AEROSPACE REPORT NO.
ATR-74(7341)-4

Operations Analysis (Study 2.1)

Shuttle Upper Stage Software Requirements

DRA

Prepared by
Advanced Mission Analysis Directorate
Advanced Orbital Systems Division

15 July 1974

Prepared for OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW-2575

Systems Engineering Operations

THE AEROSPACE CORPORATION

(NASA-CR-139597) OPERATIONS ANALYSIS (STUDY 2.1): SHUTTLE UPPER STAGE SOFTWARE REQUIREMENTS (Aerospace Corp., El Segundo, Calif.) 113 p HC $8.75 CSCL 22A G3/30

N74-32253

Unclas
17149

Aerospace Report No.
ATR-74(7341)-4

OPERATIONS ANALYSIS (STUDY 2.1)

Shuttle Upper Stage Software Requirements

Prepared by

Advanced Mission Analysis Directorate
Advanced Orbital Systems Division

15 July 1974

Systems Engineering Operations
THE AEROSPACE CORPORATION
El Segundo, California

Prepared for

OFFICE OF MANNED SPACE FLIGHT
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D. C.

Contract No. NASW-2575

Aerospace Report No.
ATR-74(7341)-4

OPERATIONS ANALYSIS (STUDY 2.1)

Shuttle Upper Stage Software Requirements

Prepared

R. R. Wolfe
NASA Study 2.1 Director
Advanced Mission Analysis Directorate

Approved

L. R. Sitney, Assoc. Group Director
Advanced Mission Analysis Directorate
Advanced Orbital Systems Division

FOREWORD

This report presents the results of subtask 3 of Study 2.1, Operations Analysis. This subtask has as its primary objective the investigation of software costs related to Shuttle upper stage operations with emphasis on the additional costs attributable to space servicing. The increased complexity of automated space servicing, beyond current development and recurring costs could be excessively high.

Historically, software development efforts for space programs have been difficult to scope. This has been due in part to the lack of firm requirements at the outset, resulting in numerous unscheduled revisions and retest cycles. This report addresses this problem as well as several other factors which influence the ability to predict software costs for the Shuttle upper stage.

Specific interest is directed at the additional cost and complexities associated with space servicing of automated payloads by the upper stage in a preprogrammed mode. Consideration is also given to manned interactive support at the Mission Control Center and the associated impact this would have on software design and cost.

This subtask of Study 2.1 represents approximately 10% of the total effort. Companion reports are being published on other subtasks and a final report will be published at the end of the contract period. Study 2.1 is one of three study tasks being conducted under NASA Contract NASW-2575 in FY 1974. The NASA study director is Mr. V. N. Huff, NASA Headquarters, Code MTE.

One caution should be observed. The results of any task such as this are highly dependent upon the initial set of ground rules and assumptions. The Shuttle upper stage operational concept assumed for this effort is based upon the experience gained from existing USAF programs and as such provides a rational basis for estimating software requirements. Other concepts, such as manned space operations, may result in different requirements.

ACKNOWLEDGMENTS

In performing this subtask it was necessary to communicate with several firms having experience in complex space program software developments. Firms such as System Development Corporation, IBM, TRW, Lockheed Missiles System Corporation, and the MIT Draper Laboratory, under Dr. R. Battin. A mutual concern was shown by everyone to work for improved cost estimation techniques. The response was, in all cases, positive and very much appreciated.

Additionally, recognition should be made of two individuals at The Aerospace Corporation who performed the functional analyses and supported the contractor interviews; Mr. M. Jensen for his efforts on spaceborne software, and Mr. R. Coulston for his effort on ground checkout software.

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

Space servicing of automated satellites offers the possibility of reducing future space program expenditures through improved utilization of the Shuttle and a reduction in payload procurement costs. Satellites may be serviced on orbit by removing failed or expended modules and replacing these with operational units. This function, when performed by the Shuttle upper stage, can be completely preprogrammed prior to liftoff. In addition, upper stage operations may involve servicing of several satellites on any one flight. One of the principal concerns is that the increased complexity of space servicing relative to current space operations could result in excessive upper stage software costs. This involves not only software development for spaceborne and ground systems but the recurring costs of maintaining the software system as well. Therefore, this subtask of Operations Analysis (Study 2.1) has as its primary objective the investigation of software costs related to Shuttle upper stage operations with emphasis on the additional costs attributable to space servicing.

The problem of attempting to estimate software costs for future programs is well known in nearly every field of design and development. Software is notoriously difficult to control and invariably overruns cost projections and scheduled delivery dates. Although this has occurred repeatedly, the one saving grace to date has been that the software cost, even with overruns, is a small percentage of the total design and development program involved. Therefore, unless the overrun is substantial, it may be absorbed without serious impact on the program. However, it is reasonable to expect software development for future programs to require a larger percentage of the budget for two reasons: (1) the number and complexity of functions to be computerized are increasing dramatically, thereby driving software costs up and (2) computer hardware costs continue to decrease, emphasizing the higher percentage of budget required for software. The additional complexity associated with space servicing functions, therefore, is a rational cause for concern.

The following questions are fundamental to the problem of estimating software costs:

a. What key parameters are involved with software costs?

b. Do sufficient historical data exist as a basis for extrapolation to the future?

c. What elements of the basic software development effort are applicable to servicing functions?

d. How complex is servicing compared to current satellite deployment operations?

e. Does multiple servicing materially increase the complexity?

f. Are recurring software costs significant?

The results presented in this report address these questions and provide a foundation for estimating software costs based upon historical records of similar programs as modified by a series of empirical factors. These empirical factors have been derived through research of current software cost trends and personal conversations with software development firms. In this regard, although subjective in nature, the principal factors affecting software costs are exposed for consideration. The final product is an estimate of the upper stage recurring and nonrecurring software costs for all mission phases with and without space servicing operations.

2. STUDY APPROACH

Several different approaches were attempted in an effort to achieve a "top down" method of estimating software costs but all suffered from a lack of sufficient substantive data to correlate the results with historical information. The approach which was finally selected uses an estimate of the required machine instructions and can be related to some extent with previous developmental and operational programs although it does require empirical judgment in addition. To this extent, the results can be used to arrive at a realistic estimate of software costs for comparison with the total program effort. The approach selected to estimate software costs consists of the following two efforts:

a. Estimate upper stage operational software requirements by functional analysis.

b. Develop software cost factors based upon historical data and a survey of current computer firm practices.

The necessary information involved with estimating upper stage software requirements is shown schematically in Figure 2-1. Several mission classes were selected for analysis, each having increasingly complex software requirements. The first mission of interest is one of deploying payloads in geosynchronous orbit with an expendable upper stage, similar to current Titan IIIC transtage operations. Airborne computer software requirements for this type of mission are well defined in terms of words of instructions, computer capacity, etc., and therefore offer a basis for comparing additional functions leading up to and including space servicing operations.

Subsequent missions include deployment of a payload in geosynchronous orbit with a recoverable upper stage. The initial part of the operation is similar to the first mission but the added complexities of guidance reinitialization, retrofire, and rendezvous with the Shuttle in low-earth orbit are required. The next step that follows incorporates retrieval of payloads, requiring a rendezvous and docking capability in

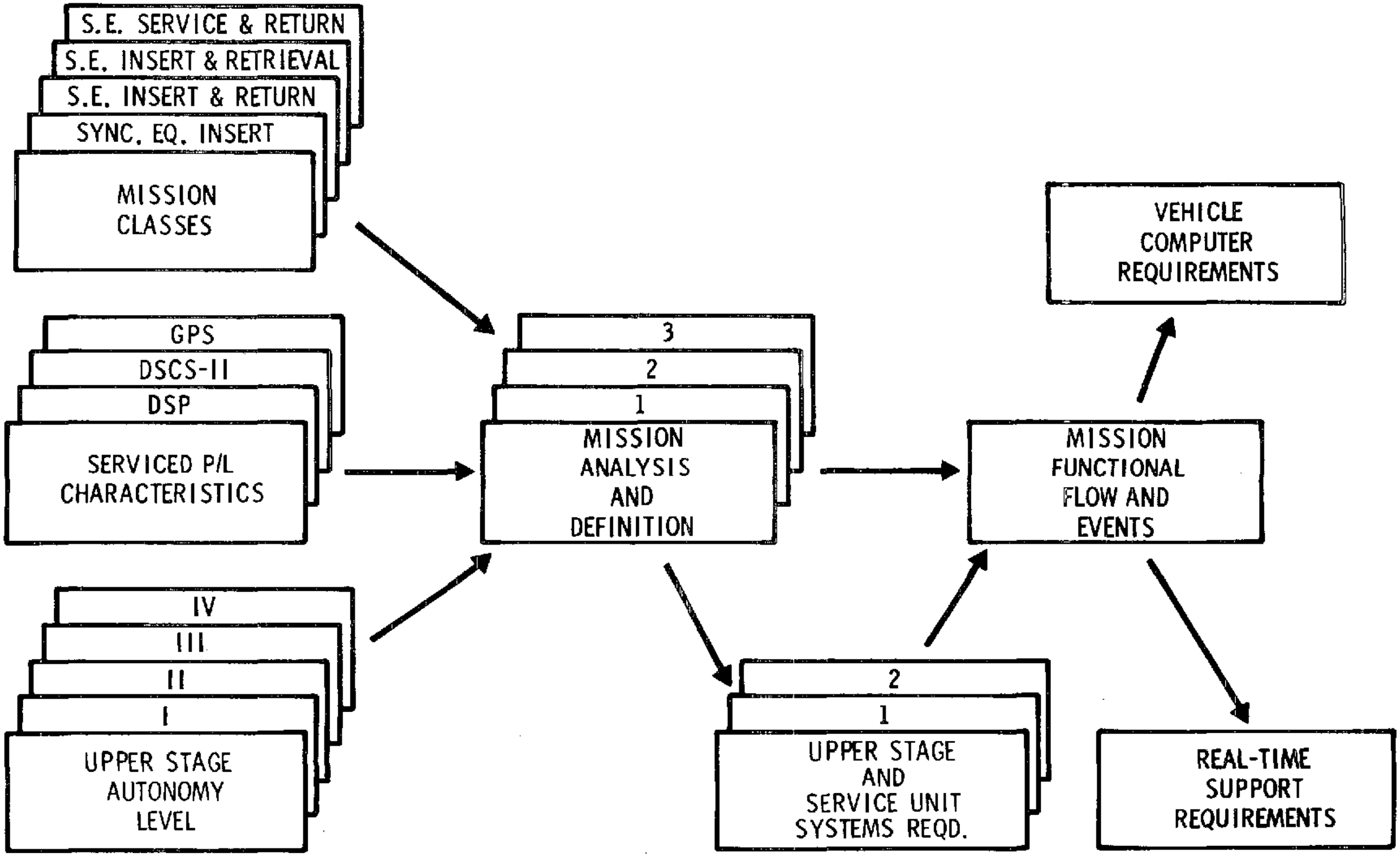


Figure 2-1. Payload/Tug Software

geosynchronous orbit. The final step for geosynchronous operations involves servicing up to as many as four different satellites at different longitude positions. Modules are removed and replaced in each satellite and the upper stage returns to the Shuttle with the failed space replaceable units (SRUs). The commonality between missions is obvious and, consequently, the subsequent functional analysis addresses primarily the multiple service mission as compared to expendable upper stage operations. Visibility is maintained such that each degree of complexity is readily seen as it contributes to the total effort.

In this approach, the only unique point relative to payload definitions is that the design approach selected is based upon preliminary designs of several DOD and NASA satellites developed at The Aerospace Corporation. The satellites must obviously be of a serviceable design and are assumed to be three-axis stabilized to permit rendezvous and docking. Further definition is provided in the following section, Ground Rules and Assumptions.

Selection of an upper stage configuration is not particularly important to this effort. The functions to be performed are essentially independent of the stage performance capability, at least as far as software requirements are concerned. However, it is important to specify the level of autonomy employed by the upper stage as well as service equipment interfaces, since these directly relate to all phases of software usage. Numerous studies have been performed, but the level of autonomy is currently undefined as are rendezvous and docking sensors and equipment. Therefore, it was necessary to define an upper stage concept that could be employed for space servicing. To aid this process a top level contingency analysis was performed (Ref. 1). In this way it was possible to arrive at a reasonable level of autonomy as well as a definition of typical instrumentation and equipment for the upper stage. This then cascades into various software requirements for ground checkout and flight operations support, impacting on both design,development, testing and engineering (DDT&E)

and recurring software costs. The rationale, guidelines, and assumptions selected are provided in the next section.

The remainder of the elements shown in Figure 1-1 relate to developing timelines, sequences of events, and functional analyses of the various missions. By this approach, based on prior program experience, it is possible to delineate the number of words of instructions. Although this process is reasonably straightforward for spaceborne computer operations (because of definable boundary conditions), it becomes very difficult for ground and flight support systems. In this regard, various contractor study results were employed where appropriate, and integrated with the experience obtained from USAF launch vehicle and satellite test operations. This then represents the approach employed to arrive at software requirements for upper stage operations with and without a space servicing capability.

The second phase of the study approach performed in parallel with the functional analyses consisted of performing research on existing software cost data. There already exists a repository of data at The Aerospace Corporation relative to SAMSO programs of the past. However, with the exception of the Titan IIIC program, it is difficult to relate the resultant cost to an initial set of requirements. The same was true of data obtained from computer and software firms. Historically, software requirements continually change during the program development with little traceability to original cost estimates.

Each contractor tends to cost software in a somewhat different manner. In each instance, when reviewing this subject, the contractor was asked what he would base his cost estimate on if a proposal were requested for a program such as the Shuttle upper stage. Many factors are involved, but no consistent trend was obvious. Perhaps the first point of note was the question by the contractors of, "What budget has been allocated?" It then appeared that to the greatest extent possible the contractors would attempt to scope the interpretation of software requirements to fit the budget. This is not altogether without reason since most contracts in the past have had a very loose definition of software requirements. The contractor is then left with the positive injunction to develop an operational system within whatever budget allocation is provided. In essence, no one has any general set of guidelines. Each program is assessed separately depending upon market conditions.

Several empirical factors do however appear to influence the contractor response. These factors are generally recognized by the majority of the industry, but little quantifiable data exists to produce a firm set of relationships. The approach taken here then is to employ rational factors for the several variables involved, to arrive at a reasonable upper bounds of software costs. The factors involved are discussed in Section 5. The values employed in the final cost estimate are defined in Section 6. Since judgment is involved, the values selected must be considered in light of the study objectives to estimate the software cost and the impact of servicing operations on the Shuttle upper stage program. Further effort is desirable to research other programs in an attempt to quantify these parameters and develop firm software cost estimating relationships.

3. GROUND RULES AND ASSUMPTIONS

Before the functional analysis of upper stage operations can be considered, the specific ground rules and assumptions used in this analysis must be clearly defined. Basically, there are three major points which are somewhat arbitrary but which form the foundation of the analysis which follows.

a. Servicing operations are a support function performed in response to a payload user request.

b. Upper stage operations require only a minimal interface with the Shuttle.

c. The service unit is essentially a self-contained entity programmed to perform the service function after docking with the payload has been accomplished.

In considering the overall operational concept, it is important to distinguish between the roles of the service agency versus those of the payload user agency. The user is the only one qualified to diagnose an off-nominal condition for a given satellite to determine if servicing is required. It is the user who must specify the unit to be replaced. It is also the user who must perform the system checkout after the satellite has been serviced. The same procedures would be employed as those at the time of initial deployment to bring the satellite to an operational state. Consequently, this capability must reside with the user and,therefore, no checkout capability exists with the service unit. Although the software problem of isolating a failed component may be significant, it should not be affected by the satellite's being reconfigured for space servicing. Essentially, the same information is required for subsystem monitoring and diagnosis, whether of an expendable or serviceable configuration. The failure must be isolated before corrective action can be initiated. With space servicing, it is only necessary to isolate the failure to the SRU, since bench testing can be performed in the laboratory after return of the module. Therefore, there may actually be a reduction in monitoring requirements. The important point is that any such software requirements have been disassociated from the logistics operation and consequently have no impact on upper stage software requirements.

It is also assumed that the payload user has the responsibility for placing the payload in a serviceable configuration. All satellites to be serviced are assumed to be attitude-stabilized in all three axes. However, a given satellite may be earth-oriented, sun-oriented, or inertially fixed in space, depending upon its mission. Rendezvous and docking instrumentation must be incorporated into the satellite design and, consequently, the orientation must be known to support the acquisition mode of the upper stage. For the purpose here, the satellite is assumed to be pre-positioned for rendezvous and docking as shown in Figure 3-1. In addition, all appendages are assumed to be retracted out of the path of approach of the upper stage. Once the payload has been placed in the proper position, the user communicates this information to the servicing operations center authorizing the rendezvous to proceed. In this way, there is no crosslink or interface between spaceborne communications systems. This is particularly important when considering the wide variety of payloads and user agencies which may be involved.

Interfaces between the upper stage and the Shuttle are considered only insofar as the software problem is concerned. There must be an RF command link from the Shuttle to the upper stage to support stage retrieval. Space servicing operations are assumed to have no impact on this link. The Shuttle is assumed to have the capability to initialize the upper stage guidance system at the time of deployment. It is assumed that the same RF link is employed. As an alternative, the Tracking and Data Relay Satellite (TDRS) system may be used for the same functions. In either case, the airborne computer must be capable of being updated relative to its own state vector. Also, signal conditioning and encoding/decoding of all telemetry information must be employed. These functions are all common to any Shuttle upper stage independent of servicing operations.

It is further assumed that there is no interface between the upper stage and the payloads when a service unit is employed. It is feasible to deploy payloads on a service mission, but the only physical interface with the payload is derived from the service unit, not the upper stage. On the

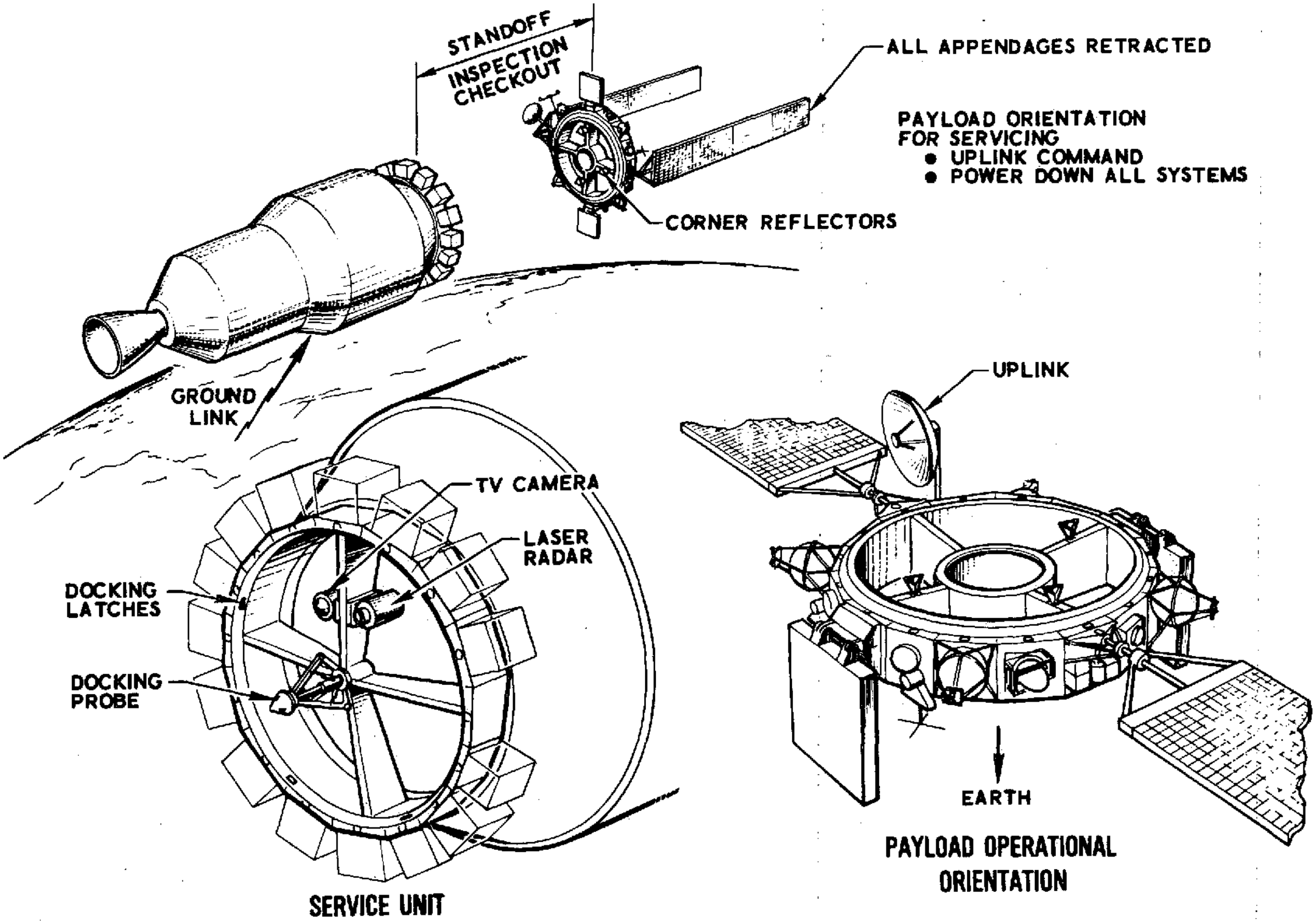


Figure 3-1. Space-Servicing Operations

other hand, the service unit is assumed to have both a physical and signal interface with the upper stage.* This is shown in Figure 3-2. Rendezvous and docking sensors must be located near the forward face of the service unit to provide an unobstructed view of the payload.

The service unit is initialized by the upper stage computer when rendezvous has been achieved and a hard dock established. Rendezvous is performed by aid of a laser radar with corner reflectors on the payload. The airborne functions are normally automated with a command override capability from mission control. In the event system errors preclude acquisition, it will be necessary to update the guidance system from the ground. A standoff maneuver is performed when the upper stage is approximately 30 to 50 meters from the payload. Visual monitoring is then performed via the television receiver to assure that the payload is in a proper configuration for docking. If so, the docking and servicing sequence is enabled by ground command and the functions are performed automatically. An override and backout capability exists at all times by virtue of manned interactive support through the command receiver.

Under normal circumstances, the upper stage maneuvers to the payload via signals from sensors mounted on the service unit. A hard dock is performed and verified by on-board sensors. The verification signal energizes the service unit sequence to initiate servicing. The necessary sequence of events and time periods involved are shown in Figure 3-3. It is assumed that the service unit docking attachments snub the payload to a properly indexed position. Otherwise, indexing could be performed by the service unit to seek a known detent position. The position of the payload relative to the service unit must be known and controlled to effect a proper changeout of modules. This series of events, to remove and replace SRUs, is assumed to be preprogrammed in the service unit sequencer prior to launch. There is no uncertainty involved unless a verification signal fails to allow the full sequence of events to be executed. The manned interactive command override would then be necessary.

---

* For the purpose of this task the MSFC Full Capability Tug has been assumed as the upper stage. The results are applicable to other configurations as well.

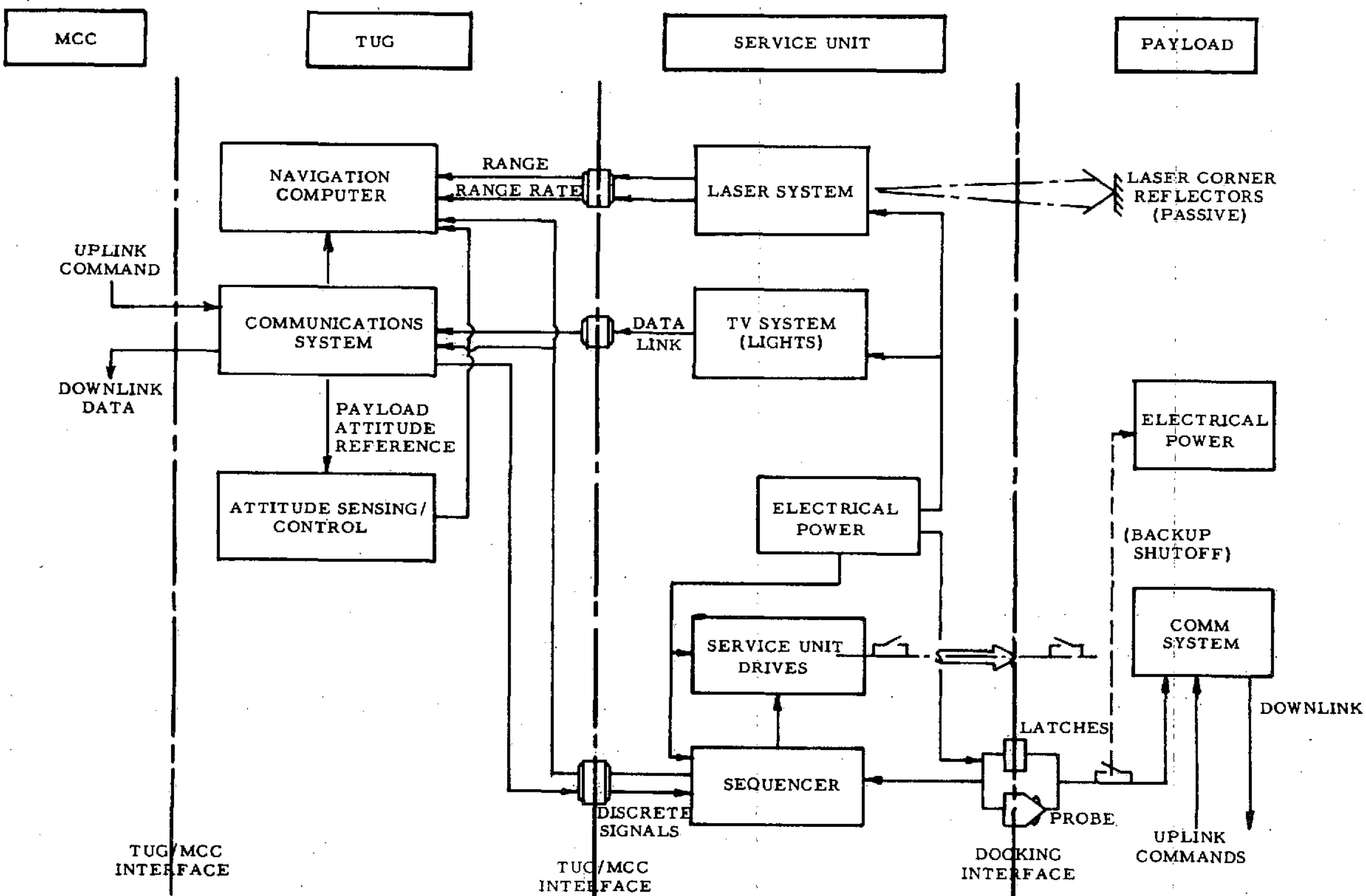


Figure 3-2. Space-Servicing Interface Schematic

SERVICE OPERATIONS					
NO	FUNCTION	SU	TUG	PL	MCC
1	COMPLETE ORBIT INSERTION		X		
2	VERIFY ACTION TO PL USER				X
3	PAYLOAD REORIENT COMMAND			X	
4	PAYLOAD RETRACT APPENDAGES			X	
5	REORIENT TUG FOR ACQUISITION		X		
6	INITIATE LASER SEARCH	X			
7	LOCK ON - INITIATE CLOSURE	X			
8	STANDOFF MANEUVER		X		
9	POWER DOWN PAYLOAD			X	
10	INSPECT PL DOCKING CONFIG.				X
11	AUTHORIZE FINAL CLOSURE				X
12	PERFORM FINAL CLOSURE		X		
13	DOCK WITH PAYLOAD		X		
14	SNUB DOCKING LATCHES	X			
15	VERIFY POSITIVE LATCH			X	
16	INITIATE SERVICE SEQUENCE	X			
17	SERVICE UNIT INDEXES TO DETENTE	X			
18	SU INDEXES FREE POSITION	X			
19	FAILED SRU REMOVED	X			
20	SU INDEXES TO REPLACEMENT	X			
21	SRU REPLACE ACTION	X			
22	RECYCLE AS NECESSARY	X			
23	SU VERIFY INTERFACES	X			
24	SU RELEASE DOCK LATCHES	X			
25	TUG REINITIALIZES NAV SYSTEM		X		
26	PERFORM STANDOFF MANEUVER		X		
27	POWER UP PAYLOAD			X	
28	EXTEND APPENDAGES			X	
29	PERFORM PL CHECKOUT			X	
30	AUTHORIZE NEXT MANEUVER				X
31	INITIATE SEQUENCE OF OPS	X			

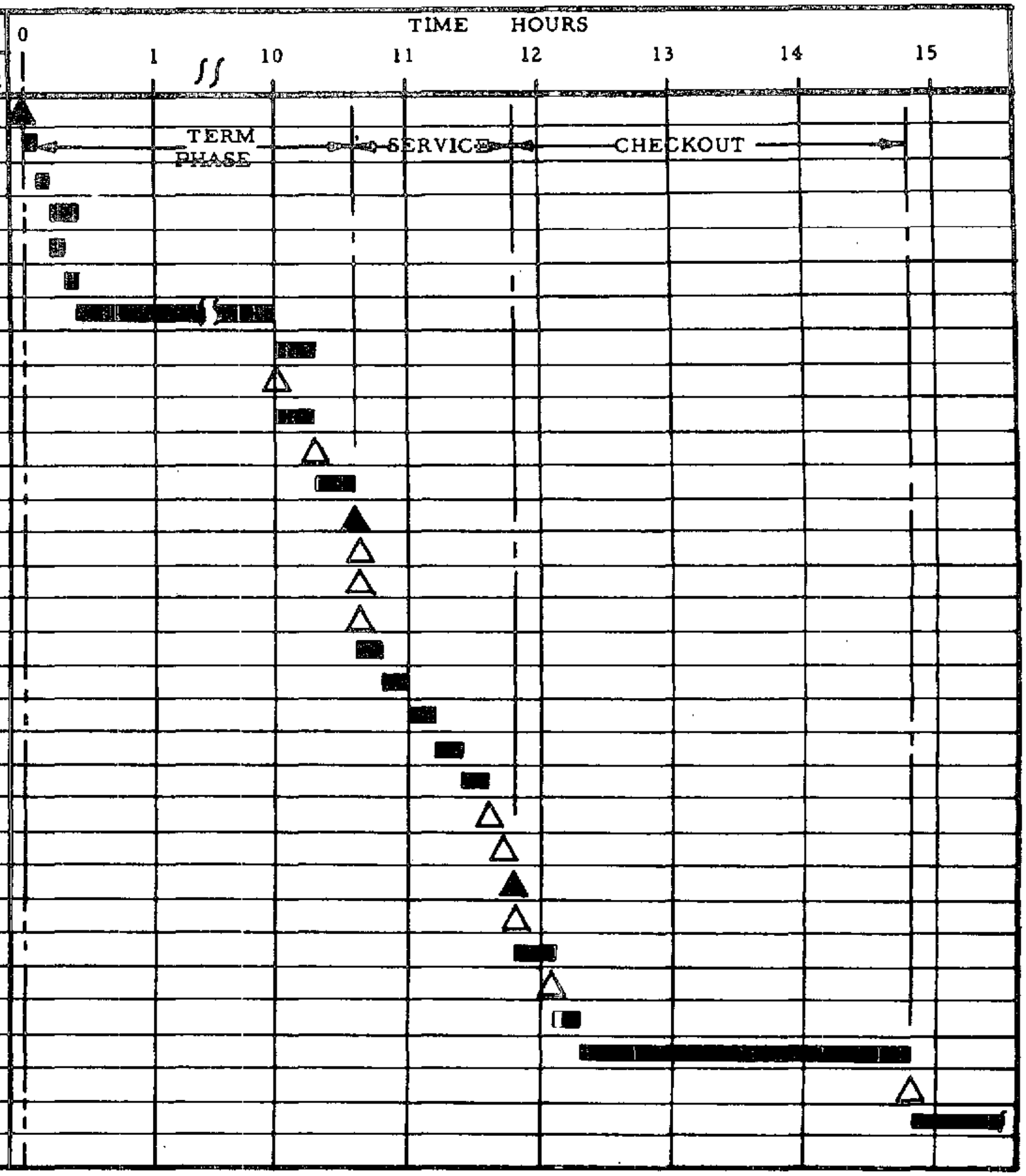


Figure 3-3. Rendezvous and Docking Sequence

After servicing one payload, the upper stage performs another standoff maneuver while payload checkout is performed by the payload user. To minimize the possibility of electrical shorts during servicing, the payload is powered down. After standoff has been achieved, the payload is again powered up and all retracted appendages returned to their operational position. If checkout is unsatisfactory, the payload user must decide the next course of action and relay this to the Shuttle operations control center. He may elect to recycle all events, call up other SRUs on the next servicing mission, or, if performance allows, return the payload for laboratory inspection. The payload user is the only agency capable of making these decisions, recalling that Shuttle operations are assumed to be a supporting role. In this way also, the upper stage airborne system is not complicated by checkout routines for various payloads.

