MSC Internal Note
No. 70-FS-2

UNCLASSIFIED

MSC INTERNAL NOTE No. 70-FS-2

PROGRAMMED GUIDANCE EQUATIONS
for
LUMINARY 1C
MANNED LM EARTH ORBITAL
AND LUNAR PROGRAM

Prepared by
Flight Software Branch

FLIGHT SUPPORT DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Approved: ______________________
J. C. Stokes, Jr., Chief
Flight Software Branch

Approved: ______________________
Lynwood C. Dunseith, Chief
Flight Support Division

UNCLASSIFIED

Page Change Record

This document is a complete re-issue of MSC Internal Note No. 69-FS-4, "Programmed Guidance Equations for LUMINARY 1B Manned LM Earth Orbital and Lunar Program," dated September 1969, updated to reflect the information in the LUMINARY 1C program to be flown on the Apollo H2 mission. The coding changes between LUMINARY 1B and 1C are indicated by a vertical line to the left of the affected area.

This document supercedes all previous issues and changed pages will be based on this issue in the future.

Abstract

The information presented in this document on the LUMINARY 1C guidance program was produced with the intention that it be used together with a symbolic tabulation of the program. The information is divided into a series of separate sections, each of which describes a basic area of guidance computation and contains a list of definitions of variables and constants used in that area of the program. In order to assist the user in finding the computations in which he is interested, summaries of each section have been included, and all routine tags used in this document (generally identical to but a subset of those in the program listing) are indexed at the end of the document along with a list of references to each routine listed. A list of references to flagwords and channels has been included as well, as a supplement to the list of references to variables and constants supplied in the program listing itself.

The program from which this document was prepared is identified "LUM131A Revision 3" and was released on January 23, 1970 for fabrication of the LM Guidance Computer memory ropes for the Apollo H2 mission.

Because of the purposes for which the information in this document was originally prepared, and the methods used in its production, this material should not be used as definitive information on the LUMINARY 1C program but as an aid in the reading and understanding of the program listing. If definitive information is required, the G&N contractor is the proper source for it.

Table of Contents

Introduction 1

Summaries of Individual Sections 3

Notation and Terminology 7

Major Variables 19

Erasable Memory Initialization 25

Channels, Flagwords and Other Discrete
Information Registers 37

List of References to Flagwords and Channels . . . 75

Sections

- Alignment of the Inertial Sub-System . . . ALIN - 1
- Ascent Guidance . . . . ASCT - 1
- Attitude Maneuver Routines . . . . . . ATTM - 1
- Burn Control Routines . . . . BURN - 1
- Conic Subroutines . . . . CONC - 1
- Coordinate Transformations . . . . . . COOR - 1
- Digital Autopilot Control Routines . . . DAPA - 1
- Digital Autopilot Phase Plane Logic . . DAPB - 1
- Data Input/Output Routines . . . . . . DATA - 1
- Descent Guidance . . . . DESC - 1
- Display Interface Routines . . . . . . DINT - 1
- Display and Keyboard Interface Logic . . DSKY - 1
- Extended Verbs . . . . EXVB - 1
- IMU Computations . . . . IMUC - 1
- Program Interrupts . . . . INTR - 1
- Mathematical Functions,Executive,Waitlist. MATX - 1
- Orbital Integration . . . . ORBI - 1
- Program Service Routines . . . . PGSR - 1
- Radar Control Routines . . . . RADR - 1

Rendezvous Navigation . . . . RNAV - 1
Servicer . . . . SERV - 1
Up and Down Telemetry . . . . TELE - 1
Testing Routines . . . . TEST - 1
Targeting - Lambert . . . . TRGL - 1
Targeting - External Delta-V . . . . . TRGX - 1

Tables

Program and Routine List (equivalence definition) . W - 3
Noun List . . . . W - 5
Alarm Codes . . . . W - 15
Checklist Codes . . . . W - 19
Option Codes . . . . W - 21
Job Priorities . . . . W - 23

Index of Routines . . . . X - 1

Introduction

Under the egis of the Program Development Group, Apollo Guidance Program Section, Flight Software Branch of MSC, in order to facilitate the reading of the detailed symbolic listing, a "Programmed Guidance Equations Document" has been prepared for the "LUMINARY" program. A major purpose of this document has been to provide more effective identification and analysis of various program performance features and to permit more effective review of published computer program documentation.

During reviews of previous programs written for the Apollo Guidance Computers, it was found desirable to assemble a set of working-paper information on the equations actually programmed for these flights. This material has proven to be useful to the various groups associated with these flights, in that it can be used to bridge the gap between the extreme detail of the program listing and the occasional lack of detail available elsewhere on the guidance equations. Consequently, the material on the following pages has been assembled in a fashion similar to that used for previous programs and follows the same general format.

Certain aspects of the program are quite complex, and this programmed guidance equation material should not be considered as a substitute for actual study of the program symbolic listing itself. No complete set of equation information was available from the G&N contractor against which the programmed equations could be validated, and in the interest of timely publication, the review of the assembled document against the program assembly has not been as detailed as would be desired.

The program assembly listing which was used to prepare this programmed equation information bears the heading print:

GAP: ASSEMBLE REVISION 003 OF AGC PROGRAM LUM131A BY NASA 2021112-111

and is dated January 23, 1970. The function of virtually all the program steps of interest to the flight is described either on the following pages, or, for general computer system control, in TRW Working Paper 3420.5-27 (revision 2).

Recipients of this document are cautioned against misusing it as a definitive description of the "LUMINARY" guidance equations. Instead, it might be used to achieve a better understanding of the program assembly listing, since it is intended as an aid in review of the listing, not as a substitute for it. Definitive guidance equation information can be provided only by the G&N contractor through the appropriate MSC channels.

A great deal of credit goes to TRW Systems MTCP Tasks A-90 and A-201 (Support of Apollo Guidance Program and Guidance Document Review) personnel, in particular Mr. William C. Koelsch, who conducted a similar review of the "SUNDANCE" program. This document has drawn heavily upon the results of that review and could not be published at this time without the earlier work done by TRW Systems.

Summary of Individual Sections

The contents of this description of the "LUMINARY" guidance and control equations are divided into twenty-five semi-independent sections, each of which is assigned a four letter code. Pages are numbered consecutively within each section and the sections themselves are arranged in alphabetical order by code. Familiarity with the information in "Notation and Terminology" is helpful in understanding the somewhat specialized type of notation used in describing the program, but each section includes a list of "Quantities in Computations" which is intended to describe all variables and constants in the section that are not described in the list of "Major Variables".

Alignment of the Inertial Subsystem (ALIN)

Programs used to align the ISS to any of several specified alignments and to compute the "reference to stable-member" transformation matrix, based on measurements of the positions of celestial bodies with respect to the spacecraft.

Ascent Guidance (ASCT)

Programs used to initiate, control, and terminate the LM ascent from the lunar surface (P12) and aborts from powered descent (P70 and P71).

Attitude Maneuver Routines (ATTM)

Computations performed to determine the axis about which spacecraft rotation should take place and the magnitude of the rotation to go from present vehicle attitude to final attitude, including logic controlling the maneuver rates and DAP interface.

Burn Control Routines (BURN)

Programs used to initiate, control, and terminate all three types of LM burns (RCS, APS, and DPS).

Conic Subroutines (CONC)

Subroutines used by navigation and targeting routines to compute various conic parameters.

Coordinate Transformations (COOR)

Subroutines defining the transformations between the several coordinate systems used by the LGC, including routines for determination of lunar and solar position and selenographic latitude and longitude.

Digital Autopilot Control Routines (DAPA)

P-axis and Q,R-axis RCS control for free or powered flight; Q,R-axis Gimbal Trim System. (Includes jet fail monitor.)

Digital Autopilot Phase Plane Logic (DAPB)

Equations used to compute jet accelerations, jet firing times and disturbing accelerations; deadband selection and astronaut interface routines.

Data Input/Output Routines (DATA)

Logic used to control display and loading of various LGC registers under control of the DSKY or internal programs, including noun definitions.

Deseent Guidance (DESC)

Programs used to initiate, control, and terminate the LM powered descent maneuvers (P60's).

Display Interface Routines (DINT)

Routines governing program use of the DSKY and the priority of displays.

Display and Keyboard Interface Logic (DSKY)

Routines defining the mechanics of interpreting inputs from the DSKY keyboard or uplink and for changing the status of numerical character lights on the display.

Extended Verbs (EXVB)

Definition of the functions of the extended verbs (40-99) including those controlling block updates of LGC E-memory, AGS initialization, calculation of TFF and rendezvous displays.

IMU Computations (IMUC)

Computations associated with the Inertial Measurement Unit, including those for controlling CDU pulse outputs, for checking and setting IMU modes and switching between them, for accelerometer and gyro compensation, and for gyro torquing.

Program Interrupts (INTR)

Short description of all eleven program interrupts; the routines associated with program interrupt No. 4.

Mathematical Functions, Executive, Waitlist (MATX)

Various built-in trigonometric, logarithm and root extraction functions used by the programs, and some of the logic associated with the operations, such as "Establish," or "Call."

Orbital Integration (ORBI)

The equations used for precision integration of the state vector, and logic to switch between powered flight and coasting flight navigation.

Program Service Routines (PGSR)

Routines used for initialization of the LGC, re-initialization in case of restart, and for change of program (major mode). Also includes alarm routines.

Radar Control Routines (RADR)

Routines controlling the positioning and reduction of data from the Landing Radar and the Rendezvous Radar.

Rendezvous Navigation (RNAV)

Programs using the rendezvous radar to update both state vectors maintained in the LGC and to update the "error transition matrix" based on tracking of the CSM.

Servicer (SERV)

Average-G navigation routines and other routines used for burn control and monitoring.

Up and Down Telemetry (TELE)

Uplink character processing and computations performed for periodic downlink transmission, including a brief summary of the information on the downlink.

Testing Routines (TEST)

Computer self-test routines and externally initiated tests.

Targeting - Lambert (TRGL)

Burn targeting using the Lambert computation of velocity-to-be-gained.

Targeting - External Delta-V (TRGX)

Burn targeting for constant attitude burns.

Notation and Terminology

Because of some of the special design features of the Apollo Guidance Computer, a set of special notation and terminology has been found useful in describing the equations programmed for this computer. In most cases, this notation and terminology follows that which seems to be employed by the G&N contractor, and also follows that which was used in documents previously prepared on Block 1 and Block 2 programs.

The following document may be found useful for supplemental information on the symbolic listing, and for a more detailed discussion of the computer hardware and general computer system control:

3420.5-27, "Apollo Guidance Program Symbolic Listing Information for Block 2," Revision 2, dated 27 June 1968, prepared under MTCP Task A-90.

Copies of that document, together with revision information as it is published, may be obtained from the Flight Software Branch of MSC.

Numbers

A. General

The guidance computer is designed with a memory word length of 15 bits (plus a sixteenth bit, not sensed by the program, used to achieve "odd parity", i.e., an odd number of binary ones in the total 16-bit word). It is also designed as a fractional machine, so that all numbers in the computer are less than one: "equation values" greater than 1.0 are accommodated by suitable scaling, as described below. Arithmetic is all one's complement except in special instances where two's complement is required for hardware interface information. There is no hardware floating point capability in the computer, although a facsimile to floating point computations is sometimes used (and identified as "quasi-floating point") in the program, particularly when the quantities involved can have a large dynamic range.

