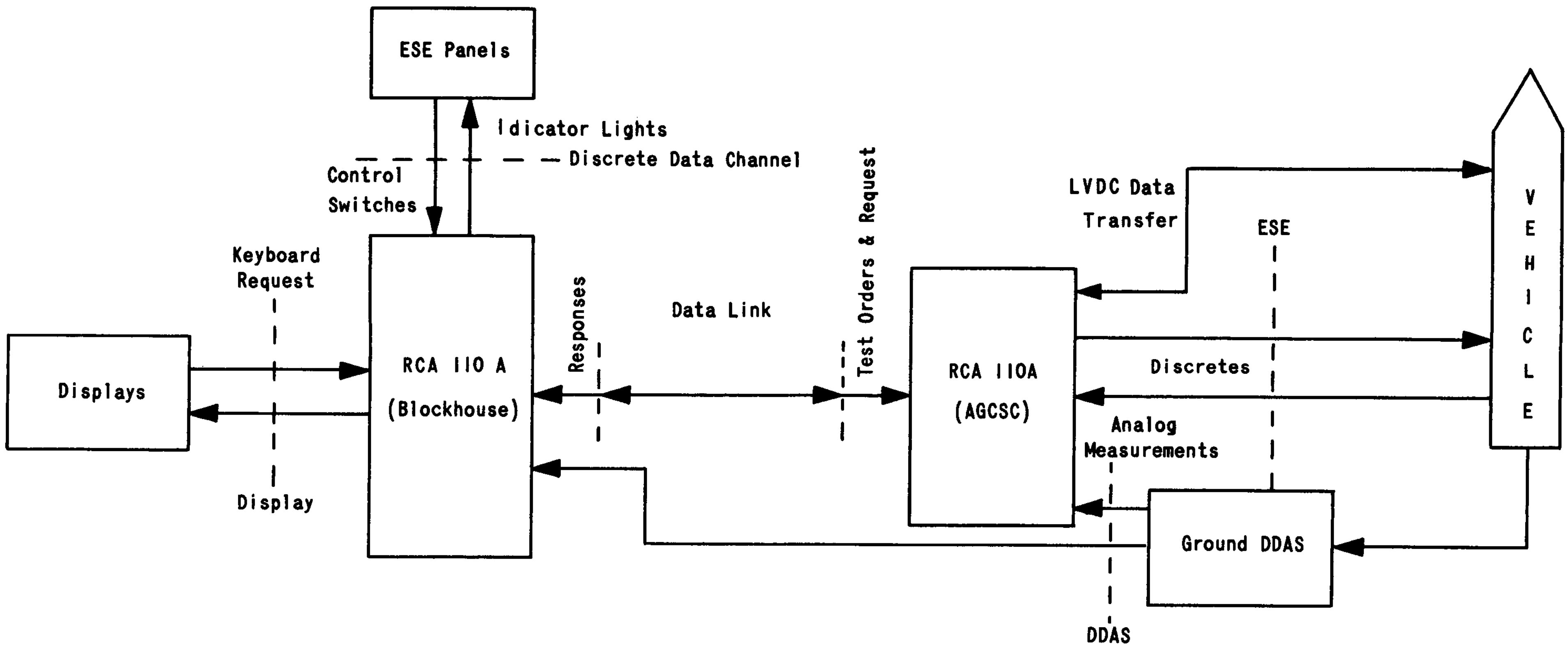
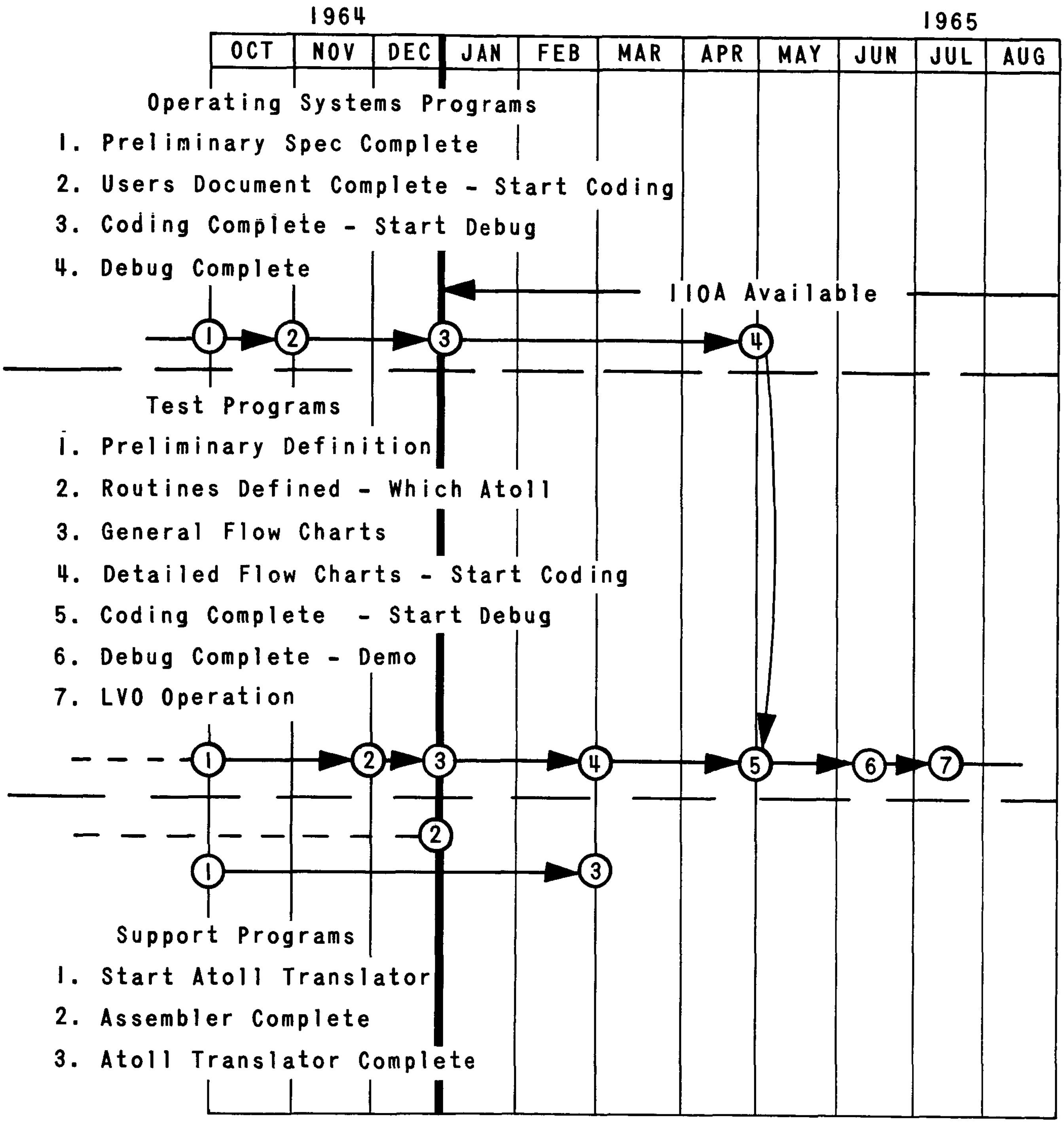


FIGURE 6
ACE-S/C STATION



Either Computer Can Request Any Data in Stream Includes PCM FM/FM

FIGURE 7 CONFIGURATION AND DATA FLOW SATURN COMPUTER COMPLEX



SCHEDULE SA- 201 *

FIGURE 8 SATURN LAUNCH COMPUTER COMPLEX

* This Chart By Astrionics Laboratory
(As of October 1964)