Performance analyses have shown that the upper stage may be capable of servicing four or more payloads on a single flight, if the flight is limited to seven days. This will vary with the final selection of the stage configuration which is to be made sometime in the near future. For the purpose of this study, to assess software requirements, it is assumed that a capability exists to service up to four satellites in geosynchronous orbit. Therefore, it is necessary to provide an update capability for the upper stage guidance system.

This can easily be achieved in one of two ways. The ephemeris of each payload is always well known as a result of long periods of tracking. This ephemeris can be automatically assumed by the upper stage prior to initiating transfer to a second satellite position. All errors accumulated by the airborne system up through the servicing period would therefore be nulled. This would have a minimal impact on the airborne software. The same routines and functions would be employed for each maneuver, the only potential changes being limited to coefficients to reflect mass and inertia changes. Since these are seldom critical, they can be preprogrammed.

The alternative is to provide updates via tracking data through the upper stage command receiver. For high-altitude operations, the existing tracking networks provide sufficient coverage for this function. The TDRS would be employed for low-altitude operations. In either case, the spaceborne system must have the capability to receive, decode, read, and verify the input to the airborne computer. This requires a relatively small but not insignificant number of software instructions and, therefore, will be incorporated when estimating software requirements.

One final point remains to be established regarding manned-interactive support at the Shuttle Operations Center. A review of the upper stage functions indicates that for the most part, under normal conditions, all airborne functions can be automated. This builds upon current practice for Transtage, Centaur, and Agena operations, as examples. The upper stage guidance system is assumed to be sufficiently accurate to place the upper stage within laser acquisition range of the payload to be serviced. Except for the standoff inspection maneuver, all rendezvous and docking functions can be automated with relative ease and should not present any severe technical problems. However, the capability to automate onboard systems to respond to contingency situations could severely impact the overall software requirements. If, on the other hand, a high degree of monitoring and control is exercised from the ground, the operations control center would experience a severe software impact. This position was resolved by performing a top level contingency analysis, which is documented as Reference 1. A brief summary is provided below to indicate the influence on scoping the software problem.

The contingency analysis is based upon an analogy by fault tree application to the basic question, "What can cause space servicing of a payload to fail?" A search is then made to categorize all the paths leading to this failure condition. The failure paths can subsequently be traced to hardware elements, from which evolve safety design requirements. In this particular application, it is not necessary to define equipment design approaches, but to determine whether sufficient cause exists to justify manned-interactive operations. Secondly, if man is involved, to what degree is unique software support required?

Failure of the service mission can result from four unique events in the total sequence of operations:

a. The upper stage fails to complete rendezvous and docking.

b. Rendezvous and docking are completed successfully, but the service unit fails.

c. Servicing of the payload is completed, but the service unit/payload combination fails to undock.

d. During the approach, standoff, undock, or other programmed maneuvers, a catastrophic collision occurs between the upper stage and the payload.

In each instance, the failure condition may be brought about by failures in the upper stage, the service unit, or the payload. A typical breakdown of failure causes is given in Figure 3-4, showing payload failures that could result in a catastrophic collision with the upper stage. This is one of 12 such trees. The emphasis of this one in particular is that failures could occur which result in unknown obstructions in the path of approach of the upper stage, e.g., a pressure vessel fracture (mechanical failure) damaging a retraction mechanism. Indications to the payload user may erroneously indicate that a successful retraction has occurred. This is only pertinent if the appendage obstructs the docking path; hence the "and" gate requiring two conditions to exist simultaneously. These events can be broken down further for specific payloads (and upper stage designs) to arrive at reliability and safety criteria relative to redundancy and probability of occurrence.

It is sufficient for this effort to indicate that conditions could arise over the broad spectrum of payloads which would jeopardize the upper stage or the payload if completely automated servicing were conducted. Manned interaction is necessary to support contingencies, but there is no need to take an active role in the nominal servicing functions. This approach forms the basis of the following assumptions to be employed in sizing the software problem:

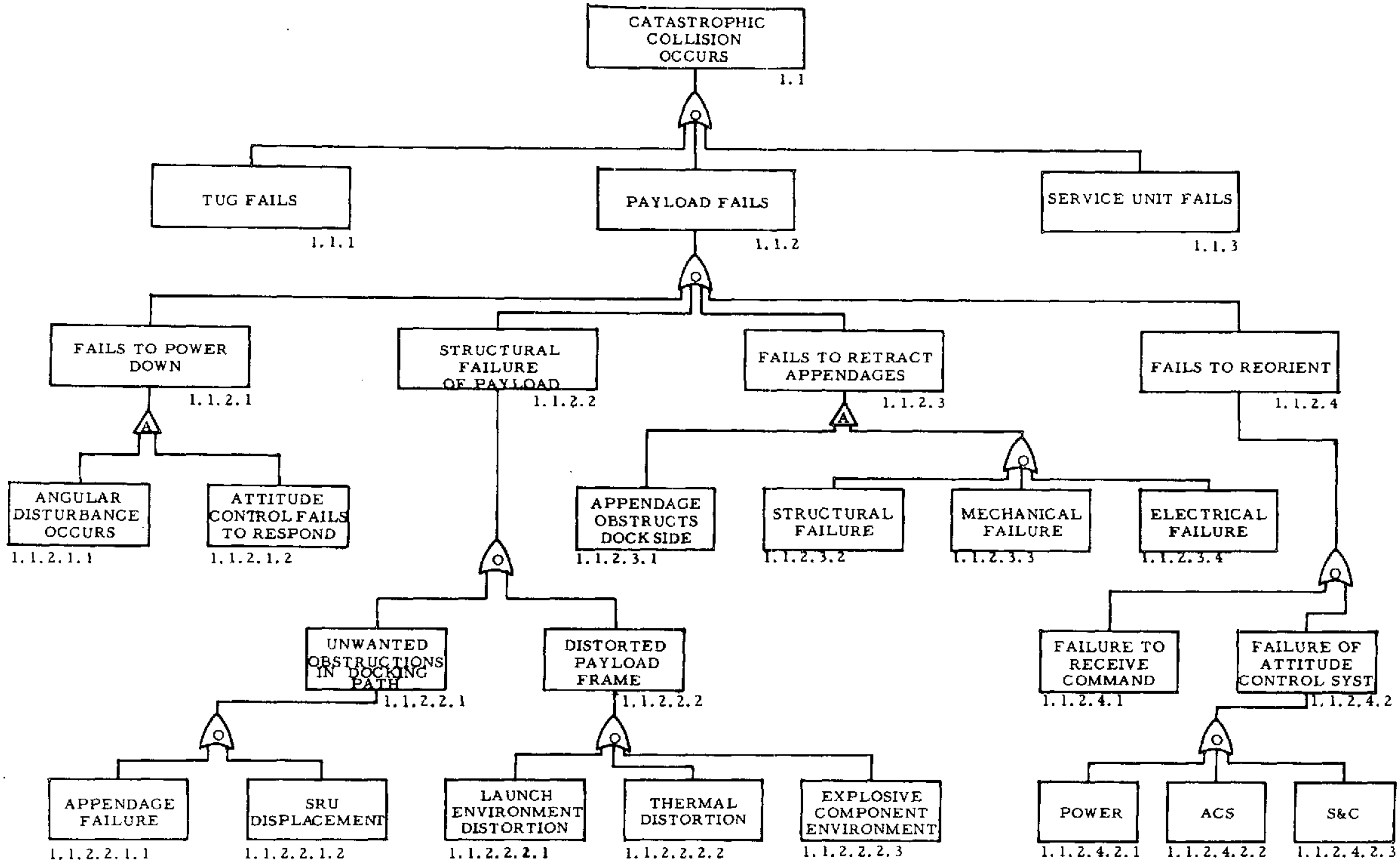


Figure 3-4. Contingency Analysis

a. All nominal functions of rendezvous, dock, SRU replacement, and undock are performed in an automated manner.

b. Inspection, command override, backout initiation, guidance updates, and other go/no-go decisions are made by manned interactive support at the control center.

c. The Operations Control Center has a unique monitoring capability only as required for go/no-go decisions and is an adjunct to Shuttle operations, so that fundamental software support and other interface information is readily available.

This completes the definition of the operations concept, ground rules and assumptions, except for ground checkout. It has been assumed that ground operations at the NASA Kennedy Space Center (KSC) will incorporate a system similar to the Launch Processing System (LPS) defined in Reference 2. Therefore, only those functions related to the upper stage/service unit checkout and launch preparation will be considered. All other support functions (i.e., fueling, control alignment, etc.) are assumed to be available.

One final caution should be observed. The operational concept employed in these ground rules is based upon experience gained from existing USAF programs. In these programs, preprogrammed automated operations are repeatedly employed. For instance, orbital insertion of payloads requires no direct action by the mission control center after lift off. A different philosophy exists at the NASA Mission Control Center, wherein manned space operations have necessitated close support at the MCC. Consequently, altering the ground rules to abide by this concept could possibly affect the conclusions presented later in this report. If the shuttle upper stage is to require close support from the MCC then it is recommended that the software cost projections be reevaluated.

4. FUNCTIONAL ANALYSES

Only one mission has been analyzed in detail. This was mission No. 3, geosynchronous servicing of four payloads. An upper stage similar to the full capability Tug was assumed. The remaining missions are primarily subsets of this one case. Data from the current Titan IIIC program were used to establish functional breakdowns and commonality of computer operations. The functional analysis was carried to the third level to derive software requirements. These requirements are then integrated to arrive at a total software budget, recognizing that many functions are common and do not require separate and distinct programs. Conversations with NASA indicate that the Tug computer has been sized to approximately 50K words of instruction. This serves as a basis for comparison as the functional analysis progresses.

The software development process is shown schematically in Figure 4-1. There are three basic functions involved: spaceborne systems, flight support operations, and ground checkout operations. Other items, such as crew training simulators, etc., were not considered but may be substantial in the final analysis of overall Shuttle System operations. The development cycle for each area is somewhat repetitive. Preliminary system design is performed to arrive at a hardware definition. From this effort evolves a preliminary software budget for the vehicle computer, instrumentation, and flight support. As the design effort progresses, three separate software specifications evolve: the integrated spaceborne requirements, the instrumentation list, and the flight support software specification. Each area of development is iterative in nature requiring a constant reevaluation of software requirements versus implementation.

Coding of software normally begins after the software specification has been firmed up. Generally, from this point on, contractors feel qualified to estimate software costs. However, experience has shown that a significant amount of effort is required during the design phase to develop equations, interface relationships and sequences of operations. The results

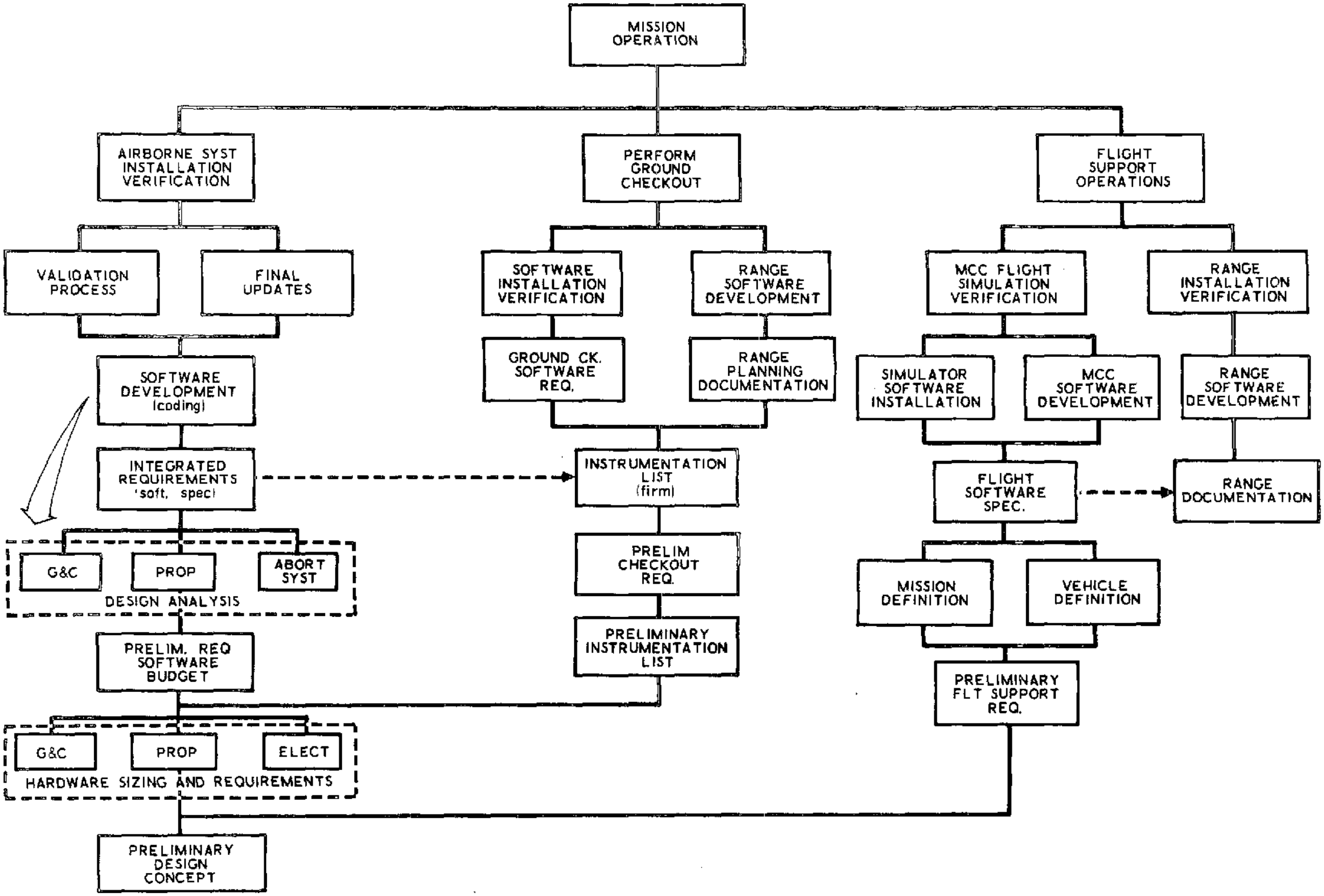


Figure 4-1. Single Mission Typical Software Development

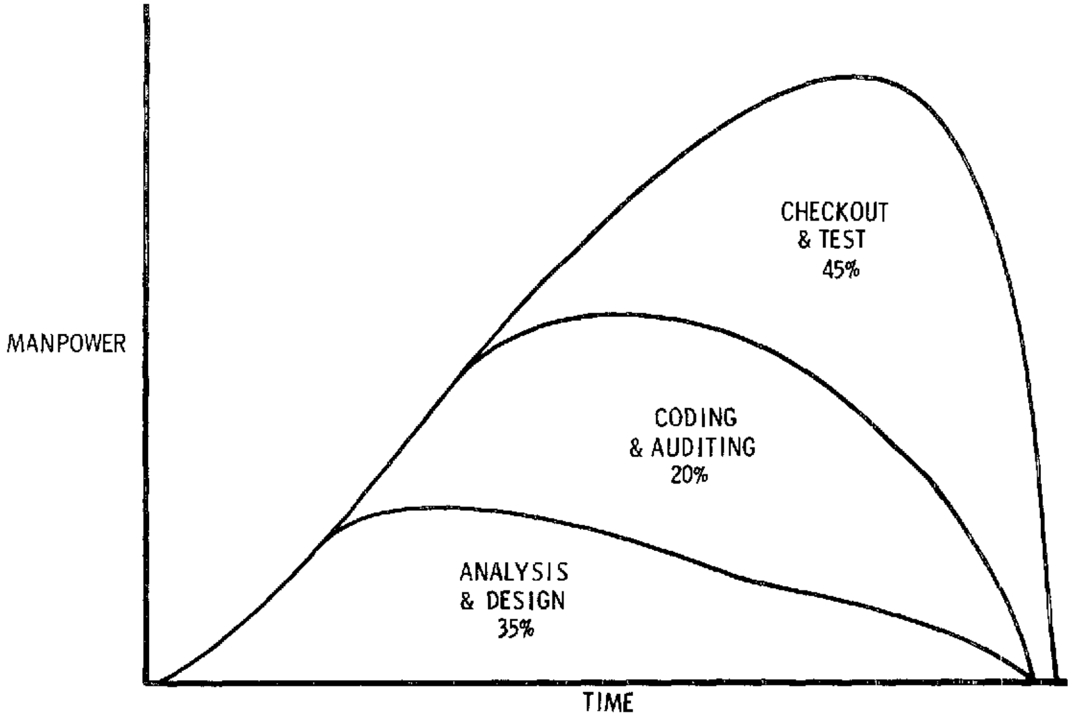
of one survey (Ref. 3) indicate that software development can be subdivided into three elements:

a. Analysis and design
b. Coding and auditing
c. Checkout, validation and testing

The results of the survey are reasonably consistent for the three applications tabulated in Figure 4-2. For the purposes of this analysis, it has been assumed that 35% of the effort will be devoted to analysis and design of the software algorithms, 20% to coding, and 45% to test and checkout. This allocation will in reality vary with each functional element considered in the reference mission analysis. However, for the purpose of the total software effort, the above distribution appears reasonable. It will be shown later that this breakdown of the software development cycle is necessary if the empirical factors for estimating software costs are to be employed.

The software requirements were derived from a functional analysis of mission three, geosynchronous servicing of up to four satellites at different longitude placements. The top level (Level 1) flow is shown in Figure 4-3. Each function is developed to the third level for the three software regions of interest: spaceborne, ground, and flight support software. A sample of this development is shown in Figure 4-4 for the function "Deploy Tug." At this point it is possible to estimate the software requirements to perform each function. The emphasis is primarily on those additional functions required to support space servicing.

The results of the functional analysis are provided as Appendix A and are summarized in Tables 4-1 and 4-2. Table 4-1 indicates the various software modules required to perform the identified list of upper stage functions. Table 4-2 provides a similar summary for ground checkout operations. The flight support software at the mission control center is more difficult to define and consequently is derived in a different manner, as explained later.



PROJECT	ANAL & DES	CODE & AUD	CHK & TEST
GEMINI	36	17	47
SAT V	32	24	44
APOLLO	22	26	52*

* HEAVY

Figure 4-2. Distribution of Software Effort

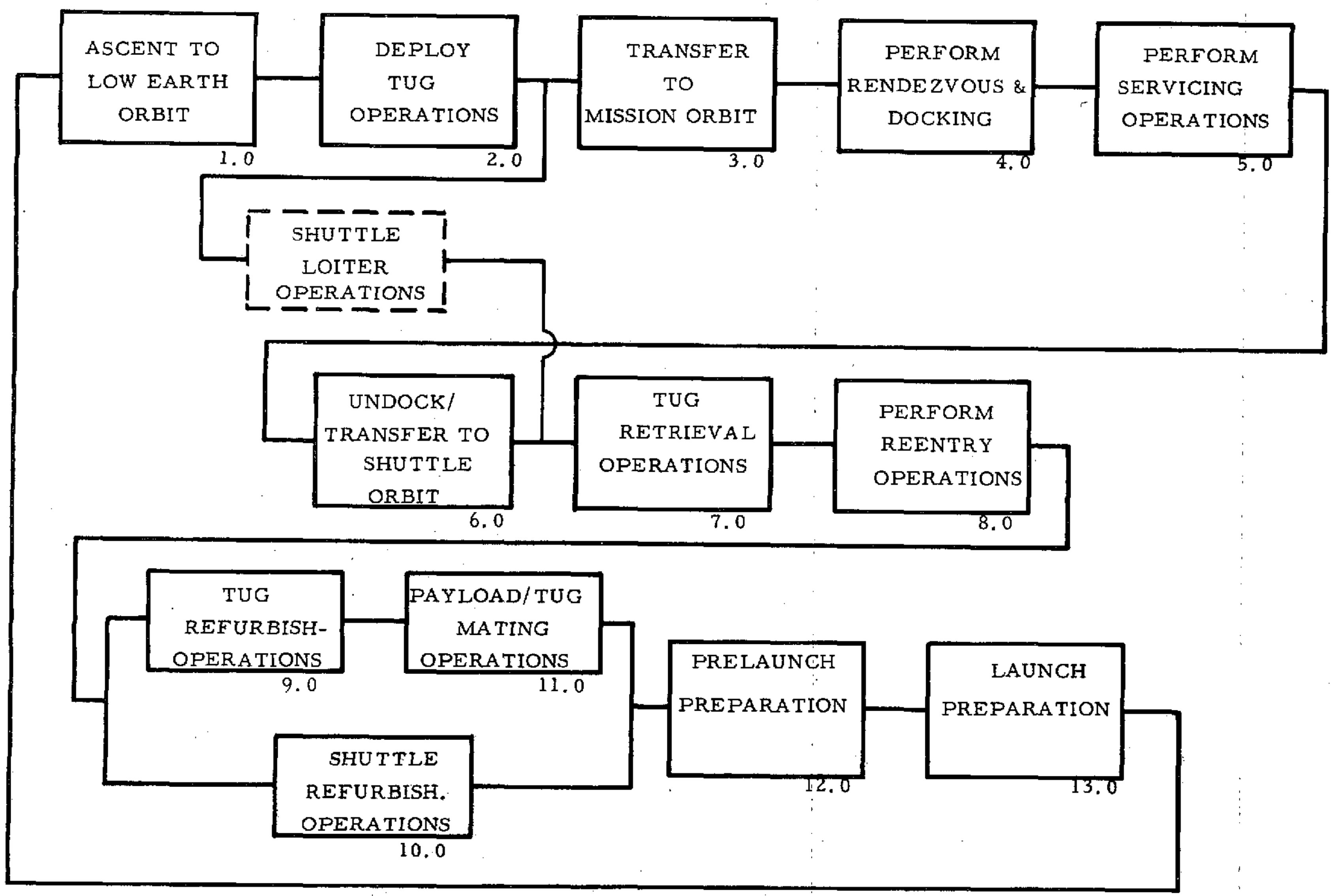


Figure 4-3. Top Level Functional Flow - Mission 3

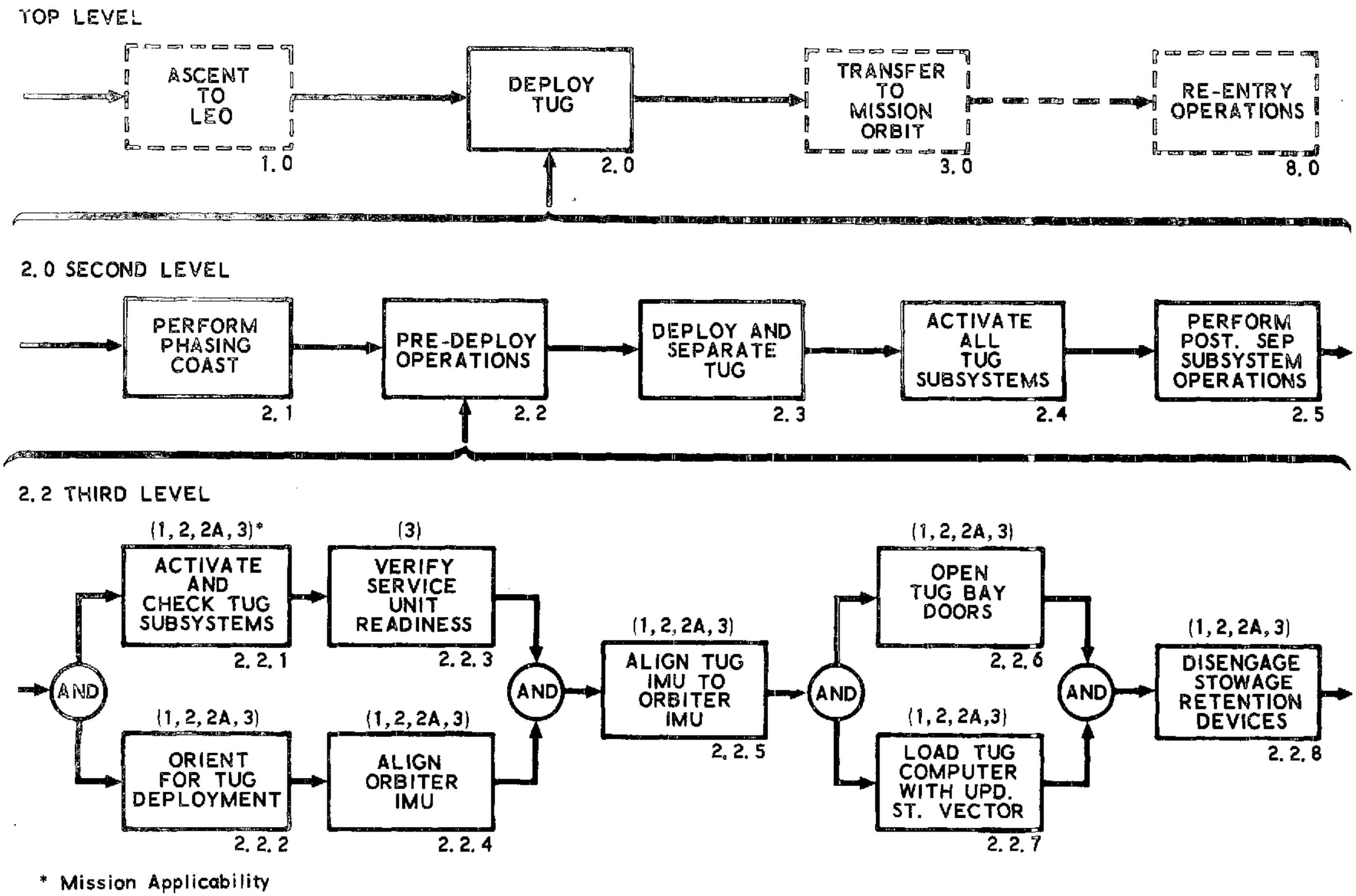


Figure 4-4. Sample Functional Flow Hierarchy

Table 4-1. Functional Flow Links to Computer Program Requirements

FUNCTIONAL FLOW ELEMENT	PROGRAM MODULE: EXECUTIVE	NAV (inertial) and (update)	GUIDANCE	DIGITAL AUTOPILOT (powered and coast)	T/M FORMATTING	PROP UTIL	COMMUN PROCES	STAR ALIGN	SEQUENCING (CMD, discretes)	O/B C/O	HAZ ANAL	HORIZ SENS PROC	REND GUID	DOCKING	SERV MGT	UTIL SUBROUT AND CONSTRAINTS
1.0 IMU ALIGNMENT - SYSTEM C/O	X	X			X		X		X	X						X
2.0 COMPUTER LOADING	X						X		X							X
3.0 APS ACTIVATION	X					X			X							X
4.0 RECEIVE AND EXECUTE MISSION ENABLE	X								X							X
5.0 MANEUVER TO AQUIRE HORIZON	X	X		X		X			X			X				X
6.0 COMPUTE BURN PARAMETERS	X		X			X			X							X
7.0 MANEUVER TO REQUIRED ATTITUDE	X	X		X					X							X
8.0 MAIN ENGINE BURN (2)	X	X	X			X			X	X						X
9.0 ACTIVATE DOCKING SUBSYSTEM	X								X					X		X
10.0 ORIENT FOR PAYLOAD DOCKING	X			X		X			X							X
11.0 ACQUIRE AND LOCK-ON TO PAYLOAD	X								X				X			X
12.0 DETERMINE RANGE AND RANGE RATE	X								X				X			X
13.0 DETERMINE RENDEZVOUS INTERCEPT MVRS	X								X				X			X
14.0 APS BURN	X	X	X			X			X				X			X
15.0 VERIFY PAYLOAD ORIENT. FOR DOCKING	X				X		X		X					X		X
16.0 DOCK WITH PAYLOAD	X								X					X		X
17.0 DESPIN DOCKING MECHANISM	X								X					X		X
18.0 CONNECT PAYLOAD UMBILICALS	X								X							X
19.0 CONDUCT PAYLOAD SERVICING	X								X						X	X
20.0 UNCOUPLE AND SEPARATE PAYLOAD	X								X							X
21.0 MANEUVER TO SAFE DISTANCE	X	X	X	X				X	X			X				X
22.0 CHECKOUT P/L AND TUG-GND LINK	X				X		X		X	X						X
23.0 REPEAT 8 TO 22 FOR 3 PAYLOADS	X								X							X
24.0 RECEIVE NAV UPDATE	X	X					X		X							X
25.0 RECEIVE AND STORE COMMAND SEQUENCE	X						X		X							X
26.0 MAIN ENGINE BURN (2)	X	X	X	X		X			X							X
27.0 ESTAB COMMUNICATION WITH ORBITER	X						X		X							X
28.0 TRANSFER FLT CONTROL OR ORBITER	X			X					X							X
29.0 DEACTIVATE AND SAFE MAIN PROP	X					X			X							X
30.0 VERIFY ALL SUBSYSTEMS SAFE	X								X			X				X
31.0 APS INHIBIT	X								X							X
32.0 STOW APPENDAGES	X								X							X
33.0 PASSIVATE SUBYSTEMS	X								X							X