The 15-bit word is divided into a sign bit and 14 magnitude bits, with the bits numbered from the sign (#15) to the least significant magnitude bit (#1) as shown:

Value:	Sign	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}	2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}
Number:	15,	14,	13,	12,	11,	10,	9,	8,	7,	6,	5,	4,	3,	2,	1

Once the stored value of a number in the computer register (evaluated using the bit weights shown) has been determined, its equation value may be determined by multiplying the stored value by some power of two which is called the "scale factor":

> The scale factor of a quantity is the power of two by which the number in the computer (considered as a fraction in the range between -1 and +1) must be multiplied to obtain its equation value. For convenience, the scale factor is shown as "Bxx", where "xx" is some positive or negative integer.

For example, if a word has a scale factor B14, this means that the individual bits have a weight 2^{14} (or 16384) times the values shown in the above table. A counter in the program with this scale factor, therefore, would have its least significant magnitude bit (or least increment) equal to 1 ($2^{-14} \times 2^{14} = 1$). See Appendix A of 3420.5-27 (Revision 2) for more details.

In many instances, the word length of 15 bits, permitting a number to be expressed to 1:±16384 (1 part in ±16384), is insufficient to give the required precision in the computations. In these instances, a double precision number (stored in two consecutive memory cells) is used. Scale factor information for double precision numbers has the same interpretation as for single precision numbers, and the less significant half of the word has weights that are 2^{-14} times the weights shown above. A quantity that is double precision with a scale factor of B28, for example, would have a least significant bit of 1 and a maximum value of (2^{28} - 1), or 268,435,455. In a few cases (usually involving time information), triple precision quantities are required, which follow similar rules.

In cases where it is not reasonably self-evident, the single, double, or triple precision nature of the quantity is specified (by subscripts sp, dp, and tp, respectively).

In addition to the scale factor information, it is necessary to know the units in which the quantities are expressed. Times are usually expressed in units of "centiseconds" (0.01 seconds). The navigation equations give a position output in units of meters and a velocity output in units of meters per centisecond. Angles are usually expressed in units of revolutions (of 360°). Where not reasonably obvious, units on the quantities are given.

Unless otherwise specified, numbers with explicit values are quoted in decimal. A notation such as "1E-5" means 10^{-5}. Octal numbers are indicated by a subscript of 8 (such as 34_8 for octal 34); binary numbers have a subscript 2 (such as 11100_2 for 34_8 expressed in binary). The quantity +0 is 00000_8; -0 is 77777_8 (in some cases, the distinction is important, since a computer instruction can distinguish between them).

B. Constants

Fixed scalar constants are denoted by K:xxx, where xxx is a symbol of arbitrary type and length, selected generally for similarity with the program notation (capital letters) or for mnemonic usefulness (small letters). A subscript can be used to indicate one of several constants stored in a table.

Constants stored in erasable memory (so they can be changed without hardware implications) are not distinguished from variables. Some are set by the fresh start or restart routines; some are loaded as part of the erasable load.

In a few cases, vector or matrix constants are employed. They are so designated by an underline or brackets as are vector and matrix variables.

C. Variables

Variables are generally designated by several upper case capital letters, sometimes with subscripts (the symbol K: is reserved for constants). Lower case letters sometimes appear with the symbol for a variable, and have the following special meanings:

d: A division indication appears as part of the symbol in the program (1dANET appears in the listing as 1/ANET).

i: A suitable number or capital letter is substituted as defined in the related equation information for "i" (VACiUSE = VAC3USE if i = 3, etc.)

m: A minus sign appears as part of the symbol in the program (mTPER appears in the listing as -TPER).

p: A plus sign appears as part of the symbol in the program (pMGA appears in the listing as +MGA).

An underline of the first character of a symbol means a quantity with several components, frequently a vector but sometimes a quantity having vector-type properties, such as gyro compensation about different axes. Individual components of a vector are identified by a subscript using the same symbol as for the vector, but without the underline: TS_x, for example, would be the X component of the vector $\underline{T}S$.

Matrices are designated by enclosing their identification mnemonics in brackets: [REFSMMAT] for instance. Interpretive language operations in the guidance computer (see TRW Working Paper 3420.5-27 (Revision 2)) permit convenient manipulations of 3x3 matrices whose elements are stored double precision. Therefore, larger matrices are handled nine elements at a time. Both pre-multiplication and post-multiplication of a vector by a matrix are incorporated (post-multiplication is equivalent to pre-multiplication by the transpose). Elements of a 3x3 matrix are stored "first row, then second row, then third."

D. Addresses

The value of an address is designated by the symbol used in the program, enclosed in quotation marks: without the quotation marks, the contents of the cell with that address would be indicated. In order to improve presentation clarity, many of the program-step symbols are omitted from the equations, and others may not precisely correspond to the program divisions actually located by the symbol in question. It is sometimes necessary to refer to a computer address as actually packed into a 15-bit word: for this, the term "CADR" (see 3420.5-27 (Revision 2)) is employed (erasable and fixed memory CADR formats differ).

A number of subroutines are used within the program, each of which require information on the "main" program to which program control must be transferred at the end of the subroutine. For clarity in showing the computation flow performed, the retention of the necessary return address information is shown explicitly in a few places. The "return address" is the address to which control is to be transferred after completion of the subroutine; the "calling address" is the address from which transfer to the subroutine was made. In many cases, the return address is one greater than the calling address, but in some instances, such as transfers to the "ALARM" routine, the cell following the calling address contains information pertinent to the subroutine (such as the alarm pattern), and therefore the return address may be several address locations after the calling address. To save program steps, in some cases the complement of the address may actually be used by the program for storage purposes; but this fact, since it has no effect on the computation flow, is not indicated in the equations.

E. Subscripts

Subscripts are used for relative addressing or for informational purposes. As relative addresses, they index one of several variables or constants stored in a table with only one explicit address. $DSPTEM1_1$ indicates the cell after DSPTEM1, and $DSPTEM1_2$ indicates the cell after $DSPTEM1_1$. Sometimes the subscript contains the complete address and any arbitrary erasable memory is selected by the notation E_{ADR} where the address of the cell desired is stored in ADR. When used for informational purposes, the subscript does not change the address of the cell but merely indicates its nature. The following subscripts are frequently used:

sp: Single precision

dp: Double precision

tp: Triple precision

ms, ls: More and less significant halves of a double precision number

x, y, z: First, second and third components of a three-dimensional vector

2 (with a number): Binary

8 (with a number): Octal

(Numbers without a subscript are quoted in decimal)

11, 12, 13, 14, 15, ...: Elements of the first row of a matrix

21, 22, 23, 24, 25, ...: Elements of the second row of a matrix

31, 32, 33, 34, 35, ...: Elements of the third row of a matrix

41, 42, 43, 44, 45, ...: Elements of the fourth row of a matrix

etc.

Program Control

Three types of program sub-units occur within the complete program:

a) A subroutine, which performs a certain function and then returns control to the program sub-unit which called it (subroutines, of course, may have other subroutines within them).

b) A task, which is a short sequence of computations performed based on a time criterion, or upon some external signal.

c) A job, which is a program entity (such as targeting computations, steering computations, processing of a keyboard character, etc.) of long duration which must be done in a definite sequence. (Accelerometer data, for example, must be corrected for biases before navigation computations are performed, and navigation before steering commands, so they all form part of the same job.)

Time-dependent tasks are implemented by a "waitlist" system (see section VIIA of 3420.5-27), for which the programmer merely specifies the time delay (from "now") when he wants a computation done, and the starting address of that computation. The time delay has a least increment of 10 milliseconds. Unless interrupts are inhibited, a program interrupt (which can be caused by a signal from the telemetry system, the uplink, the "waitlist" hardware, separate ("T4RUPT") waitlist-type hardware, a keyboard input,etc.) causes suspension of a job and performance of the task. Tasks, however, are not subject to interruption by other tasks, but continue to completion. There is also a hardware monitoring function which can cause a "program/hardware restart" which, if necessary, could interrupt a task: in general, however, these should not be encountered.

Jobs are sequenced with the aid of a priority system (see section VIIB of 3024.5-27(Rev. 2)) and are performed only if no tasks must be performed. If no "productive" computations are required, then a "dummy job" is performed, which checks periodically for the availability of a job to be performed, and of course is subject to interruption for a task. A job can be "established" (put into a list to be selected when its priority is sufficiently high) by another job or by a task. A job can be "put to sleep" to wait for some external event (such as an uplink input), the occurrence of which will "awaken" the job. Jobs can optionally be assigned a set of "working storage" erasable memory cells, called a "VAC area" (see Section VID of 3420.5-27 (Revision 2)).

A program "step" is a step in the computation sequence shown on the page, and should not be confused with a line of the program assembly listing.

The following program-control terms are employed:

a. Awaken a job: Cause a job (if any) with the indicated starting address to be restored to its original priority after a period of being "asleep", during which the performance of the job was suspended by making its priority negative.

b. Call "XXX" in yy seconds: Cause a task with starting address of "XXX" to be performed in yy seconds from the present time (yy has a least increment of 10 milliseconds).

c. Delay yy seconds: Cause the present string of computations to be suspended for yy seconds, and then restart at next line. Delays to a Waitlist Task are implemented by a waitlist call to the following step, and then an end task; delays for a job are achieved by using the "DELAYJOB" routine.

d. End job: Terminate performance of the job, and transfer control to an executive routine to initiate performance of the job which has the highest priority of those remaining. The "dummy job" is the only one which is not ended in this fashion.

e. End task: Terminate performance of the waitlist-initiated task, and transfer control to a routine which checks for other waitlist tasks, causing resumption of previous computations if there are no such tasks.

f. Establish "XXXX": Enter the job with starting address "XXXX" in the priority list to be performed when appropriate (each job has a priority associated with it, not necessarily shown in the equations). A job can optionally be established with or without a working storage (VAC) area (not generally shown in the equations).

g. If: Carry out the indicated manipulations provided the indicated conditions are satisfied. The "indicated manipulations" are "indicated" by being indented, if the "If" statement is followed by a colon, or by being on the same line as an "If" statement followed by a comma. Should the conditions not be satisfied or after performing "indicated manipulations" that do not end with a "Proceed to...", continue on in sequence at the next non-indented line.

h. Perform "XXXX": Cause a subroutine with starting address "XXXX" to be entered. The specific memory cell to which the subroutine returns control depends in some cases on the purpose of the routine, but generally is the step after the "Perform" instruction.

i. Proceed to "XXXX": Cause the program step with address "XXXX" to be the next one to be performed, and continue the performance of the program from that point.

j. Proceed to XXXX: Cause the program step whose address is stored in cell XXXX to be the next one performed, and continue the performance of the program from that point.

k. Put to sleep: Cause the present job's computations to be suspended (by making its priority negative) until some event takes place causing the job to be "awakened." Differs from End job and Establishing a job in that the VAC area and Job Register Set are retained. When a job is put to sleep, a starting address identification is provided for use in awakening the job (and also to specify the starting point for the job when it is performed).

l. Resume: Resume computations which were interrupted to perform the task (used for tasks not initiated through waitlist means, cf. End task).

m. Return (or "Return via XXX" where the return address has been stored in XXX): Return to the subroutine's calling program (which in some cases could have been either a task or a job).

n. When no transfer instruction occurs at the end of a routine, computations continue down the page through subsequent routine or routines.

In addition to the above terms, a special notation is used with display interface routines:

Proceed to "GOFLASH"
(If terminate, ...; if proceed, ...; if other response, ...)

or

Perform "GOFLASHR"
(If terminate, ...; if proceed, ...; if other response, ...)