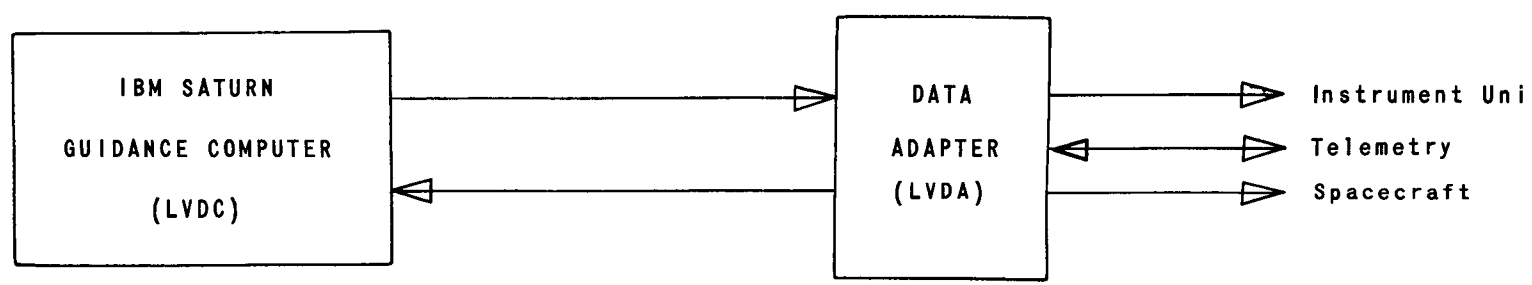
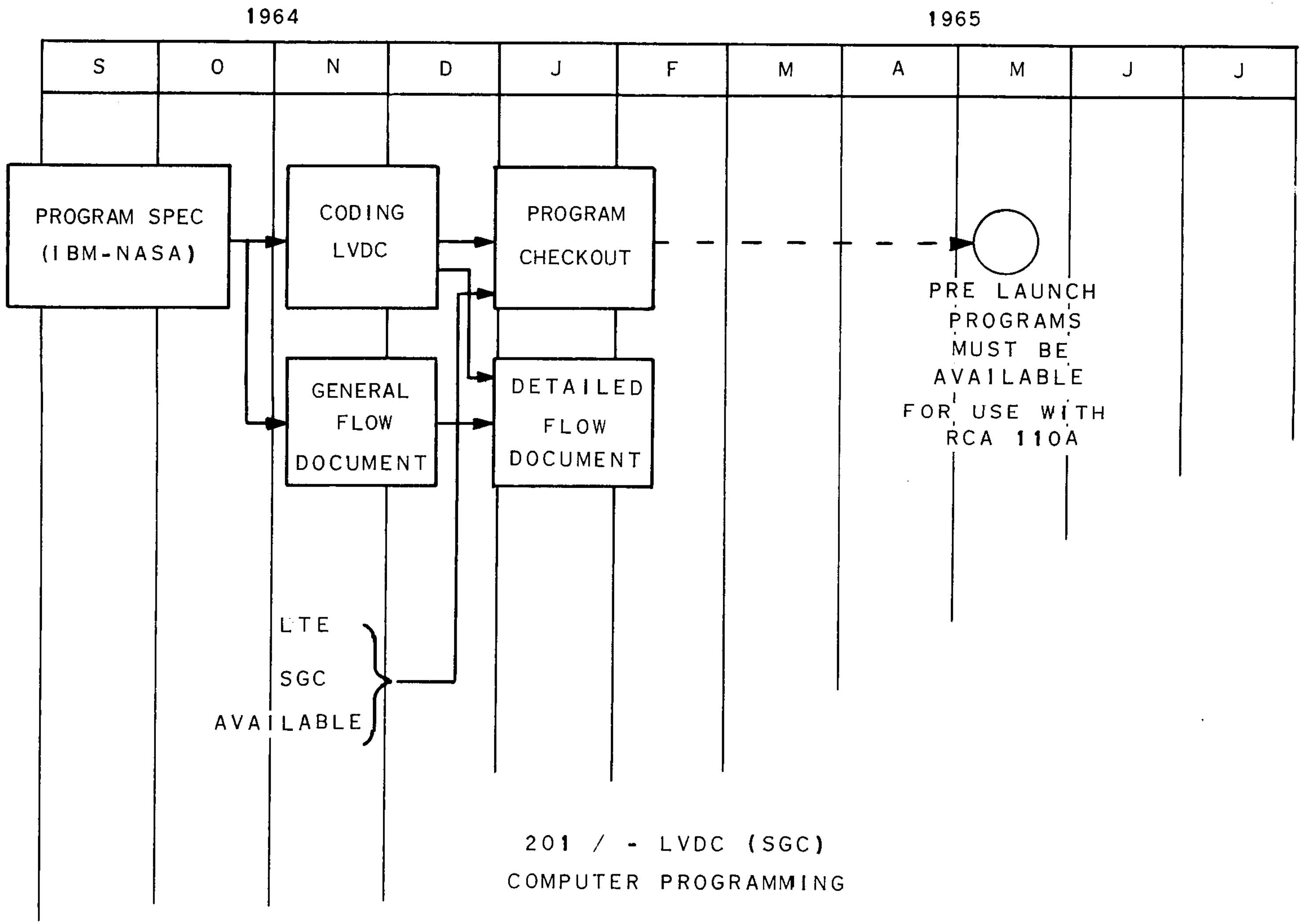