Table 4-2. Ground Checkout Functions

FUNCTION		APPLICATION							
FFBD	TITLE	Process Control	Interface Stimulus	Program Prep.	Prelaunch Checks	Launch Checks	PRIMARY INSTR.	SECONDARY INSTR.	TOTAL
9.0	Tug Refurbishment								150,000
	• Safe Systems	X					35,000	5,000	
	• Process Maint. Data	X					100,000	-	
	• Prep. Service Unit		X	X			5,000	5,000	
11.0	Payload Tug/Mating Ops								335,000
	• Tug Interface Verify		X	X			50,000	15,000	
	• Service Unit Interface		X	X			50,000	20,000	
	• Joint Sim. Flight			X			150,000	50,000	
12.0	Prelaunch Preparation								195,000
	• Orbiter Interface Checks		X		X		15,000	10,000	
	• Comb. Sim. Flight				X		130,000	40,000	
13.0	Launch Preparation								235,000
	• Preflight Tug/SU Cks.					X	35,000	10,000	
	• Load Tug Computer			X			50,000	10,000	
	• Perform Countdown			X		X	100,000	30,000	
	TOTAL						720,000	195,000	915,000

- Primary Instructions are those required for a baseline reference mission
- Secondary Instructions are required to accommodate additional missions

The results of Table 4-1 have been integrated to arrive at an estimate of the total number of instructions required for the spaceborne computer. These results are shown in Table 4-3 for various levels of complexity. If the upper stage is employed for satellite deployment operations alone, it is estimated that 27, 000 words of instruction in machine language will be required. Increasing the time on orbit inherently increases the complexity of the navigation functions, raising this value to 35, 000 words of instruction. Incorporating a rendezvous and docking capability requires an additional 5500 words for a total of 40, 000 words of instruction. These values form the basic requirement for the upper stage spaceborne computer within the ground rules and constraints specified in Section 3.

It should also be recognized that for a large number of computer applications some savings in core storage can be achieved through "packing" words together. The above estimates have been based upon a 32-bit word length, however, in many instances an 8 or 16-bit word is adequate for an instruction. Therefore, a packing formula has been developed, based upon prior experience, to take advantage of this technique. By improved packing, the core storage can be reduced by approximately 40%. A further assumption is made relative to use of a higher order language (HOL). It is anticipated that the current trend toward higher order languages will continue, with the end result requiring a slight increase in storage requirements. Although the higher language improves the programmability, it inherently requires a modest increase in machine instructions (10%). The total memory size, based on a 32-bit word, is then estimated to be approximately 27, 000 words of instruction.

In addition, space servicing will require a modest increase over and above this value to accommodate discrete commands, the standoff maneuver, backout and recycle capability, and integration of the sequencer unit outputs. It is estimated that these functions would require no more than 1000 additional words of instruction. By comparison, the service

Table 4-3. Estimation of Servicing Software Cost (Spaceborne System)

PROGRAM MODULE	2 DAY DEPLOY ONLY	6 DAY AUTON. NAV.	6 DAY AUTON., REDEZ. DOCK
INFLIGHT EXECUTIVE	2500	3000	3000
NAVIGATION (INERTIAL)	1500	2500	2500
GUIDANCE	2000	2000	2000
DIGITAL AUTOPILOT (POWERED)	2000	2000	2000
DIGITAL AUTOPILOT (COAST)	1500	1500	1500
TELEMETRY FORMATTING	1500	1500	1500
PROPELLANT UTILIZATION	1000	1000	1000
COMMUNICATIONS	1500	1500	1500
STELLAR ALIGNMENT	3000	3000	3000
SEQUENCING	1500	2500	3000
ON BOARD CHECKOUT	4000	5000	6000
HAZARD ANALYSIS	2500	3000	3500
NAVIGATION UPDATE	1000	2000	2000
HORIZON SENSOR (INCLUDES FILTER)	N/A	2500	2500
RENDEZVOUS GUIDANCE/TARGETTING	N/A	N/A	3000
DOCKING PROGRAM	N/A	N/A	2000
UTILITY SUBROUTINES AND CONSTANTS	1500	2000	2500
TOTAL (T)	27,000	35,000	40,500
SHORT INSTRUCTION PACKING $T/5 + (4/5)\,T/2 = 0.6T$	16,200	21,000	24,300
HOL INCREASE (10%)	1,620	2,100	2,430
MEMORY SIZE (32 BIT WORDS)	17,800	23,100	26,730

⊙ INCREASE DUE TO SERVICING FUNCTIONS

/ TUG AIRBORNE COMPUTER ------------ 1000 INSTRUCTIONS

/ SERVICE UNIT SEQUENCER ------------ 2000 INSTRUCTIONS unit sequencer should require no more than 2000 words of instruction. These very modest increases are based upon the ground rule that the servicing functions are primarily discrete on-off signals for removing and replacing SRUs. All checkout functions are left to the payload user.

Ground support system requirements are considerably more difficult to estimate. The ground operations functions consist of: (1) payload, upper stage, and service unit preparation and interface verification, (2) prelaunch checkout of the service unit, and (3) post-landing safing and SRU removal operations. Installation and verification of the upper stage computer program and the service unit sequencer functions are also required. The total number of software instructions is placed between 800,000 and 1,200,000 words. The foundation for estimating the ground system software is based upon the NASA concept of a Launch Processing System (LPS) as defined in Reference 2. The LPS concept alters the historical approach of ground systems from special purpose hardwired consoles to a centralized computer control center for management of all launch site functions. In consequence, the impact of upper stage checkout and service unit installation should be minimal relative to the overall software requirement, representing less than 10% of the total estimated software effort associated with the LPS concept. This value is in reasonable agreement with the Centaur experience (Appendix B) which has one of the few nearly automated checkout systems.

Mission control center operations are even less well defined than ground support operations. The functions to be performed can be identified, such as navigation updates, sequencer override, and if necessary, subsystem status in addition to visual monitoring. The entire data stream must be decommutated, formatted and stored for display callup. These functions are, for the most part, also required for Shuttle support and Tug operations independent of servicing. The servicing functions will draw upon a repository of available programs for support with a minimal amount of new software required. For this reason, it was not possible to scope the mission control center software requirements with any degree of certainty and an alternate approach was employed for the purpose of this study.

It was determined, through conversations with IBM personnel under contract to NASA Jet Propulsion Laboratory, that mission control software in support of deep space probes and test programs required between 20,000 and 30,000 new words of instruction over and above the existing software system. This is also in reasonable agreement with the USAF Satellite Control Facility experience for introducing new programs into the software system. This value therefore appears to be reasonable for upper stage support over and above the functions normally required for Shuttle operations.

In summary, the total software requirements in terms of words of instructions, to achieve an operational capability to perform space servicing are:

Item	Tug	Service Unit
Spaceborne Software	30,000	2,000
Ground Checkout/Support	1×10^6	100,000
Mission Control/Flight Support	30,000	5,000

Recurring software costs are anticipated to be relatively low by comparison with such programs as Apollo, Centaur or Titan IIIC. For these programs, each mission was essentially unique, and as such, required modification of the basic software programs. The Shuttle system, by virtue of its flexibility to accommodate numerous mission operations, should not require any extensive recurring software effort. The capability to support a wide variety of operations is inherent in the basic system development. Some effort will be involved in routine validation of software program constants prior to each flight and undoubtedly minor algorithm changes will occur from time to time. The basis for estimating recurring costs of software will, therefore, be based upon prior history from the Titan IIIC and Centaur Programs (Appendix B). This point is discussed further in the next section.

5. SOFTWARE COSTING SURVEY

A survey of several software development firms was performed in an effort to establish a cost basis for the requirements presented in the previous section. There exists within The Aerospace Corporation a certain level of experience in cost estimation of software programs. However, although program costs can be identified, it is extremely difficult to relate these costs to an initial set of requirements. Therefore, there exists a great deal of uncertainty in cost estimation relationships for software. For this reason, it is suggested at the outset that software costs should be considered within a band of reasonability subject to numerous factors which influence the final estimate. In the end, it is primarily a matter of judgment in estimating software costs.

The approach selected is to first provide the background data existing within The Aerospace Corporation, including published and unpublished results. The uncertainties in this data will be pointed out and where possible related to other reference information. This can then be related to data obtained from the MIT Draper Laboratory for the Apollo program. These results are then considered in light of the primary factors associated with software cost uncertainties. With this background, it is then possible to estimate the upper stage DDT&E and recurring software costs for space servicing operations.

After attempting several different approaches, it was found that words of instruction in machine language provide a reasonable basis for estimating software costs. The first set of data is derived from an Aerospace study in 1973 (Ref. 4) with the results of this effort summarized in Figure 5-1. Man-months of effort are plotted against the number of instructions as a point of reference. The man-months provided relate to the total effort involved including design, coding, and testing of the software product. However, in some cases the reference points shown represent only part of the known program effort for which documentation can be found. For instance, the Saturn V airborne system requires

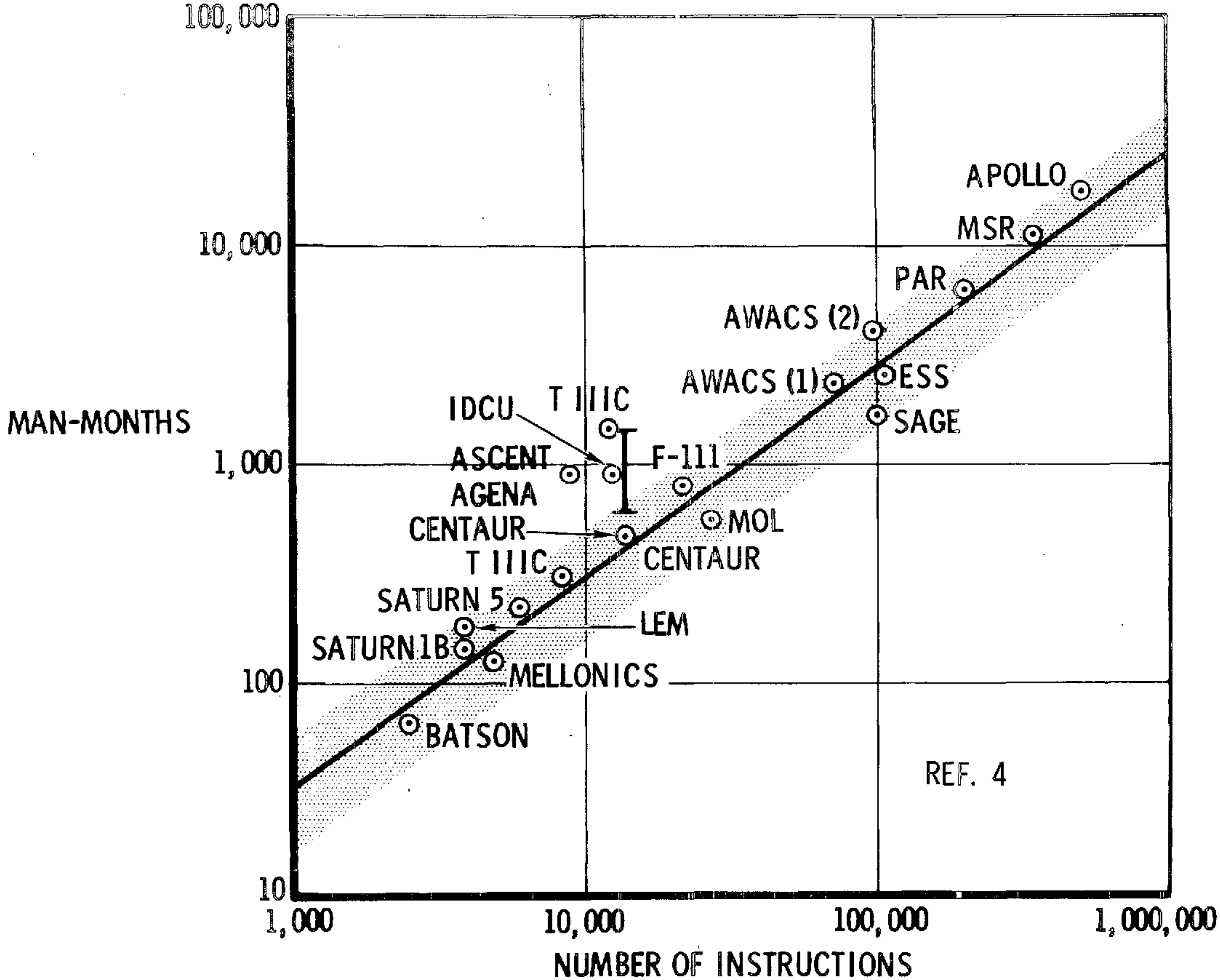


Figure 5-1. Software Costs

considerably more than 7000 words, however, the only point of reference for which cost has been provided is this particular increment. This data was originally derived from a System Development Corporation survey (Ref. 5).

It should also be noted that early Titan IIIC data indicated 400 man-months of effort for 8000 words of instruction. This data point provides a reference point for the Cost Estimating Relationship (CER) but fails to incorporate the total program costs. The total costs are better represented by the band of 800 to 1200 man-months for 12,000 to 13,000 words of instruction. This estimate reflects the fact that up to four different contractors were involved in this effort and, therefore, the total cost is more representative of the cost of the software development. In any program of this type, a significant cost will always be associated with integration of all the contractor efforts. The same will be true with the upper stage, and although NASA may perform the integration role, there still will be an associated cost. In the case of the Titan IIIC example, the original software development performed by The Martin Marietta Corporation was estimated to be approximately two million dollars. The inclusion of associated contractor efforts (Delco, Logicon, Aerospace) raises this cost to approximately five million dollars (1200 man-months).

Another important point to make is that often NASA or other agencies will inherently pay for software development which is never used. Programs will be coded, checked out, and subsequently discarded because the initial requirements are no longer valid. A point of reference is provided by the Apollo program. The software development cycle is shown in Figure 5-2 as provided by the MIT Draper Laboratory. Their records (Ref. 6) indicate that approximately 160,000 words of instruction were developed for the Apollo program through the first lunar landing. This was estimated to be approximately $45 million dollars for software alone, or $280 per word of instruction. However, MIT developed, coded, tested

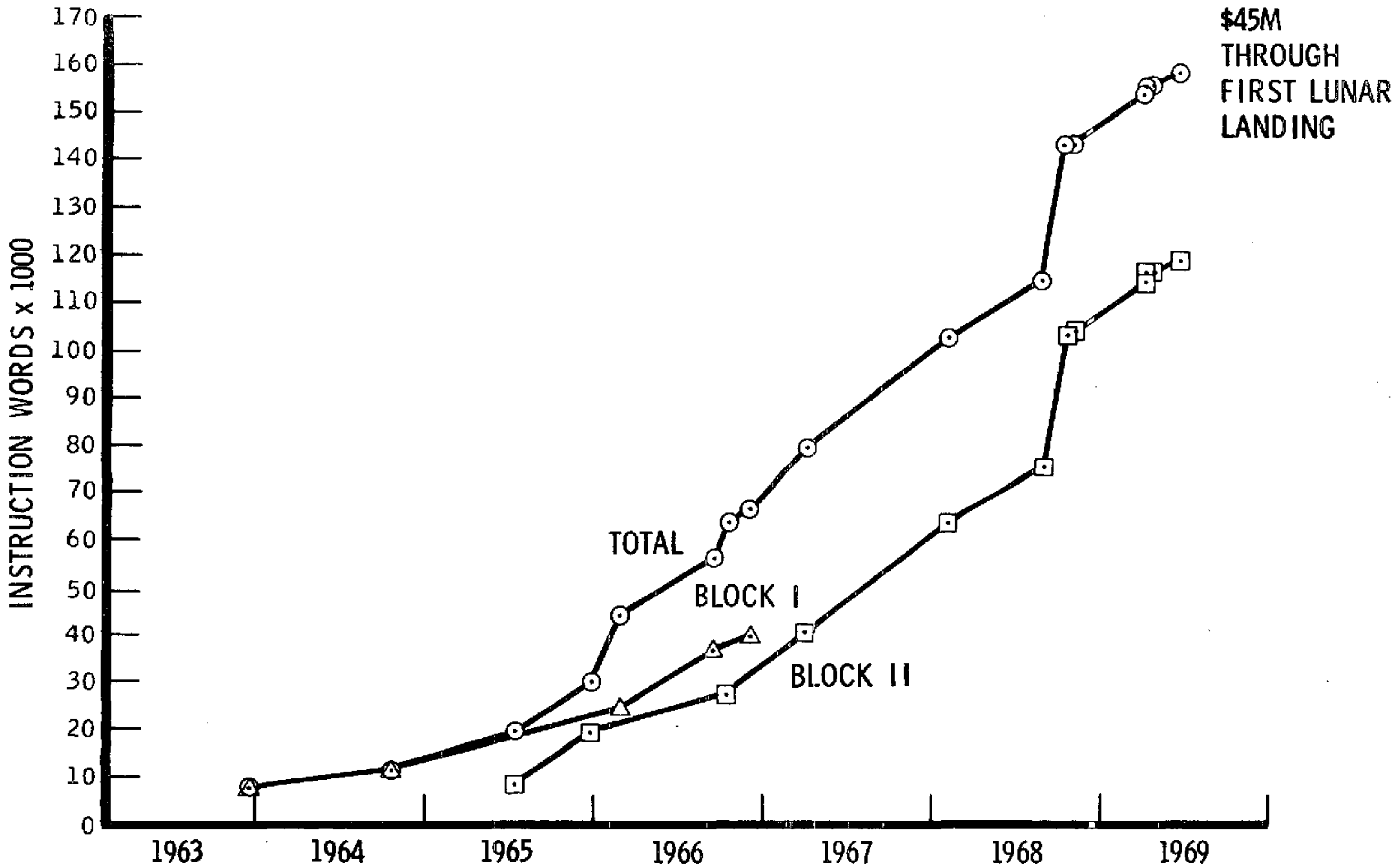


Figure 5-2. Apollo Software Development

and delivered over 500, 000 words of instruction (Ref. 7) to NASA for Apollo. Because of changing specifications, new ideas for various algorithms and computer modifications, there was a large portion of developed software that was never used in an operational sense.

This then becomes a better reference for estimating the cost per word of instruction. The ratio then becomes approximately $90 per word of instruction. This point lies remarkably close to the CER curve of Figure 5-1. It appears reasonable to expect the cost per word of instruction to decrease as the program size increases. The Apollo program was somewhat unique in that the original specifications called for a 4000-word memory capacity. This was subsequently modified, but computer capacity was a continual problem during the entire development cycle. For large programs of this type, it should be expected that the average cost of software would be relatively low. In the lower range of programs, the ratio is between $200 and $300 per word of instruction, depending upon the degree of integration involved.

The cost relationship of Figure 5-1 cannot be used alone without further consideration. Because of the uncertainties associated with nearly all software development programs, it is necessary to define additional factors influencing software cost and provide some judgment in arriving at an estimate for the Shuttle upper stage. As a result of discussions with numerous contractors, the following factors were determined to be representative of the uncertainties associated with estimating software development costs based solely upon number of instructions:

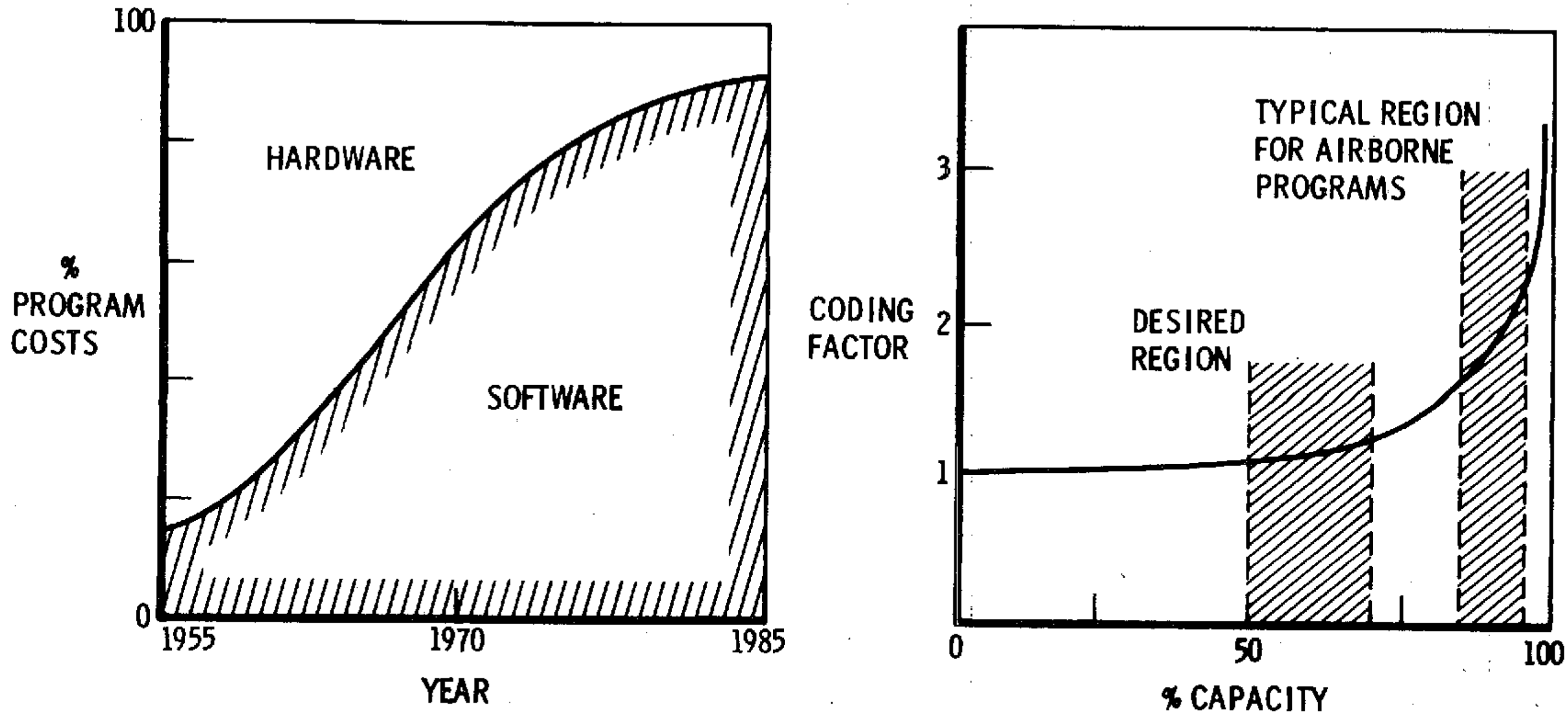
a. Computer capacity
b. Complexity of program
c. Prior experience or history
d. Language employed
e. Degree of integration effort required

The number of words of instruction is treated as the basic cost of developing software. If there are no hidden problems, it should be possible to estimate a word count for any series of functions to be performed. This has been done in the previous sections for upper stage operations involving space servicing. The basic cost of developing the spaceborne software is estimated to be 30,000 words of instruction, requiring 892 man-months of effort. If a man-month is taken at a cost of $4000, the cost of developing the basic software will be $3.56 million dollars. Manpower costs will inherently rise with time; hence, it may be desirable to increase this value to reflect costs in the 1980 time period. However, because the magnitude of other uncertainties appears to be considerably more significant, the inflationary effect of man-month charges is neglected for the time being.

The additional effort to accommodate space servicing functions is estimated to be 92 man-months with an associated cost of $0.37 million dollars. This then provides an unadjusted basic software development cost of $3.95 million dollars. Taken as a ratio for a point of reference, this provides $123/word of instruction. However, the remaining factors will have a substantial impact on these values.

Computer capacity has been recognized for many years as a major factor in software cost overruns. In past programs, the major concern, in terms of cost and weight, has been the spaceborne computer. Therefore, rigid controls were placed on the computer design long before the software had been sized correctly. This invariably led to software programs which exceeded the computer capacity requiring various overlay procedures, reprogramming and redesign of the software to remain within the hardware constraints.

In the future, this trend should be reversed. Hardware costs (relative to the same performance) have been and should decrease for some time to come. Reference 3 points out that in the 1980 to 1985 time period the software costs of an operational system will be three to four times the hardware costs. This is illustrated in Figure 5-3. The



REF: RAND REPORT
RM-6213-PR
JAN 1970

Figure 5-3. Computer Capacity Impact on Software Costs principal message is that computer capacity in terms of weight and cost should no longer be allowed to constrain the software because the system cost will favor large excesses in capacity to minimize the chance of placing constraints on the software.

In the same figure, another curve is presented from Reference 3. There are no historical data points to verify the shape of this curve, but the fundamental characteristic is accepted by all the software firms surveyed. As the software needs approach the capacity limits of the computer, more and more work is required to code a set of requirements. It is not unreasonable to expect this cost to double or go even higher. Invariably the contractor response was to prefer computer sizing of approximately twice the software instruction count.

The Shuttle upper stage computer has not been selected as of this time, but estimates obtained from NASA Marshall Space Flight Center indicate a projected capacity of 50,000 words. The estimated software requirements from this study then represent a capacity utilization of 60%. Without further definition of the upper stage and the subsystem redundancy requirements, the only position to be taken at this time is that computer capacity will be sufficient to disregard a further increase in the coding effort. However, if future estimates result in a substantial increase in the capacity utilization, it is recommended that the curve of Figure 5-3 be used to account for the increased effort required. In this event, the factor would be applied to all three phases of software development. More effort would be required for design, coding, and certainly for test and validation.

The next parameter to be considered is the "complexity" factor. This is a highly subjective term but one recognized as important in estimating costs. When surveying various software firms, each indicated that his response to an RFP would depend a great deal on the "complexity" of the software and "prior experience" with the type of effort requested.

Since these are judgment factors, they have been ranked ranging from complex manned operations such as Apollo, to less complex automated satellite operations.

Manned systems tend to require very flexible programs to accommodate all identifiable contingencies, with numerous redundant paths and self-check capabilities. Automated satellite operations, similar to the USAF Satellite Control Facility, tend to be nontime critical with limitations imposed simply by the onboard computer capacity. Other programs, such as the Titan III-Centaur, are completely preprogrammed and many functions are time critical.

Upper stage operations, including servicing are judged to lie between the Apollo level of complexity and the Titan IIIC program. Many functions can and should be preprogrammed, similar to the Titan system, but the capability must exist for manned interactive support thereby increasing the level of complexity somewhat. For the purpose of this effort, it is estimated that upper stage software requirements will be 50% more complex than the Titan IIIC system. Since the Centaur data point lies near the curve of Figure 5-1, this appears to be a reasonable point of reference to correlate the complexity parameter.