X = Y

End job

In the first case, control is transferred to the display interface routine and the program is terminated until the astronaut responds. In the second case, a display interface routine is established and the subsequent steps (X = Y) are executed before the astronaut responds to the display (usually). The four possible astronaut responses that are recognized are: "Terminate" via verb 34; "Proceed" via verb 33 or the proceed button; "Resequence" via verb 32; and "Enter" by pushing the enter button in response to a "please perform" type verb or after loading data via one of the load verbs. The last two types of response will initiate the "if other response" branch.

A capability also exists to "kill" a task that has been inserted in the "waitlist" but has not yet been executed. It is simply removed from the waitlist.

Operations

Several mathematical manipulations are available to the programmer and are indicated either by special characters or by lower case symbols. The trigonometric, logarithm and square root functions are described in the MATX section.

1) Standard Arithmetic Operations

+ : plus

- : minus

  : (blank) multiply; scalar by scalar, vector by scalar, etc.

/ : divide; scalar by scalar; vector by scalar

|A|: Magnitude of scalar (absolute value)

|$\underline{A}$|: Magnitude of vector (length)

2) Special scalar operations (see MATX section)

$\sqrt{A}$	: square root
arccosA	: double precision arc cosine of A
arcsinA	: double precision arc sine of A
$\arcsin_{sp}A$	: single precision arc sine of A
cosA	: double precision cosine of A
$\cos_{sp}A$	: single precision cosine of A
sinA	: double precision sine of A
$\sin_{sp}A$	: single precision sine of A
lnA	: natural logarithm of A

3) Vector operations

* ($\underline{A}$ * $\underline{B}$)	: vector cross product
· ($\underline{A}$ · $\underline{B}$)	: vector dot product
$\underline{A}^2$	: square of length of $\underline{A}$
[A] $\underline{B}$ or $\underline{B}$ $[A]^T$	: premultiply the vector $\underline{B}$ by the matrix [A](3x3; interpretive instruction MXV). The X component of the answer is equal to the dot product of the first row of [A] with $\underline{B}$, etc.

$\underline{B}$ [A] or $[A]^T$ $\underline{B}$: post multiply the vector $\underline{B}$ by the matrix [A] (3 x 3, interpretive instruction VXM). The X component of the answer is equal to the dot product of the first column of [A] with $\underline{B}$, etc.

4) Bit operations

Switch bit a of B to 1 or 0: Set or reset bit a of erasable memory register B.

Set bit a of A = bit b of B: Set or reset bit a of erasable memory register A according to whether bit b of B is set or reset.

Invert bit a of A: Set bit a of A = the complement of bit a of A.

$\wedge$ (A $\wedge$ B): Bit by bit logical product. If a given bit in both A and B is a binary 1, then the result is a binary 1; otherwise, the result is a binary 0.

$\sim (\tilde{B})$: Bit by bit complement of B.

5) cycle: Shift in a cyclic fashion, with bits "spilling out" of one end of the register appearing at the other end. For example, bit 14 cycled left 2 places, since the word length of the computer is 15 bits for data, becomes bit 1, as does the same bit cycled right 13 places (or, in this case, shifted right 13 places).

6) limit: Cause the maximum value (usually K:posmax) to be stored if the quantity or its computed value exceeds that maximum. (Unless otherwise specified, the magnitude is limited, with sign information preserved; frequently done by means of a check for overflow.)

7) modulo: Form a quantity, for A modulo B, equivalent to the value of B times the remainder from (A/B). For example, 380° modulo one revolution is 20°.

8) overflow: Exceed the capacity of the computer register (i.e. the maximum value of the number allowed by its scaling). The interpretive language (section VI of 3420.5-27 (Revision 2)) has a special cell which is set if such an overflow is encountered, and which may be sensed to cause program branching. The divide instruction in the interpretive language, if the numerator exceeds the denominator (using the numbers in the computer register), sets the answer to the maximum capacity of the computer register with the proper sign.

9) quasi-floating point: Carry out a computation (usually involving a division) by, in general, normalizing both numerator and denominator before performing the division (with suitable provisions to avoid division overflow), and then shifting the result the appropriate number of places. Normalization involves shifting a number so that there are no leading magnitude zeroes, and counting the number of shifts required.

10) set A = B and B = A: Exchange the contents of "A" and "B".

11) signA: Complement the accumulator if A is negative; otherwise, leave the accumulator alone. Unless otherwise specified, if A = 0 the accumulator is also left alone, i.e. 0 is a "positive" number.

12) shift: Shift in a non-cyclic fashion, with exclusion of the sign bit: bits "spilling out" of the least significant end of the register (for shifting right) are lost. Cf. "cycle". Vacated bits are set to sign bit.

13) sign agreement: Force the signs of the various parts of a multiple-precision word to be the same.

14) unit($\underline{A}$ + $\underline{B}$): Form or use a unit vector from the vector information specified. The scale factor of the unit vector when formed is B1. The magnitude of the vector is left in push-down list address 36D and its square in 34D. Operation "overflows" if magnitude of vector (before forming unit) is less than 2^{-21} as stored in computer register: if a cross product of 2 unit vectors involved, this would be a "true value" of 2^{-19}, or about 0.002 mr.

15) $\underline{A} = (a_1, a_2, a_3)$: Form a vector with x, y, and z components = a_1, a_2 and a_3.

16) $B = (b_1, b_2)$: Form a double precision number B with more significant half = b_1, less significant half = b_2.

17) The difference between two angles expressed in two's complement form can be performed by a special instruction to produce a result in one's complement form.

Major Variables

(and constants)

The following quantities are used at several points in the program; a general knowledge of their significance will be valuable in any effort to understand the program.

$CADRFLSH_i$ (i=0,1,2), CADRSTOR, DSPLIST: Single precision address storage registers in the display interface routines. Three internally generated displays can be handled by the display interface routines at any one time, one "priority" display, one "mark/extended verb" display and one "normal" display. The addresses of the routines requesting these displays are stored in $CADRFLSH_0$, $CADRFLSH_1$ and $CADRFLSH_2$ respectively. If the active display finds that the astronaut is using the DSKY (Display and Keyboard Assembly), it is put to sleep and its address stored in DSPLIST until the astronaut releases the DSKY. If the active display requires astronaut response, it is put to sleep and its address stored in CADRSTOR until the response is received.

$\underline{C}$DU (CDU_x, CDU_y, CDU_z): LGC input counters incremented directly from the Coupling Data Unit to maintain LGC knowledge of the position of the outer, inner and middle gimbal angles of the Inertial Measurement Unit. These counters are coupled directly to the "Read Counters" in the three ICDU channels of the CDU, and the two counters associated with each gimbal angle are incremented simultaneously. In order to synchronize the LGC counters with the Read Counters, the LGC counters must be set to zero while the ICDU Read Counters are maintained at zero by setting bit 5 of channel 12. The gimbal angle data thus maintained is scaled B-1 in units of revolutions (one least increment is equivalent to 2^{-15} revolution) and is in two's complement form.

The manipulation of two's complement numbers in the LGC maintains a periodic regularity by propagating an overflow bit into the sign bit. In the case of angles scaled B-1 in units of revolutions, the overflow bit represents one-half of a revolution (180 degrees), and angular sums over one half of a revolution in magnitude are "automatically" adjusted to lie within the range $-\frac{1}{2} \leq \text{angle} < \frac{1}{2}$. For example: $0.35 + 0.42 = -0.23$ ($126.0^\circ + 151.2^\circ = -82.8^\circ$).

CDU_s, CDU_t: LGC input counters incremented directly from the Coupling Data Unit to maintain LGC knowledge of the Rendezvous Radar shaft and trunnion angles. These counters are coupled to the Read Counters in the two RRCDU channels of the CDU and are exactly like the CDU-linked counters described above. The RRCDU Read Counters are zeroed by bit 1 of channel 12.

CDUD: Single precision vector containing the latest specification of desired IMU gimbal angles for the Digital Autopilot, scaled B-1 in units of revolutions and stored in two's complement form. See the description of LGC two's complement included with the definition of CDU.

Channels: Fifteen bit interface registers for input and output of discrete information. See TRW Working Paper 3420.5-27 (Revision 2) for more precise and complete information. Channels 5,6,11-16 and 30-33 are described in the section entitled "Channels, Flagwords and Other Discrete Information Registers."

1,2: Identical to L and Q registers; see MATX section.

3,4: Standby Clock; continues to count when the LGC is put into standby mode (see "P06").

7: Bits 15-8 and 4-1 have no function; bits 7-5 are the fixed-memory address extension bits (SUPERBNK); see MATX section.

10: Channel used by "T4RUPT" to set the relays in the DSKY; bits 15-12 contain the relay address; bits 11-1 contain the desired setting. Called OUT0 in the programs; see INTR section.

34,35: Two channels loaded with information to be telemetered by the downlink; see TELE section.

DELCDU: Single precision vector subtracted from CDUD every 100 milliseconds to control the rate during an automatic attitude maneuver, scaled B-1 in units of revolutions and stored in two's complement form. See the description of LGC two's complement included with the definition of CDU.

DELV: Double precision sensed-change-in-velocity vector, scaled B14 in units of centimeters per second and expressed in stable member (IMU) coordinates. (One pulse from the accelerometer is equivalent to one centimeter per second; DELV is loaded directly from the accelerometers.)

DISPDEX: Single precision index controlling the periodic and otherwise independent display (by "CLOKTASK") of burn parameters. See BURN section.

DNLSTCOD: Single precision index (range 0-5) indicating which of the downlists is to be telemetered, scaled B14 and unitless. Loaded by various programs to select the proper downlist. See K:DNTABLE in TELE section.

$DSPTAB_i$ (i=0-10): Computer storage for the DSKY illumination relay settings, complemented when they are changed to signal to the "T4RUPT" routine that the relays are to be re-set as soon as possible. A numerical record of what is displayed via $DSPTAB_{10,9,8}$ is maintained in MODREG, VERBREG and NOUNREG. The contents of XREG, YREG and ZREG do not necessarily reflect what is currently being displayed via $DSPTAB_{7-0}$. See DSKY and DATA sections.

$DSPTAB_{11}$: Single precision flagword whose bits designate relays to be set to illuminate lamps on the DSKY. Bit 9 lights the "program alarm" (PROG) lamp when set; bit 8 lights the "tracker fail" (TRACKER) lamp when set; bit 6 lights the "gimbal lock warning" (GIMBAL LOCK) lamp when set; bit 5 lights the "LR altitude fail" lamp when set; bit 4 lights the "no attitude" (NO ATT) lamp when set (via "T4RUPT"); bit 3 lights the "LR velocity fail" lamp when set.

E_{ADR}: Notation used to denote any cell in erasable memory whose address is stored in the quantity used as the subscript (here denoted by ADR). If access is required to an E-memory cell in a switched bank other than that in which the program is operating, the EBANK register must be re-set first.

FLAGWRD0-11: Single precision registers whose bits are used individually for storage of on-off/yes-no type information. The flagwords are described in the section entitled "Channels, Flagwords and Other Discrete Information Registers" and references to the flagwords are listed in the section entitled "List of References to Flagwords and Channels."

i, j, k, n, p, s, t, v: Letters used to denote various temporary storage cells in the programs, usually indexes scaled B14 and unitless.

K:dvtoacc: Constant implicit in the 2 second navigation cycle, scaled B-1 in units of seconds to the minus one power. Equation value: $\frac{1}{2}$. If the navigation cycle is changed to something other than 2 seconds, this constant will have to be changed to an explicit one wherever it appears.