FIGURE 9 HARDWARE AND DATA FLOW SATURN GUIDANCE COMPUTER



201 / - LVDC (SGC)
COMPUTER PROGRAMMING

FIGURE 10 PRE-LAUNCH PROGRAMS

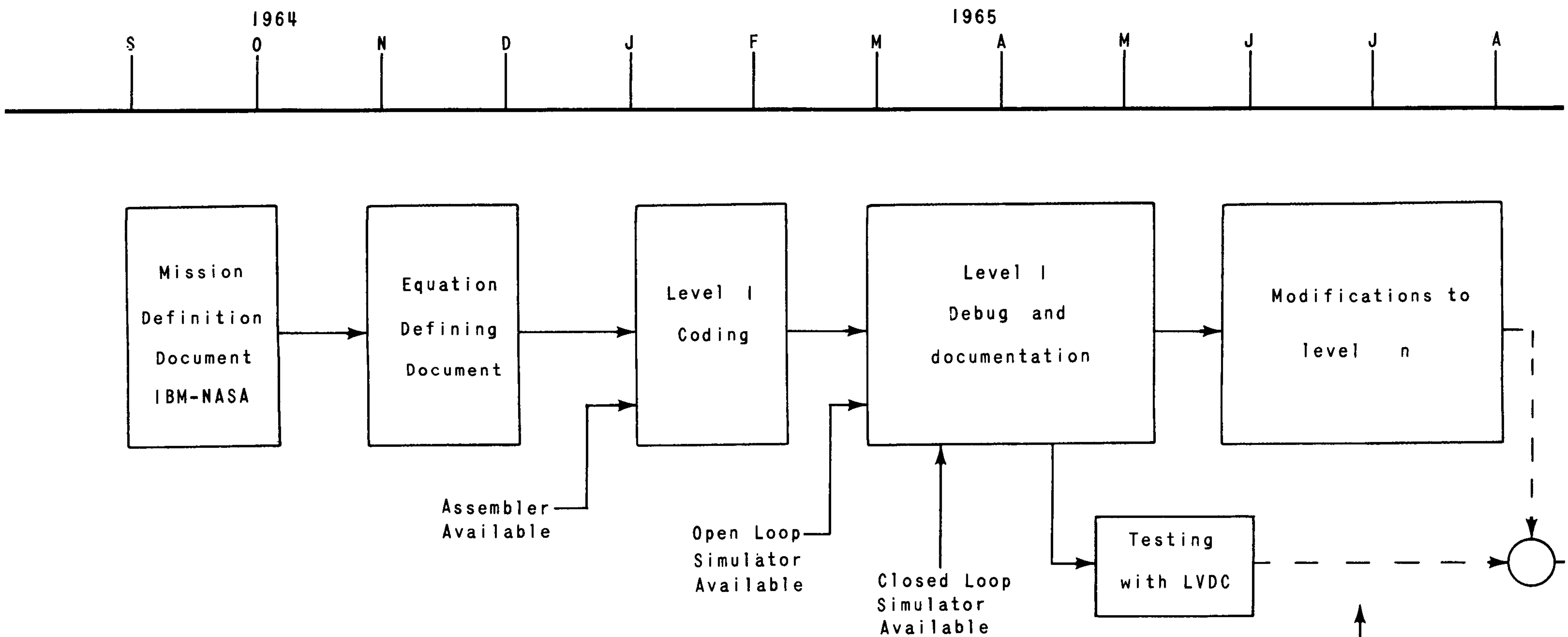


FIGURE II
201 - LVDC (SGC)
COMPUTER PROGRAMMING

Flight Programs

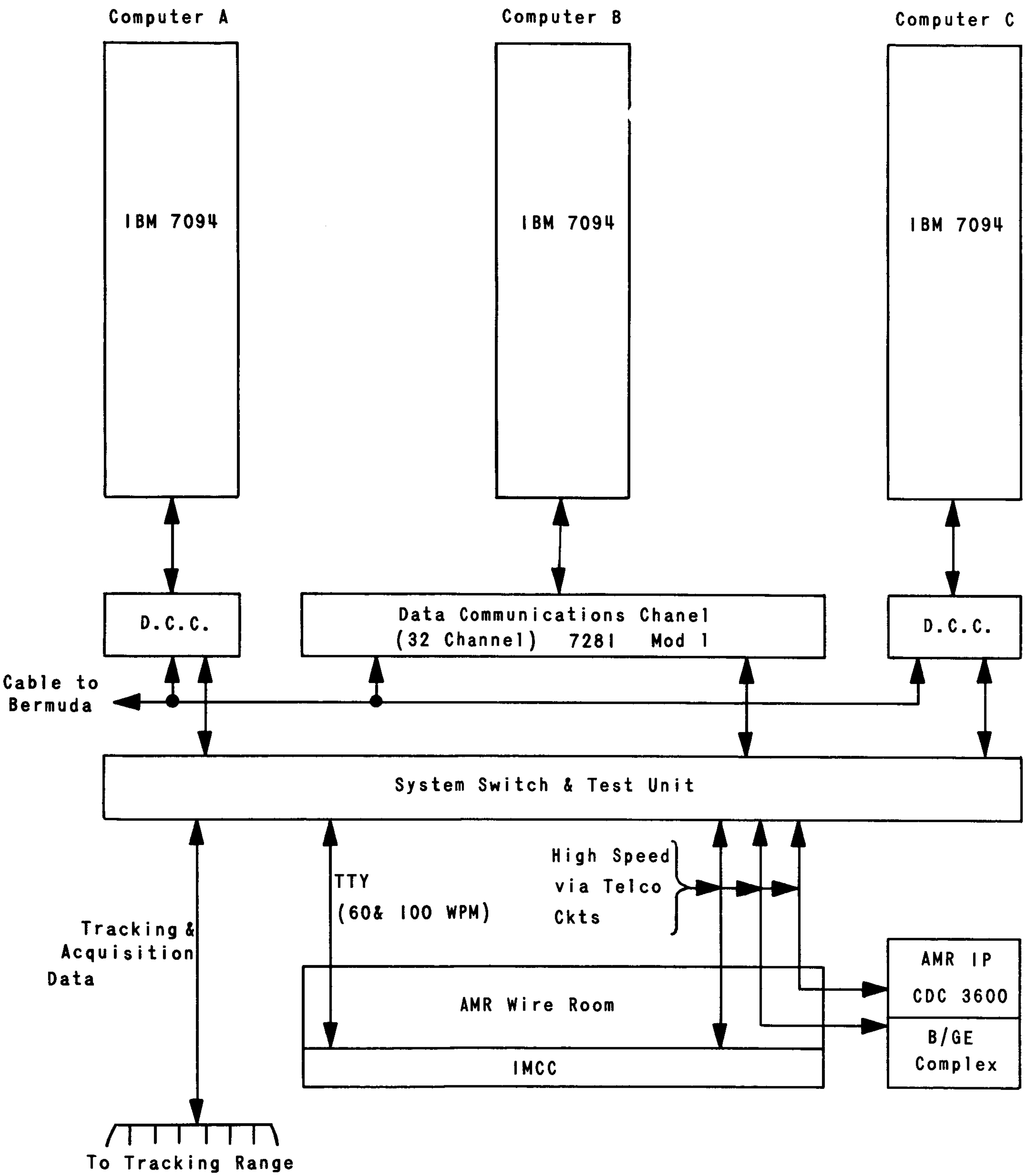
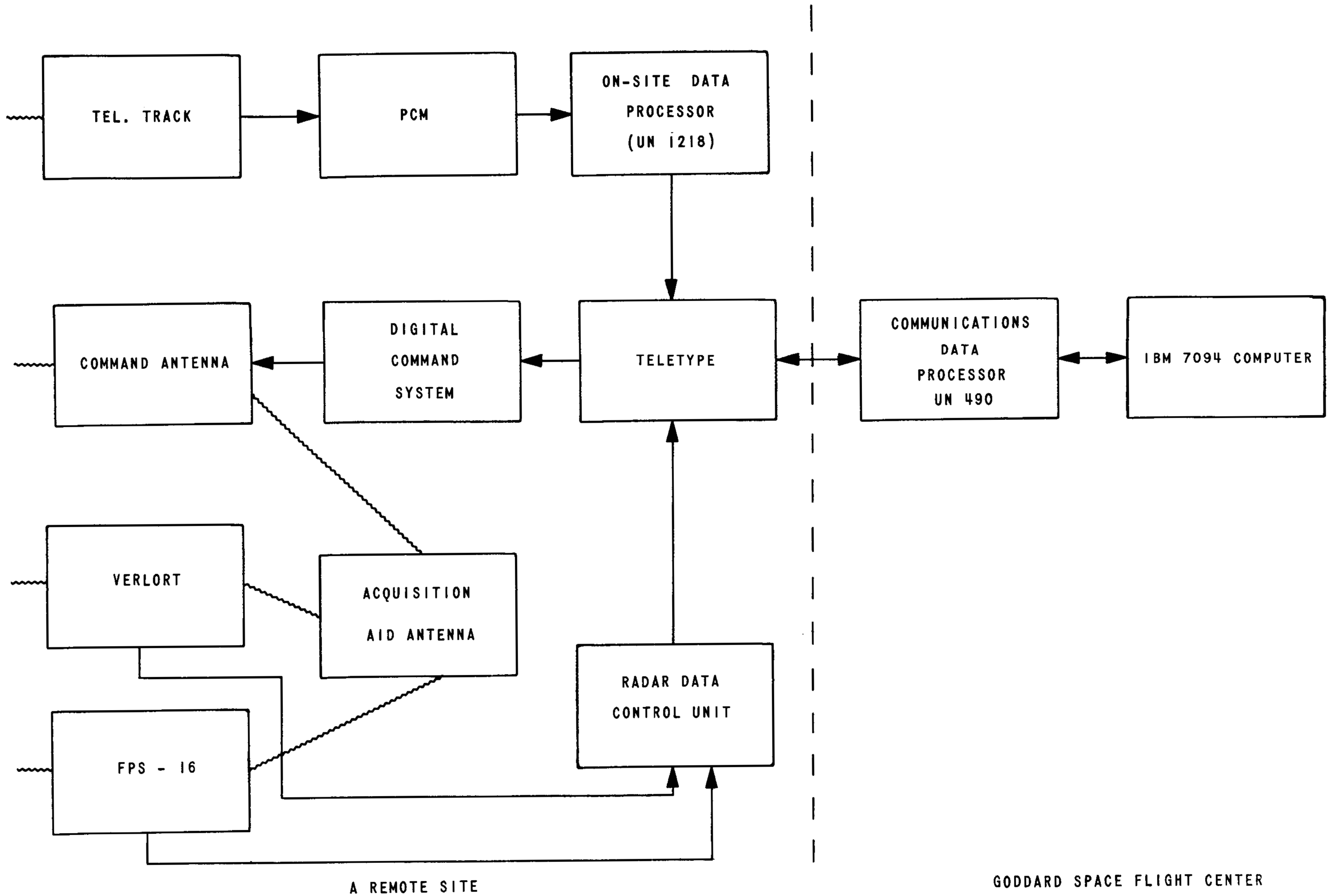