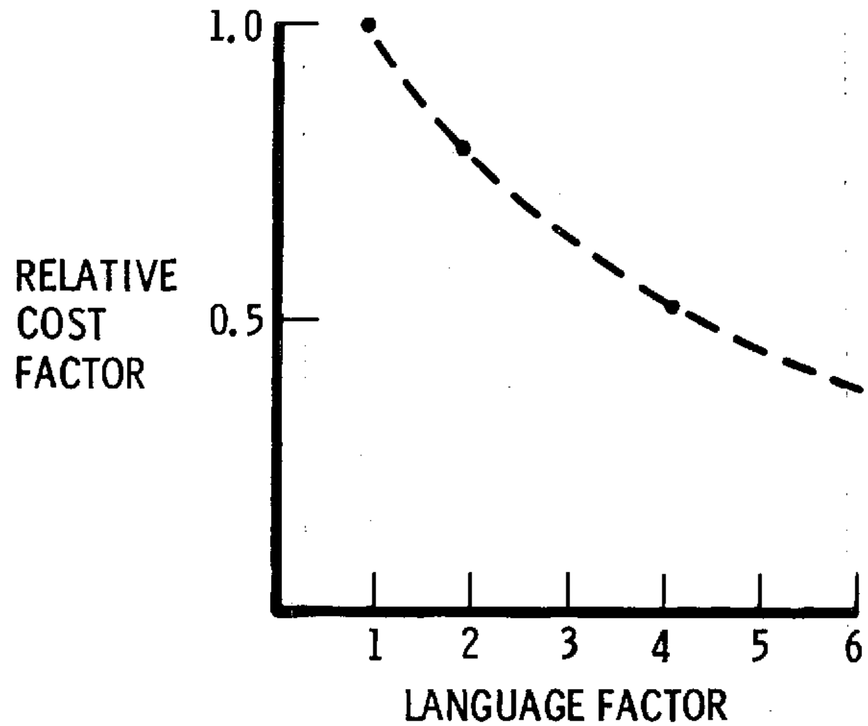
In addition, most contractors in the software field are considered to be competent to develop software for the Shuttle upper stage. Most of these have some degree of "prior" experience from which they can respond with confidence. For example, the carryover of the Atlas Centaur software experience to the Titan III Centaur is estimated to be as high as 80% utilization of prior efforts. However, since space servicing represents an increase in the operational dimensions of the upper stage, it appears prudent to provide some extra margin for lack of prior experience in this particular area. A 20% factor over and above the basic word count has been selected. It also appears reasonable that this prior experience factor would be reflected in all three phases of software development: design, coding and testing.

The next parameter of interest is the selection of a higher order language for coding the software requirements. Higher order languages are inherent in future programs and the question is not whether there will be one but rather what type will be selected. These languages will in general always result in a larger core storage, as implied in Table 4-3. However, this effect is more than compensated by the reduction in software design effort to implement and to validate the coding effort. Fortran itself represents a form of higher order language as compared to machine language programming employed in early applications. Estimates vary considerably but, for example, one word in Fortran can be equated to two or more words of machine code. Jovial is estimated on the average to result in a four-to-one ratio. Figure 5-4 provides an estimate of the coding factor for various languages and then relates this to an overall relative cost factor. The general feeling among contractors was that to expect a reduction in manpower beyond 50% would be unrealistic, no matter what language is assumed. The exact shape of the relationship is unknown but experience indicates it should be somewhat similar to that of Figure 5-4. For the purpose of this analysis, it was assumed that a higher order language would be employed [similar to Houston Assembly Language (HAL)] and that a 30% reduction in the software coding and validation could be realized.

Finally, it is necessary to consider the problem of software integration. If the software requirements are firm and if only one contractor is involved with the software development, then this effect should be minimal. However, again the historical experience has proven otherwise for programs of the magnitude of Titan IIIC or the Shuttle upper stage. In attempting to place some value on the impact of integration, consideration was given to the two efforts which bound the upper stage development, the Titan IIIC and the Apollo programs. In the Apollo program, MIT performed the integration role for the airborne system working with other contractors and NASA to develop specifications, consider hardware problems, and

- HIGHER ORDER LANGUAGE SHOULD REDUCE SOFTWARE CODING EFFORT
  / ASSUMES DEVELOPED COMPILER EXISTS

LANGUAGE	FACTOR
MACHINE	1:1
FORTRAN	2:1
JOVIAL	4:1
GOAL	4:1
HPL	10:1



- ASSUMES NO CONSTRAINTS DUE TO MACHINE CAPACITY
  / HOL INCREASES WORD COUNT
  / HOL REDUCES COST PER MACHINE INSTRUCTION
  - APPLICABLE TO CODING EFFORT ONLY

Figure 5-4. Software Language

resolve conflicts. As time passed, NASA assumed more and more of this responsibility but the effect is the same. On any program of this complexity and magnitude, some allowance in software development costs must be made to account for integration of subelements into a composite program. It is estimated that this effort was in the neighborhood of 50% of the basic software development cost.

The Titan IIIC program also required a sizable integration effort, but since the basic effort was smaller than Apollo, the ratio of total cost to that of the prime contractor is higher. The addition of Delco, Logicon, and Aerospace efforts to those of the Martin Marietta Corporation (a prime contractor) resulted in software costs rising from approximately $2.5 million dollars to $5 million dollars. This is not to imply that the costs were not justified. The effort required to integrate all elements of the software (including validation, testing, installation, etc.) is sizable and although it does not of itself produce code, it is a cost which must be recognized and accounted for. Hence, for the Titan IIIC program, the ratio of the total cost to that of the basic program cost is a factor of two.

The Shuttle upper stage will also require a great deal of integration effort whether performed by NASA or a contractor. In either event, the costs will be reflected against the software development. This integration charge will be reflected in all phases of development, although it could be heaviest in the test and validation areas. Therefore, for the purpose of estimating software costs, it will be assumed that the upper stage will have approximately the same level of integration effort as the Titan IIIC program. An allowance of 100% above the basic software development will therefore be employed.

The overall effect of these parameters is significant. To summarize, the basic spaceborne software development cost was estimated to be

$3.93 million dollars. Adjustments for other influencing parameters are summarized as:

Computer capacity	No impact
Complexity factor	50% allowance
Prior experience	20% allowance
Language	30% reduction
Integration	100% allowance

The adjusted spaceborne software cost is then estimated, on the basis of $4000 per man-month, to be as given below:

Total estimated cost ($4000/MM)	Basic	Adjusted
Without servicing (892 MM)	$3.56M	$8.98M
Servicing increment (92 MM)	$0.37M	$0.93M
Total	$3.93M	$9.91M

A similar approach may be taken for estimating ground checkout and support program costs. Although the same variables influence the overall cost, there is less definition of the impact due to the lack of historical data. There are very few systems which have employed any substantial amount of computerized control. The only data developed from the survey is presented in Figure 5-5. This figure provides data points for both ground checkout and mission control center programs. The Centaur data is developed from Appendix B. The Agena data is derived from conversations with Lockheed Missile and Space Company (LMSC) and Aerospace Corporation personnel. The referenced JPL data is derived by conversations with IBM personnel (the principal developer of the programs).

The Centaur program is the only one that approaches automated checkout and even this falls short of what is planned for the Shuttle upper stage. The Agena data is actually a composite or integrated effect of

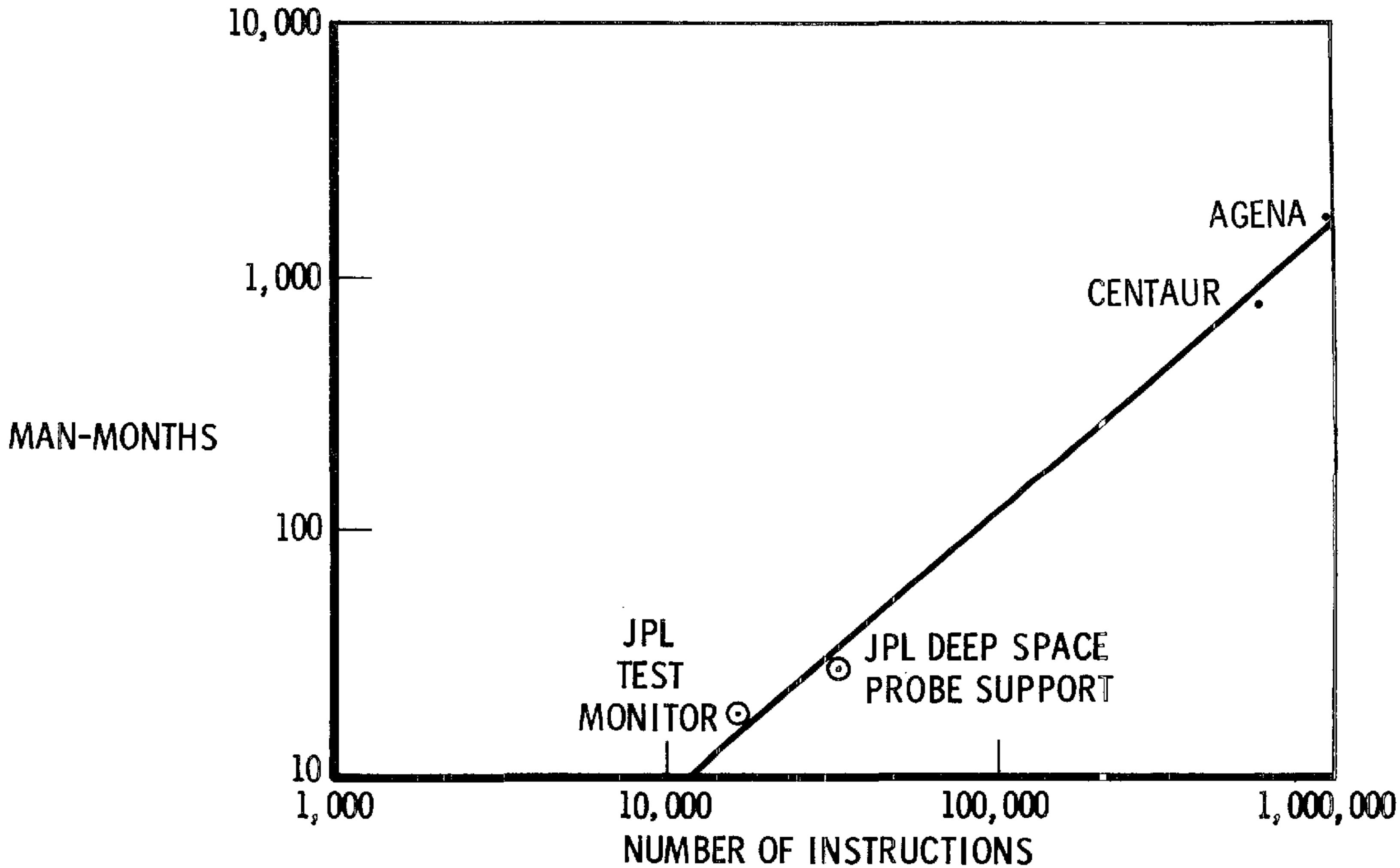


Figure 5-5. Ground Support Software

several years of development, modification, and redevelopment, with an integrated output of approximately one million words of code. This is in the range of what will be required for the upper stage, as explained in the previous section. The average cost is approximately seven dollars per word of instruction, considerably less than for spaceborne systems.

The total cost for the ground checkout and support system is estimated to be $7.2 million dollars, based upon 1800 man-months at $4000 per man-month. For the purpose of this analysis and because the degree of integration and other factors are not readily definable, this value has been increased to $10 per word, giving approximately a 30% margin. The upper stage ground software cost is then placed at approximately $10 million dollars. Space servicing systems will require some support but it should be relatively minimal. It is estimated to be in the range of 10% of the basic requirement or one million dollars. The total for ground support software is then $11 million dollars.

Flight support software costs are even more vague in that much of the software must already exist for Shuttle operations. Again, a cost of $10 per word of instruction (to be conservative) appears reasonable in the region of interest of Figure 5-5. The flight support software is then estimated to cost $300,000. An additional increment is provided to accommodate a command and control uplink/downlink of 5000 words. Total flight support software development costs then become $350,000.

Recurring software costs represent a new dimension altogether. Recurring costs are real but seldom are recorded. The principal reason is that there is no way to define the requirements for new or modified programs. It is generally assumed that the original program development was complete and therefore recurring support is not required. What little experience that exists indicates otherwise. The Titan IIIC program appears to be a reasonable example. Each time a new mission is developed (nearly each flight), it becomes necessary to develop new guidance coefficients, new discrete profiles, and different timelines. These changes must be verified, installed, and checked. On the average, this is estimated to cost approximately $100,000 for each new mission definition. The data of Appendix B indicates Centaur follow-on flight software costs to be lower than this, approximately $26,000 to $53,000.

If a conservative value of $100,000 is employed for an estimated 10 upper stage flights per year, the recurring mission software cost would be one million dollars annually. This should serve as an upper bound, since existing studies show an average of six to eight flights per year, and the costs should be more in line with Centaur operations. Assuming a cost associated with space-servicing to be 10% of the nominal recurring cost, it is possible to arrive at the total software cost estimate provided in Table 5-1. Note values have been supplied for recurring flight support software costs representing a lower bound or threshold cost of support. Some cost will be accrued but the value should be low enough to be negligible for this analysis.

In the search for supporting data relative to Mission Control Center operations, it was possible to obtain actual records from IBM at Houston. This record provides, by the month, the actual man power charges for Apollo and Skylab programs from inseption to phase out. The records also provide computer operating hours. This data is very helpful in evaluating the general character of support required for large complex programs but there is no way to relate these costs to a reference set of initial requirements. For this reason, it is difficult to extrapolate this data to future requirements. However, because it does represent one of the few sources of firm data for large programs, it has been incorporated into this report as Appendix C to serve as a reference for any future work on software costs.

Table 5-1. Software Costs Summary

TYPE	DDT&E $M		*RECURRING $M/YR	
	TUG	SERV	TUG	SERV
SPACEBORNE	8.98	0.93	1.00	0.10
GROUND SUPPORT	10.00	1.00	1.00	0.10
FLIGHT SUPPORT	0.30	0.05	0.10**	0.05**
TOTAL	19.28	1.98	2.10	0.25
	$21.26		$2.35/YR	

*ESTIMATED FROM T-III AND SCF EXPERIENCE

**ESTIMATED THRESHOLD VALUES

6. SUMMARY AND CONCLUSIONS

In summary, it is to be expected that the software development costs for the Shuttle upper stage will be approximately $20 million dollars. Space servicing should account for a little over 10% of this value. Recurring software is estimated to be a little over two million dollars per year in support of a very active upper stage program.

However, a few final remarks are necessary to place these results in proper perspective. The software development costs for the upper stage are not insignificant; however, they do not appear to be unreasonably high either. Software costs should amount to no more than 5 or 10% of the total program DDT&E. This will, of course, depend to some extent on the final configuration selected. The degree of redundancy, flight support, and manned interactive participation will have a significant influence on these costs. However, probably the most important factors to be considered are the firmness of the software specification and the degree of integration required. The very size of the NASA organization and the inherent involvement of numerous Centers can easily lead to major problems in integrating all elements of the software. In addition, support for a large number of satellite programs will also pose severe problems in deriving a firm specification.

It should also be kept in mind that there are a large number of shortcomings with this analysis. When surveying contractors for data, a great deal of sympathy was received but little substantive data. In general, the contractors agreed that the factors employed in this analysis represent the real essence of the problem. The problem is in quantifying these parameters to provide some uniformity of results. Understandably, each contractor has developed methods of his own to estimate software costs, but these tend to be proprietary and probably have as many factors influencing their results as has this report. The cooperation of the contractors was very gratifying, as was support from within the NASA organization. Everyone agreed that there is a real need for improved cost estimating techniques.

It is also generally agreed that methods must be found to reduce future software development costs. New techniques need to be explored, higher order languages developed, and possibly structured programming employed. Programs are becoming so complex and so costly that unless these or other techniques are employed, the probability of achieving an operational program within any type of budget projections will be extremely remote.

REFERENCES

1. Study 2.1 Operations Analysis, Space Servicing Contingency Analysis, ATR-74(7341)-5, The Aerospace Corporation, El Segundo, California (to be published).

2. Launch Processing System Concept Document, Directorate of Design Engineering, KSC-DD-LPS-007, NASA John F. Kennedy Space Center (Revised 11 January 1974).

3. Some Information Processing Implications of Air Force Space Missions: 1970 - 1980), RM-6213-PR, The Rand Corporation, Santa Monica, California (January 1970).

4. Attack Assessment Software Development Cost Analysis, ATM-73(3085)-12, The Aerospace Corporation, El Segundo, California (25 June 1973).

5. SDC Technical Memoranda, TM-(L)-3405/002/00, Santa Monica, California, (7 March 1967).

6. Rankin, Daniel Allen, A Model of the Cost of Software Development for the Apollo Spacecraft Computer, Alfred P. Sloan School of Management, MIT, Cambridge, Massachusetts (June 1972).

7. Apollo Guidance Computer Words Through Apollo 11, Memo to N. Sears from J. Kernan, The Charles Stark Draper Laboratory, MIT, Cambridge, Massachusetts, (3 February 1971).

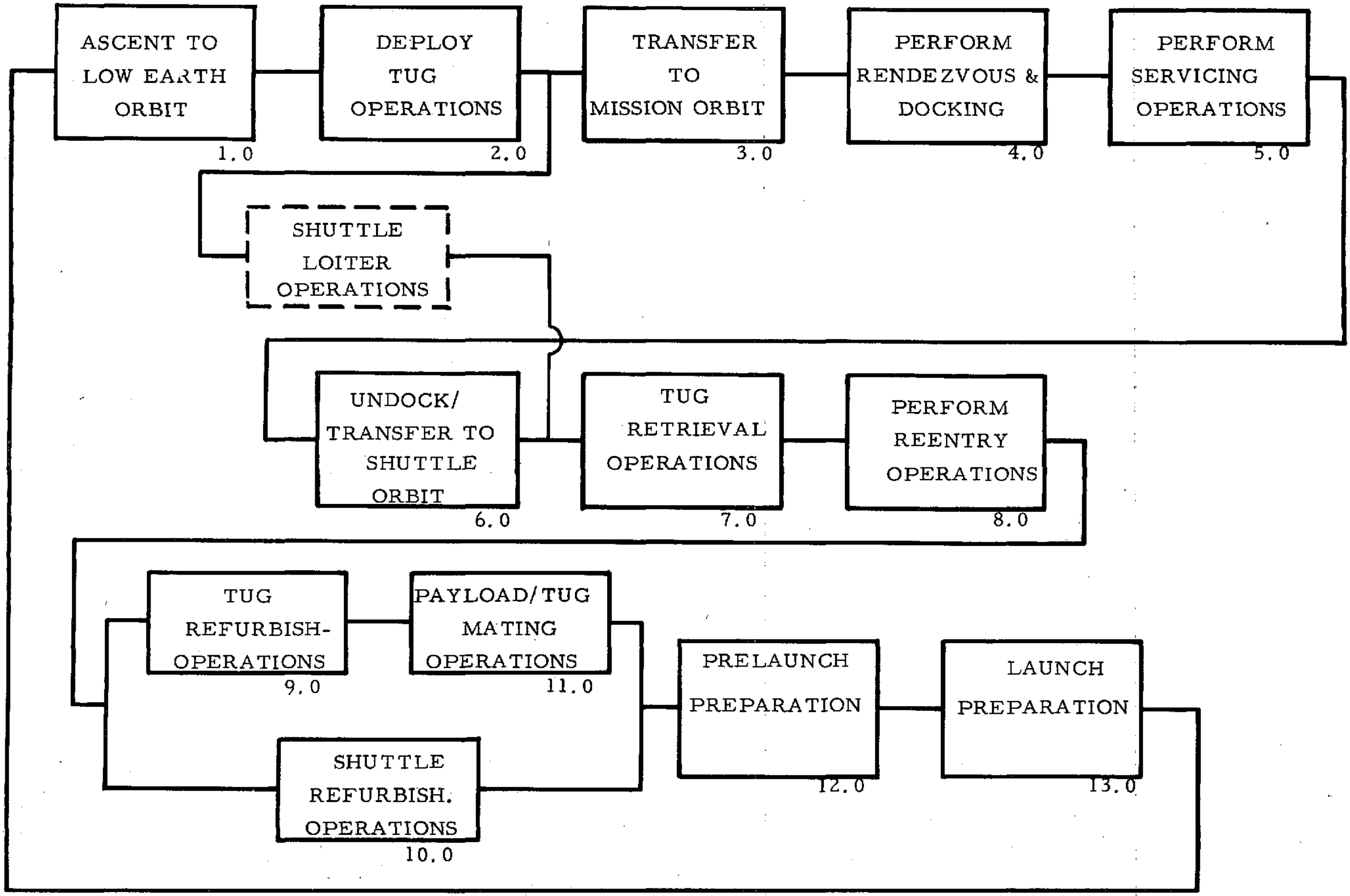
APPENDIX A

FUNCTIONAL FLOW BLOCK DIAGRAMS

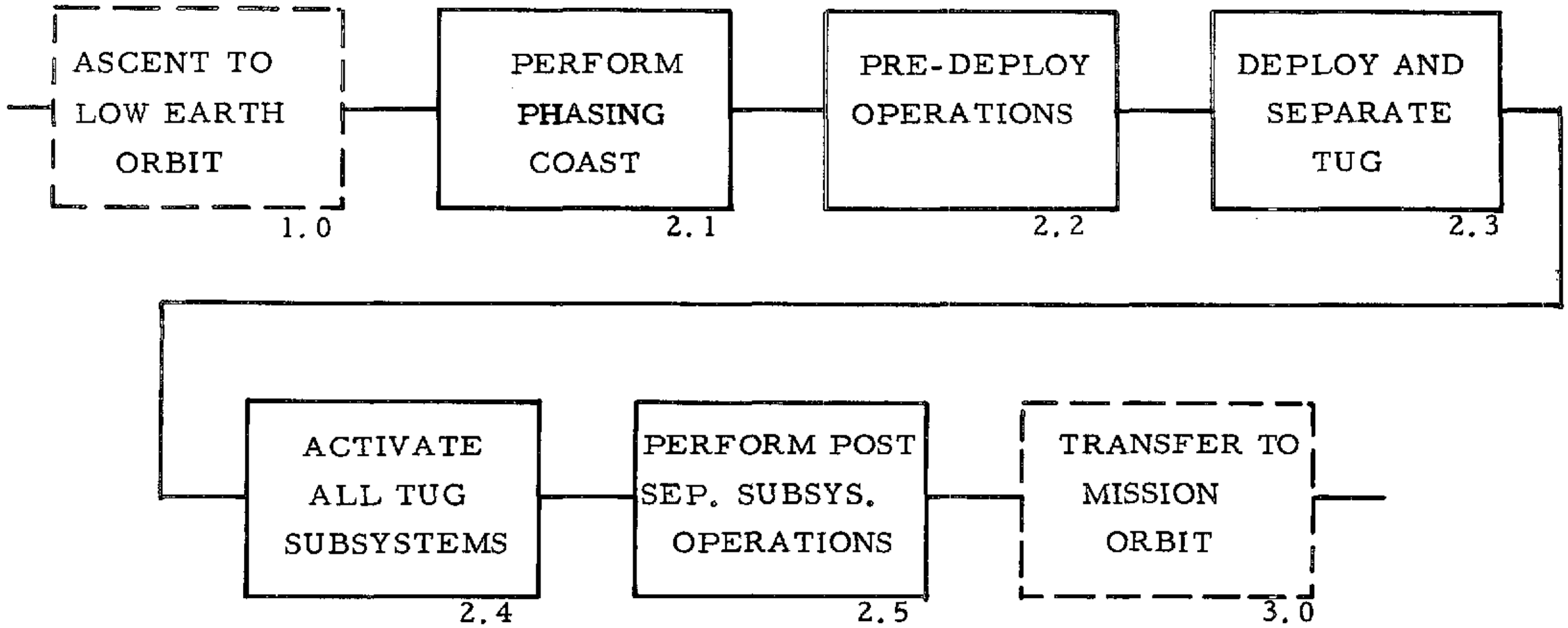
GEOSYNCHRONOUS ORBIT

SPACE SERVICING

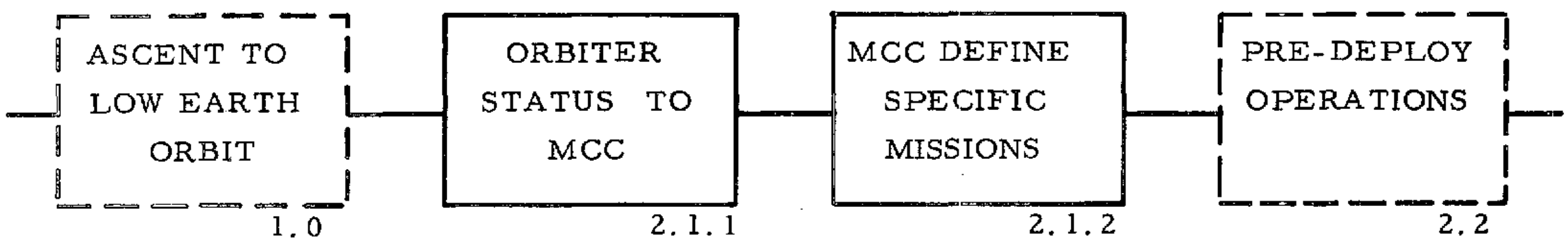
TOP LEVEL FUNCTIONAL FLOW - MISSION 3



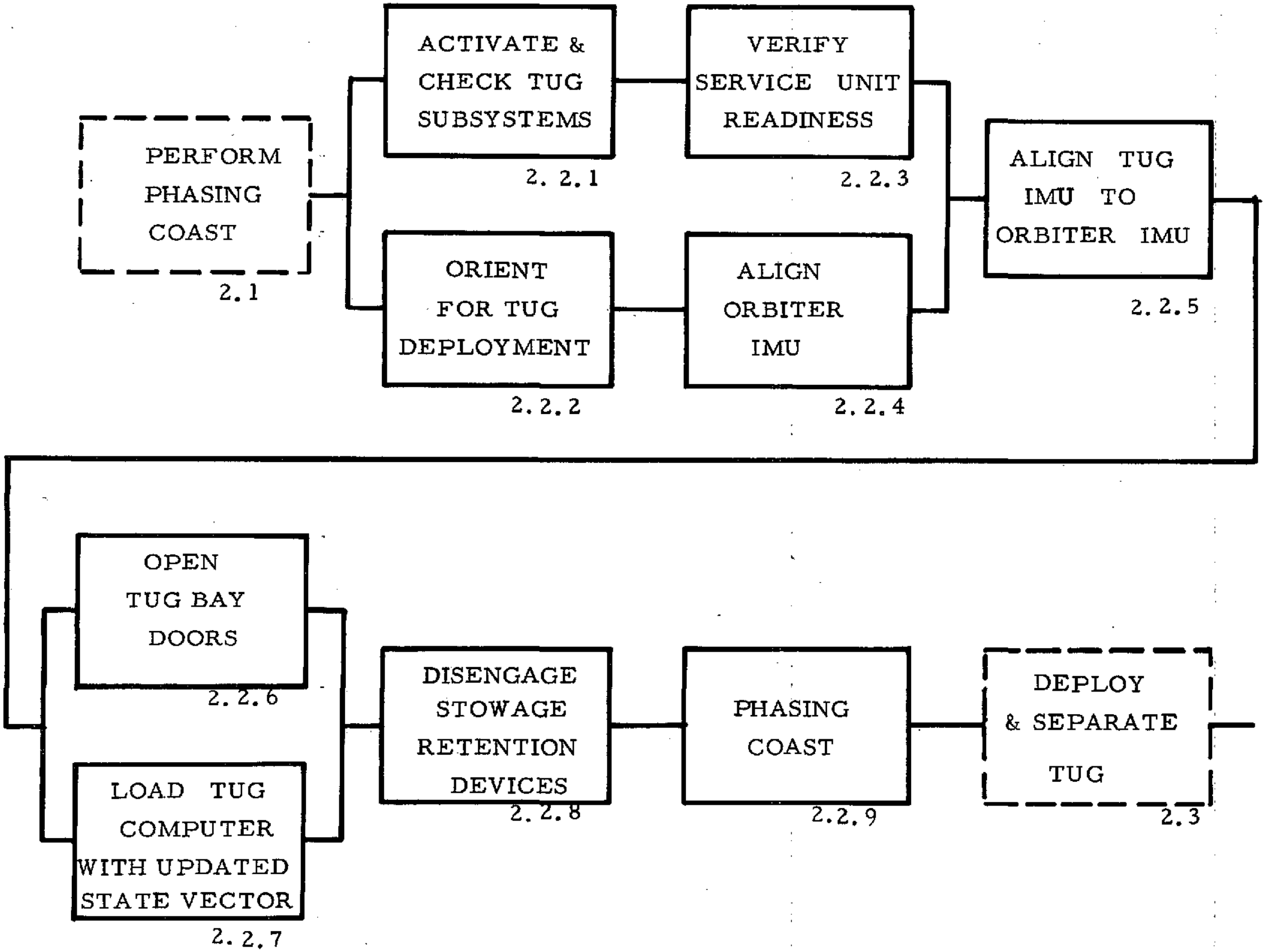
FFBD BLOCK 2.0 - DEPLOY TUG - (2nd LEVEL FUNCTIONS)