K:posmaxsp, K:posmaxdp: Notation for the maximum positive value that can be stored in a single or double precision LGC register. The stored value of the constants is 37777_8 and 37777_8 37777_8 respectively. The equation value of each is determined by the scale factor of the variable involved. If B is the scale factor, the equation values are:

$$K{:}posmaxsp = 2^B (1 - 2^{-14})$$

$$K{:}posmaxdp = 2^B (1 - 2^{-28})$$

K:VxxNxx: Single precision constant verb-noun code. The two-digit decimal noun number is stored in bits 1-7 of the constant; the two-digit decimal verb number is stored in bits 8-14.

$MPAC_i$ (i=0-7): Multiple precision accumulator and storage used by jobs coded in interpretive language (via the interpretive decoder) and sometimes by jobs coded in basic language. A set of eight single precision cells associated with each job and used exclusively by that job. When a job is put to sleep or is interrupted by a job of higher priority, $MPAC_0$-$MPAC_7$ are saved as part of the "job core" reserved for that job, and they are re-set exactly as they were when the interrupted job is re-established.

MUDEX, PBODY: Single precision indexes used to differentiate between constants and program branches that differ according to whether the center of attraction is the earth (0) or the moon (8 and 2 respectively). See CONC section and ORBI section respectively.

PRIOTIME: Single precision time when a priority display is activated, used to enforce the two second delay before the response to a priority display is accepted (to avoid the problem of a response to a just interrupted normal display being interpreted as the response to the priority display.) See DINT section.

RATT, VATT, TAT: State vector output from orbital integration with constant scaling (B29, B7, B28) or variable scaling (RATT1, VATT1, TAT) in units of meters, meters per centisecond, and centiseconds.

RCVCSM, RCVLEM: The permanent state vectors for the CSM and LM contain six double precision vectors and three double precision scalars. They are listed below along with the name of the equivalent variable used in the precision integration of each.

LM	CSM	Integration
RCVLEM	RCVCSM	RCV
VCVLEM	VCVCSM	VCV
TCLEM	TCCSM	TC
DELTALEM	DELTACSM	TDELTAV
NUVLEM	NUVCSM	TNUV
TETLEM	TETCSM	TET
RRECTLEM	RRECTCSM	RRECT
VRECTLEM	VRECTCSM	VRECT
XKEPLEM	XKEPCSM	XPREV

[REFSMMAT]: Double precision, 3x3 transformation matrix, scaled B1 and unitless. Defined such that Asm = [REFSMMAT] Arf where A is a vector expressed in stable member and reference coordinates respectively. (Other transformation matrices are not continuously maintained but only generated when needed.)

RN, VN, PIPTIME, R-OTHER, V-OTHER, TETCSM: Double precision vectors describing the navigation state of the LM and CSM respectively, scaled nominally (B29, B7, B28).

TDEC1: Double precision time input to orbital integration routines (GET) specifying the endpoint of the integration, scaled B28 in units of centiseconds. (Only input necessary if the permanent state vector is used as the origin.)

TEPHEM: Triple precision elapsed time from the beginning of the nearest Besselian year to the time when the LGC clock (TIMENOW) is zeroed, for use in the calculation of ephemerides; scaled B42 in units of centiseconds and included in the pre-launch erasable load.

THETAD: Single precision vector containing the final desired gimbal angles that define a desired orientation of the spacecraft with respect to the Inertial Measurement Unit (IMU) for large attitude maneuvers or IMU alignment; scaled B-1 in units of revolutions and stored in two's complement form. See the description of LGC two's complement included with the definition of CDU.

TIG: Double precision time of ignition (or predicted cutoff time, once the engine has been ignited) input to the burn programs from the targeting programs. The parameters required of the targeting programs by the burn programs (in addition to the LM state and mass estimate, which are assumed always available) are:

External Delta-V (XDELVFLG = 1)	Lambert Targetted (XDELVFLG = 0)
TIG	TIG
RTIG	RTARG
VTIG	TPASS4
DELVSIN	NORMSW (FL7, bit 10)
DELVSAB	

TIME3, TIME4, TIME5, TIME6: LGC clocks in addition to TIMENOW which are used to control interrupts of one kind or another. See INTR section.

TIMENOW: Double precision computer clock, incremented every centisecond (one hundredth of a second) by the LGC oscillator; scaled B28 in units of centiseconds.

TS (and various transmutations such as TS1, TSnoun TS_{12}, TSvec): Real or dummy temporary storage cells used for convenience in describing the performance of the equations. When used as a communication cell between routines, TS generally represents what the program transfers via the accumulator.

VGPREV: Double precision previous value of velocity-to-be-gained vector, program notation also "VGTIG," scaled B7 in units of meters per centisecond.

UPSVFLAG: Single precision flag loaded with a state vector update (address of UPSVFLAG is just before that of RRECT) to indicate whether the update is for the LM or CSM state and whether it is in moon-centered or earth-centered reference coordinates. See ORBI section.

XSC, XSM, XSCD, etc.: Double precision unit vectors in the directions of the X spacecraft axis, the X stable member axis, the desired X spacecraft axis, etc. Such unit vectors are always scaled B1 and unitless, but they may be expressed in various coordinate systems. An effort has been made to indicate the coordinate system in cases where it was not immediately obvious by adding small letters (sc, sm, rf) to the tag; the notations YSCsm, YSCsc, and YSC all represent the same cell - the indication of coordinate system is purely explanatory.

Erasable Memory Initialization

The quantities listed below constitute the "erasable memory load" which supplements the initialization performed by verb 36 (fresh start routine "SLAP1") in order to prepare the LGC erasable memory for the beginning of the mission. The list shows the absolute address of each quantity in the list (single, double or triple precision) in ECADR form (EBANK in bits 11-9; address = 1400_8 + bits 8-1); the tag assigned to that address by this document; the tag assigned to that address by the LUMINARY program if it differs from that used in this document; the scale factor and the units which the program assumes when handling each quantity; and the section of this document in which the quantity is defined.

Following this alphabetical list are the erasable memory quantities listed in order of increasing ECADR.

ECADR	Tag (alternate tag)	Scale	Units	Section
01516 01517	2LATE466	B28	centiseconds	DESC
02566 02567	ABTRDOT	B7	meters/centisecond	ASCT
01463	ADIAX	B-6	gyro pulses/cm per sec	IMUC
01464	ADIAY	B-6	gyro pulses/cm per sec	IMUC
01465	ADIAZ	B-6	gyro pulses/cm per sec	IMUC
01466	ADSRAX	B-6	gyro pulses/cm per sec	IMUC
01467	ADSRAY	B-6	gyro pulses/cm per sec	IMUC
01470	ADSRAZ	B-6	gyro pulses/cm per sec	IMUC

(Because one gyro pulse is equivalent to 2^{-21} revolutions, the above six quantities could also be assumed to be scaled B-26 in units of revolutions.)

ECADR	Tag (alternate tag)	Scale	Units	Section
02020 02021	AGSK	B28	centiseconds	EXVB
02512	AHZLIM	B-4	m/cs^2	DESC
03404	$AOTAZ_1$ (AOTAZ)	B-1	revolutions (2's comp)	ALIN
03405	$AOTAZ_2$ (AOTAZ+1)	B-1	revolutions (2's comp)	ALIN
03406	$AOTAZ_3$ (AOTAZ+2)	B-1	revolutions (2's comp)	ALIN
03407	$AOTAZ_4$ (AOTAZ+3)	B-1	revolutions (2's comp)	ALIN

ECADR	Tag (alternate tag)	Scale	Units	Section
03410	$AOTAZ_5$ (AOTAZ+4)	B-1	revolutions (2's comp)	ALIN
03411	$AOTAZ_6$ (AOTAZ+5)	B-1	revolutions (2's comp)	ALIN
03412	$AOTEL_1$ (AOTEL)	B-1	revolutions (2's comp)	ALIN
03413	$AOTEL_2$ (AOTEL+1)	B-1	revolutions (2's comp)	ALIN
03414	$AOTEL_3$ (AOTEL+2)	B-1	revolutions (2's comp)	ALIN
03415	$AOTEL_4$ (AOTEL+3)	B-1	revolutions (2's comp)	ALIN
03416	$AOTEL_5$ (AOTEL+4)	B-1	revolutions (2's comp)	ALIN
03417	$AOTEL_6$ (AOTEL+5)	B-1	revolutions (2's comp)	ALIN
	(2's comp indicates that these quantities are stored in two's complement form, not the usual one's complement form)			
03400 03401	ATIGINC	B28	centiseconds	TRGL
03373	AZBIAS	B-1	revolutions	DESC
01711 01712	AZO	B0	revolutions	COOR
02570 02571	COSTHET1	B2	unitless	ASCT
02572 02573	COSTHET2	B2	unitless	ASCT
01327	CSMMASS	B16	kilograms	DAPB
02520 02521	DELQFIX	B24	meters	SERV
03425	DELTTFAP	B17	centiseconds	DESC
02474 02475	DESIGNRX (RIGNX)	B24	meters	DESC
02476 02477	DESIGNRZ (RIGNZ)	B24	meters	DESC
02472 02473	DESIGNV (VIGN)	B10	meters/centisecond	DESC

ECADR	Tag (alternate tag)	Scale	Units	Section
02504 02505	DESKIGNV (KIGNV/B4)	B18	centiseconds	DESC
02500 02501	DESKIGNX (KIGNX/B4)	B4	unitless	DESC
02502 02503	DESKIGNY (KIGNY/B8)	B-16	$meters^{-1}$	DESC
03011	DKDB	B15	$revolutions^{-1}$	DAPB *
03005	DKKAO3N	B14	unitless	DAPA *
03004	DKOMEGAN	B14	unitless	DAPA *
03003	DKTRAP	B-3	revolutions/second	DAPA *
02634 thru 02641	DLAND	B24	meters	DESC
03113	$DOWNTORK_0$ (POSTORKP)	B5	seconds	DAPA
03114	$DOWNTORK_1$ (NEGTORKP)	B5	seconds	DAPA
03115	$DOWNTORK_2$ (POSTORKU)	B5	seconds	DAPA
03116	$DOWNTORK_3$ (NEGTORKU)	B5	seconds	DAPA
03117	$DOWNTORK_4$ (POSTORKV)	B5	seconds	DAPA
03120	$DOWNTORK_5$ (NEGTORKV)	B5	seconds	DAPA
01350	E32C31RM	B80	$meters^6/centisecond^2$	ORBI
01347	E3J22R2M	B58	$meters^5/centisecond^2$	ORBI
01356	ELBIAS	B-1	revolutions	DESC
02432 02433	$GAIN_0$ (GAINBRAK)	B0	unitless	DESC
02466 02467	$GAIN_{28}$ (GAINAPPR)	B0	unitless	DESC
03000	HIASCENT	B16	kilograms	DAPB *
02507	HIGHCRIT	B14	DPS throttle pulses	DESC
02514	HLROFF	B24	meters	SERV
03012	IGNAOSQ	B-2	$revolutions/second^2$	BURN
03013	IGNAOSR	B-2	$revolutions/second^2$	BURN