FIGURE 12 GOODARD SPACE FLIGHT CENTER COMPUTER COMPLEX

FIGURE 13 CADFISS DATA FLOW



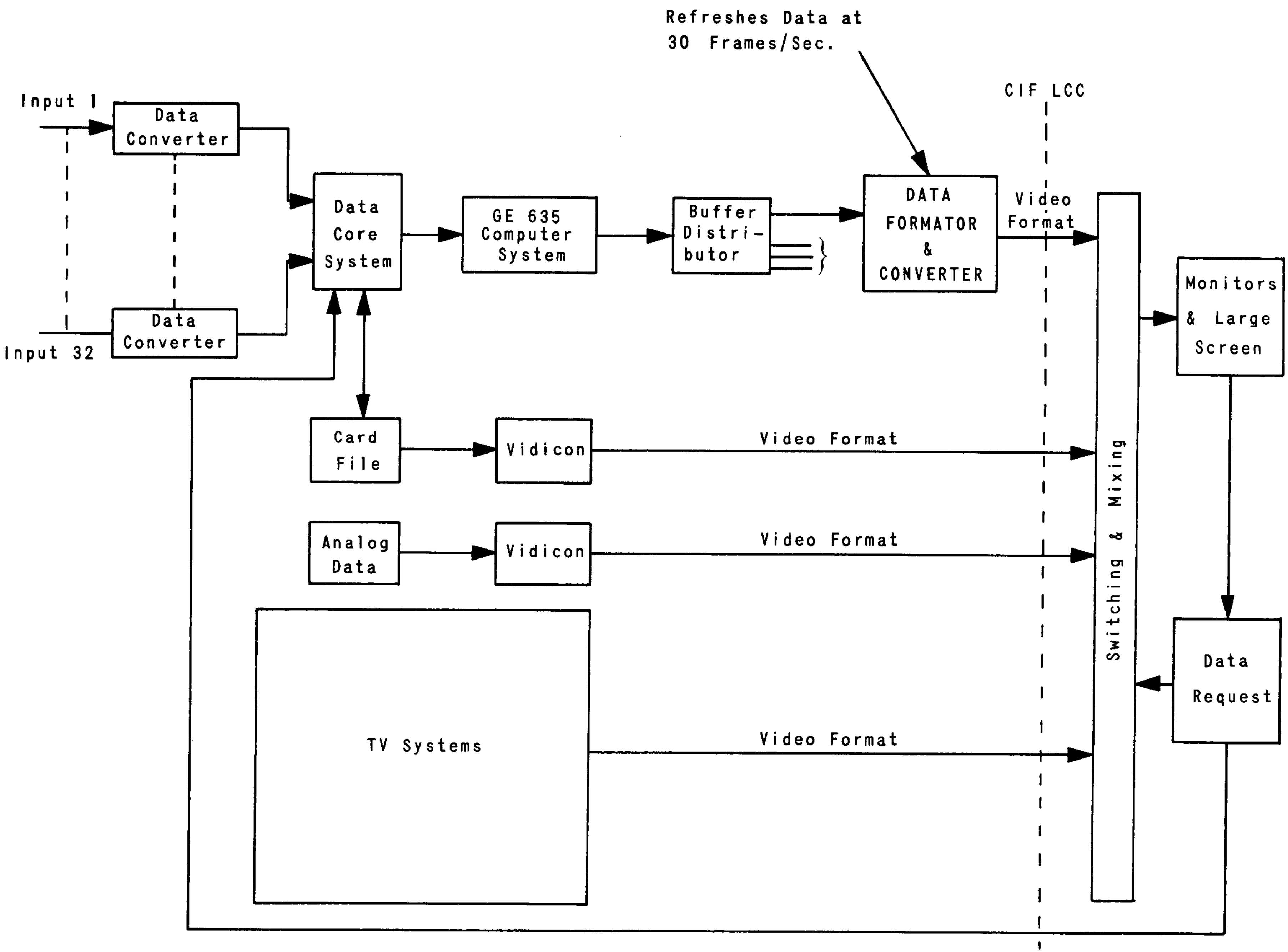


FIGURE 14 REAL TIME DATA DISPLAY SYSTEM

APPENDIX B

Computer Program Development Milestones

In any programming process, several end items and events are identified which might serve as milestones for scheduling and progress monitoring of specific programming tasks. Once for a specific task the appropriate milestone events are identified, close management control can be affected of the project. This section designates which types of programming milestone events are appropriate for such monitoring and discusses criteria for defining completion of milestone events in progress reporting.

B1.0 Some Suggested Milestones

Several detailed software development milestone events can be monitored by the customer (NASA centers) in order to achieve close management control of the programming process. Several such milestone events which might be applicable to specific programming tasks are described below. The sequence of these events is shown in the attached figure.

This list of milestone events must, of course, be tailored to each specific programming task. For less complex programming tasks than that of the example shown in the attached figure, several of these events may not be appropriate and/or may be merged for convenience. Hence, this list can best serve only as a general guideline for defining milestone events.

(1) System Performance Requirements Document Available

The system performance requirements is a document or set of documents which defines the overall system functions which shall be implemented by the computer-based system under development. It describes the operational capability, functional objectives, physical limitations, and other idiosyncrasies of the system requirements, and it distinguishes between which system functions shall be implemented in hardware, which in software, and which manually. It represents a deliverable end item of the programming process. It is the NASA Center's responsibility to assure that these requirements become available to the program development contractor.

(2) Software System Design Requirements Document Complete

The software system design requirements is a document or set of documents detailing the tasks of software development or modifications to existing software required to perform the system functions which are to be implemented by computer programming. It is an extension of the system performance requirements document of (1) above, relating strictly to the computer programs and personnel (operator) position requirements of the computer-based system. When not incorporated as part of the system performance requirements document of (1) above, it represents a separate deliverable end item of the programming process.

(3) Software System Performance and Preliminary Design Specifications Document Complete

The software system performance and preliminary design specifications is a document or set of documents which contains the technical performance specifications, requirements, factors, plans, and procedures necessary to permit design and implementation of the software subsystems. It contains identification of the operational software subsystem functions, utility and support software subsystem requirements, data base requirements, software testing requirements, and what existing software subsystems (if any) can be modified toward these goals. It represents a deliverable end item of the programming process.

(4) Computer Hardware Characteristics Frozen

Before detailed software design can be done, the computer hardware characteristics must be specified, fixed, and available to the software development contractor. This event does not necessarily represent a deliverable end item, but it is critical to software design.

(5) Utility, Support, and Operational Software Subsystems Design Specifications Documents Complete

The software subsystems design specifications are documents or sets of documents, one for each of the utility, support, and operational software subsystems as appropriate, which contain the detailed design of the subprograms to comprise the subsystem, the specific subfunctions to be performed by each individual subprogram, and the inputs, outputs, interfaces, and logic structure of each. These documents represent deliverable end items of the programming process.