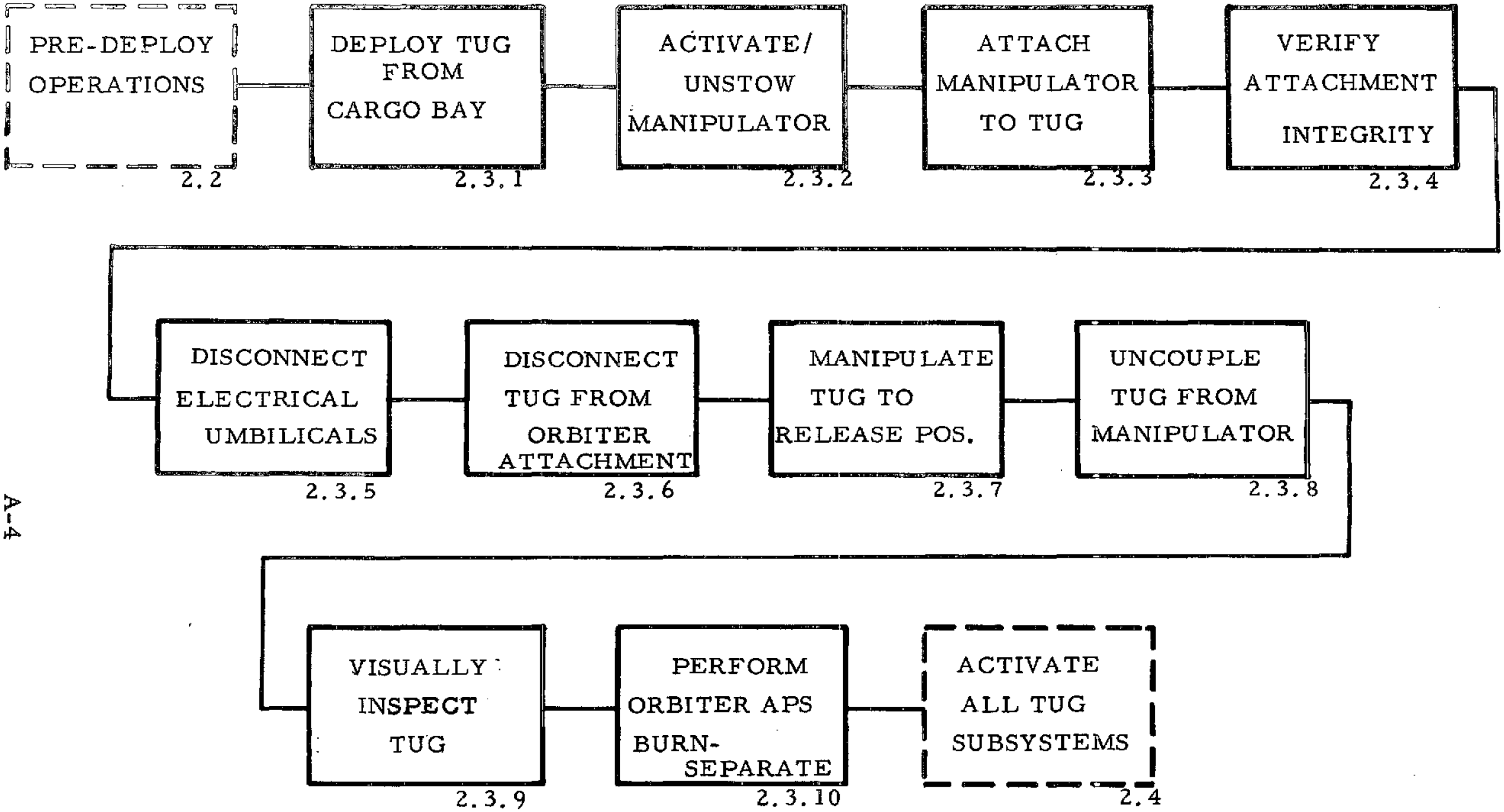
FFBD BLOCK 2.1 - PERFORM PHASING COAST - (3d LEVEL FUNCTIONS)



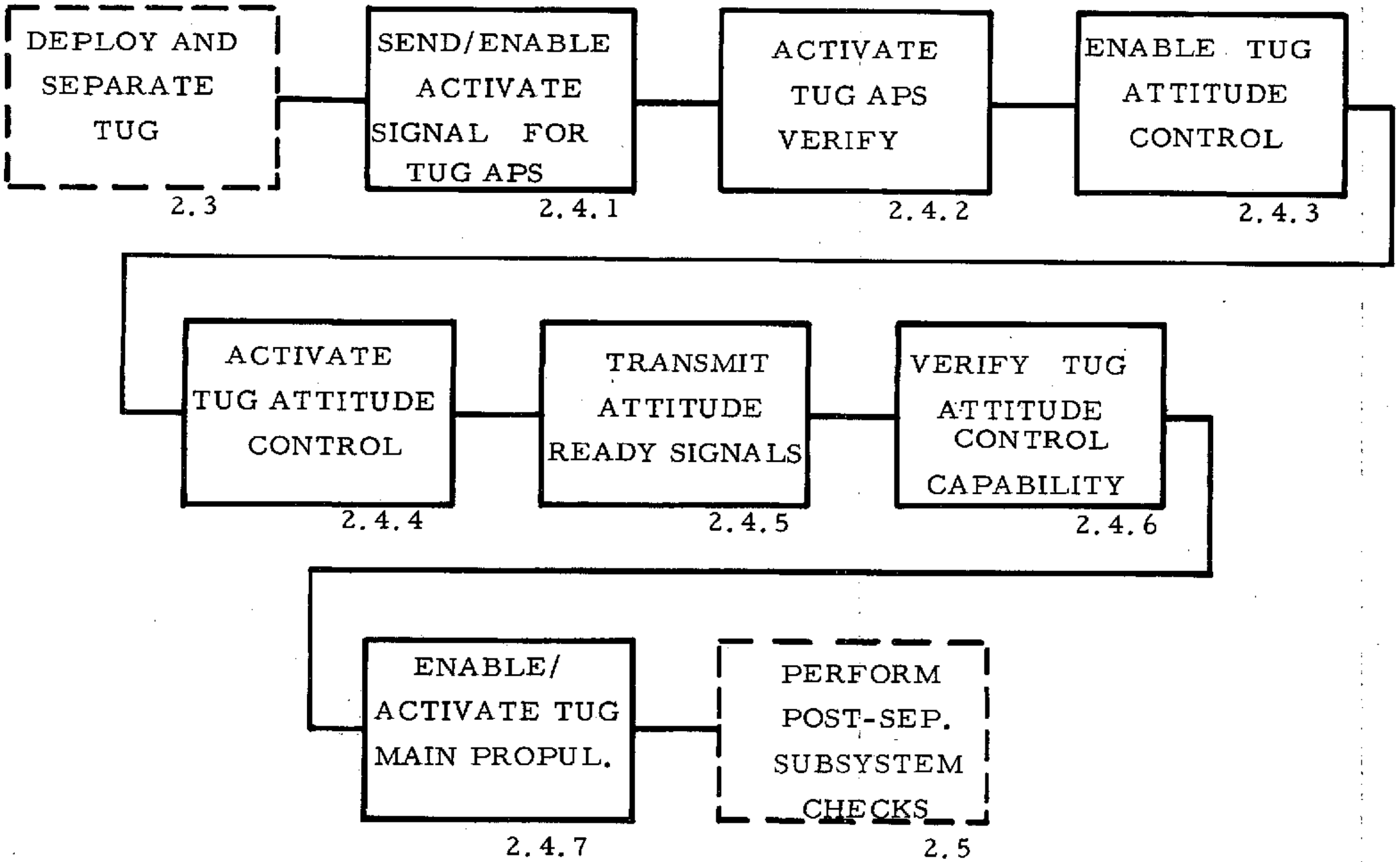
FFBD BLOCK 2.2 - PRE - DEPLOY OPERATIONS - (3rd LEVEL FUNCTIONS)



FFBD BLOCK 2.3 - DEPLOY AND SEPARATE TUG - (3d LEVEL FUNCTIONS)

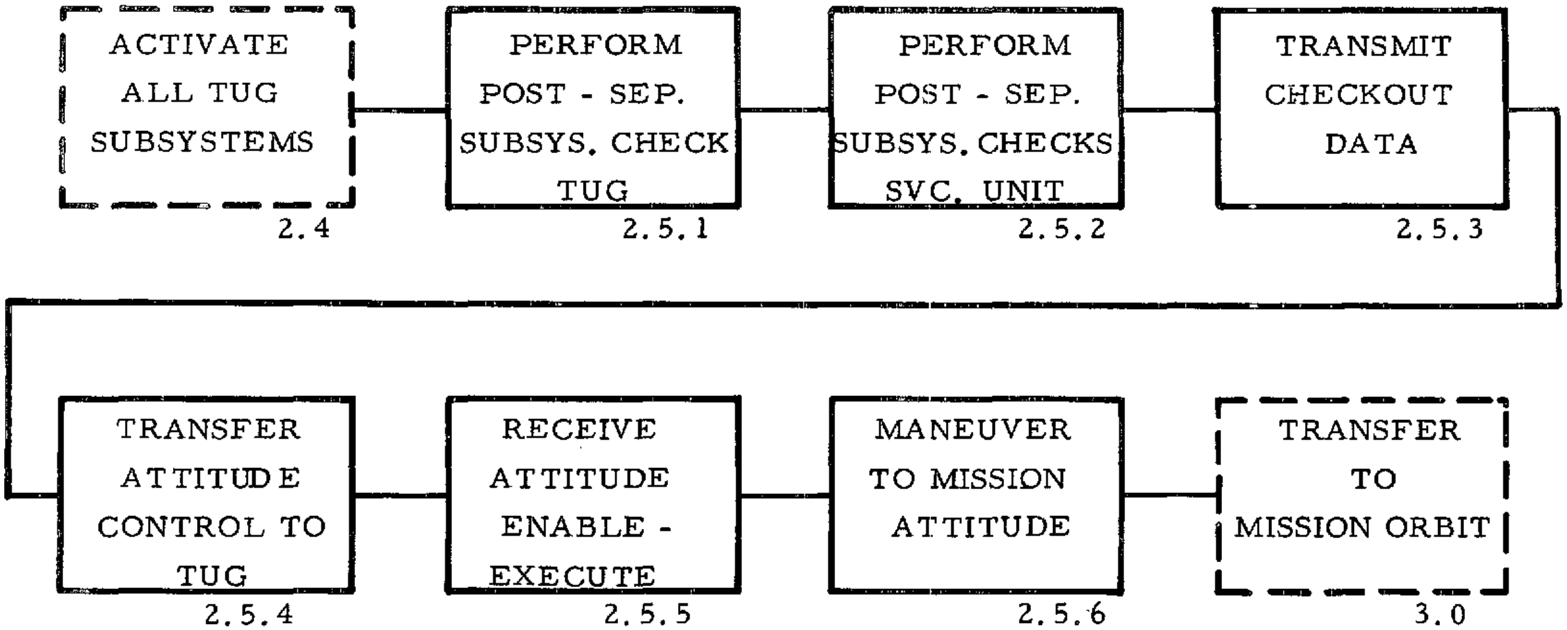


FFBD BLOCK 2.4 - ACTIVATE ALL TUG SUBSYSTEMS - (3d LEVEL FUNCTIONS)

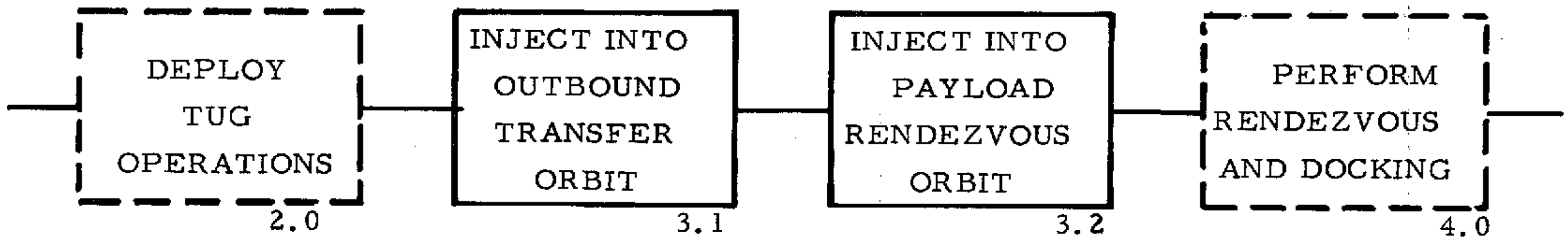


FFBD BLOCK 2.5 - PERFORM POST-SEP. SUBSYSTEM OPERATIONS

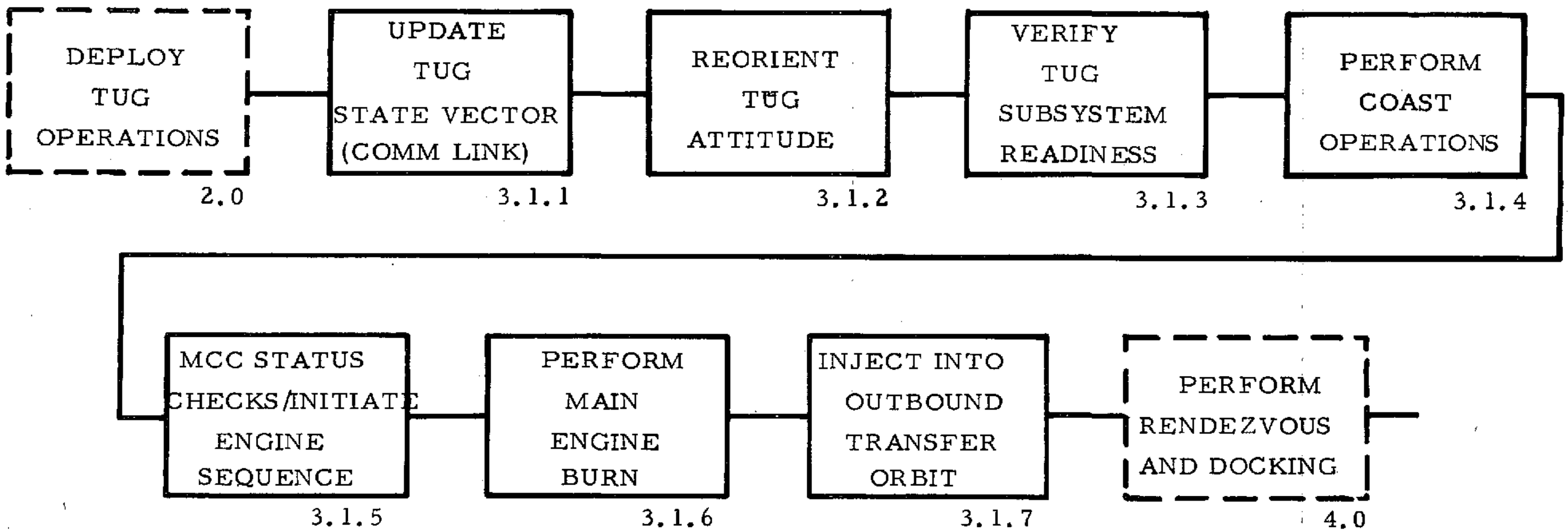
(3d LEVEL FUNCTIONS)



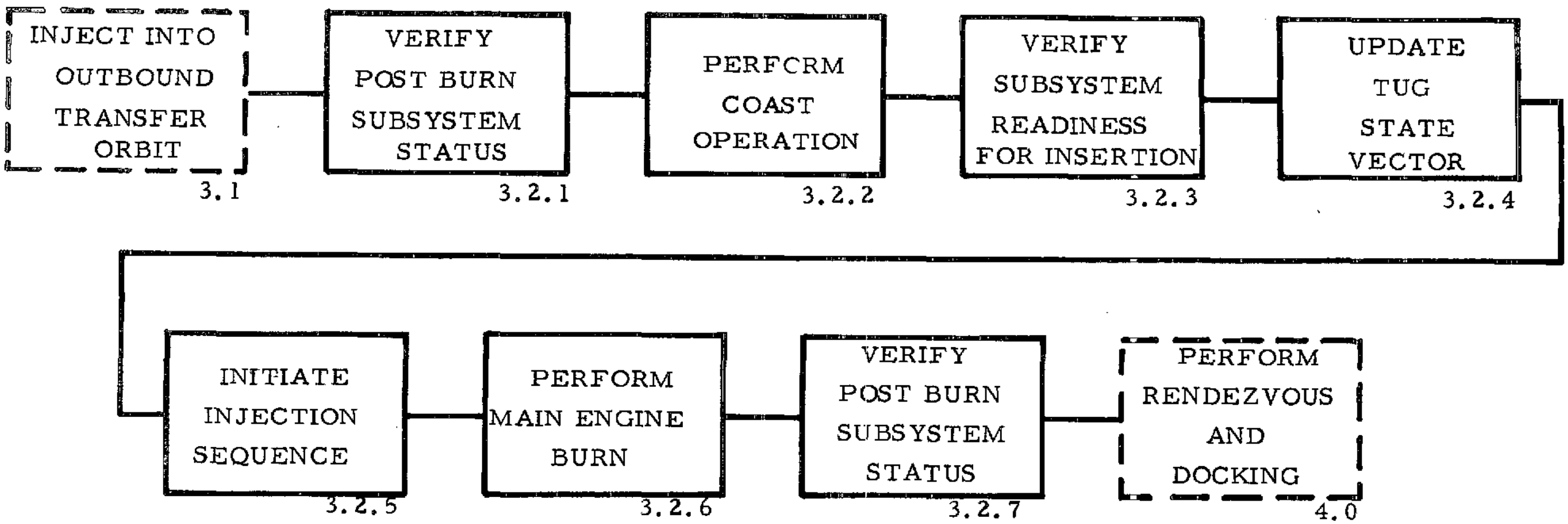
FFBD BLOCK 3.0 - TRANSFER TO MISSION ORBIT - (2nd LEVEL)



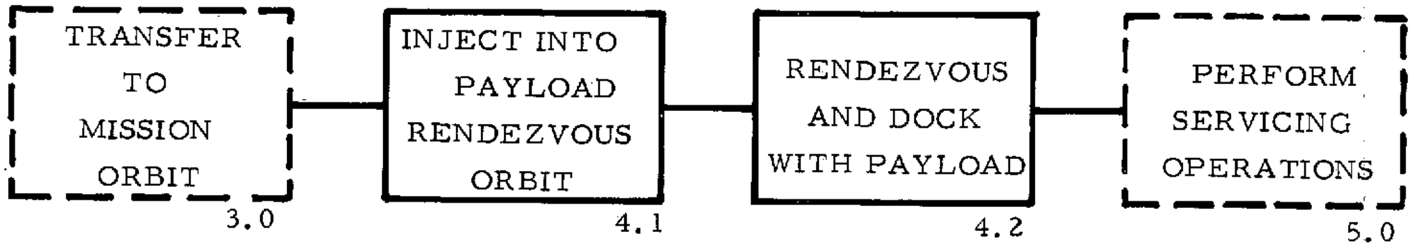
FFBD BLOCK 3.1 - INJECT INTO OUT BOUND TRANSFER ORBIT - (3rd LEVEL)



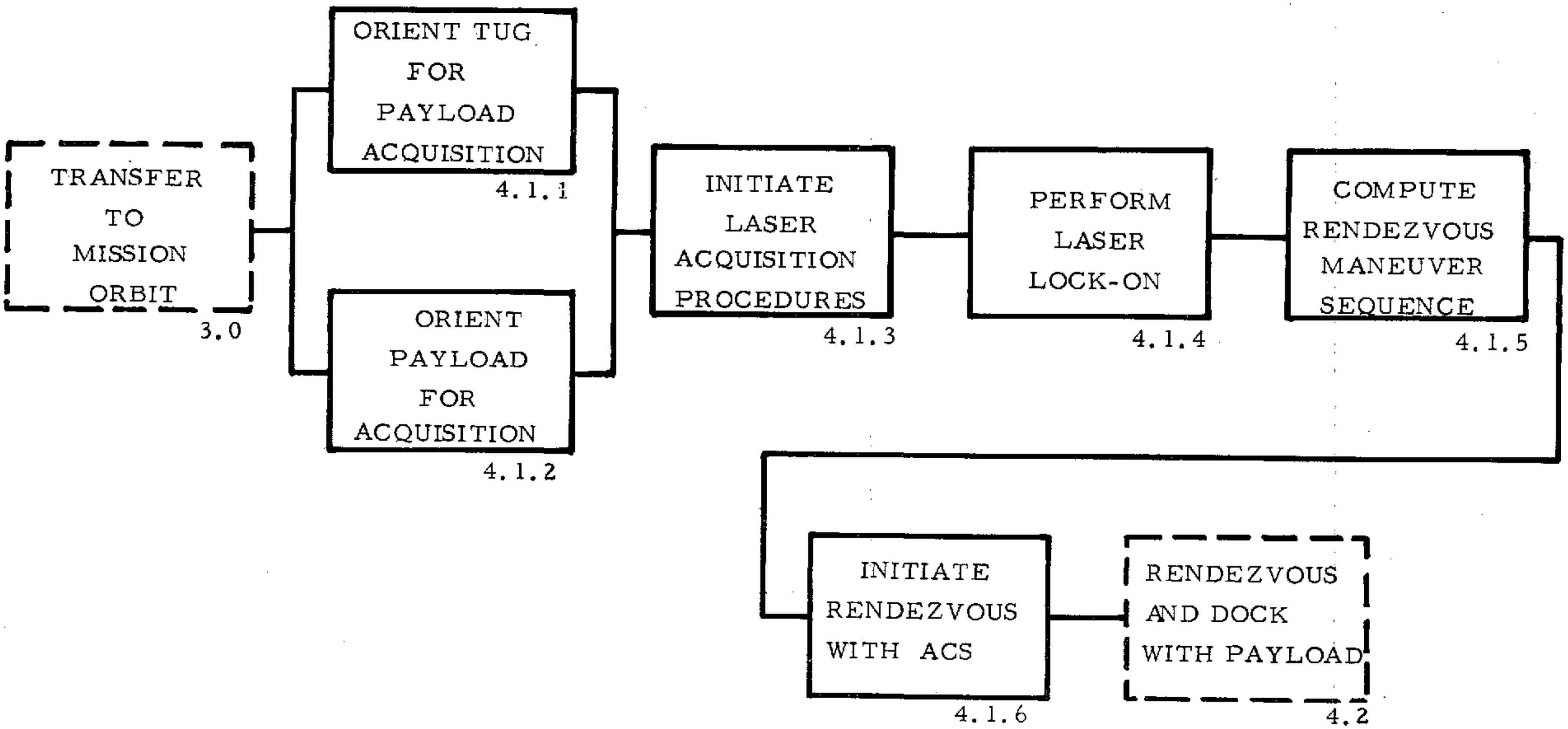
FFBD BLOCK 3.2 - INJECT INTO PAYLOAD RENDEZVOUS ORBIT - (3rd LEVEL)



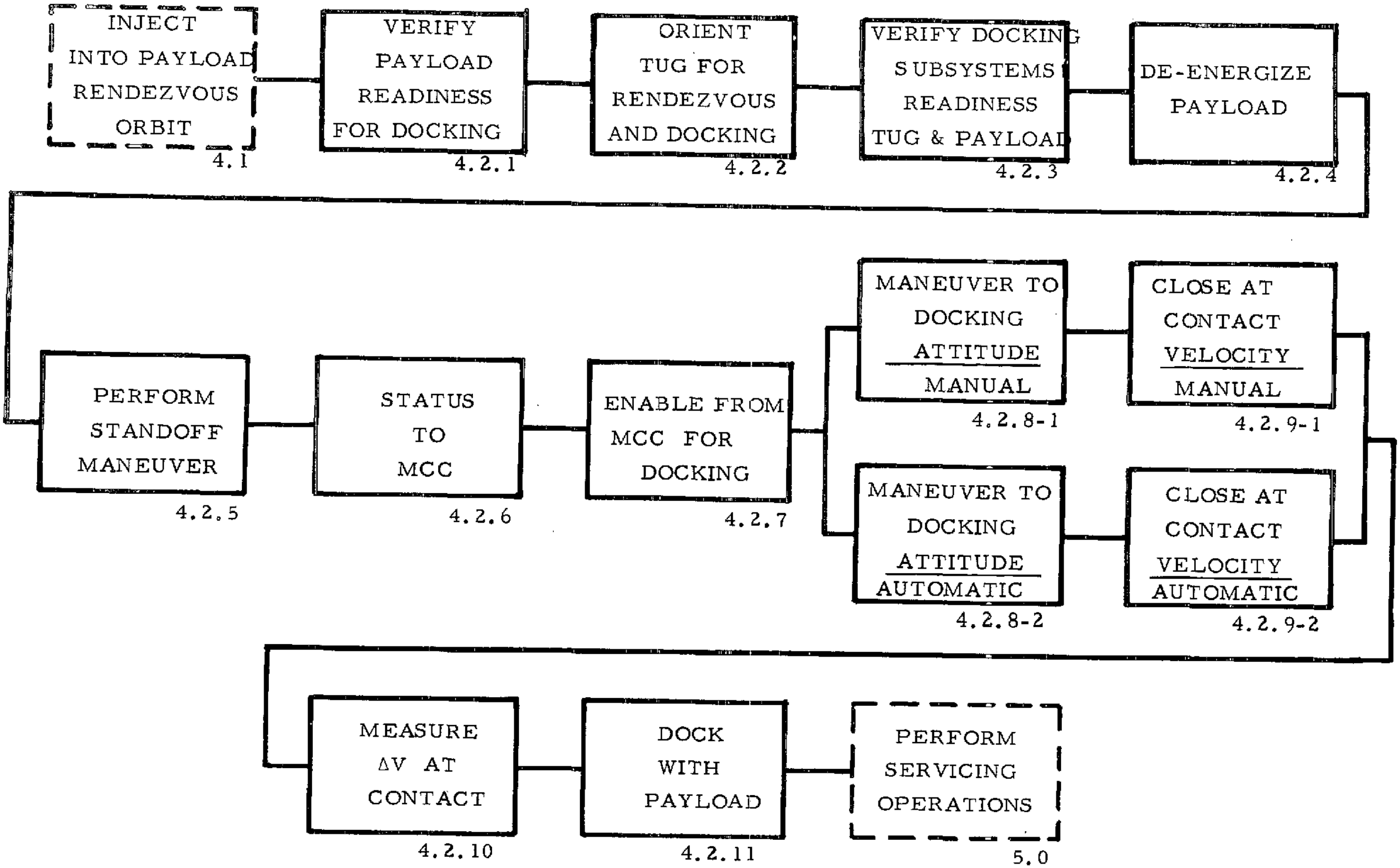
FFBD BLOCK 4.0 - PERFORM RENDEZVOUS AND DOCKING - (2nd LEVEL)



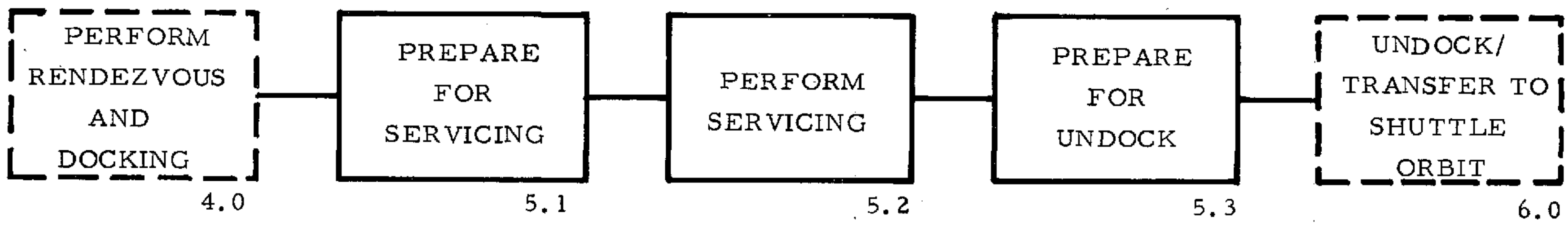
FFBD BLOCK 4.1 - INJECT INTO PAYLOAD RENDEZVOUS ORBIT - (3rd LEVEL)



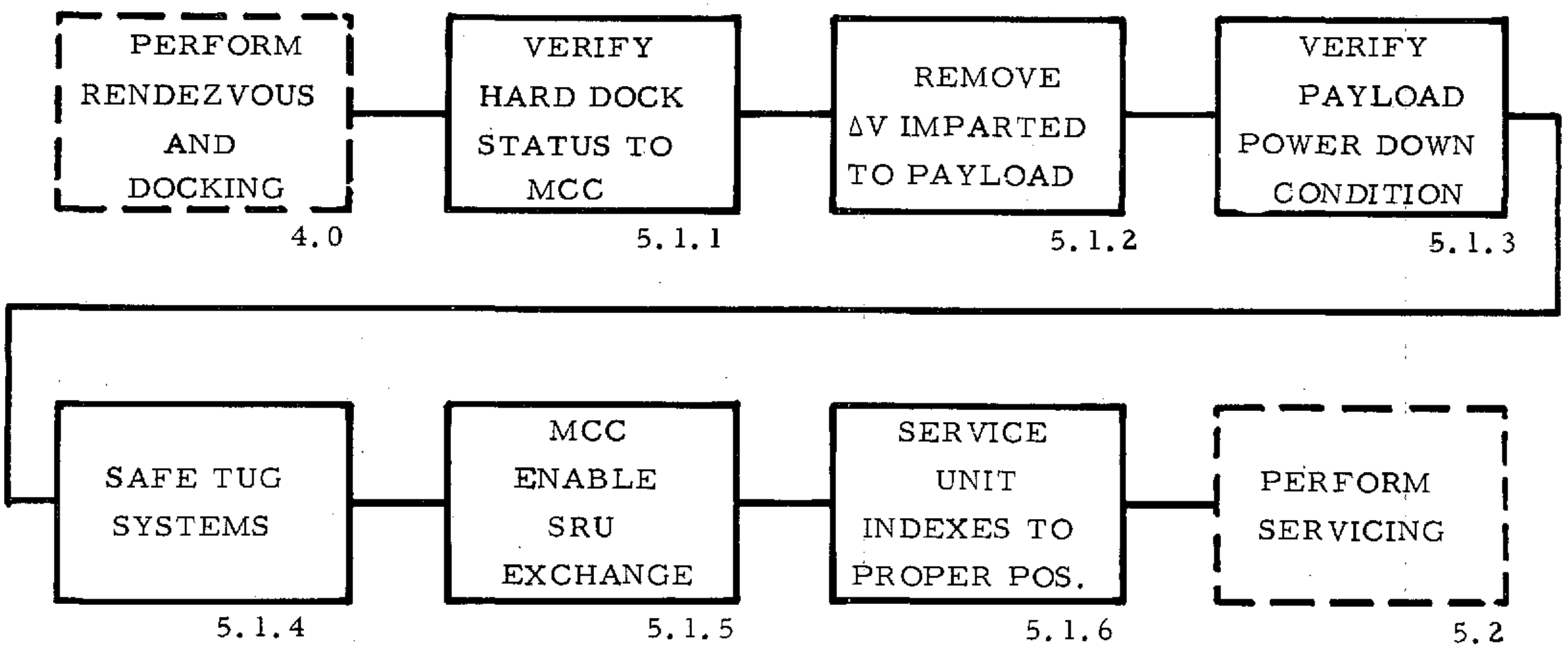
FFBD BLOCK 4.2 - RENDEZVOUS AND DOCK WITH PAYLOAD - (3rd LEVEL)