ECADR	Tag (alternate tag)	Scale	Units	Section
02550 02551	J1PARM	B24	meters	ASCT
02554 02555	J2PARM	B24	meters	ASCT
02552 02553	K1PARM	B24	meters/revolution	ASCT
02556 02557	K2PARM	B24	meters/revolution	ASCT
	* See note on page ASCT-14.			
02542 02543	LAGdTAU (LAG/TAU)	B0	unitless	DESC
03426	LEADTIME	B17	centiseconds	DESC
01326	LEMMASS	B16	kilograms	DAPB
02012 thru 02017	LM504 (504LM)	B0	radians	COOR
03010	LMKAOSN	B14	unitless	DAPA *
03007	LMOMEGAN	B14	unitless	DAPA *
03006	LMTRAP	B-3	revolutions/second	DAPA *
02506	LOWCRIT	B14	DPS throttle pulses	DESC
02522	$LRALPHA_1$ (LRALPHA)	B-1	revolutions (2's comp)	SERV
02524	$LRALPHA_2$ (LRALPHA2)	B-1	revolutions (2's comp)	SERV
02523	$LRBETA_1$ (LRBETA1)	B-1	revolutions (2's comp)	SERV
02525	$LRBETA_2$ (LRBETA2)	B-1	revolutions (2's comp)	SERV
03420	LRHMAX	B14	meters	SERV

ECADR	Tag (alternate tag)	Scale	Units	Section
02527	LRVF	B7	meters/centisecond	SERV
02526	LRVMAX	B7	meters/centisecond	SERV
03421	LRWH	B0	unitless	SERV
02530	$LRWV_0$ (LRWVZ)	B0	unitless	SERV
02531	$LRWV_1$ (LRWVY)	B0	unitless	SERV
02532	$LRWV_2$ (LRWVX)	B0	unitless	SERV
02533	$LRWVF_0$ (LRWVFZ)	B0	unitless	SERV
02534	$LRWVF_1$ (LRWVFY)	B0	unitless	SERV
02535	$LRWVF_2$ (LRWVFX)	B0	unitless	SERV
02536	LRWVFF	B0	unitless	SERV
01243 01244	MASS	B16	kilograms	SERV DAPB
02546 02547	MAXFORCE	B12	kg meters/centisecond^2	DESC
02544 02545	MINFORCE	B12	kg meters/centisecond^2	DESC
01460	NBDX	B-5	gyro pulses/centisecond	IMUC
01461	NBDY	B-5	gyro pulses/centisecond	IMUC
01462	NBDZ	B-5	gyro pulses/centisecond	IMUC
	(B-5 gyro pulses/cs equivalent to B-26 revolutions/cs)			
01452	$PIPABIAS_x$ (PBIASX)	B-3	PIPA counts/centisecond	IMUC
01454	$PIPABIAS_y$ (PBIASY)	B-3	PIPA counts/centisecond	IMUC
01456	$PIPABIAS_z$ (PBIASZ)	B-3	PIPA counts/centisecond	IMUC
01453	$PIPASCF_x$ (PIPASCFX)	B-9	unitless	IMUC
01455	$PIPASCF_y$ (PIPASCFY)	B-9	unitless	IMUC

ECADR	Tag (alternate tag)	Scale	Units	Section
01457	$PIPASCF_z$ (PIPASCFZ)	B-9	unitless	IMUC
03002	PITTIME	B14	centiseconds	DAPB
03402 03403	PTIGINC	B28	centiseconds	TRGL
02511	QHZ	B0	unitless	DESC
01351 01352	RADSKAL	B21	LR low scale altitude bits/meter/cs	SERV
02562 02563	RAMIN	B24	meters	ASCT
01770 01771	RANGEVAR	B-12	unitless	RNAV
01772 01773	RATEVAR	B-12	unitless	RNAV
02022 thru 02027	$\underline{R}$LS	B27	meters	CONC
02004	RMAX	B19	meters	RNAV
02537	RODSCALE	B-7	meters/centisecond	DESC
03001	ROLLTIME	B14	centiseconds	DAPB
03427	RPCRTIME	B17	centiseconds	SERV
03430	RPCRTQSW	B1	unitless	SERV
01774	RVARMIN	B12	$meters^2$	RNAV
02010	SHAFTVAR	B-12	$radians^2$	RNAV
01353	SKALSKAL	B0	unitless	SERV
02416 thru 02423	$\underline{T}ARGADG_0$ (ADG, ABRFG)	B-4	$meters/centisecond^2$	DESC
02452 thru 02457	$\underline{T}ARGADG_{28}$ (AAPFG)	B-4	$meters/centisecond^2$	DESC

ECADR	Tag (alternate tag)	Scale	Units	Section
02402 thru 02407	$TARGRDG_0$ (RDG, RBRFG)	B24	meters	DESC
02436 thru 02443	$TARGRDG_{28}$ (RAPFG)	B24	meters	DESC
02410 thru 02415	$TARGVDG_0$ (VDG, VBRFG)	B10	meters/centisecond	DESC
02444 thru 02451	$TARGVDG_{28}$ (VAPFG)	B10	meters/centisecond	DESC
02510	TAUHZ	B11	centiseconds	DESC
02540 02541	TAUROD	B9	centiseconds	DESC
02434	$TCGF_0$ (TCGFBRAK)	B17	centiseconds	DESC
02470	$TCGF_{28}$ (TCGFAPPR)	B17	centiseconds	DESC
02435	$TCGI_0$ (TCGIBRAK)	B17	centiseconds	DESC
02471	$TCGI_{28}$ (TCGIAPPR)	B17	centiseconds	DESC
03423	$TEND_0$ (TENDBRAK)	B17	centiseconds	DESC
03424	$TEND_1$ (TENDAPPR)	B17	centiseconds	DESC
01706 01707 01710	TEPHEM	B42	centiseconds	COOR
01570 01571	TETCSM	B28	centiseconds	ORBI
01642 01643	TETLEM	B28	centiseconds	ORBI
02560 02561	THETCRIT	B0	revolutions	ASCT

ECADR	Tag (alternate tag)	Scale	Units	Section
02400 02401	TLAND	B28	centiseconds	DESC
03431 03432	TNEWA	B28	centiseconds	BURN
02513	TOOFEW	B14	unitless	DESC
02011	TRUNVAR	B-12	$radians^2$	RNAV
02426 02427	$TTFADGZ_0$ (ABRFG* and ADG2TTF+0)	B-4	$meters/centisecond^2$	DESC
02462 02463	$TTFADGZ_{28}$ (AAPFG* and ADG2TTF+28)	B-4	$meters/centisecond^2$	DESC
02430 02431	$TTFJDGZ_0$ (JBRFG* and JDG2TTF+0)	B-21	$meters/centisecond^3$	DESC
02464 02465	$TTFJDGZ_{28}$ (JAPFG* and (JDG2TTF+28)	B-21	$meters/centisecond^3$	DESC
02424 02425	$TTFVDGZ_0$ (VBRFG* and VDG2TTF+0	B13	meters/centisecond	DESC
02460 02461	$TTFVDGZ_{28}$ (VAPFG* and VDG2TTF+28	B13	meters/centisecond	DESC
01713 01714	$UNITW_x$ (mAYO)	BO	unitless	COOR
01715 01716	$UNITW_y$ (AXO)	BO	unitless	COOR
03371 03372	VELBIAS	B6	meters/centisecond	SERV
02005	VMAX	B7	meters/centisecond	RNAV
01775	VVARMIN	B-12	$meters^2/centisecond^2$	RNAV
02000	WRENDPOS	B14	meters	RNAV
02001	WRENDVEL	BO	meters/centisecond	RNAV
02002	WSHAFT	B-5	radians	RNAV
02006	WSURFPOS	B14	meters	RNAV

ECADR	Tag (alternate tag)	Scale	Units	Section
02007	WSURFVEL	B0	meters/centisecond	RNAV
02003	WTRUN	B-5	radians	RNAV
01700 thru 01705	X789	*	radians	RNAV
	(*Scaling is B5 for earth and B3 for moon)			
02564 02565	YLIM	B24	meters	ASCT
03422	ZOOMTIME	B14	centiseconds	BURN

* These quantities are also loaded by the fresh start routine entered * from verb 36.

In addition to the quantities listed on the previous pages, the indicated bits of the following flagwords must be padloaded as they are not initialized by the fresh start ("SLAP1") routine.

FLAGWRD3	bit 13	(REFSMFLG)
FLAGWRD8	bit 8	(SURFFLAG)
	bit 11	(LMOONFLG)
	bit 12	(CMOONFLG)
FLGWRD10	bit 13	(APSFLAG)

The following pad loaded **variables** are listed by ascending ECADR. The tag name is that given by this document.

ECADR	Tag	ECADR	Tag
01243-4	MASS	01700-5	X789
01326	LEMMASS	01706-10	TEPHEM
01327	CSMMASS	01711-2	AZO
01347	E3J22R2M	01713-4	$UNITW_x$
01350	E32C31RM	01715-6	$UNITW_y$
01351-2	RADSKAL	01770-1	RANGEVAR
01353	SKALSKAL	01772-3	RATEVAR
01356	ELBIAS	01774	RVARMIN
01452	$PIPABIAS_x$	01775	VVARMIN
01453	$PIPASCF_x$	02000	WRENDPOS
01454	$PIPABIAS_y$	02001	WRENDVEL
01455	$PIPASCF_y$	02002	WSHAFT
01456	$PIPABIAS_z$	02003	WTRUN
01457	$PIPASCF_z$	02004	RMAX
01460	NBDX	02005	VMAX
01461	NBDY	02006	WSURFPOS
01462	NBDZ	02007	WSURFVEL
01463	ADIAX	02010	SHAFTVAR
01464	ADIAY	02011	TRUNVAR
01465	ADIAZ	02012-7	LM504
01466	ADSRAX	02020-1	AGSK
01467	ADSRAY	02022-7	RLS
01470	ADSRAZ	02400-1	TLAND
01570-1	TETCSM	02402-7	$TARGRDG_0$
01642-3	TETLEM		

ECADR	Tag
02410-5	TARGVDG$_0$
02416-23	TARGADG$_0$
02424-5	TTFVDGZ$_0$
02426-7	TTFADGZ$_0$
02430-1	TTFJDGZ$_0$
02432-3	GAIN$_0$
02434	TCGF$_0$
02435	TCGI$_0$
02436-43	TARGRDG$_{28}$
02444-51	TARGVDG$_{28}$
02452-7	TARGADG$_{28}$
02460-1	TTFVDGZ$_{28}$
02462-3	TTFADGZ$_{28}$
02464-5	TTFJDGZ$_{28}$
02466-7	GAIN$_{28}$
02470	TCGF$_{28}$
02471	TCGI$_{28}$
02472-3	DESIGNV
02474-5	DESIGNRX
02476-7	DESIGNRZ
02500-1	DESKIGNX
02502-3	DESKIGNY
02504-5	DESKIGNV
02506	LOWCRIT
02507	HIGHCRIT
02510	TAUHZ
02511	QHZ
02512	AHZLIM

ECADR	Tag
02513	TOOFEW
*** 02514	HLROFF
02516-7	2LATE466
02520-1	DELQFIX
02522	LRALPHA$_1$
02523	LRBETA$_1$
02524	LRALPHA$_2$
02525	LRBETA$_2$
02526	LRVMAX
02527	LRVF
02530	LRWV$_0$
02531	LRWV$_1$
02532	LRWV$_2$
02533	LRWVF$_0$
02534	LRWVF$_1$
02535	LRWVF$_2$
02536	LRWVFF
02537	RODSCALE
02540-1	TAUROD
02542-3	LAGdTAU
02544-5	MINFORCE
02546-7	MAXFORCE
02550-1	J1PARM
02552-3	K1PARM
02554-5	J2PARM
02556-7	K2PARM
02560-1	THETCRIT
02562-3	RAMIN

ECADR	Tag
02564-5	YLIM
02566-7	ABTRDOT
02570-1	COSTHET1
02572-3	COSTHET2
02634-41	DLAND
03000	HIASCENT
03001	ROLLTIME
03002	PITTIME
03003	DKTRAP
03004	DKOMEGAN
03005	DKKAOSN
03006	LMTRAP
03007	LMOMEGAN
03010	LMKAOSN
03011	DKDB
03012	IGNAOSQ
03013	IGNAOSR
03113-20	$DOWNTORK_{0-5}$
03371-2	VELBIAS
03373	AZBIAS
03400-1	ATIGINC
03402-3	PTIGINC
03404-11	$AOTAZ_{1-6}$
03412-7	$AOTEL_{1-6}$
03420	LRHMAX
03421	LRWH
03422	ZOOMTIME
03423	$TEND_0$
03424	$TEND_1$
03425	DELTTFAP
03426	LEADTIME
03427	RPCRTIME
03430	RPCRTQSW
03431-2	TNEWA

Channels 1-4, 7, 10, 34, and 35 are discussed in the list of Major Variables

Channel 5

Bits 15-9 have no significance; bits 8-1 are set to command RCS jet firings and reset to terminate the firing.