(6) Utility, Support, and Operational Software Subsystems Coding and Documentation Complete

Separate programming milestones events should be monitored for the coding and documentation completion of each of the utility, support, and operational software subsystems. If appropriate, the operational software subsystem may be coded, documented, and program tested in two or more groups or packages of subprograms; where this is the case, the completions of each package could be monitored as separate milestone events. The sets of subprogram documents of each subsystem are deliverable end items.

(7) Utility and Support Software Subsystem Testing Complete

Two programming milestone events, not representing deliverable end items, are the completion of both piecemeal program debugging and subsystem overall testing of the utility and support software subsystems. These events denote the availability of these two software subsystems for use in checkout of the operational software subsystem and are critical events in the programming process.

(8) Operational Software Subsystem Programs Debugging Complete

This is a milestone event, not representing a deliverable end item, which denotes completion of individual subprograms debugging of the operational software subsystem and the availability of those subprograms for simulated environment testing as a group. If the operational software subsystem is being developed in packages, separate such milestone events could be monitored for each package.

(9) Operational Software Subsystem Simulated Environment Testing Complete

This is a milestone event, not representing a deliverable end item, designating completion of testing, in a simulated environment, of the operational software subsystem as an integrated group of subprograms. If the operational software subsystem is being developed in packages, separate such milestone events could be monitored for each package.

(10) Utility, Support, and Operational Software Subsystems Acceptance Test Specifications Document Complete

These three milestone events are the completion of the acceptance test plans documents for the utility, support, and operational software subsystems. Software subsystem acceptance test plans describe in detail the testing requirements which must be satisfied for official "buy-off" or acceptance of the software subsystem by the customer (NASA Center) and release of the program development contractor from responsibilities for implementation efforts on the subsystem. These acceptance test plans are deliverable end items in the programming process.

(11) Personnel Positions Handbooks and User's Manual Document Complete

The personnel position handbooks and user's manual, if appropriate, is a deliverable end item document or set of documents prepared for each type of operator position in the system and provides all necessary information of the performance of that position. It is primarily a complete reference document suitable as a personnel training manual, and is designed to meet the needs of the operator console stations having computer-generated displays, manual switch actions, and/or external communication capabilities.

(12) Utility, Support, and Operational Software Subsystems Acceptance Testing Complete

These are three milestone events, one for each of the utility, support, and operational software subsystems, denoting the completion of acceptance testing as defined in the acceptance test plan documents of each subsystem. These milestone events represent deliverable end items of the programming process in the sense that they represent acceptance or "delivery" of the subsystems by or to the NASA Center.

(13) User's Installation Equipment and Facilities Available

The availability of all equipment and facilities at the user's installation is a milestone event which is critical for system integration of the overall hardware-software-environment - personnel system. If the software is to be integrated in several different user's installations, separate such milestone events should be monitored for each. As part of the programming process, such milestone events do not represent deliverable end items.

(14) Start and Completion of Overall Hardware - Software - Environment - Personnel System Integration Tests

The integration of the operational and support software subsystems, system hardware, external environment, and operator personnel into a total working system and the testing of that total system is a major activity of the programming process before the system becomes fully operational or the mission is flown. Both the start and completion of such testing should be monitored as separate milestone events. Such events do not represent deliverable end items of the programming process.

(15) Final Insertion of System Parameters

The software system data base requirements are represented in milestone events (3) and (5) listed above, and implementation of the data base in computer-accessible form is represented in milestone event (8) listed above. However, final delivery of the exact system parameter numerics to be used in final system use or actual mission flight is an additional distinct milestone event critical for system use. Since such numerics may come from NASA Centers other than that responsible for the particular system installation, this milestone event represents a deliverable end item.

(16) Operational Software Subsystem and All Documents Updated per Test Results

A final milestone event to be monitored for the operational software subsystem is the completion of all modifications to the software subsystem and previously delivered subsystem documents as are required as a result of software malfunctions and oversights which were exposed during the software subsystem and overall system testing. Insofar as established configuration management procedures involve documented change orders, these updating milestone events represent deliverable end items in the programming process.

B2.0 Monitoring and Reporting Milestones

At the present time, two tools are being used throughout NASA for reporting status and progress on development activities: PERT (Program Evaluation and Review Technique) and SARP (Schedules and Review Procedure). Both of these tools are applicable to software development management, but each has certain weaknesses. SARP is better suited for high-level status monitoring where summarization is needed of the status of uncompleted tasks. However, PERT is better suited for detailed software development monitoring and management control, since it permits graphical presentation of status of several activities in parallel which is much the case of computer program coding and debugging, and it is better than SARP for reporting status of activities which are essentially modifications to existing systems. The figure attached can be used as a pattern for software PERT charts.

The discussions of this section, on the other hand, are oriented toward SARP as a reporting tool, i.e., milestones (clearly definable points in time during the development activities) are emphased and not the activities themselves. Care must be exercised in identifying those such milestones which are appropriate to specific software design and procurement tasks. There appear to be two classes of software projects pertaining to Apollo: those which involve essentially new software development, and those which envolve essentially minor modifications to existing or previously developed computer program systems. SARP milestones for the new software projects should where possible be patterned after the "generalized" milestones discussed above. For the modifications type software projects, however, not all of these are appropriate, although certain critical stages are still involved:

(1) Specifications of the system (new or modified) in which the modified software is required.

(2) Specifications of the required software modifications.

(3) Implementation of the software modifications (coding, debugging, documenting).

(4) Checkout and acceptance of the modified software.

For essentially new software development tasks, the following milestone identification discussion applies:

Of the sixteen (16) types of software milestone events listed above as appropriate for customer (NASA Center) monitoring of the programming process, six (6) are of sufficient criticality in the programming process to be monitored by NASA headquarters and to be subjected to progress reporting to NASA headquarters by the NASA centers responsible for each specific programming task. These six are:

(1) <u>System Performance Requirements Document Available</u>

The completion of a document generated by the NASA Centers describing to the contractor the job to be done by the computer based systems as it affects computer programming. This milestone event represents a deliverable end item of the computer programming process. (See B1.0 (1) above.)

(2) <u>Software System Performance and Preliminary Design Specifications Document Complete</u>.

The completion of a document by the contractor describing the way in which he plans to do the job specified by the system performance requirements document. This milestone event represents a deliverable end item of the computer programming process. (See B1.0 (3) above.)

(3) Operational Software Subsystem Programs Debugging Complete

The point in time when coding and debugging of the operational software subsystem is complete on subprogram basis. At this time, integration of the program packages can begin. This milestone event does not represent a deliverable end item of the computer programming process. (See B1.0 (8) above.)

(4) Operational Software Subsystem Simulated Environment Testing Complete

The point in time when the subprograms have been integrated and tested as a complete program. This is the time when the contractor feels the operational software subsystem is ready for acceptance testing. This milestone event does not represent a deliverable end item of the computer programming process. (See B1.0 (9) above.)

(5) Operational Software Subsystem Acceptance Testing Complete

The point in time when the NASA Center has agreed that the acceptance testing of the operational software system has been successfully completed. This milestone event represents a deliverable end item of the computer programming process in the sense that it represents acceptance or "delivery" of the program by or to the customer. (See B1.0 (12) above.)

(6) Completion of Overall System Integration Testing

Completion of overall system operational testing in which the software is integrated as part of the overall system in which it operates. Upon the completion of testing the system is considered available for use. This milestone event does not represent a deliverable end item of the computer programming process. (See B1.0 (14) above.)