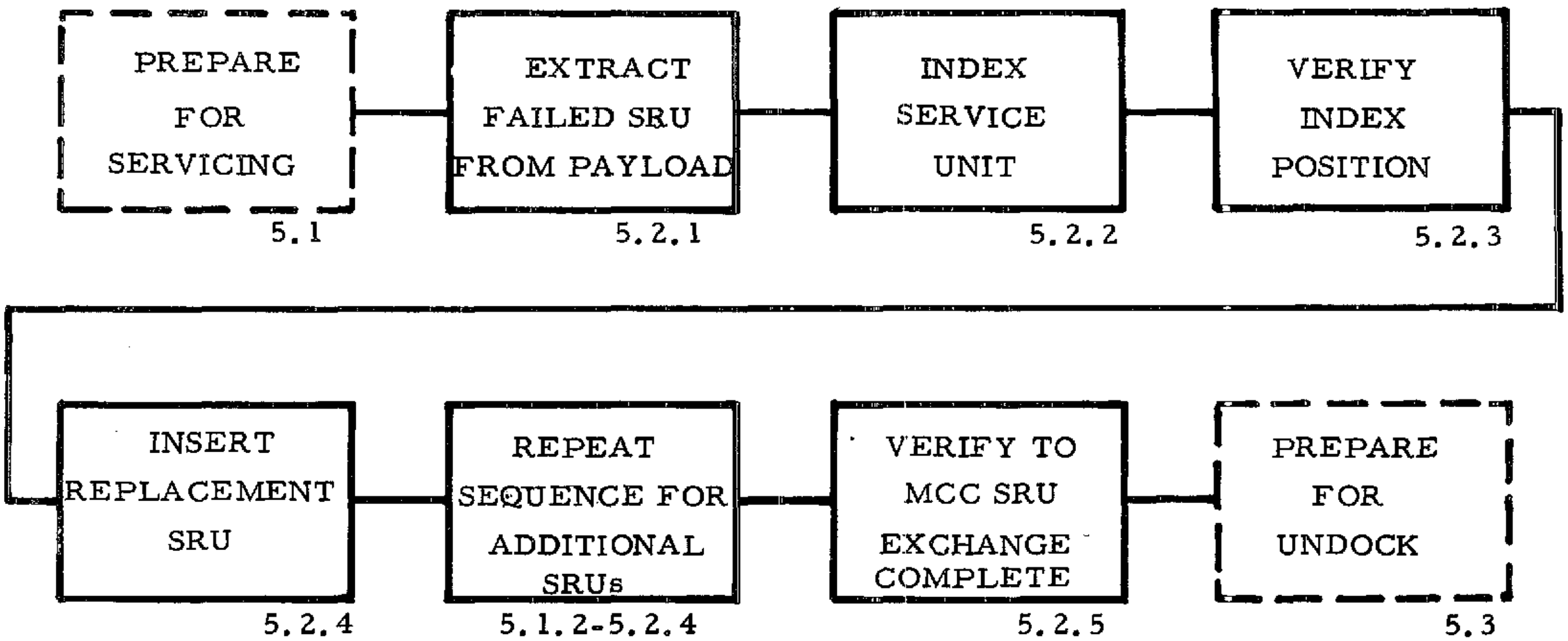
FFBD BLOCK 5.0 - PERFORM SERVICING OPERATIONS - (2nd LEVEL)



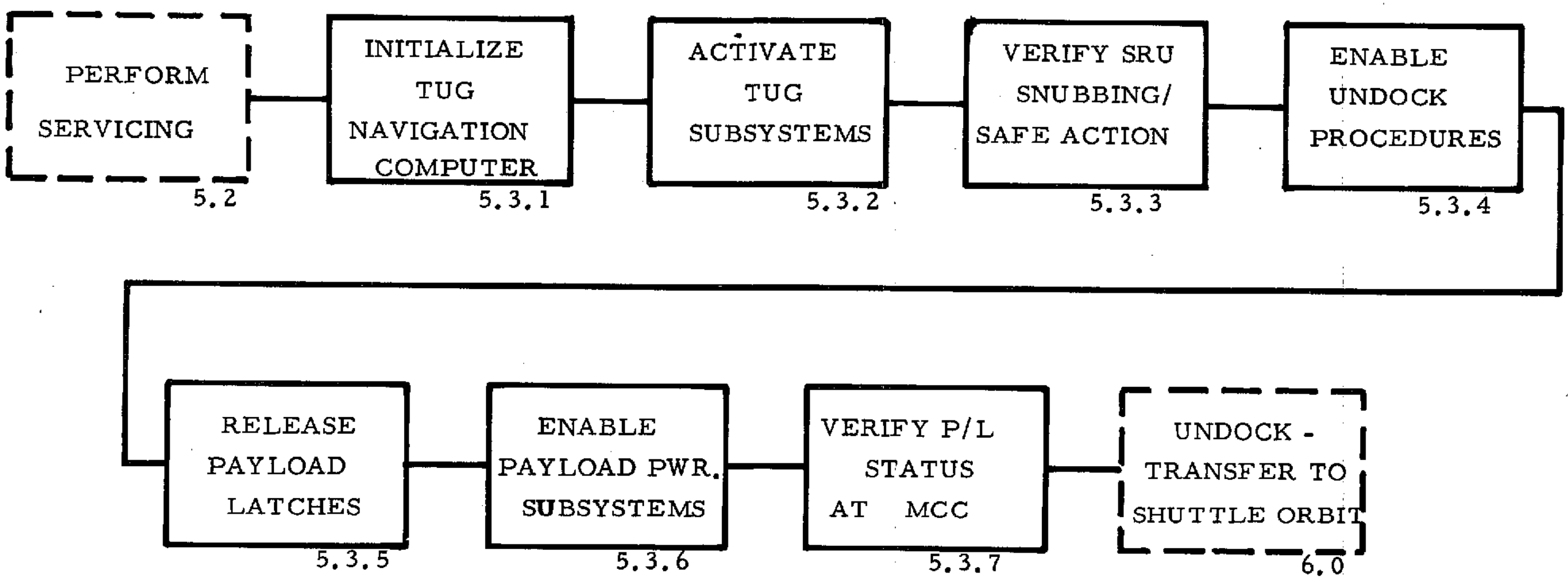
FFBD BLOCK 5.1 - PREPARE FOR SERVICING - (3rd LEVEL)



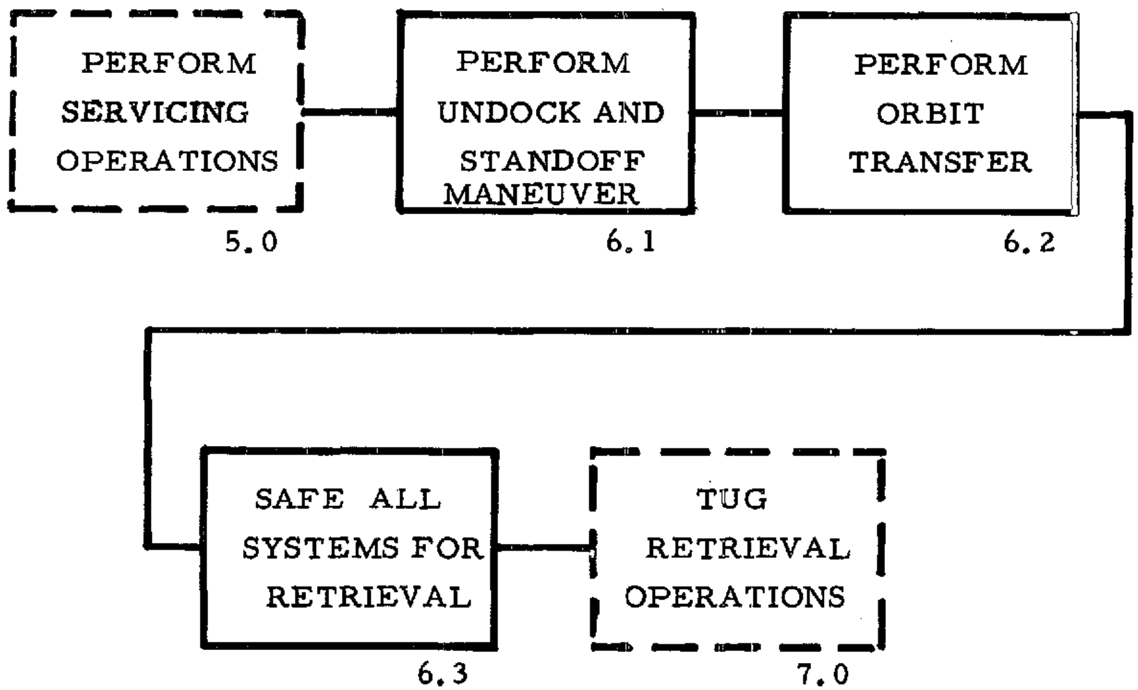
FFBD BLOCK 5.2 - PERFORM SERVICING - (3rd LEVEL)



FFBD BLOCK 5.3 - PREPARE FOR UNDOCK - (3rd LEVEL)

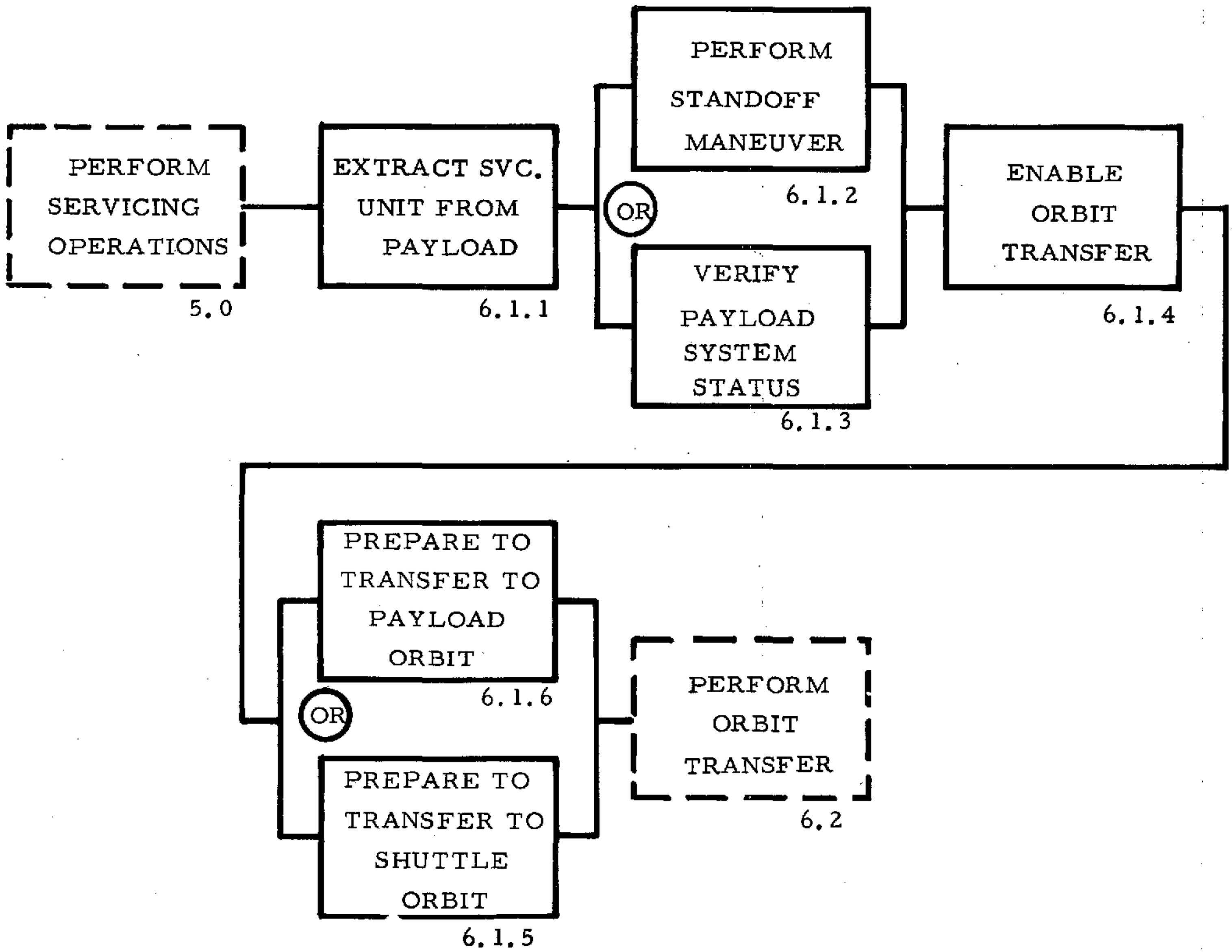


FFBD BLOCK 6.0 - UNDOCK/TRANSFER TO SHUTTLE ORBIT - (2nd LEVEL)

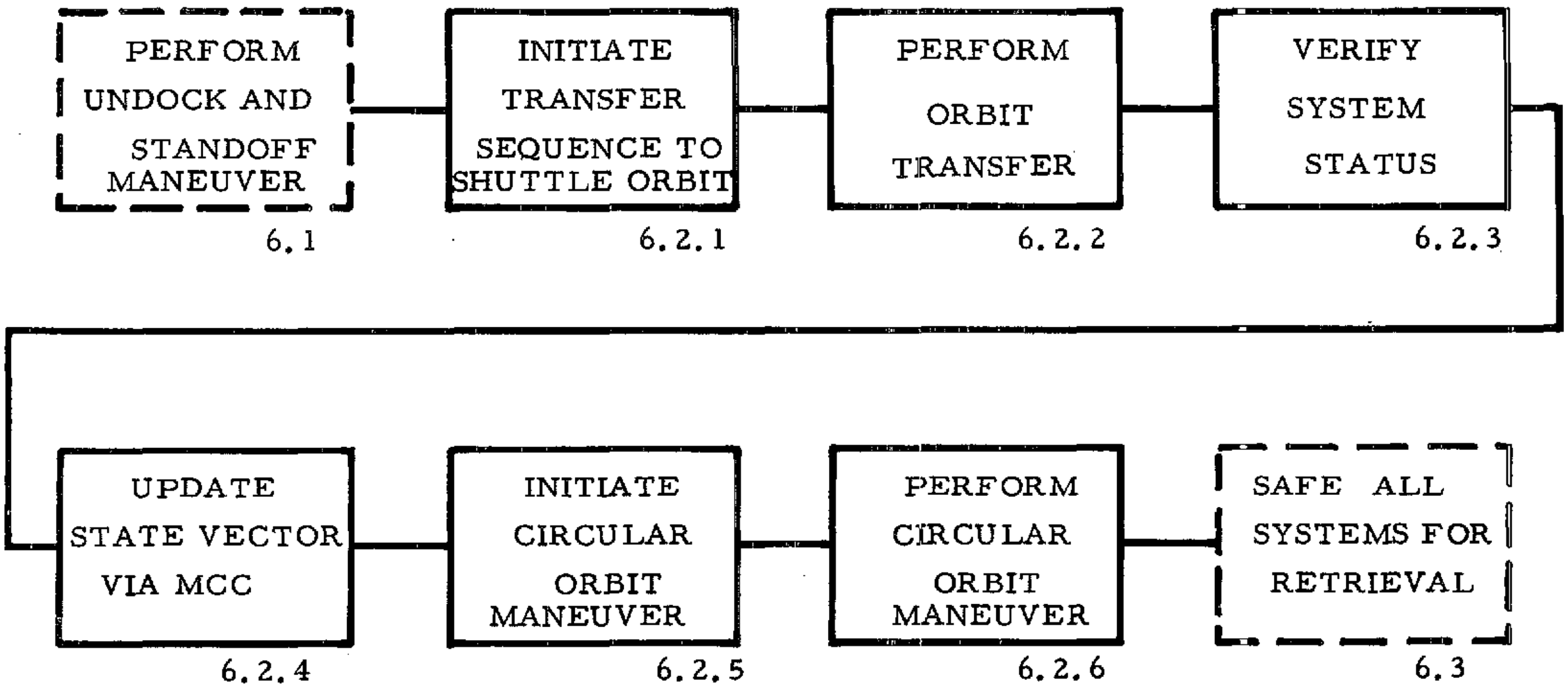


NOTE: FFBD Blocks 7.0 and 8.0 have no impact on Tug software

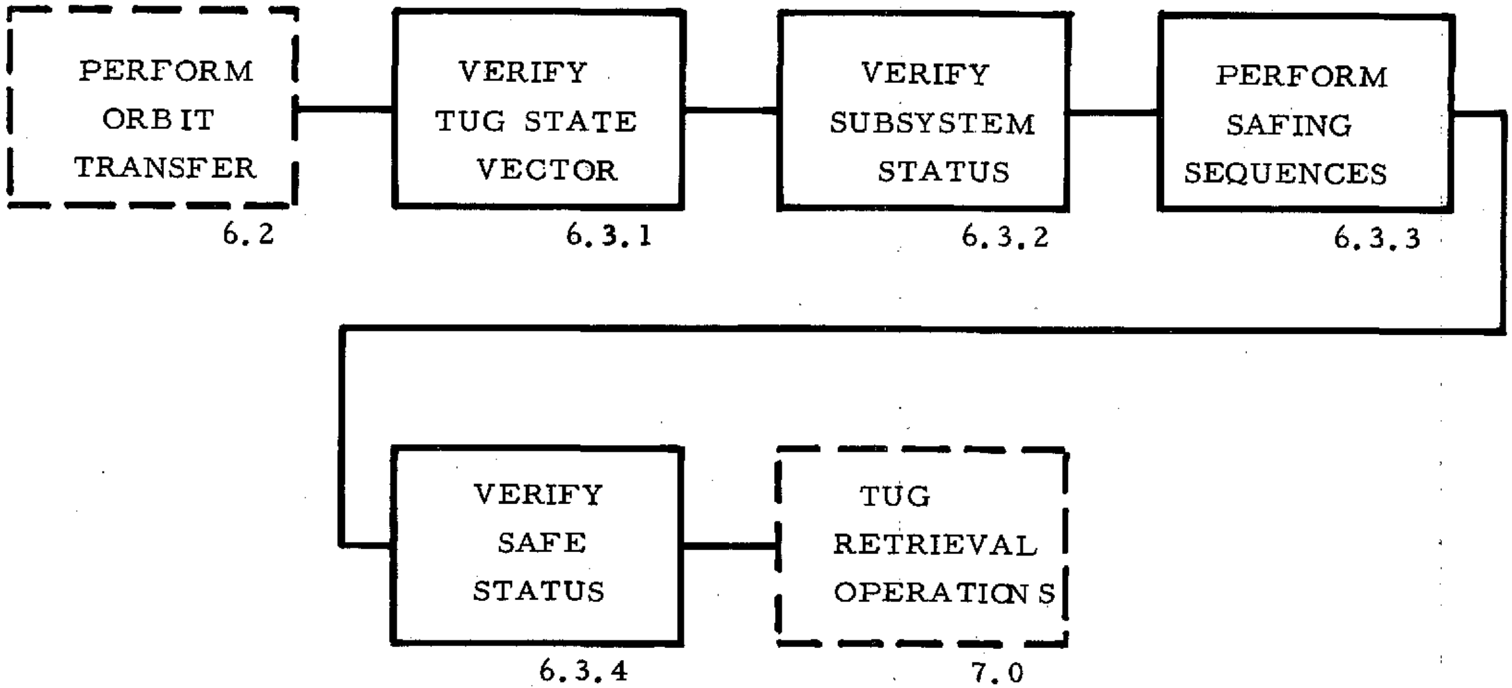
FFBD BLOCK 6.1 - PERFORM UNDOCK & STANDOFF MANEUVER - (3rd LEVEL)



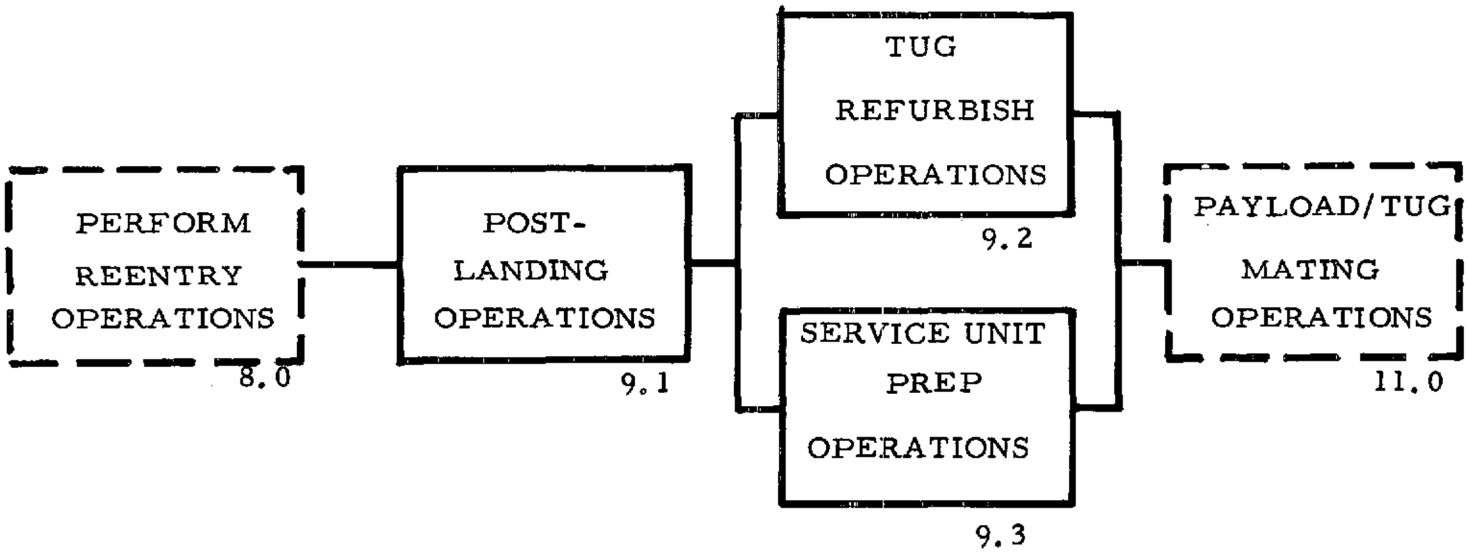
FFBD BLOCK 6.2 - PERFORM ORBIT TRANSFER - (3rd LEVEL)



FFBD BLOCK 6.3 - SAFE ALL SYSTEMS FOR RETRIEVAL - (2nd LEVEL)

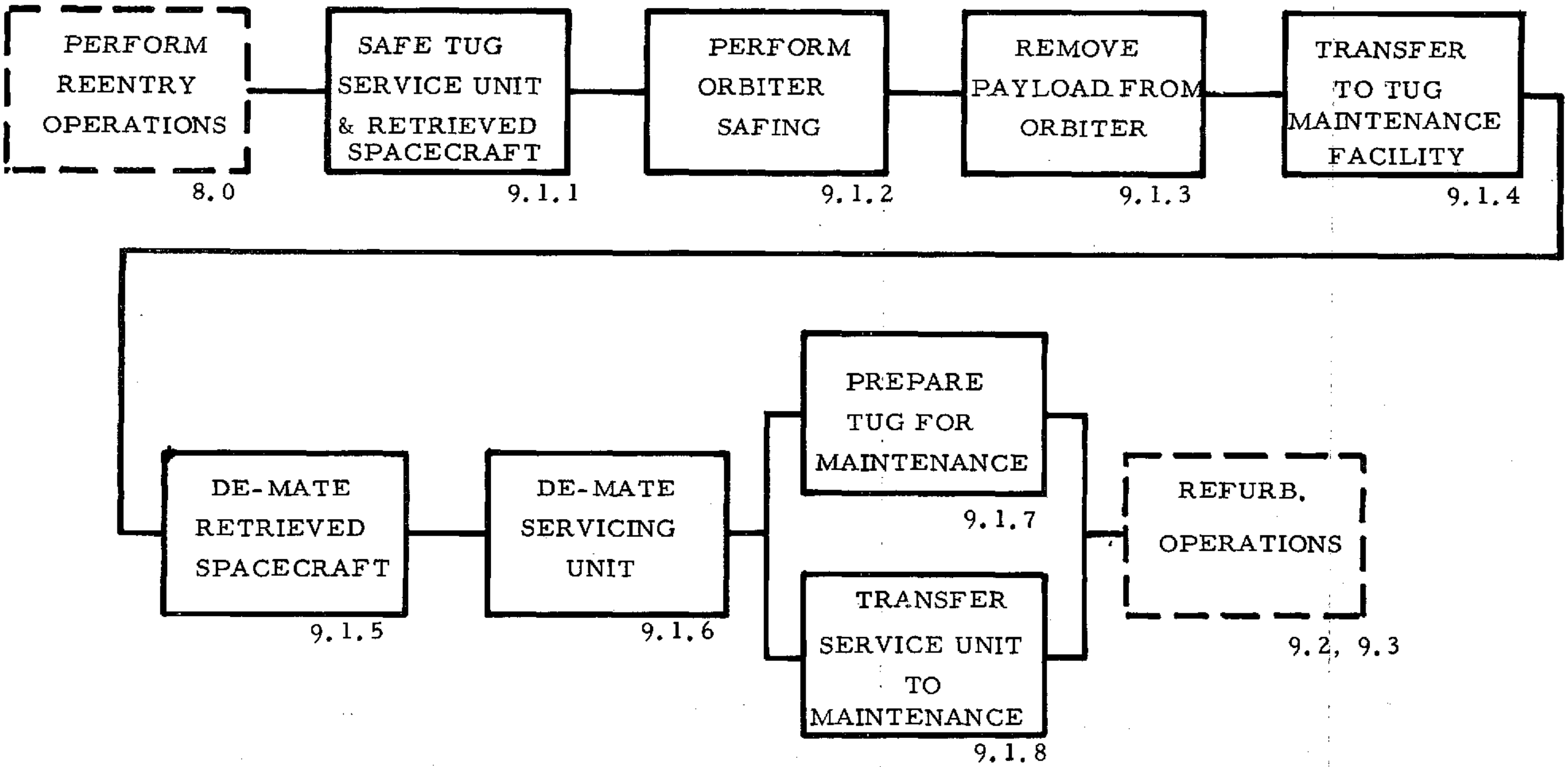


FFBD BLOCK 9.0 - TUG REFURBISHMENT - (2nd LEVEL)

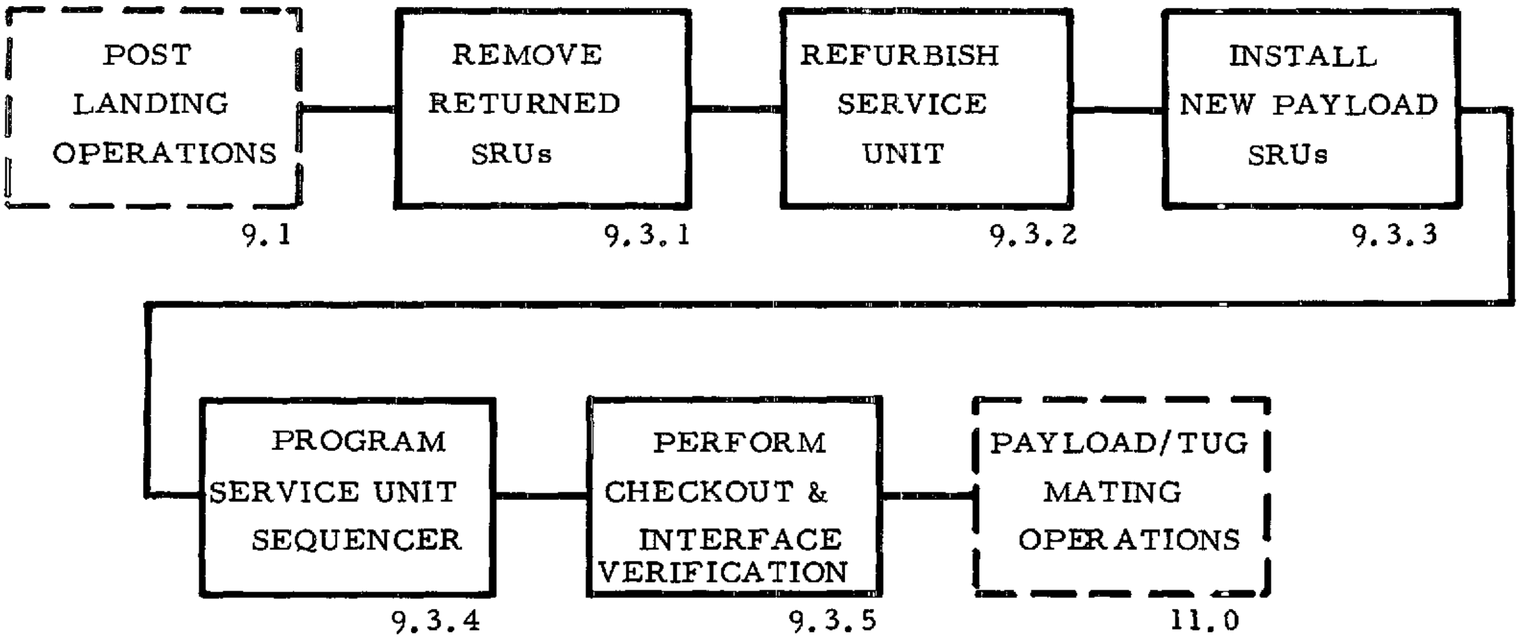


NOTE: FFBD BLOCKS 9.2 and 10.0
HAVE NO IMPACT ON TUG SOFTWARE

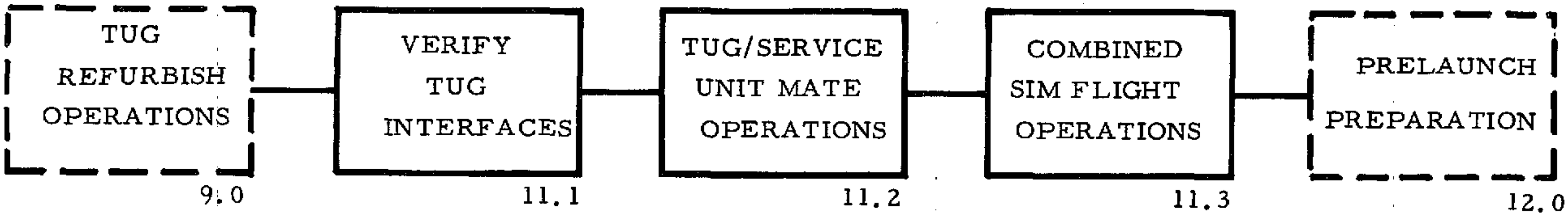
FFBD BLOCK 9.1 - POST-LANDING OPERATIONS - - (3rd LEVEL)