Bit	Code	Jet Number	System	Rotation Effect	Translation Effect
8	1D	14	B	+U	+X
7	1U	13	A	-U	-X
6	2D	10	A	+V	+X
5	2U	9	B	-V	-X
4	3D	6	B	-U	+X
3	3U	5	A	+U	-X
2	4D	2	A	-V	+X
1	4U	1	B	+V	-X

Channel 6

Bits 15-9 have no significance; bits 8-1 are set to command RCS jet firings and reset to terminate the firing.

Bit	Code	Jet Number	System	Rotation Effect	Translation Effect
8	1S	16	B	-P	+Y
7	4S	4	A	+P	-Y
6	3S	8	A	-P	-Y
5	2S	12	B	+P	+Y
4	2F	11	A	-P	+Z
3	1F	15	A	+P	-Z
2	4F	3	B	-P	-Z
1	3F	7	B	+P	+Z

Channel 11

Bits 15, 12, 11, and 8 are spare.

Bits 14 (engine off) and 13 (engine on) are assigned to the main engine on/off function. The normal engine-off command configuration is 10_2; the normal engine-on command configuration is 01_2. The following information concerns the performance of the LM when either of the two other possible binary states (00_2 or 11_2) occurs (as in a hardware restart - "GOPROG").

"All Block 2 computers have the engine-on and engine off discretes in bits 13 and 14 respectively of channel 11. If the LEM Descent engine sees a 1,1 condition (both output transistors conducting) or a 0,0 condition (both output transistors non-conducting), it will ignore the signal and remain in the state it was previously in. This allows the computer to zero all the output bits during a restart and not shut the engine off. There is no time limit as to how long an improper state (1,1) or (0,0) can last with the descent engine.

"The LEM ascent engine will be turned on by an erroneous 1,1 condition which lasts longer than 1 millisecond. Therefore the LGC must be programed to set the bits to the proper state within 0.5 millisecond following recovery from a restart."

It is assumed that the ascent engine will remain off if staging occurs with the bits in the configuration 00_2.

A fresh start sets bit 14 to 1 and sets remaining bits to 0.

Meaning when set and reset (channel 11 is an output or command channel)

10 1 - Caution Reset signal: resets the flip-flop holding the Restart lamp in the energized state.
0 - Allow the Restart lamp to light

9 1 - Test connector discrete used in bench tests
0 -

7 1 - Light the "Operator Error" lamp (automatic flash)
0 - Extinguish the "Operator Error" lamp

6 1 - Start flash of verb and noun registers on the DSKY
0 - Stop the verb-noun flash

5 1 - Light the "Key Release" lamp (automatic flash) to request key release
0 - Extinguish the "Key Release" lamp

4		1 - Light the "Temperature Caution" lamp 0 - Extinguish the "Temperature Caution" lamp
3		1 - Light the "Uplink Activity" lamp 0 - Extinguish the "Uplink Activity" lamp
2		1 - Light the "Computer Activity" lamp 0 - Extinguish the "Computer Activity" lamp
1		1 - Light the "ISS Warning" lamp 0 - Extinguish the "ISS Warning" lamp

Channel 12

Bit and initial value (fresh start)

Bit	Initial value	Meaning (channel 12 is an output or command channel)
15	0	Bit energizes a latching relay that signals that the ISS turn-on delay is complete, removing the signal from bit 14 of channel 30 and switching the ISS into the normal operate mode. Reset after remaining set for about ten and one fourth seconds.
14	0	1 - Enable RR lock-on and automatic tracking. 0 - Disable Rendezvous Radar lock-on and automatic tracking.
13	0	Command provided via a DSKY relay to change landing radar from position #1 to position #2. (Returned to position #1 by a spacecraft switch.)
12	0	Bit set to cause rotation of the DPS bell around the +Z LM axis to produce a negative angular jerk around the +Z axis (-R).
11	0	Bit set to cause rotation of the DPS bell around the -Z LM axis to produce a positive angular jerk around the +Z axis (+R).
10	0	Bit set to cause rotation of the DPS bell around the +Y LM axis to produce a negative angular jerk around the +Y axis (-Q).
9	0	Bit set to cause rotation of the DPS bell around the -Y LM axis to produce a positive angular jerk around the +Y axis (+Q).
8	0	Bit set to display inertial data.

7	0	spare
6	0	1 - Enable ICDU Error Counters - coarse align, or display on FDAI. 0 - Disable ICDU Error Counters (3).
5	0	1 - Zero the ICDU Read Counters; force the ICDU gimbal angle follower counters to zero. 0 - Allow the ICDU Read Counters to follow the IMU gimbal angles, incrementing the CDU counters in the LGC as they do.
4	0	1 - Enable coarse align of the IMU; connect the ICDU Read Counters with the ICDU Error Counters so that the latter may be decremented as the IMU is coarse aligned. 0 - Disable coarse align of the IMU.
3	0	Bit not set in LUMINARY program.
2	0	1 - Enable Rendezvous Radar CDU Error Counters. 0 - Disable RRCDU Error Counters (2).
1	0	1 - Zero the RRCDU Read Counters. (Like bit 5) 0 - Allow the RRCDU Read Counters to follow the RR position angles.

Channel 13

Bit and initial value (fresh start)

		Meaning (channel 13 is an output or command channel)
15	0	Bit set to 1 to permit cell 00031g (TIME6) to be decremented by 1 each 0.000625 second. When cell has been reduced to -0, the next decrement resets bit to 0 and causes program interrupt #1.
14	0	Bit set to 1 to permit the RCS jet fail switches or the DPS gimbal fail switch to cause interrupt #10; always 0 in LUMINARY.
13	0	Bit set to 1 to permit signals from the translational hand controller to cause interrupt #10; always 0 in LUMINARY.
12	0	Bit set to 1 to permit signals from the rotational hand controller to cause interrupt #10.
11	0	Bit set to 1 to cause the PRO key on the DSKY to be interpreted as a "standby" key and put the LGC into standby mode.

Channel 13 (Continued)

10 0 Bit set to 1 to test the DSKY lights and relays not otherwise accessible to the software: energizes the Restart, Standby and Computer Warning lamps (the latter through a "warning filter").

9 0 1 - Initiate readout of analog-to-digital converters associated with the displacement of the rotational hand controller into cells RHCP, RHCQ, and RHCR (42_8-44_8).
0 - Stop readout of RHC analog-to-digital converters

8 0 1 - Enable input to RHCP, RHCQ, and RHCR from rotational hand controller analog-to-digital converter
0 - Disable input to RHCP, RHCQ, and RHCR

7 0 Bit used as the "word order code" bit (first bit in the 40-bit downlink sequence sent from the LGC containing digital data) for telemetry.

6 0 Bits used to block all inputs to INLINK; not set in LUMINARY.

5 0 Not used in LUMINARY.

4 0 Bit set to 1 to initiate transmission of radar information to the LGC. Bit is reset to 0 when program interrupt #9 is generated after the end of the pulse train from the radar to cell 46_8 (RNRAD).

3-1 Bits set to determine the routing of radar information into RNRAD when bit 4 is set. Information into RNRAD is: RR range information if bits 3-1 are 001_2; RR range rate if 010_2; LR X-velocity if 100_2; LR Y-velocity if 101_2; LR Z-velocity if 110_2; and LR altitude information if bits 3-1 are 111_2. Bits are initially 000.

Channel 14

Bit and initial value (fresh start)

Meaning (channel 14 is an output or command channel)

15 0 Bit set to 1 to cause output pulses (at a 3200 pps rate) to be generated from CDUXCMD, cell 00050_8. When cell is counted down to zero, the bit is reset, stopping the pulses. The ICDU Error Counter is loaded by these pulses if bit 6 of channel 12 is 1.

Channel 14 (Continued)

14	0	Bit set to 1 to cause output pulses to be generated from CDUYCMD, cell 00051_8. Like bit 15.
13	0	Bit set to 1 to cause output pulses to be generated from CDUZCMD, cell 00052_8. Like bit 15.
12	0	Bit set to 1 to cause output pulses (at a 3200 pps rate) to be generated from CDUTCMD, cell 00053_8. When cell is counted down to 0, the bit is reset, stopping the pulses. The RRCDU Error Counter is loaded by these pulses if bit 2 of channel 12 is 1.
11	0	Bit set to 1 to cause output pulses to be generated from CDUSCMD, cell 00054_8. Like bit 12.
10	0	Bit set to 1 to specify "gyro activity": it causes the pulse train whose magnitude is in cell 00047_8, GYROCMD, to be sent with polarity and destination specified by bits 9-7 of this channel, if bit 6 of this channel is 1. Bit reset after the pulses are sent.
9	0	1 - Gyro torquing pulses from GYROCMD specify a negative torque. 0 - Gyro torquing pulses from GYROCMD specify a positive torque. (Other pulse-type outputs from the computer have the polarity indicated by the polarity of the information in the counter cell itself.)
8-7		Bits used to specify the axis for gyro compensation information from GYROCMD. Conventional output sequence is inner (Y), middle (Z), and outer (X). The settings of bits 8 and 7 are: 00_2 for no output; 01_2 for X-axis gyro; 10_2 for Y-axis gyro; 11_2 for Z-axis gyro. Bits are initially 000.
6	0	Bit set to 1 to enable the power supply that produces the torquing pulses used to torque the gyros. Generally remains set after the first gyro torquing operation.
5	0	Not used in LUMINARY.
4	0	Bit set to 1 to cause output pulses to be generated from cell 00055_8 (THRUST) for use in controlling the position of the descent engine throttle.

3 0 Bit set to 1 to initiate shifting of data from cell 00060_8 (ALTM) to spacecraft indicator for altitude or altitude rate information. (See bit 2 of this word.) Bit reset to 0 just after start of data shift.

2 0 Bit set to 1 to indicate that altitude rate information is being shifted from cell 00060_8; if bit is 0 altitude information is being shifted from cell 00060_8.

1 0 Not used in LUMINARY.

Channel 15

Bits 15-6 have no function.

Bits 5-1 contain the five-bit binary keycode generated by the depression of one of the keys on the DSKY (Display and Keyboard Assembly). The depression of any key causes program interrupt #5 which reads the keycode immediately, while the key is depressed. The release of the key resets channel 15 and resets an interrupt trap (#15) to re-enable the interrupt.

Channel 16

Bits 15-8, 2 and 1 have no function.

Bit 7 is set to 1 if an increase in the rate of descent is desired by the crew (i.e. a lower thrust). Generated by moving a rate-of-descent switch in the -X direction (towards the engine).