B3.0 Criteria For Reporting Milestone Achievement

Reporting progress on such programming milestone events for monitoring is misleading without succinct agreements on definitions of what constitutes achievement of each type of milestone event. The following definitions could be used to this end:

A software document should be considered complete only when it has been reviewed by the customer (NASA Center), approved, published, and is available.

A set of software documents should be considered complete only when the complete list identifying the documents which comprise the set has been agreed upon by the customer (NASA Center) and each document of that list is complete by the above criterion of completion.

A software subsystem (set of subprograms) should be considered to have been completely coded when the complete list identifying the subprograms has been formulated by the contractor and agreed upon by the customer (NASA Center), and each subprogram thus identified has been completely coded, key punched, assembled without fatal diagnostics, and documented.

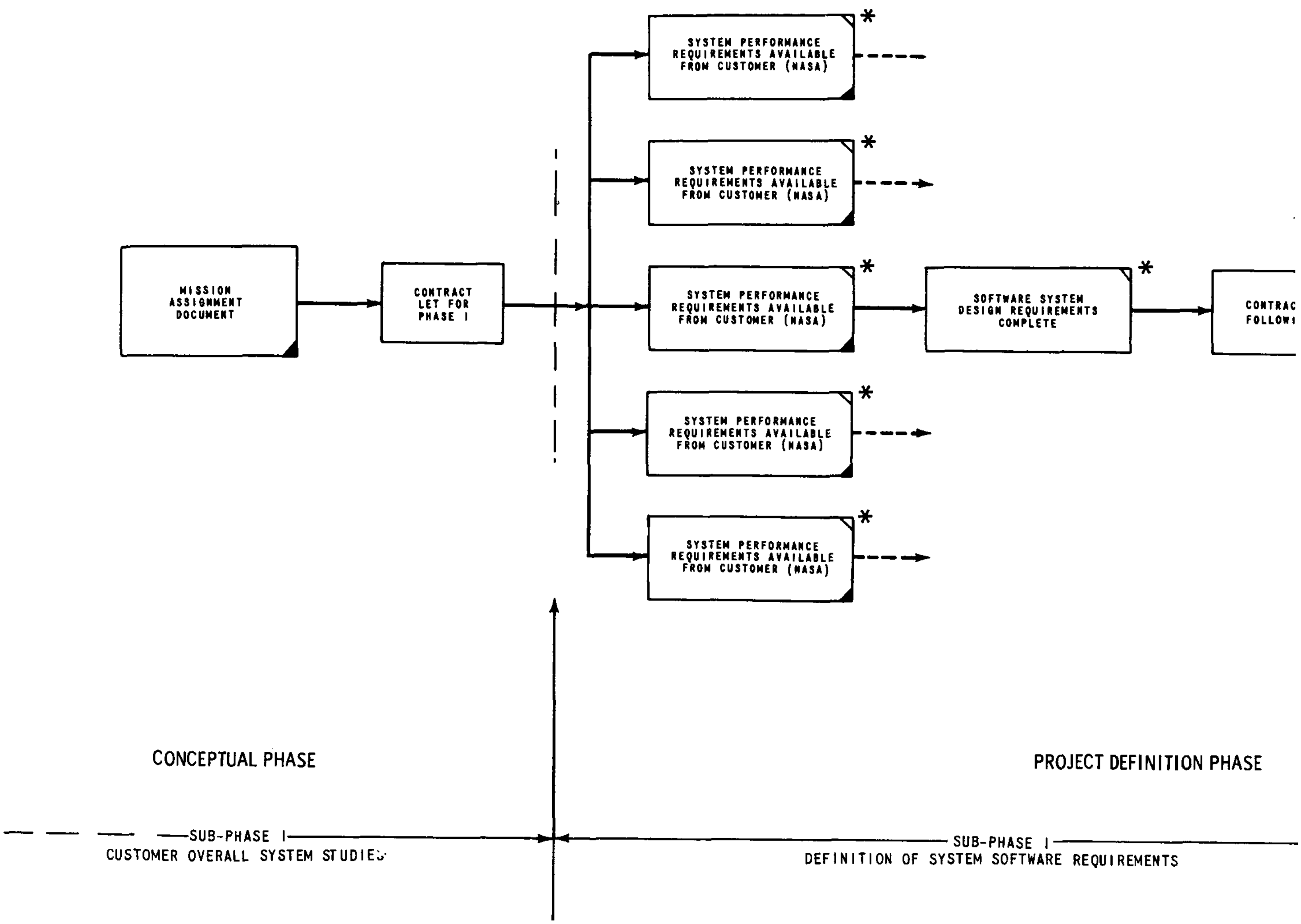
A software subsystem (set of subprograms) should be considered to have been completely (piecemeal) programs debugged when, in the judgment of the contractor, each subprogram of the agreed-upon list of subprograms of the subsystem is ready for overall subsystem simulated environment testing.

A software subsystem should be considered completely overall tested (or simulated - environment tested) when, after the subsystem has been completely (piecemeal) program debugged, the customer (NASA Center) and contractor concur that the simulated environment test objectives have been satisfied.

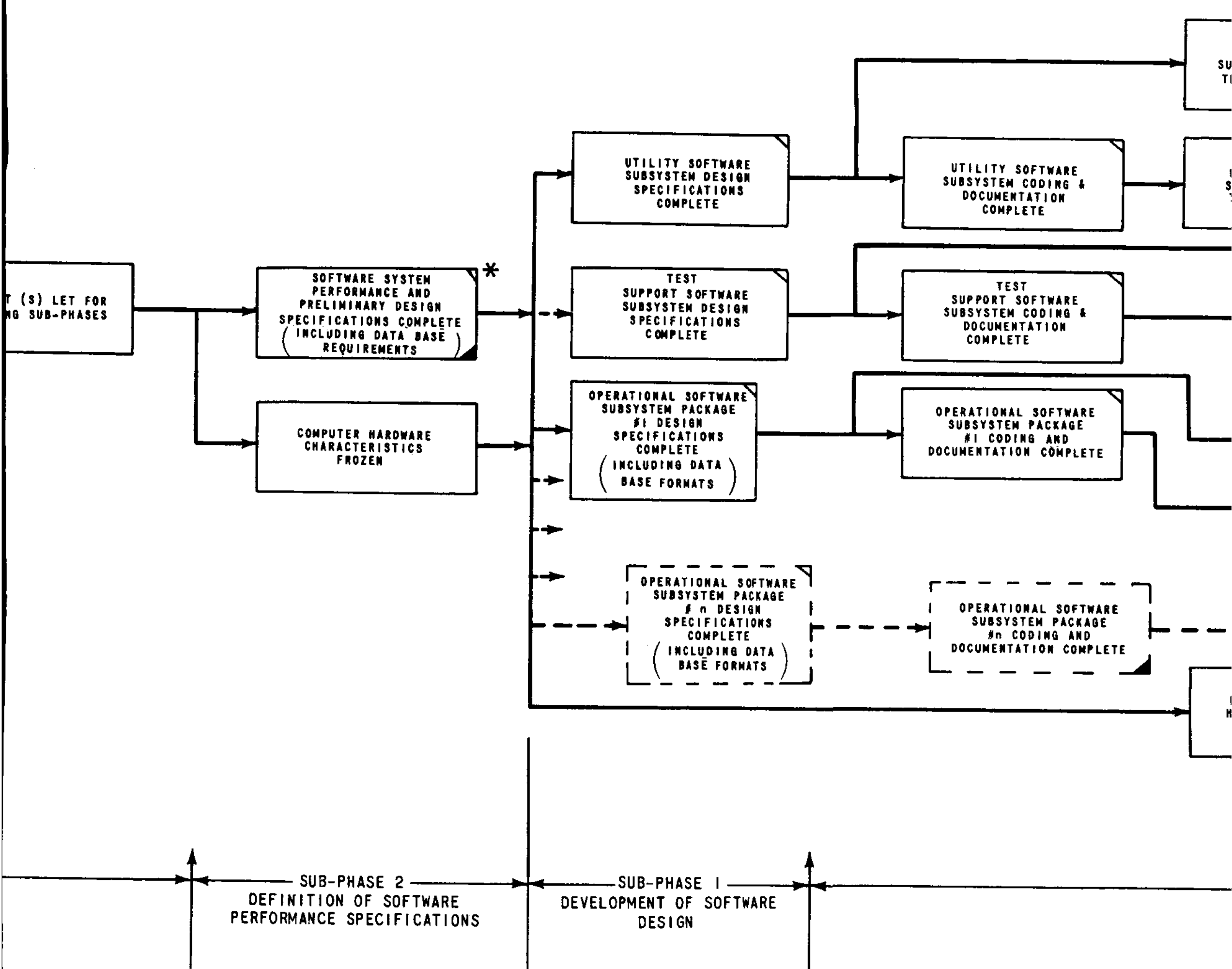
A software subsystem should be considered completely acceptance tested when, after completion of the subsystem simulated-environment testing and after the subsystem acceptance test plans documents have been completed, the customer (NASA Center) and contractor concur that the acceptance test objectives have been fully satisfied.