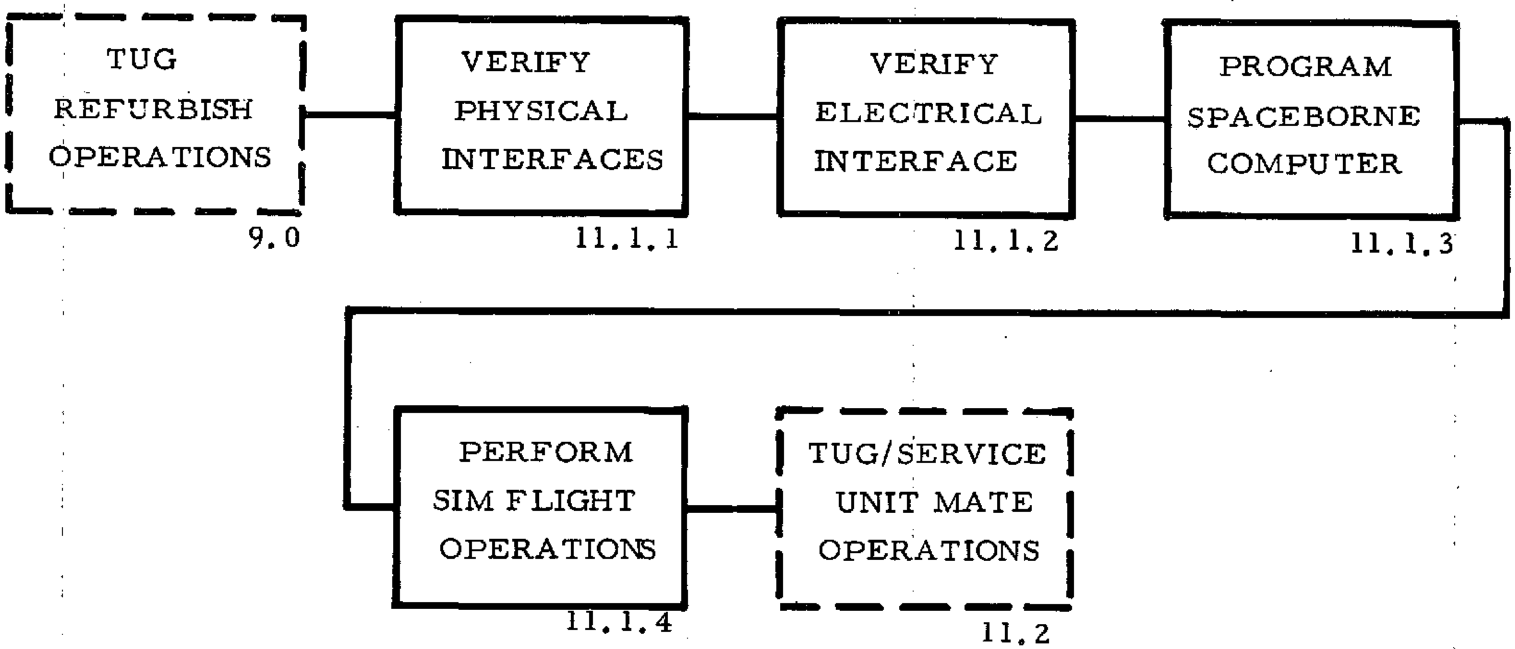
FFBD BLOCK 9.3 - SERVICE UNIT PREP OPERATIONS



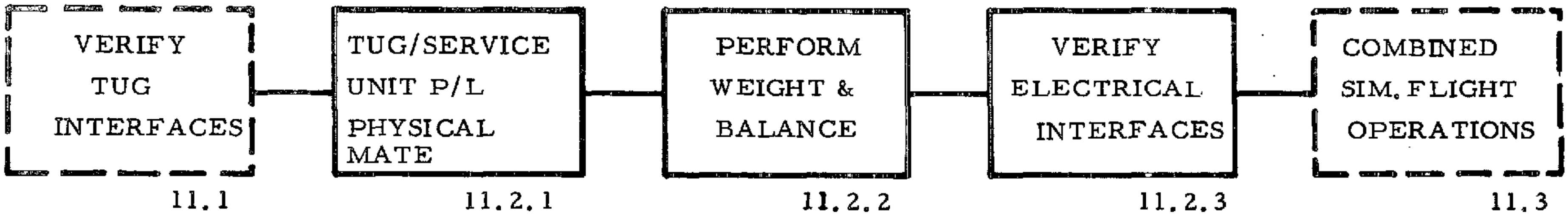
FFBD BLOCK 11.0 - PAYLOAD/TUG MATING OPERATIONS - (2nd LEVEL)



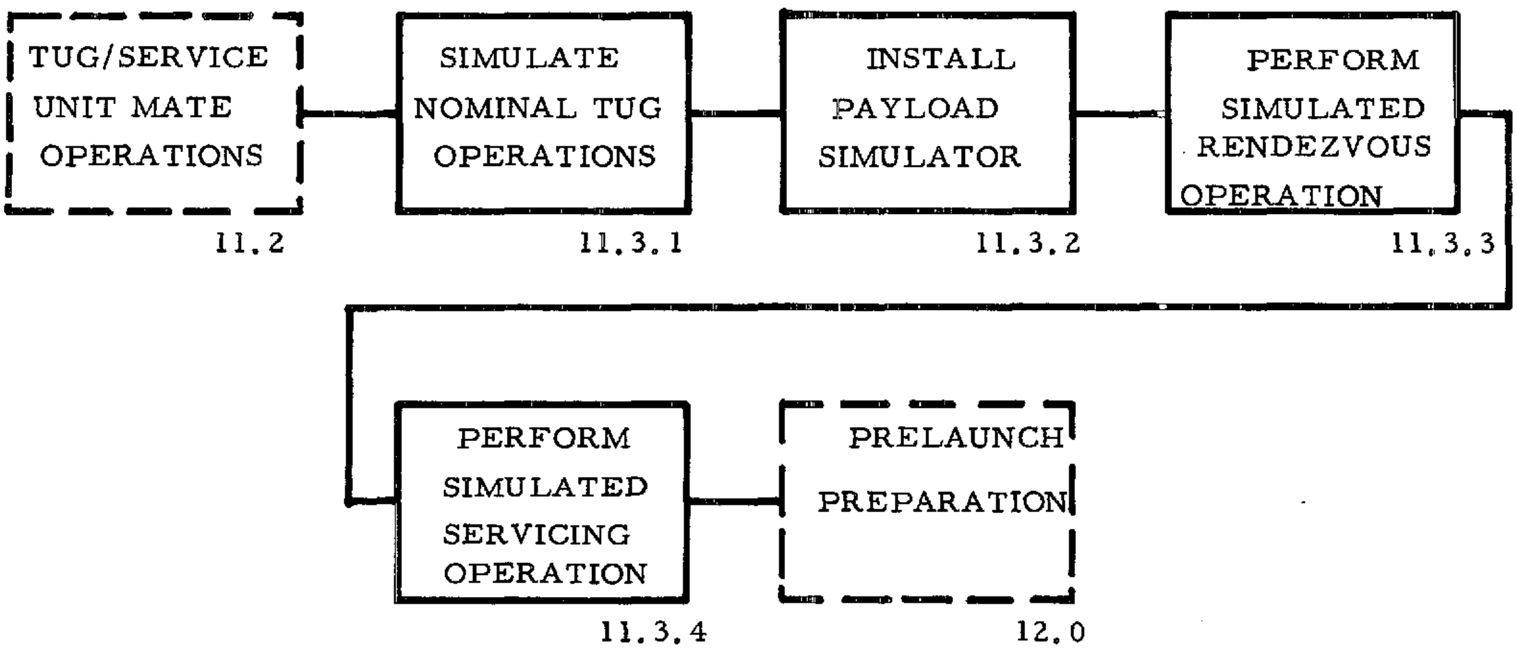
FFBD BLOCK 11.1 - VERIFY TUG INTERFACES - (3rd LEVEL)



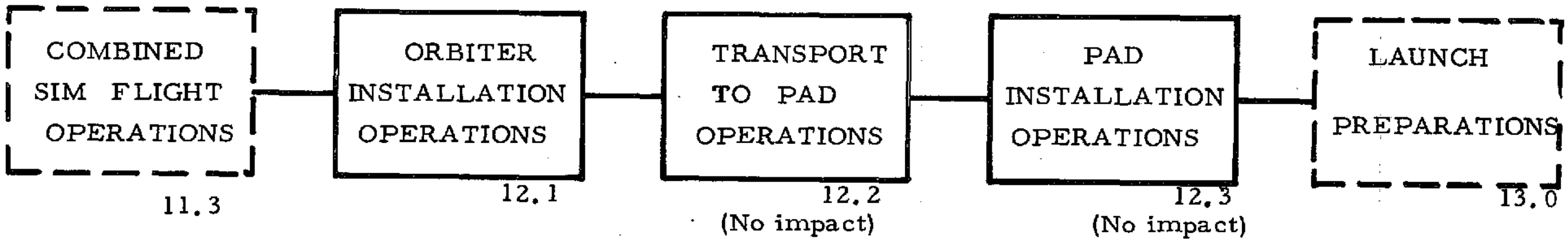
FFBD BLOCK 11.2 - TUG SERVICE UNIT MATE OPERATIONS - (3rd LEVEL)



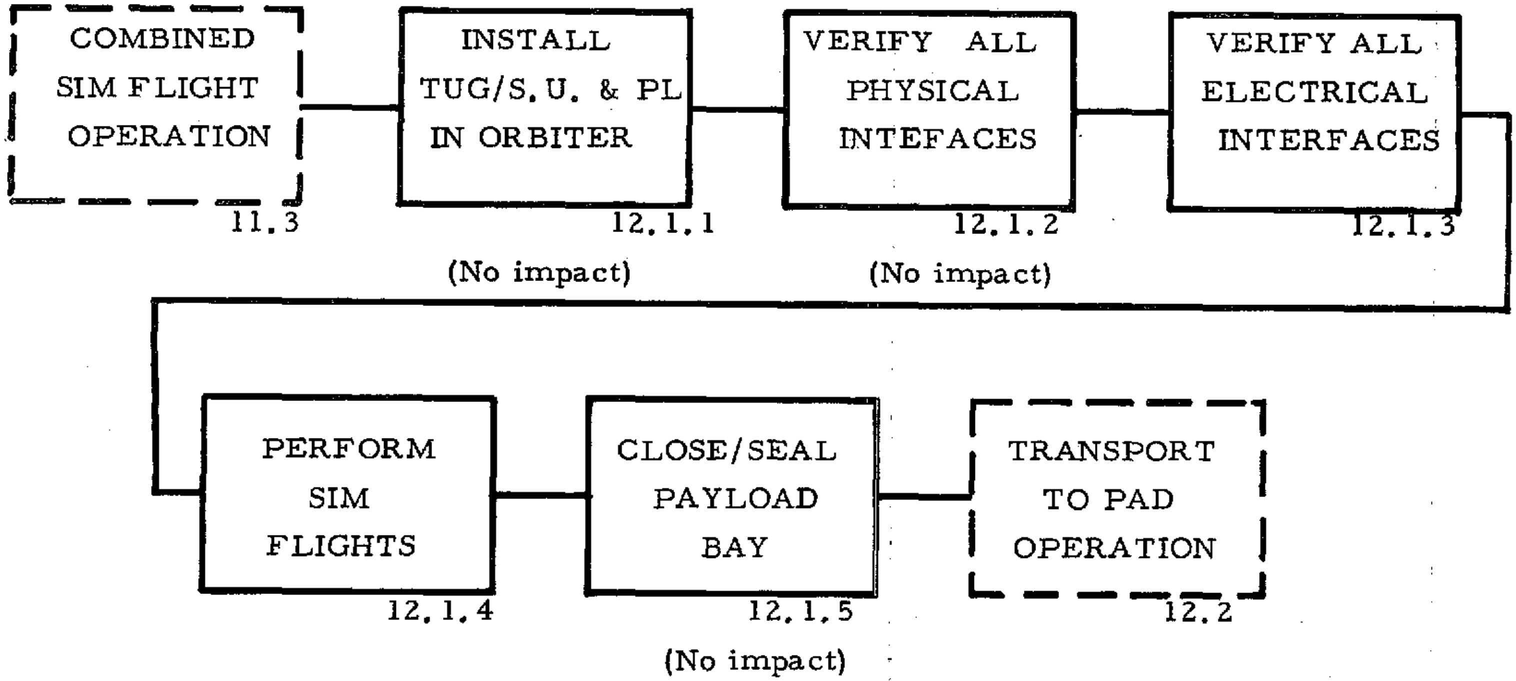
FFBD BLOCK 11.3 - COMBINED SIM FLIGHT OPERATIONS - (3rd LEVEL)



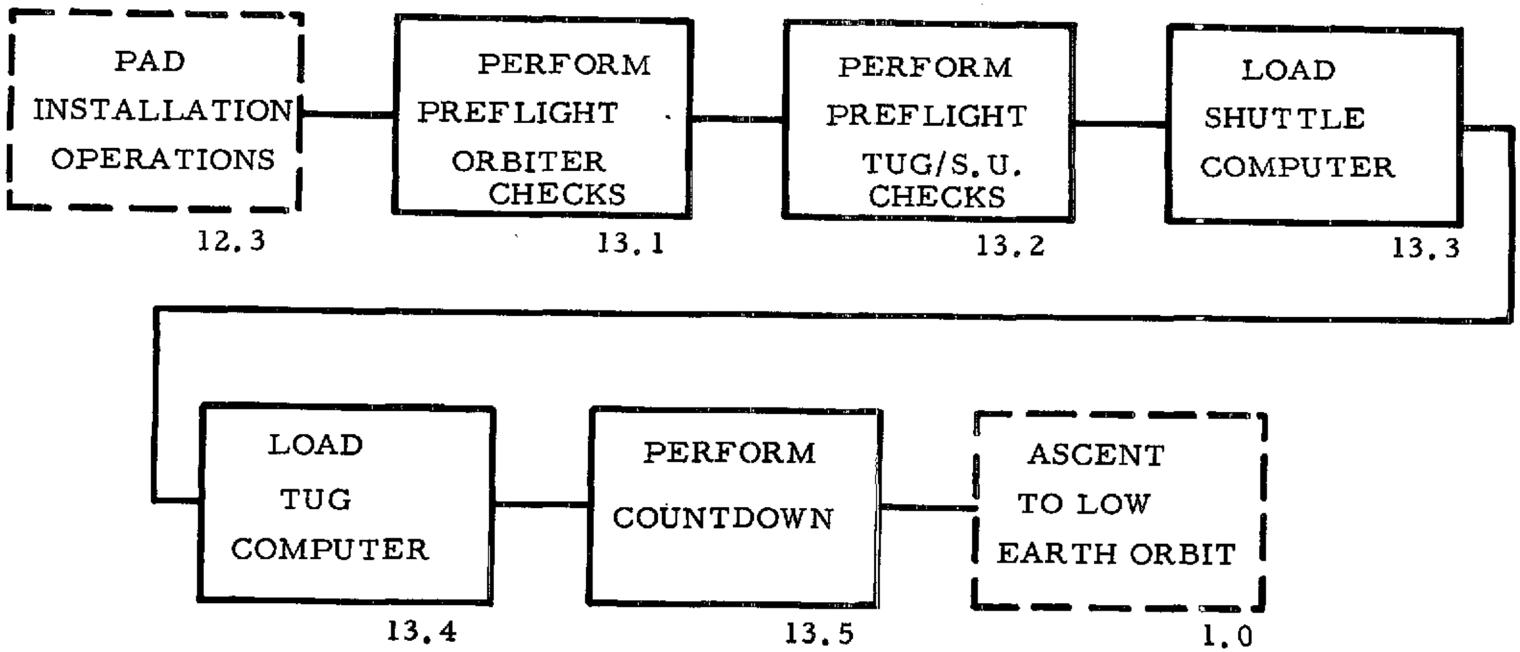
FFBD BLOCK 12.0 - PRELAUNCH PREPARATION - (2nd LEVEL)



FFBD BLOCK 12.1 - ORBITER INSTALLATION OPERATIONS - (3rd LEVEL)



FFBD BLOCK 13.0 - LAUNCH PREPARATION - (2nd LEVEL)



APPENDIX B

D-1 CENTAUR SOFTWARE DEVELOPMENT COSTS

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LEWIS RESEARCH CENTER
CLEVELAND, OHIO 44135

APR 25 1974

REPLY TO ATTN OF: 9200(NW)

Mr. Robert R. Wolfe
NASA Study Director
The Aerospace Corporation
Post Office Box 92957
Los Angeles, CA 90009

Subject: D-1 Centaur Software Development Costs

In response to your letter of March 15, 1974, the following information is forwarded for work being performed under Contract NASW-2575.

A. Introduction

The cost data contained herein are formulated with respect to:

1. Airborne software development and checkout costs for the D-1 Centaur program. These costs reflect actual man-hour and computer hour expenditures for the combined Atlas/Centaur (D-1A) and Titan/Centaur (D-1T) programs.

2. Non-recurring and recurring software costs associated with mission variations. These costs reflect expected expenditures, based upon past experience, for planetary and orbital missions.

The software development costs for the D-1 Centaur program reflect the work required to create the software and program it for the on-board digital computer whose function is to control various aspects of vehicle guidance, navigation, PCM TLM formatting, sequencing, and control requirements. A modular software concept was formulated to facilitate software variations needed to support a variety of mission and vehicle configurations. Costs reflect combined D-1A and D-1T program expenditures since this work was performed in the same time frame by the same personnel and is in most cases applicable to both configurations.

The software costs for mission variations are based upon past experience and knowledge of required task scopes on the D-Centaur program. The data have been updated to reflect recent D-1 program experience and are expressed for first and follow-on flight for planetary and orbital type missions.

B. Ground Rules/Assumptions

1. Software costs contained herein are those associated with the development and checkout of on-board airborne computer programs for the 16K Digital Computer Unit (DCU) used for Atlas/Centaur and Titan/Centaur missions. Trajectory and performance, stability and control, and loads analyses are not interpreted as "software" and are not included. (These are estimated to be 55,000 hours and 1,000 computer hours.)

2. D-1A (Atlas/Centaur) and D-1T (Titan/Centaur) software development occurred during the same basic time period and, therefore, corresponding software costs were not individually segregated for each of these programs. A judgement/experience factor would indicate that either the D-1A or D-1T program would cost about 85 percent of the total cost, if done on a separate basis.

3. Software checkout costs are included herein. These costs are for development of Airborne Computer Software, Computer Controlled Launch Set (CCLS), and Flight Acceleration Profile (FAP) checkout software.

4. The SDS 930 computer is used to perform software checkout (CCLS and FAP). This computer is provided to Convair Aerospace Division as GFP.

C. Software Cost Data

1. D-1 Centaur Program Software Development Cost

Category	Man-Hours	Computer Hours	
		Cyber-70	Analog.
• Airborne Computer Software	69,284	395	32
• CCLS Factory Checkout Software	85,863	13	658
• CCLS ETR Checkout Software	8,580	---	---
• FAP Checkout Software	33,166	6	231
Totals	196,893	414	921

Notes:
a. Airborne computer software costs include support to software checkout activities listed.

b. Above data reflect D-1 software costs through delivery (DD-250) of the first D-1A and D-1T vehicle. Final flight program adjustments and flight support activities at ETR are not included.

2. Mission Costs for Airborne Computer Software

	Man-Hours		Computer Hours (Cyber-70)	
	Low	High	Low	High
Planetary Mission				
• First flight	4,800	10,000	26	69
• Follow-on flight	1,100	2,200	6	18
Orbital Mission				
• First flight	3,500	8,100	20	53
• Follow-on flight	500	1,100	1	2

Notes:
a. Low and high values shown reflect limits of program complexity, based on past experience.

D. Titan IIIE (Non-recurring)

A limited review was conducted for the Titan IIIE engineering analysis effort performed in support of the Titan/Centaur vehicle integration and mission support tasks. The Titan/Centaur integration tasks include trajectories, aerodynamics, venting, propulsion, staging, environmental stability, propellant and stability analysis.

Man-hours	74,000 hours
Computer hours	400 hours

E. Titan IIIE (Recurring) First Mission

The mission peculiar engineering analysis includes trajectory and performance, aerodynamic, propulsion, flight control, integrated loads, stability and range safety analysis.

Man-hours	25,600 hours
Computer hours	300 hours

We trust that these data will be helpful to you in your current study.

Andrew J. Stofan
Manager, Titan/Centaur Project Office

APPENDIX C

APOLLO/SKYLAB

MANPOWER, COMPUTER TIME REPORT

The following attachments have been provided by The International Business Machines Corporation, Houston, Texas, in response to a request for historical records related to the NASA Mission Control Center. These records represent one of the few available for distribution and provide a valuable insight into the level of support required for highly complex manned space operations. Use of the data is limited because the records are not traceable to an initial set of requirements. However, IBM has prepared a brief description of the scope of effort involved which provides a frame of reference for interpreting the level of effort and computer time shown. The data may therefore be useful for future efforts addressing software development and recurring costs, especially as related to flight operations support or Mission Control Center activities. It was therefore felt advisable to include the information provided for reference purposes.

APOLLO/SKYLAB

MANPOWER, COMPUTER TIME REPORT

05/15/74

International Business Machines Corporation
1322 Space Park Drive
Houston, Texas 77058

Attached is the history of the manpower and computer time for Apollo and Skylab

Attachment 1 shows the manpower expended by month in thousands of hours. Attachment 2 contains the parameters necessary to convert hours to equivalent man years and equivalent man months. Man hours expended has been broken into three major categories:

- o hours expended on Apollo programming systems
- o hours expended on Skylab programming systems
- o hours expended on Non Mission related activities

No attempt has been made or should be made to pro-rate the Non Mission hours to Apollo or Skylab programming because of the change of responsibilities in the Engineering and Operations categories between Apollo and Skylab. Caution should be used when interpreting the manpower for Apollo during 1971 and 1972 for the following reason. Manpower expended against Apollo during the maintenance period was exceptionally low. Though a large number of people was required in order to maintain competence and coverage of these large systems, these people were able to spend a significant portion of their time working on Skylab thus reducing the charges to Apollo.

Attachment 3 contains the computer time expended by month and is also broken down into the same three major categories as was above with man hours. Because of a unique characteristic of the OCCURS System, a significant amount of computer time was charged to utility programs. For convenience this time has been pro-rated between Apollo and Skylab in order to determine the total computer time expended for Apollo Programming and for Skylab Programming.

IBM's responsibility in support of the Mission Control Center RTCC consisted of the development, coding and validation of software operations on the five IBM 360-75 computers used in the RTCC with two exceptions; communications network interfaces, which were managed by Univac and LMSC, and programming of a subsidiary CDC computer which served as a data base storage unit. The IBM effort reported in the following tables included the major portion of the software development and recurring support costs for the RTCC. As such, it represents a reasonable base for projecting future levels of support for large complex programs, similar to Apollo and Skylab. In addition, to IBM's effort, other contractors also supported the RTCC. Philco-Ford was responsible for all control and display hardware and overall operational support which included contracted support from CDC, Univac and LMSC. However, the principal effort regarding software development was performed by IBM.

The IBM effort included the development of all software for mission operations control, simulator operations, support of the Real Time Operating System (RTOS), activity scheduling, General Software Support Computing, and earth resources interactive processor operations, as well as general computer operations support. On Apollo four of the five computers were primary with the fifth held in reserve. On Skylab four computers were used. One computer was dedicated to data storage functions for incoming experiment data. A second computer was used for retrieval of information from the mass data storage. The third was dedicated to the "Activity Scheduling Program" and the fourth to Simulation and Earth Resources. In general, each computer was capable of containing 1.1 million instructions of code and 4 million words of data storage.

As a result of the complex nature of the programs required for the Apollo and Skylab programs, it is recommended that further information regarding the data of this appendix be referred directly to IBM, Houston.