Bit 6 is set to 1 if a decrease in the rate of descent is desired by the crew (i.e. a higher thrust). Generated by moving a rate-of-descent switch in the +X direction.

Bits 5-3 are set by depression of the mark reject, Y mark, and X mark buttons. The depression of any of these buttons causes program interrupt #6 which reads the information on channel 16 immediately while the button is depressed. The release of the button resets channel 16 and resets interrupt trap #16A to re-enable interrupt #6. Bits 5-3 are processed by the "MARKRUPT" routine.

A special capability is programmed into the restart routine to enable the astronaut to extricate the LGC from a multiple-restart loop. The restart program exits to the fresh start program if it senses that the mark reject button and the error reset key (channel 15 code 22_8) are depressed simultaneously.

Channel 30

Bit	Meaning (channel 30 is an input or information channel)
15	1 - Stable Member temperature outside design limits. 0 - Stable Member temperature within design limits. (Connected directly to lamp controlled by bit 4 of channel 11)
14	1 - IMU power-on switch off; or, IMU power-on switch on and IMU in normal operate mode ("turn-on delay complete" discrete sent from the LGC)(bit 15 of channel 12).

Bit	Meaning (Channel 30 is an input or information channel)
14	0 - IMU power-on switch on and IMU caged, waiting for "turn-on delay complete" discrete from the LGC. (The discrete from the LGC sets a relay which switches the IMU to normal operate mode.)
13	1 - IMU good. 0 - IMU fail due to excessive servo errors or degradation of 3200 pps or 800 pps supply . (See IMUMON routine.)
12	1 - ICDU good. 0 - ICDU fail (due to excessive errors or low voltage). (See IMUMON routine.)
11	1 - IMU not caged by crew switch setting. 0 - IMU caged by setting of crew switch. (See IMUMON routine.)
10	1 - SCS control of spacecraft. (Panel switch) 0 - PGNCS control of spacecraft.
9	1 - IMU power-on switch switched to off (panel 100). 0 - IMU power-on switch switched to on.
8	spare
7	1 - RRCDU good. 0 - RRCDU fail (due e.g. to excessive errors or low voltage). (See RRCDUCHK routine.)
6	1 - Display of inertial data from the computer is not desired by the crew. 0 - Display of inertial data from the computer is desired by the crew. When the appropriate information has been loaded by the program, bit 8 of channel 12 is set to 1.
5	1 - LGC DPS throttle commands disabled. 0 - LGC DPS throttle commands enabled.
4	Bit sensed 0 is "ABORT STAGE" command.
3	1 - Engine not armed. 0 - Engine armed.
2	1 - Ascent configuration. (not examined in Luminary) 0 - Ascent and descent configuration.

Channel 30 (Continued)

Bit	Meaning (channel 30 is an input or information channel)
1	Bit sensed 0 is "ABORT" command.

Channel 31

Bit	Meaning (channel 31 is an input or information channel)
15	1 - RHC (Rotational hand controller) is in detent. 0 - RHC is displaced from detent.
14	1 - The PGNS Mode Control switch is not set to "Auto". 0 - The PGNS Mode control switch is set to "AUTO", indicating that the LGC has complete authority for control of the spacecraft.
13	1 - The PGNS Mode Control switch is not set to "Attitude Hold". 0 - The PGNS Mode Control switch is set to "Attitude Hold", indicating that LGC Digital Autopilot authority is limited to rate damping.
12	1 - -Z translation not commanded via the THC (Translational Hand Controller). 0 - -Z translation commanded via the THC.
11	1 - +Z translation not commanded via the THC. 0 - +Z translation commanded via the THC.
10	1 - -Y translation not commanded via the THC. 0 - -Y translation commanded via the THC.
9	1 - +Y translation not commanded via the THC. 0 - +Y translation commanded via the THC.
8	1 - -X translation not commanded via the THC. 0 - -X translation commanded via the THC.
7	1 - +X translation not commanded via the THC. 0 - +X translation commanded via the THC.

Channel 31 (Continued)

Bits 6-1 carry discrete information about the displacement of the Rotational Hand Controller (RHC - also denoted by ACA). The bits are normally 1 and are set to 0 by the RHC to indicate the following directions of desired rotation:

Bit	6	5	4	3	2	1
Desired Rotation	-R	+R	-P	+P	-Q	+Q

Channel 32

Bits 15, 13, 12, and 11 of channel 32 are spares.

Bit 14 is activated by the "Proceed" key on the DSKY (formerly the "Standby" key and still functional as such when program 6 has enabled standby). It is normally set (1) and is reset to 0 only while the proceed key is depressed. It is examined every 120 milliseconds by the "T4RUPT" program and functions like a verb 33 except when program 6 has enabled standby. Proceed is rejected if V21, V22 or V23 on DSKY.

Bit 10. Bit state of 0 indicates that the descent engine gimbal failure monitor detects an apparent gimbal fail in the pitch or roll gimbal trim system. Not used by the LUMINARY program.

Bit 9 is sensed as 0 if action is taken by the crew to turn off the descent engine gimbal system. This prevents bits 12-9 of channel 12 from having any effect on the flight and causes the software to avoid using the gimbal.

Bits 8-1 are normally set (1). They are reset by astronaut panel switches that disable RCS jets (indicate that they are not functional). Each switch disables two jets, one with thrust around the P axis and one with thrust around the Q or R axis. The bits disable the following jets (see "RCSMONIT" routine):

Bit	Jet Numbers	Action (rotation)	Quad	System	Code
8	10	+V	2	A	2D
	11	-P	2	A	2F
7	9	-V	2	B	2U
	12	+P	2	B	2S
6	13	-U	1	A	1U
	15	+P	1	A	1F

Channel 32 (Continued)

Bit	Jet Numbers	Action (rotation)	Quad	System	Code
5	14	+U	1	B	1D
	16	-P	1	B	1S
4	6	-U	3	B	3D
	7	+P	3	B	3F
3	1	+V	4	B	4U
	3	-P	4	B	4F
2	5	+U	3	A	3U
	8	-P	3	A	3S
1	2	-V	4	A	4D
	4	+P	4	A	4S

Channel 33

Bit	Meaning (channel 33 is an input or information channel)
15	1 - Computer oscillator operating; reset to 1 by channel load instruction. 0 - Computer oscillator failure; flip-flop that can be set by a power transient momentarily interrupting the oscillator. Not sensed by the LUMINARY program.
14	Flip-flop sensed as 0 if a "computer warning" indication has been produced (e.g. multiple restarts, counter fail, voltage fail in standby, or alarm test by bit 10 of channel 13). Reset to 1 by channel load instruction. Not sensed by LUMINARY program.
13	Flip-flop input sensed as 0 if a PIPA fail indication generated by the PIPA (accelerometer) electronics. Reset to 1 by channel load instruction.
12	Flip-flop input sensed as 0 if a telemetry end pulse occurs too soon after the previous pulse (faster than 100 pps). Reset to 1 by channel load instruction.
11	Flip-flop input sensed as 0 if an input bit to cell 00045_8 (INLINK) is rejected due to an excessive bit rate (faster than 6400 pps). Reset to 1 by channel load instruction.
10	Not examined by the LUMINARY program.
9	1 - Landing radar high scale. 0 - Landing radar low scale.

Channel 33 (Continued)

Bit	Meaning (channel 33 is an input or information channel)
8	1 - At least one of the three LR velocity trackers not locked on. 0 - Landing radar velocity data good.
7	1 - Landing radar not in position 2. 0 - Landing radar in position 2.
6	1 - Landing radar not in position 1. 0 - Landing radar in position 1.
5	1 - LR range tracker or rear velocity-beam tracker (2) not locked on. 0 - Landing radar range (altitude) data good.
4	1 - RR range tracker and frequency tracker not both locked on. 0 - RR range and range rate data good.
3	1 - Rendezvous radar range high scale. 0 - Rendezvous radar range low scale.
2	1 - RR power off or RR mode switch not in the "LGC" position. 0 - RR on and under LGC control; can be positioned via cells 53_8-54_8.
1	spare

FLAGWRD0

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	NEED2FLG	1 - Error needles driven with LGC DAP computed body rates 0 - Error needles driven with attitude errors
14	(0)	JSWITCH	1 - Integration of W-matrix 0 - Integration of state vector
13	(0)	MIDFLAG	1 - Integration with secondary body and solar perturbations (should remain zero in LUMINARY) 0 - Integration without solar perturbations
12	(0)	MOONFLAG	1 - In the sphere of influence of the moon 0 - In the sphere of influence of the earth
11	(0)	P21FLAG	1 - Use base vectors already calculated 0 - 1st pass -- calculate base vectors
10	(0)	FSPASFLG	1 - First pass through reposition routine 0 - Not first pass through reposition routine
9	(0)	P25FLAG	1 - P25 in operation (preferred tracking attitude) 0 - P25 not in operation
8	(0)	IMUSE	1 - IMU in use (being switched, torqued or aligned) 0 - IMU not in use
7	(0)	RNDVZFLG	1 - P20 or P22 has been enabled 0 - P20 or P22 has not been enabled
6	(0)	RRNBSW	1 - RRTARGET in navigation base coordinates 0 - RRTARGET in stable member coordinates
5	(0)	LOKONSW	1 - Radar lock-on desired 0 - Radar lock-on not desired
4	(0)	NEEDLFLG	1 - Display total attitude error 0 - DAP following error displayed
3	(0)	FREEFLAG	A temporary flag used for utility purposes in many routines by P51 and P52; by lunar and solar ephemerides.
2	(0)	R10FLAG	1 - R10 outputs data to altitude and altitude rate meters only 0 - Output of 1 condition plus forward and lateral velocity on cross pointers

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
1	(0)	OLDESFLG	1 - R29 (powered flight RR designate routine) gyro command loop requested 0 - R29 (powered flight RR designate routine) gyro command loop not requested

FLAGWRD1

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	NJETSFLG	1 - Two jet RCS burn 0 - Four jet RCS burn
14	(0)	DIDFLAG	1 - Inertial data is available 0 - Perform data display initialization functions
13	(0)	ERADFLAG	1 - Compute earth radius for Fischer ellipsoid; use stored moon radius (never set in LUMINARY) 0 - Compute moon radius; use stored earth radius (pad radius) (in latitude-longitude routines)
12	(0)	RODFLAG	1- If in P66, normal operation continues. Restart clears flag 0 - If in P66, reinitialization is performed and flag is set
11	(0)	Spare	
10	(0)	R61FLAG	1 - R61 (preferred tracking attitude routine) LEM to be operated 0 - R65 (fine preferred tracking attitude routine) LM to be operated
9	(0)	spare	
8	(0)	VEHUPFLG	1 - Update CSM state vector 0 - Update LM state vector
7	(0)	UPDATFLG	1 - State vector updates from tracking allowed 0 - Updates from tracking not allowed
6	(0)	NOUPFLAG	1 - Neither CSM nor LEM state vector may be updated 0 - Either CSM or LEM state vector may be updated
5	(0)	TRACKFLG	1 - Tracking allowed 0 - Tracking not allowed
4	(0)	spare	
3	(0)	SLOPESW	1 - Iteration with bias method 0 - Iteration with Regula Falsi method
2	(0)	GUESSW	1 - No starting value for iteration 0 - Starting value for iteration exists

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
1	(0)	Spare	

FLAGWRD2

Bit and initial value (fresh start)