A programming task or computer-based system should be considered completely integrated tested when, after each subsystem has been completely acceptance tested and the system integration test plans documents have been completed, the customer (NASA Center) and the contractor concur that the hardware-software-environment-personnel system integration test objectives have been fully satisfied.

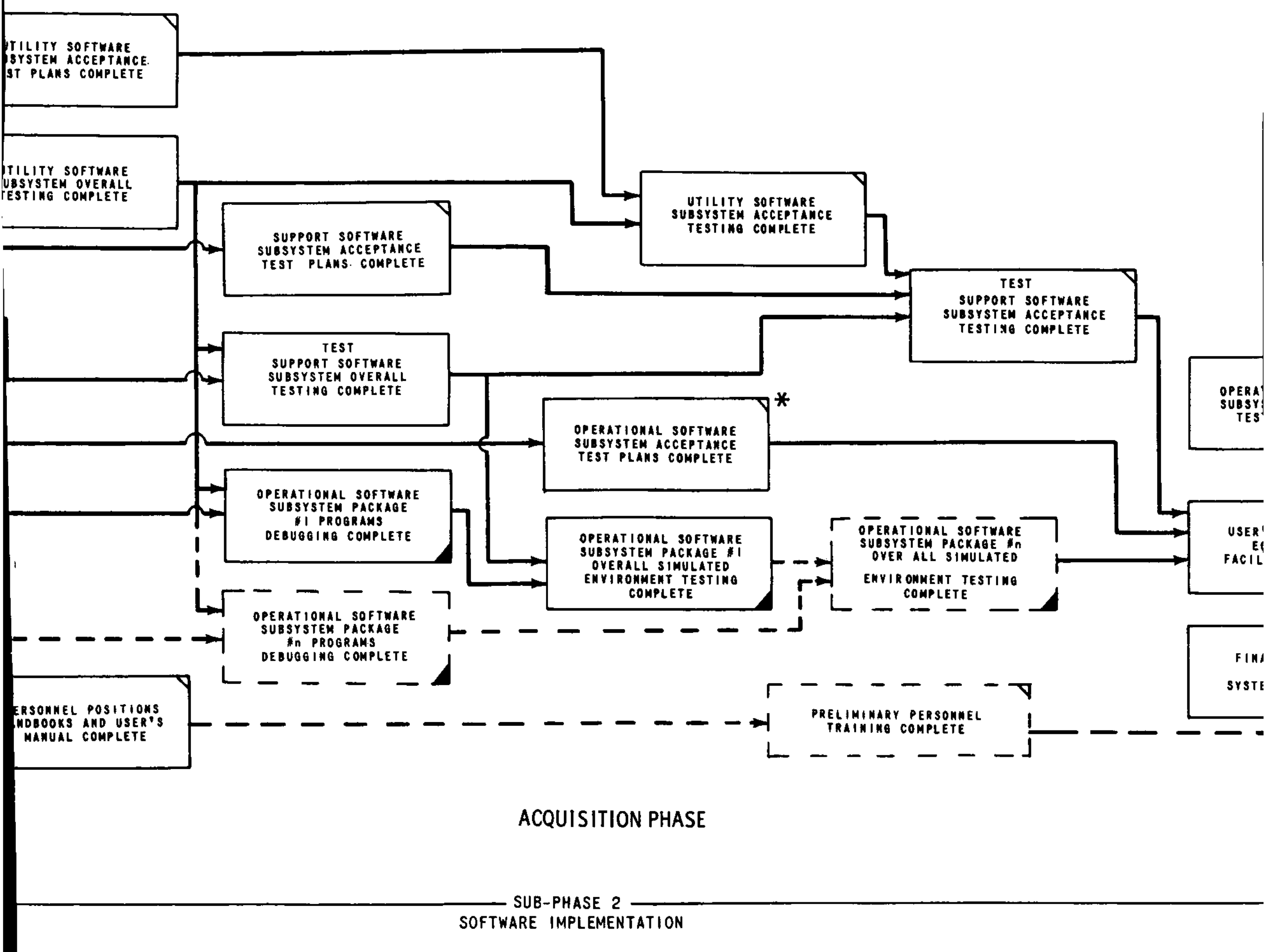
A software subsystem and all documents should be considered completely updated when all change orders pertaining to software errors and design oversights which have become manifest during testing have been completed in accordance with established configuration management procedures.



MISSION ASSIGNMENT DOCUMENT
CONTRACT LET FOR PHASE I
SYSTEM PERFORMANCE REQUIREMENTS AVAILABLE FROM CUSTOMER (NASA)
SYSTEM PERFORMANCE REQUIREMENTS AVAILABLE FROM CUSTOMER (NASA)
SYSTEM PERFORMANCE REQUIREMENTS AVAILABLE FROM CUSTOMER (NASA)
SYSTEM PERFORMANCE REQUIREMENTS AVAILABLE FROM CUSTOMER (NASA)
SYSTEM PERFORMANCE REQUIREMENTS AVAILABLE FROM CUSTOMER (NASA)
SOFTWARE SYSTEM DESIGN REQUIREMENTS COMPLETE
CONTRAC
FOLLOW
*
CONCEPTUAL PHASE
PROJECT DEFINITION PHASE
SUB-PHASE I
CUSTOMER OVERALL SYSTEM STUDIES
SUB-PHASE I
DEFINITION OF SYSTEM SOFTWARE REQUIREMENTS