RTCC MANPOWER CY 70 (thousands of hrs) 1965

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP													
Mission									13.9	12.6	15.2	27.2	68.9
Checkout									1.4	1.9	2.1	5.4	10.8
GSSC									4.6	5.4	5.5	14.4	29.9
RTOS									4.8	4.9	5.4	10.7	25.8
Sys Anal									1.4	1.4	1.7	2.5	7.0
Total									26.1	26.2	29.9	60.2	142.4
SKYLAB													
Mission													
Checkout													
GSSC													
RTOS													
Sys Anal													
Terminal													
Total													
NON-MISSION													
Project Off.									4.0	3.4	4.1	7.5	19.0
Tech Serv									2.7	2.9	3.3	7.5	14.9
Engineering									4.7	5.0	5.5	7.5	22.7
Math									1.0	.6	.7	1.4	3.7
Maint and Op									12.9	13.0	14.3	18.0	58.2
Earth Res													
SLS													
Total									25.3	24.9	27.9	41.9	118.5

Attachment 1

RTCC MANPOWER CY 70 (thousands of hrs) 1966

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP													
Mission	3.8	16.6	23.1	21.3	24.3	30.0	23.3	24.9	30.1	25.0	23.8	36.5	282.7
Checkout	-	3.9	4.7	3.7	4.0	4.4	3.6	3.7	4.8	4.4	3.8	5.3	46.3
GSSC	.2	7.6	9.9	9.5	11.6	14.4	11.7	12.9	16.5	14.4	14.0	19.1	141.8
RTOS	5.0	7.4	7.7	6.3	7.3	7.6	6.5	6.5	7.4	6.1	5.0	8.2	81.0
Sys Anal	1.4	1.6	2.2	2.0	2.0	2.3	1.7	2.4	2.9	2.3	1.8	2.9	25.5
Total	10.4	37.1	47.6	42.8	49.2	58.7	46.8	50.4	61.7	52.2	48.4	72.0	577.3
SKYLAB													
Mission													
Checkout													
GSSC													
RTOS													
Sys Anal													
Terminal													
Total													
NON-MISSION													
Project Off.	0.5	3.8	3.7	3.7	2.2	3.6	2.7	5.0	6.6	5.6	3.4	6.1	46.9
Tech Serv	2.5	3.3	4.0	2.9	3.5	4.0	3.5	4.6	4.6	3.5	3.0	5.1	44.5
Engineering	4.1	6.0	8.2	6.3	5.6	6.2	4.1	5.5	6.1	5.1	4.1	5.9	67.2
Math	-	.8	1.1	.9	1.0	1.1	.8	.9	1.9	.9	.6	1.1	11.1
Maint and Op	10.3	14.6	17.9	15.3	17.4	20.8	18.5	20.3	23.1	20.2	15.1	28.5	222.0
Earth Res													
SLS													
Total	17.4	28.5	34.9	29.1	29.7	35.7	29.6	36.3	42.3	35.3	26.2	46.7	391.7

RTCC MANPOWER CY 70 (thousands of hrs) 1967

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP													
Mission	27.8	29.6	35.4	31.3	30.9	34.8	26.2	29.7	43.0	31.2	28.7	33.8	382.4
Checkout	4.0	4.6	5.5	5.3	4.2	5.5	4.4	4.0	5.8	4.3	.3	4.6	52.5
GSSC	14.4	15.2	19.7	16.6	15.9	16.6	13.7	15.3	18.5	15.8	6.2	17.4	185.3
RTOS	6.2	6.1	7.3	6.4	6.5	7.1	6.2	8.4	10.5	8.7	8.2	9.4	91.0
Sys Anal	1.6	2.5	2.9	2.8	2.9	2.9	2.8	3.4	3.9	2.9	2.8	3.5	34.9
Total	54.0	58.0	70.8	62.4	60.4	66.9	53.3	60.8	81.7	62.9	46.2	68.7	746.1
SKYLAB													
Mission													
Checkout													
GSSC													
RTOS													
Sys Anal													
Terminal													
Total													
NON-MISSION													
Project Off.	4.3	6.7	8.8	7.3	6.8	6.3	4.6	5.1	6.5	5.1	5.3	5.4	72.2
Tech Serv	3.0	3.6	4.5	3.6	3.6	4.4	3.6	3.9	3.7	3.5	3.4	3.9	44.7
Engineering	6.0	5.5	5.8	4.5	4.3	3.8	4.0	3.5	4.9	4.9	4.5	5.2	56.9
Math	.8	.8	1.0	.7	1.0	1.1	.7	.7	.9	.8	.6	.7	9.8
Maint and Op	16.4	6.5	24.4	18.5	18.2	20.8	18.5	19.0	20.9	16.5	15.0	19.5	214.2
Earth Res													
SLS													
Total	30.5	23.1	44.5	34.6	33.9	36.4	31.4	32.2	36.9	30.8	28.8	34.7	397.8

RTCC MANPOWER CY 70 (thousands of hrs) 1968

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP			.7	1.4	1.2	1.3	.6	1.0	1.9	1.8	2.3	2.6	14.8
Mission	28.1	30.1	38.7	28.5	32.1	33.4	26.7	30.3	20.3	30.2	27.4	29.7	355.5
Checkout	3.7	4.7	5.6	4.0	4.4	5.3	3.9	4.2	5.9	4.9	5.3	8.8	60.7
GSSC	14.8	16.4	19.6	15.1	16.3	17.3	13.3	14.7	17.1	14.9	14.1	15.6	189.2
RTOS	7.8	9.2	11.7	8.6	9.5	10.3	8.2	9.8	11.2	10.2	9.9	11.1	117.5
Sys Anal	3.6	3.8	4.7	3.9	3.8	4.4	3.4	5.0	5.7	5.1	4.7	5.2	53.3
Total	58.0	64.2	81.0	61.5	67.3	72.0	56.1	65.0	62.1	67.1	63.7	73.0	791.0
SKYLAB													
Mission													
Checkout													
GSSC													
RTOS													
Sys Anal													
Terminal													
Total													
NON-MISSION													
Project Off.	4.6	5.3	7.3	5.3	5.9	7.6	5.6	6.3	7.2	6.0	5.3	6.3	72.7
Tech Serv	2.9	3.3	4.3	3.3	3.2	4.1	3.8	4.0	4.5	3.9	3.5	4.3	45.1
Engineering	5.2	5.5	7.3	4.3	5.2	5.2	4.3	5.2	6.0	7.7	8.1	8.8	72.8
Math	.7	.8	.9	.5	.7	.6	.4	.5	.5	.4	.4	.5	6.9
Maint and Op	15.1	16.8	21.8	15.4	17.8	21.2	16.2	23.2	22.6	20.1	18.3	23.0	231.5
Earth Res													
SLS													
Total	28.7	31.7	41.6	28.8	32.8	38.7	30.7	39.2	40.8	38.1	35.6	42.9	429.0

C-2

RTCC MANPOWER CY 70 (thousands of hrs) 1969

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP	2.3	2.3	4.6	3.6	3.7	4.8	3.4	3.2	4.1	3.3	2.9	2.9	41.1
Mission	21.7	26.4	30.8	21.1	23.1	24.2	18.3	17.7	19.7	16.0	16.5	17.4	252.9
Checkout	5.1	5.3	7.3	5.1	4.9	4.5	2.8	3.2	3.5	2.6	2.7	2.8	49.8
GSSC	12.6	14.9	17.9	12.7	14.3	15.0	8.8	10.3	12.4	9.6	8.7	9.6	146.8
RTOS	8.3	10.7	12.5	8.2	8.0	9.7	6.5	7.0	8.2	6.7	6.9	6.9	99.6
Sys Anal	4.4	5.2	5.3	3.5	4.0	4.1	3.0	4.1	4.2	3.4	2.9	2.3	46.4
Total	54.4	64.8	78.4	54.2	58.0	62.3	42.8	45.5	52.1	41.6	40.6	41.9	636.6
SKYLAB													
Mission													
Checkout													
GSSC													
RTOS													
Sys Anal													
Terminal													
Total													
NON-MISSION													
Project Off.	4.7	6.1	7.7	5.3	6.1	6.7	5.0	5.5	6.7	5.2	5.1	3.2	67.3
Tech Serv	2.9	3.6	4.8	2.9	3.0	3.6	2.7	6.6	3.9	2.7	2.7	2.9	42.3
Engineering	4.2	5.4	4.7	4.5	4.5	4.1	2.5	2.9	3.9	1.3	3.9	4.5	46.4
Math	.5	.4	.4	.3	.3	.4	-	.4	.2	.2	.2	.2	3.5
Maint and Op	13.8	17.9	22.4	16.0	17.7	18.1	9.6	17.6	17.3	12.5	13.4	14.7	191.0
Earth Res													
SLS													
Total	26.1	33.4	40.0	29.0	31.6	32.9	19.8	33.0	32.0	21.9	25.3	25.5	350.5

RTCC MANPOWER CY 70 (thousands of hrs) 1970

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP	2.0	2.7	3.1	2.5	2.2	2.2	1.6	1.7	1.7	1.6	1.7	2.0	25.0
Mission	10.8	15.0	19.8	16.1	15.4	17.2	5.8	7.1	9.0	8.4	7.5	7.7	139.8
Checkout	1.8	2.2	2.4	2.0	2.5	3.5	2.7	2.7	2.8	1.4	1.7	2.0	27.7
GSSC	6.5	8.4	9.9	7.6	7.1	8.0	5.9	7.2	8.6	6.8	7.0	6.5	89.5
RTOS	4.5	6.3	7.7	6.6	6.3	6.9	3.9	4.2	5.4	5.0	4.7	5.0	66.5
Sys Anal	1.5	2.0	2.1	1.8	1.7	1.8	.7	.6	.9	.8	1.0	1.1	16.0
Total	27.1	36.6	45.0	36.6	35.2	39.6	20.6	23.5	28.4	24.0	23.6	24.3	364.5
SKYLAB													
Mission							.9	1.1	1.5	1.8	1.5	1.6	8.4
Checkout							.3	.6	1.3	1.2	3.2	1.2	7.8
GSSC							.7	.5	.8	1.6	1.9	2.6	8.1
RTOS							.6	1.1	1.6	1.3	1.4	1.6	7.6
Sys Anal							.7	1.2	1.8	1.3	1.4	1.7	8.1
Terminal							3.8	4.1	5.3	4.7	6.4	5.9	30.2
Total							7.0	8.6	12.3	11.9	15.8	14.6	70.2
NON-MISSION													
Project Off.	3.5	4.0	3.9	3.0	4.0	4.0	2.6	2.6	3.1	1.3	2.3	2.3	36.6
Tech Serv	1.9	2.1	3.0	2.4	1.0	2.3	1.4	1.8	1.6	1.3	1.2	1.6	21.6
Engineering	3.0	4.1	5.5	3.7	3.1	3.4	2.1	2.6	3.8	3.1	3.2	6.4	44.0
Math	.2	.1											.3
Maint and Op	9.4	12.3	14.8	12.0	10.0	13.3	3.7	10.1	13.2	11.3	11.4	20.2	141.7
Earth Res													
SLS							.5	1.0	1.2	.8	1.1	1.4	6.0
Total	18.0	22.6	27.2	21.1	18.1	23.0	10.3	18.1	22.9	17.8	19.2	31.9	250.2

RTCC MANPOWER CY 70 (thousands of hrs) 1971

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP	1.6	2.1	2.4	1.8	2.0	1.6	1.1	1.5	1.5	1.3	1.1	1.3	19.3
Mission	5.6	7.2	7.4	5.1	5.4	5.6	3.8	3.9	3.0	2.7	3.2	3.2	56.1
Checkout	1.6	2.6	3.1	1.6	1.8	1.3	1.2	1.1	1.2	1.2	1.2	.9	18.8
GSSC	4.4	5.2	5.9	3.9	3.8	2.5	1.4	1.1	1.8	1.6	.9	1.1	33.6
RTOS	3.6	5.3	6.8	4.6	5.9	3.2	2.0	2.0	1.6	1.4	1.4	2.5	40.3
Sys Anal	1.0	1.4	1.8	1.4	1.4	1.0	.7	.5	.4	.5	.2	.1	10.4
Total	17.8	23.8	27.4	18.4	20.3	15.2	10.2	10.1	9.5	8.7	8.0	9.1	178.5
SKYLAB													
Mission	1.5	2.2	4.4	4.3	6.8	7.8	7.1	7.3	10.5	8.7	8.9	7.9	77.4
Checkout	.6	.6	.9	.6	1.0	.9	.9	1.5	1.6	1.4	1.2	1.6	12.8
GSSC	2.1	3.9	5.5	4.5	6.0	8.1	7.2	8.1	9.2	7.5	8.2	9.0	79.3
RTOS	1.1	1.5	1.8	1.5	3.0	5.5	5.9	6.5	9.1	7.4	8.0	10.8	62.1
Sys Anal	1.3	1.7	2.1	1.4	1.5	2.2	1.9	2.3	2.7	2.5	1.3	2.7	23.6
Terminal	4.3	6.9	8.6	6.6	8.3	10.1	8.5	9.6	11.5	10.7	10.5	9.9	105.5
Total	10.9	16.8	23.3	18.9	26.6	34.6	31.5	35.3	44.6	38.2	38.1	41.9	360.7
NON-MISSION													
Project Off.	1.6	2.4	3.4	2.1	2.4	2.6	2.2	3.7	2.2	1.4	1.5	1.7	27.2
Tech Serv	.9	1.3	1.7	1.5	1.4	1.6	1.5	1.5	1.6	1.5	1.8	2.2	18.5
Engineering	2.2	2.8	2.9	2.0	2.1	2.2	1.3	1.7	1.6	2.0	2.0	2.1	24.9
Math													
Maint and Op	8.8	13.0	14.4	8.6	12.7	12.3	9.3	10.3	11.1	10.6	9.5	6.2	126.8
Earth Res						.1	.5	.7	1.0	.9	1.1	1.7	6.0
SLS	1.0	1.2	1.5	1.1	1.6	1.6	1.3	1.3	1.3	1.3	1.3	1.3	15.8
Total	14.5	20.7	23.9	15.3	20.2	20.4	16.1	19.2	18.8	17.7	17.2	15.2	19.2

RTCC MANPOWER CY 70 (thousands of hrs) 1972

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP	.8	1.0	1.3	1.1	1.1	1.3	.9	1.1	1.1	.6	.8	.7	11.8
Mission	1.8	1.9	2.4	2.3	1.5	1.0	.9	1.2	1.5	1.6	1.6	2.5	20.2
Checkout	.6	.8	1.2	.9	.8	.9	.7	.8	.9	.6	.7	.8	9.7
GSSC	.8	1.1	1.0	.5	.4	.2	.3	.6	.7	.1	.3	.1	6.1
RTOS	.9	1.5	1.7	1.7	1.1	.6	.4	.5	.5	.5	.4	.6	10.4
Sys Anal	.2	.2											.4
Total	5.1	6.5	7.6	6.5	4.9	4.0	3.2	4.2	4.7	3.4	3.8	4.7	58.6
SKYLAB													
Mission	6.7	11.2	14.1	10.7	11.7	14.3	9.7	12.6	16.4	14.0	14.8	17.2	153.4
Checkout	1.1	2.0	2.9	2.4	2.4	2.7	1.9	2.3	3.2	2.5	1.7	1.4	26.5
GSSC	5.8	8.0	9.8	8.1	9.1	10.9	8.1	9.0	10.8	8.3	8.1	9.6	105.6
RTOS	6.6	9.7	12.9	9.6	10.8	12.8	9.0	10.4	12.2	9.8	10.4	11.0	125.2
Sys Anal	1.8	2.6	3.7	2.5	2.6	3.1	2.3	2.6	2.9	2.2	2.2	2.5	31.0
Terminal	8.3	11.2	14.5	12.4	12.3	15.9	11.1	12.1	16.9	15.4	18.2	20.4	168.7
Total	30.3	44.7	57.9	45.7	48.9	59.7	42.1	49.0	62.4	52.2	55.4	62.1	610.4
NON-MISSION													
Project Off.	.8	1.3	1.6	1.1	1.1	1.4	1.0	1.0	1.2	1.0	.9	.9	13.3
Tech Serv	1.4	1.7	1.7	1.5	1.9	1.7	1.7	2.6	1.7	1.9	1.9	1.1	20.8
Engineering Math	1.4	1.9	2.0	1.6	1.9	2.0	1.0	1.6	1.7	1.5	1.2	1.6	19.4
Maint and Op	3.4	4.7	5.9	4.9	4.6	5.1	4.2	4.5	5.2	4.3	6.3	9.5	62.6
Earth Res	1.3	2.5	3.6	3.2	3.5	4.4	3.3	3.9	4.5	3.5	3.6	3.6	40.9
SLS	.5	.7	.5	.4	.3	.3	-	.4	.5	.3	.5	.6	5.0
Total	8.8	12.8	15.3	12.7	13.3	14.9	11.2	14.0	14.8	12.5	14.4	17.3	162.0

RTCC MANPOWER CY 70 (thousands of hrs) 1973

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
ALSEP													
Mission						1.5							1.5
Checkout	.2	.4	.3	.1	.2	.2							1.4
GSSC													
RTOS													
Sys Anal													
Total	.2	.4	.3	.1	.2	1.7							2.9
SKYLAB													
Mission	9.2	13.1	17.9	14.0	13.7	13.3	11.2	12.3	14.2	11.8	11.7	11.8	154.2
Checkout	.5	.8	1.0	.6	.8	.9	.4	.4	.4	.2	.4	.3	6.7
GSSC	5.3	7.4	9.0	6.5	6.1	3.4	4.3	4.1	4.8	4.0	3.5	3.3	61.7
RTOS	5.9	8.7	11.8	8.4	7.9	8.7	5.2	6.3	6.5	5.2	4.8	5.5	84.9
Sys Anal	1.5	2.3	2.8	2.4	2.1	2.0	1.3	1.2	1.6	1.2	1.0	1.0	20.4
Terminal	11.7	19.0	26.6	21.9	20.2	18.4	11.0	12.0	14.8	11.4	8.7	6.5	182.2
Total	34.1	51.3	69.1	53.8	50.8	46.7	33.4	36.3	42.3	33.8	30.1	28.4	510.1
NON-MISSION													
Project Off.	.2	.9	1.3	1.1	1.2	1.2	.8	1.1	1.2	1.0	.7	.9	11.6
Tech Serv	1.1	1.7	2.4	2.7	2.7	2.8	1.9	1.7	1.9	1.8	1.7	2.8	25.2
Engineering	.3	.3	.5	.3	.3	.3	.1	.2	.3	.3	.3	.3	3.5
Math													
Maint and Op	2.4	6.4	6.3	5.2	7.3	4.4	6.2	4.7	5.5	5.4	2.9	3.6	60.3
Earth Res	2.0	1.7	2.0	1.5	1.9	4.6	2.2	2.4	2.6	2.4	2.5	2.8	28.6
SLS	.4	.6	.7	.5	.4	.5	.2	.3	.4	.4	.4	.5	5.3
Total	6.4	11.6	13.2	11.3	13.8	13.8	11.4	10.4	11.9	11.3	8.5	10.9	134.5

Attachment 2

Hourly conversion factors

- o 1800 hours equals one equivalent man year
- o Conversion to equivalent man month

Month	Hours
January	100
February	140
March	170
April	140
May	140
June	170
July	140
August	140
September	170
October	140
November	140
December	210
Total	1800
Average	150

COMPUTER TIME (hrs) 1965

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission								27	249	456	531	404	1667
ALSEP													
GSSC								11	355	418	371	569	1724
CPS								9	153	176	495	523	1356
Sys Anal									1	2	7	21	31
Total								47	758	1052	1404	1517	4778
SKYLAB													
Mission													
Terminal													
GSSC													
CPS													
Sys Anal													
Total													
NON-MISSION													
M&S								2	26	29	89	72	218
Proj Mgmt									9	12	9	12	42
Engineering													
Total								2	35	41	98	84	260
TOTAL								49	793	1093	1502	1601	5038
M & S UTIL													
ProRate Apollo													
Tot Prog-Ap													
ProRate Skylab													
Tot Prog-Sky													

Attachment 3

COMPUTER TIME (hrs) 1966

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	429	481	676	809	1100	1388	1426	1632	1695	1850	1683	1565	14734
ALSEP													
GSSC	778	617	924	941	925	1048	1012	1067	1037	1107	980	987	11423
CPS	880	753	781	805	846	943	801	937	601	623	523	445	8938
Sys Anal	3	0	3	3	5	51	44	4	5	1	8	2	129
Total	2090	1851	2384	2558	2876	3430	3283	3640	3338	3581	3194	2999	35224
SKYLAB													
Mission													
Terminal													
GSSC													
CPS													
Sys Anal													
Total													
NON-MISSION													
M&S	159	145	135	126	124	262	128	132	227	172	161	161	1932
Proj Mgmt	10	21	22	20	12	11	18	37	37	26	14	13	241
Engineering													
Total	169	166	157	146	136	273	146	169	264	198	175	174	2173
TOTAL	2259	2017	2541	2704	3012	3703	3429	3809	3602	3779	3369	3173	37397
M & S UTIL													
ProRate Apollo													
Tot Prog-Ap													
ProRate Skylab													
Tot Prog-Sky													

COMPUTER TIME (hrs) 1967

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	1780	1716	1782	2142	2119	2192	2033	1585	1800	1604	1233	1484	21470
ALSEP													
GSSC	1216	1087	990	1043	1123	1060	1106	1103	1171	1058	578	824	12359
CPS	460	368	339	247	392	148	319	249	246	401	191	365	3725
Sys Anal	20	92	101	160	199	159	132	144	108	103	54	98	1370
Total	3476	3263	3212	3592	3833	3559	3590	3081	3325	3166	2056	2771	38924
SKYLAB													
Mission													
Terminal													
GSSC													
CPS													
Sys Anal													
Total													
NON-MISSION													
M&S	183	144	161	191	232	251	295	282	219	115	97	93	2263
Proj Mgmt	17	14	13	11	4	8	13	17	9	8	6	13	133
Engineering													
Total	200	158	174	202	236	259	308	299	228	123	103	106	2396
TOTAL	3676	3421	3386	3794	4069	3818	3898	3380	3553	3289	2159	2877	41320
M & S UTIL													
ProRate Apollo													
Tot Prog-Ap													
ProRate Skylab													
Tot Prog-Sky													

COMPUTER TIME (hrs) 1968

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	1678	1802	1716	1690	1917	1845	1817	1824	1648	1890	1705	1469	21001
ALSEP									6	19	27	82	134
GSSC	1005	958	1097	994	1113	987	1115	1122	1025	1190	1005	999	12610
CPS	408	643	773	585	654	721	699	676	502	560	517	463	7201
Sys Anal	120	169	157	166	177	161	95	209	135	169	137	121	1816
Total	3211	3572	3743	3435	3861	3714	3726	3831	3316	3828	3391	3134	42762
SKYLAB													
Mission													
Terminal													
GSSC													
CPS													
Sys Anal													
Total													
NON-MISSION													
M&S	112	153	150	111	100	135	155	188	201	143	130	56	1634
Proj Mgmt	13	13	12	23	23	29	30	30	44	35	48	19	319
Engineering	47							19					66
Total	172	166	162	134	123	164	185	237	245	178	189	75	2019
TOTAL	3383	3738	3905	3569	3984	3878	3911	4068	3561	4006	3569	3209	44781
M & S UTIL													
ProRate Apollo													
Tot Prog-Ap													
ProRate Skylab													
Tot Prog-Sky													

COMPUTER TIME (hrs) 1969

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	1683	1455	1266	1376	1082	1056	764	690	647	725	487	615	11846
ALSEP	172	185	206	264	266	395	180	157	201	148	116	103	2393
GSSC	873	796	867	1002	951	666	360	536	553	538	440	487	8069
CPS	361	233	352	386	358	420	276	273	207	220	242	151	3479
Sys Anal	157	116	101	93	227	113	39	58	60	54	35	25	1078
Total	3246	2785	2792	3121	2884	2650	1619	1714	1668	1685	1320	1381	26865
SKYLAB													
Mission													
Terminal													
GSSC													
CPS													
Sys Anal													
Total													
NON-MISSION													
M&S	133	103	98	148	192	279	224	326	246	240	189	220	2398
Proj Mgmt	31	26	24	19	29	35	23	25	26	36	36	53	363
Engineering													
Total	164	129	122	167	221	314	247	351	272	276	225	273	2761
TOTAL	3410	2914	2914	3288	3105	2964	1866	2065	1940	1961	1545	1554	29626
M & S UTIL									85	134	100	150	469
ProRate Apollo									85	134	100	150	469
Tot Prog-Ap	3246	2785	2792	3121	2884	2650	1619	1714	1753	1819	1420	1531	27334
ProRate Skylab													
Tot Prog-Sky													

COMPUTER TIME (hrs) 1970

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	550	485	502	427	451	476	418	436	565	537	479	474	5800
ALSEP	115	83	57	52	50	54	92	76	46	68	29	36	758
GSSC	428	401	377	463	529	503	454	371	354	331	248	158	4617
CPS	138	142	119	164	174	248	157	137	129	160	192	183	1943
Sys Anal	33	30	3	2	2	1	31	18	39	49	41	39	288
Total	1264	1141	1058	1108	1206	1282	1152	1038	1133	1145	989	890	13406
SKYLAB													
Mission							1	1	7	0	1	1	11
Terminal							59	48	41	35	41	42	266
GSSC										6	8	18	32
CPS												2	2
Sys Anal											1	4	5
Total							60	49	48	41	51	67	316
NON-MISSION													
M&S	184	171	172	182	134	150	155	171	168	276	262	311	2336
Proj Mgmt	32	28	36	48	49	65	50	70	50	52	40	15	535
Engineering							27	6	44	2	5	2	86
Total	216	199	208	230	183	215	232	247	262	330	307	328	2957
TOTAL	1480	1340	1266	1338	1389	1497	1444	1334	1443	1516	1347	1285	16679
M & S UTIL	132	150	139	137	132	149	140	130	132	214	172	139	1766
ProRate Apollo	132	150	139	137	132	149	133	124	127	207	164	129	1723
Tot Prog-Ap	1396	1291	1197	1245	1338	1431	1285	1162	1260	1352	1153	1019	15129
ProRate Skylab							7	6	5	7	8	10	43
Tot Prog-Sky							67	55	53	48	59	77	359

COMPUTER TIME (hrs) 1971

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	574	400	567	545	367	310	213	180	221	225	286	147	4035
ALSEP	39	55	62	68	50	32	56	38	54	38	26	22	540
GSSC	238	238	349	321	304	232	166	149	179	131	87	99	2493
CPS	113	133	194	221	184	200	111	134	121	129	121	51	1712
Sys Anal	52	89	106	47	44	37	4	15	10	8	1	3	416
Total	1016	915	1278	1202	949	811	550	516	585	531	521	322	9196
SKYLAB													
Mission	12	20	41	12	27	49	47	57	110	166	167	121	829
Terminal	32	42	44	42	22	20	21	38	53	85	107	74	580
GSSC	40	36	150	71	104	156	256	318	344	398	377	345	2595
CPS					6	48	124	179	147	138	137	297	1076
Sys Anal	2					22	38	72	67	53	108	82	444
Total	86	98	235	125	159	295	486	664	721	840	896	919	5524
NON-MISSION													
M&S	367	236	323	342	378	329	459	436	474	549	500	615	5008
Proj Mgmt	11	10	8	7	8	7	7	7	7	13	7	8	100
Engineering	5	4	4	4	18	9	0	21	0	0	11	3	79
Total	383	250	335	353	404	345	466	464	481	562	518	626	5187
TOTAL	1485	1263	1848	1680	1512	1451	1502	1644	1787	1933	1935	1867	19907
M & S UTIL	215	120	195	188	235	211	270	276	313	322	448	563	3356
ProRate Apollo	198	108	165	170	201	155	143	121	140	125	165	146	1837
Tot Prog-Ap	1214	1023	1443	1372	1150	966	693	637	725	656	686	468	11033
ProRate Skylab	17	12	30	18	34	56	127	155	173	197	283	417	1519
Tot Prog-Sky	103	110	265	143	193	351	613	819	894	1037	1179	1336	7043

COMPUTER TIME (hrs) 1972

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	163	117	145	77	94	115	182	151	147	137	48	13	1389
ALSEP			30	24	40	61	68	68	31	35	22	3	382
GSSC	102	99	62	44	29	19	35	51	21	38	5	3	508
CPS	72	83	55	53	28	9	4	25	2	3	66		400
Sys Anal	10	4											14
Total	347	303	292	198	191	204	289	295	201	213	141	19	2693
SKYLAB													
Mission	187	232	242	231	421	261	355	378	506	771	485	342	4411
Terminal	119	158	238	256	487	488	534	419	561	918	661	560	5399
GSSC	553	294	294	230	361	324	538	442	386	431	254	195	4302
CPS	379	227	238	184	455	407	338	387	252	247	186	160	3460
Sys Anal	105	62	40	30	42	35	32	17	17	26	19	32	457
Total	1343	973	1052	931	1766	1515	1797	1643	1722	2393	1605	1289	18029
NON-MISSION													
M&S	857	732	728	559	815	482	467	390	264	363	292	187	6136
Proj Mgmt	5	3	8	5	6	1	5	2					35
Engineering	7		1	5	9	4					6	2	34
Total	869	735	737	569	830	487	472	392	264	363	298	189	6205
TOTAL	2559	2011	2081	1698	2787	2206	2558	2330	2187	2969	2044	1497	26927
M & S UTIL	772	673	679	526	782	462	424	379	259	361	288	180	5785
ProRate Apollo	159	160	148	92	76	55	59	58	27	30	23	3	890
Tot Prog-Ap	506	463	440	290	267	259	348	353	228	243	164	22	3583
ProRate Skylab	613	513	531	434	706	407	365	321	232	331	265	177	4895
Tot Prog-Sky	1956	1486	1583	1365	2472	1922	2162	1964	1954	2724	1870	1466	22924

COMPUTER TIME (hrs) 1973

	J	F	M	A	M	J	J	A	S	O	N	D	TOTAL
APOLLO													
Mission	21	21	12	9	7	5	1	1	8	17	1		103
ALSEP													
GSSC					4	5		2	3	7	9		30
CPS	2												2
Sys Anal							8			6			14
Total	23	21	12	9	11	10	9	3	11	30	10		149
SKYLAB													
Mission	446	455	369	268	106	86	183	48	56	106	26	16	2165
Terminal	994	934	1057	1270	566	374	627	361	324	320	79	22	6928
GSSC	433	303	269	193	93	76	97	25	10	11	5	1	1516
CPS	252	198	220	218	111	79	81	48	67	64	90	12	1440
Sys Anal	24	32	34	42	18	11	11	8	7	23	11	4	225
Total	2149	1922	1949	1991	894	626	999	490	464	524	211	55	12274
NON-MISSION													
M&S	399	313	334	312	169	111	171	82	104	209	101	69	2374
Proj Mgmt													
Engineering													
Total	399	313	334	312	169	111	171	82	104	209	101	69	2374
TOTAL	2571	2256	2295	2312	1074	747	1179	575	579	763	322	124	1497
M & S UTIL	389	310	260	302	123	94	127	65	88	150	79	50	2037
ProRate Apollo	4	3	2	1	0	0	0	0	0	0	0	0	10
Tot Prog-Ap	27	24	14	10	11	10	9	3	11	30	10	0	159
ProRate Skylab	385	307	258	301	123	94	127	65	88	150	79	50	2027
Tot Prog-Sky	2534	2229	2207	2292	1017	720	1126	555	552	674	290	105	14301

THE AEROSPACE CORPORATION

EXTERNAL DISTRIBUTION LIST

(REFERENCE: COMPANY PRACTICE 7-21-1)

REPORT TITLE

Operations Analysis (Study 2.1) Shuttle Upper Stage Software Requirements

REPORT NO.	PUBLICATION DATE	SECURITY CLASSIFICATION
ATR-74(7341)-4	15 July 1974	Unclassified

MILITARY AND GOVERNMENT OFFICES	ASSOCIATE CONTRACTORS AND OTHERS

(NOTE: SHOW FULL MAILING ADDRESS; INCLUDE ZIP CODE, MILITARY OFFICE SYMBOL, AND "ATTENTION" LINE.)

NASA Scientific & Technical
Information Facility
P. O. Box 33
College Park, Maryland 20740 (3)

NASA - Headquarters
Washington, D. C. 20546

V. Huff, Code MTE (50)

R. R. Carley, Code MTE (2)

Dr. J. W. Wild, Code MTE (1)
New Technology Representative,
Code KT (1)

NASA
Mr. Duncan Collins
P. O. Box 92960
Worldway Postal Center
Los Angeles, CA 90009
Building 120, Room 1406B

AFR 80-45 DISTRIBUTION STATEMENT X'D BELOW APPLIES

☐ NO DISTRIBUTION STATEMENT
(Classified documents only)

☐ A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED

☐ B. DISTRIBUTION LIMITED TO U. S. GOV'T AGENCIES ONLY;

______________________________;
(Reason)

______________. OTHER REQUESTS FOR THIS DOCUMENT
(Date statement applied)

MUST BE REFERRED TO ______________________.
(Controlling DOD office)

APPROVED BY Robert K Wolfe DATE 7/1/74
(FOR THE AEROSPACE CORPORATION)

APPROVED BY ______________________ ____________ DATE ____________
(FOR COGNIZANT AF OFFICE) (SYMBOL)

IF LIST COMPRISES TWO OR MORE SHEETS, COMPLETE
THIS SIGNATURE BLOCK ON LAST SHEET ONLY

SHEET ______ OF ______