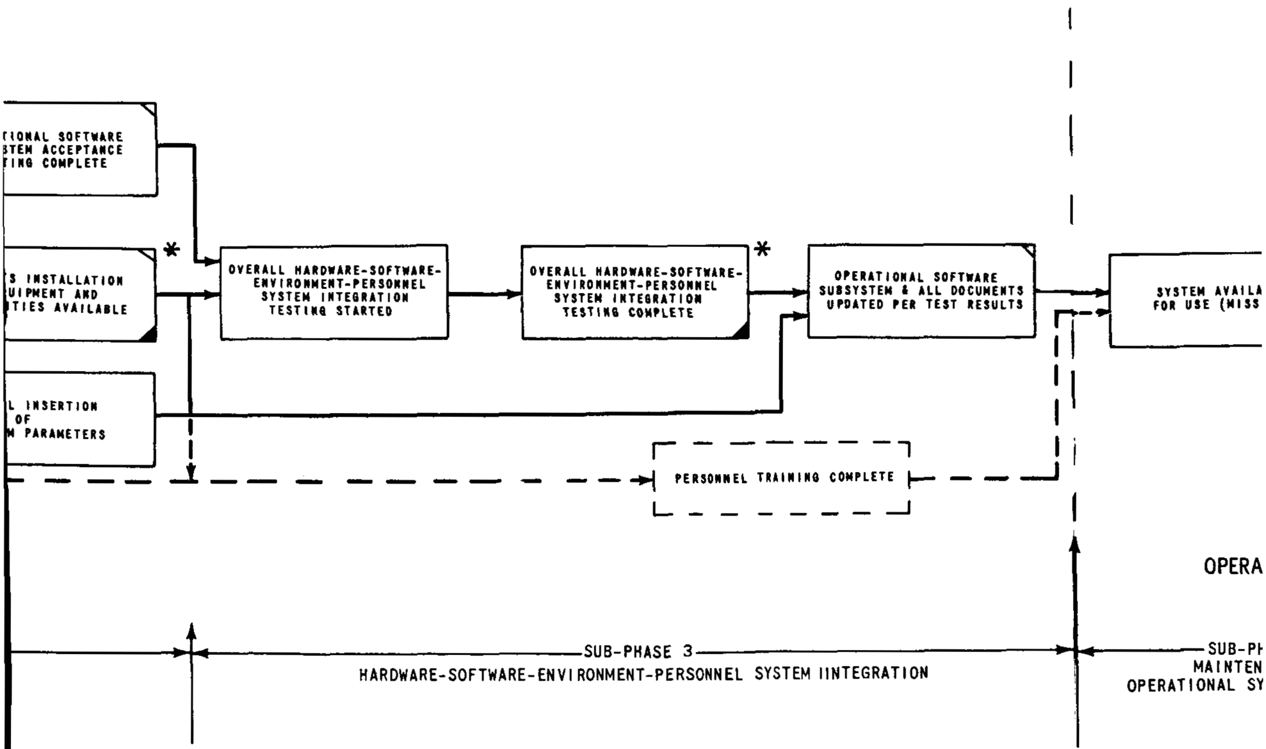

T (S) LET FOR
NG SUB-PHASES
SOFTWARE SYSTEM PERFORMANCE AND PRELIMINARY DESIGN SPECIFICATIONS COMPLETE (INCLUDING DATA BASE REQUIREMENTS)
*
COMPUTER HARDWARE CHARACTERISTICS FROZEN
UTILITY SOFTWARE SUBSYSTEM DESIGN SPECIFICATIONS COMPLETE
TEST SUPPORT SOFTWARE SUBSYSTEM DESIGN SPECIFICATIONS COMPLETE
OPERATIONAL SOFTWARE SUBSYSTEM PACKAGE #1 DESIGN SPECIFICATIONS COMPLETE (INCLUDING DATA BASE FORMATS)
OPERATIONAL SOFTWARE SUBSYSTEM PACKAGE # n DESIGN SPECIFICATIONS COMPLETE (INCLUDING DATA BASE FORMATS)
UTILITY SOFTWARE SUBSYSTEM CODING & DOCUMENTATION COMPLETE
TEST SUPPORT SOFTWARE SUBSYSTEM CODING & DOCUMENTATION COMPLETE
OPERATIONAL SOFTWARE SUBSYSTEM PACKAGE #1 CODING AND DOCUMENTATION COMPLETE
OPERATIONAL SOFTWARE SUBSYSTEM PACKAGE #n CODING AND DOCUMENTATION COMPLETE
SUB-PHASE 2
DEFINITION OF SOFTWARE PERFORMANCE SPECIFICATIONS
SUB-PHASE 1
DEVELOPMENT OF SOFTWARE DESIGN

2



TILITY SOFTWARE
BSYSTEM ACCEPTANCE
ST PLANS COMPLETE
TILITY SOFTWARE
UBSYSTEM OVERALL
ESTING COMPLETE
SUPPORT SOFTWARE
SUBSYSTEM ACCEPTANCE
TEST PLANS COMPLETE
UTILITY SOFTWARE
SUBSYSTEM ACCEPTANCE
TESTING COMPLETE
TEST
SUPPORT SOFTWARE
SUBSYSTEM ACCEPTANCE
TESTING COMPLETE
TEST
SUPPORT SOFTWARE
SUBSYSTEM OVERALL
TESTING COMPLETE
OPERATIONAL SOFTWARE
SUBSYSTEM ACCEPTANCE
TEST PLANS COMPLETE
*
OPERATIONAL SOFTWARE
SUBSYSTEM PACKAGE
#1 PROGRAMS
DEBUGGING COMPLETE
OPERATIONAL SOFTWARE
SUBSYSTEM PACKAGE #1
OVERALL SIMULATED
ENVIRONMENT TESTING
COMPLETE
OPERATIONAL SOFTWARE
SUBSYSTEM PACKAGE #n
OVER ALL SIMULATED
ENVIRONMENT TESTING
COMPLETE
OPERATIONAL SOFTWARE
SUBSYSTEM PACKAGE
#n PROGRAMS
DEBUGGING COMPLETE
ERSONNEL POSITIONS
NDBOOKS AND USER'S
MANUAL COMPLETE
PRELIMINARY PERSONNEL
TRAINING COMPLETE
OPERA
SUBSY
TES
USER
E
FACIL
FIN
SYSTE
ACQUISITION PHASE
SUB-PHASE 2
SOFTWARE IMPLEMENTATION

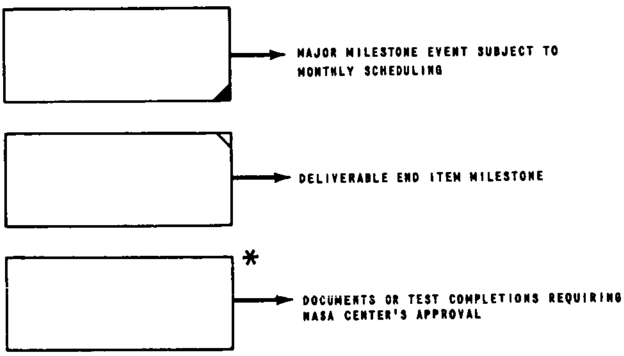
3



TIONAL SOFTWARE
STEM ACCEPTANCE
TING COMPLETE
S INSTALLATION
UIPMENT AND
TIES AVAILABLE
L INSERTION
OF
M PARAMETERS
*
OVERALL HARDWARE-SOFTWARE-ENVIRONMENT-PERSONNEL SYSTEM INTEGRATION TESTING STARTED
OVERALL HARDWARE-SOFTWARE-ENVIRONMENT-PERSONNEL SYSTEM INTEGRATION TESTING COMPLETE
*
OPERATIONAL SOFTWARE SUBSYSTEM & ALL DOCUMENTS UPDATED PER TEST RESULTS
SYSTEM AVAILA
FOR USE (MISS
PERSONNEL TRAINING COMPLETE
OPERA
SUB-PHASE 3
HARDWARE-SOFTWARE-ENVIRONMENT-PERSONNEL SYSTEM IINTEGRATION
SUB-PH
MAINTEN
OPERATIONAL SY

4

LEGEND:



.E
N)

ON PHASE

SE I
CE OF
EM SOFTWARE

MILESTONE EVENTS OF THE COMPUTER

PROGRAMMING PROCESS

FIGURE FOR APPENDIX

5

BELLCOMM, INC.

DISTRIBUTION LIST

NASA Hdqtrs.

W. M. Collins, Jr./M-BC
J. H. Disher/MAT
T. A. Keegan/MA-2
J. Kubat/MAP
R. V. Murad/MAT
S. C. Phillips/MA
E. T. Sullivan/MAP

Bellcomm, Inc.

G. A. Brill
F. C. Childs
R. E. Gradle
J. A. Hornbeck
C. A. Lovell
I. M. Ross
T. H. Thompson
J. M. West
Central File
Library
Department 2014