		Mnemonic	Meaning when 1 and 0
15	(0)	DRIFTFLG	1 - T3RUPT calls gyro compensation 0 - T3RUPT does no gyro compensation
14	(0)	SRCHOPTN	1 - Radar in automatic search option (R24) 0 - Radar not in automatic search option
13	(0)	ACMODFLG	1 - Manual acquisition by rendezvous radar 0 - Auto acquisition by rendezvous radar
12	(0)	LOSCMFLG	1 - Line-of-sight is being computed (R21) 0 - Line-of-sight not being computed
11	(0)	STEERSW	1 - Powered flight steering enabled (sufficient thrust) 0 - Powered flight steering off (insufficient thrust)
10		spare	
9	(0)	IMPULSW	1 - Minimum impulse burn (cut-off time specified) 0 - Steering burn (no cut-off time yet available)
8	(0)	XDELVFLG	1 - External delta-V VG computation 0 - Lambert (aimpoint) VG computation
7	(0)	ETPIFLAG	1 - Elevation angle supplied for P34, P74 0 - TPI time supplied for P34, P74 to compute elevation angle
6	(0)	FINALFLG	1 - Last pass through rendezvous program computations 0 - Interim pass through rendezvous program computations
5	(0)	AVFLAG	1 - LEM is active vehicle 0 - CSM is active vehicle
4	(0)	PFRATFLG	1 - Preferred attitude computed 0 - Preferred attitude not computed

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
3	(0)	CALCMAN3	1 - No final roll 0 - Final roll is necessary
2	(0)	CALCMAN2	1 - Perform maneuver starting procedure 0 - Bypass starting procedure
1	(0)	NODOFLAG	1 - V37 not permitted (do not allow a major mode change) 0 - V37 permitted (major mode change enabled)

FLAGWRD3

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	POOHFLAG	1 – P00 integration 10 minute checks are running 0 – P00 integration 10 minute checks disabled
14	(0)	GLOKFAIL	1 – Calculated middle gimbal angle exceeds 60° 0 – Checked and reset in IMU performance tests, also reset in R00.
13		REFSMFLG	1 – Reference to stable member matrix valid (protected from fresh start) 0 – Transformation matrix not valid
12	(0)	LUNAFLAG	1 – Lunar latitude-longitude conversion 0 – Earth latitude-longitude conversion
11	(1)	NOR29FLG	1 – R29 not allowed 0 – R29 allowed (rendezvous radar designate, powered flight)
10	(0)	VFLAG	1 – No star pair found during R56. 0 – Star pair found during R56
9	(0)	R04FLAG	1 – Alarm 521 suppressed 0 – Alarm 521 allowed
9	(0)	READRFLG	1 – Reading rendezvous radar data pursuant to R29 0 – Not reading rendezvous radar data pursuant to R29
8	(0)	PRECIFLG	1 – Normal integration in P00 0 – Engages 4-time step (P00) logic in integration
7	(0)	CULTFLAG	1 – Star occulted 0 – Star not occulted
6	(0)	ORBWFLAG	1 – W matrix valid for orbital navigation (never set in LUMINARY) 0 – W matrix invalid for orbital navigation
5	(0)	STATEFLG	1 – Result of integration stored in permanent state 0 – Result of integration not to be stored in permanent state
4	(0)	INTYPFLG	1 – Conic integration 0 – Encke integration (precision)

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
3	(0)	VINTFLAG	1 - CSM state vector integration 0 - LM state vector integration
2	(0)	D6OR9FLG	1 - W matrix considered 9-dimensional for integration 0 - W matrix considered 6-dimensional for integration
1	(0)	DIMOFLAG	1 - W matrix is to be used 0 - W matrix is not to be used

FLAGWRD4

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	MRKIDFLG	1 - Mark display awaiting astronaut response 0 - Mark display not awaiting astronaut response
14	(0)	PRIODFLG	1 - Priority display awaiting astronaut response 0 - Priority display not awaiting astronaut response
13	(0)	NRMIDFLG	1 - Normal display awaiting astronaut response 0 - Normal display not awaiting astronaut response
12	(0)	PDSPFLAG	1 - Make normal display priority (set by P20 for R60 display) 0 - Do not make normal display priority
11	(0)	MWAITFLG	1 - Higher priority display operating when mark display initiated; it's asleep and waiting 0 - Mark display not asleep because it's waiting for higher priority display to be completed
10	(0)	NWAITFLG	1 - Higher priority display operating when normal display initiated 0 - If normal display is asleep, it's not because another display was operating when it started
9	(0)	MRKNVFLG	1 - Mark display awaiting key release 0 - Mark display not awaiting key release
8	(0)	NRMNVFLG	1 - Normal display awaiting key release 0 - Normal display not awaiting key release
7	(0)	PRONVFLG	1 - Priority display awaiting key release 0 - Priority display not awaiting key release
6	(0)	PINBRFLG	1 - Astronaut has interferred with existing display 0 - Astronaut has not interferred with display
5	(0)	MRUPTFLG	1 - Mark display interrupted by priority display 0 - Mark display not interrupted by priority display
4	(0)	NRUPTFLG	1 - Normal display interrupted 0 - Normal display not interrupted

FLAGWRD4 (Continued)

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
3	(0)	MKOVFLAG	1 - Mark display interrupting normal 0 - Priority display interrupting mark or normal
2	(0)	spare	
1	(0)	XDSPFLAG	1 - Mark (extended verb) display not to be interrupted 0 - Mark display may be interrupted

FLAGWRD5

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	DSKYFLAG	1 - Displays sent to DSKY 0 - No displays to DSKY
14	(0)	spare	
13	(0)	SNUFFER	1 - U, V jets disabled during DPS burns (V65) 0 - U, V jets enabled during DPS burns (V75)
12	(0)	NOTHROTL	1 - Inhibit full throttle 0 - Permit full throttle
11	(0)	R77FLAG	1 - R77 is operating, suppress all radar alarms and tracker fails 0 - R77 is not operating
10	(0)	RNGSCFLG	1 - Scale change has occurred during RR reading 0 - Scale was the same before the reading and after
9	(0)	DMENFLG	1 - Measurement incorporation using 9x9 W-matrix 0 - Measurement incorporation using 6x6 W-matrix
8	(0)	ZOOMFLAG	1 - P63 throttle-up has occurred 0 - P63 throttle-up has not occurred
7	(0)	ENGONFLG	1 - Engine turned on (commanded on) 0 - Engine turned off (APS or DPS)
6	(0)	3AXISFLG	1 - Maneuver specified by three axes 0 - Maneuver specified by one axis; R60 calls "VECPOINT"
5	(0)	AORBSFLG	1 - Jets 7,15,8,and16 used for P-axis control 0 - Jets 4,12,3, and 11 used for P-axis control
4	(0)	NORRMON	1 - Bypass RR gimbal monitor 0 - Perform RR gimbal monitor
3	(0)	SOLNSW	1 - Lambert does not converge; Time-radius routine cannot solve because of near-circular orbit 0 - Lambert or Time-radius problem soluble

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
2	(0)	MGLVFLAG	1 - Local vertical coordinates computed 0 - Middle gimbal angle computed
1	(0)	RENDWFLG	1 - W-matrix valid for rendezvous navigation 0 - W-matrix not valid for rendezvous navigation

FLAGWRD6

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	S32.1F1	1 - DELVEET1 exceeds the maximum allowable 0 - DELVEET1 within maximum bound
14	(0)	S32.1F2	1 - First pass of CSI iteration 0 - Reiteration
13	(0)	S32.1F3A	See TRGX section
12	(0)	S32.1F3B	See TRGX section.
11	(0)	spare	
10	(0)	GMBDRVSW	1 - Initial positioning of DPS pitch or roll gimbal complete. 0 - Initial positioning of DPS pitch or roll gimbal not complete.
9	(0)	spare	
8	(0)	MUNFLAG	1 - Lunar Landing Average-g navigation(P12, 63, 64, P66, 67, 70 and 71) 0 - Orbital Average-g navigation (P40, 41, 42 and 47)
7	(0)	spare	
6	(0)	REDFLAG	1 - Landing site redesignation permitted 0 - Landing site redesignation not permitted
5	(0)	spare	
4	(0)	spare	
3	(0)	NTARGFLG	1 - Astronaut did overwrite delta (See "S34/35.5") 0 - Astronaut did not overwrite delta
2	(0)	AUXFLAG	1 - Providing IDLEFLAG is not set, SERVICER will exercise DVMON on its next pass 0 - SERVICER will skip DVMON on its next pass, even if the IDLEFLAG is not set. It will then set AUXFLAG

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
1	(0)	ATTFLAG	1 - LM attitude exists in moon-fixed coordinates 0 - No LM attitude available in moon-fixed coordinates

FLAGWRD7

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	ITSWICH	1 - Test Lambert answer against limits 0 - Accept next Lambert TPI search solution
14	(0)	MANUFLAG	1 - Attitude maneuver during RR search 0 - No attitude maneuver during RR search (not set in Luminary)
13	(0)	IGNFLAG	1 - Ignition time has arrived 0 - Ignition time has not yet arrived
12	(0)	ASTNFLAG	1 - Astronaut has okayed ignition 0 - Astronaut has not okayed ignition
11	(0)	SWANDISP	1 - Landing analog displays enabled 0 - Landing analog displays suppressed
10	(0)	NORMSW	1 - Unit normal input to Lambert 0 - Lambert computes its own unit normal
9	(0)	RVSW	1 - Do not compute final state vector in Time-Theta 0 - Compute final state vector in Time-Theta
8	(0)	V67FLAG	1 - Astronaut changing W-matrix initialization values 0 - Astronaut not changing values
7	(1)	IDLEFLAG	1 - Disable Delta-V monitor 0 - Enable Delta-V monitor
6	(0)	V37FLAG	1 - Servicer running 0 - Servicer not running
5	(0)	AVEGFLAG	1 - Average-G desired 0 - Average-G not desired
4	(0)	UPLOCKFL	1 - K, K-bar, K fail 0 - No KKK fail since last error reset
3	(0)	VERIFLAG	**Inverted** whenever P27 is ended with a verb 33
2	(0)	V82EMFLG	1 - Moon vicinity 0 - Earth vicinity

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
1	(0)	TFFSW	1 - Calculate time to perigee 0 - Calculate TFF

FLAGWRD8

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	RPQFLAG	1 - Position vector of secondary body not calculated 0 - Position vector of secondary body calculated
14	(0)	spare	
13	(0)	NEWIFLG	1 - First pass through integration 0 - Succeeding iteration of integration
12		CMOONFLG	1 - Permanent CSM state in lunar sphere of influence 0 - Permanent CSM state in earth's sphere (protected from fresh start)
11		LMOONFLG	1 - Permanent LM state in lunar sphere of influence 0 - Permanent LM state in earth's sphere (protected from fresh start)
10	(0)	FLUNDISP	1 - Current guidance displays inhibited 0 - Current guidance displays permitted
9	(0)	spare	
8		SURFFLAG	1 - LM on lunar surface 0 - LM not on lunar surface (protected from fresh start)
7	(0)	INFINFLG	1 - Closure through infinity required in conic solution 0 - Closure through infinity not required
6	(0)	ORDERSW	1 - Iterator uses second order minimum mode 0 - Iterator uses first order standard mode (not set in Luminary)
5	(0)	APSESW	1 - Orbit does not intersect RDESIRED (Time-Radius) 0 - Orbit does intersect REDESIRED
4	(0)	COGAFLAG	1 - No conic solution; close to rectilinear 0 - Orbit is not too close to rectilinear for solution

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
3	(0)	spare	
2	(0)	INITALGN	1 - Initial pass through P57 0 - Second pass through P57
1	(0)	360SW	1 - Transfer angle near 360 degrees 0 - Transfer angle not near 360 degrees

FLAGWRD9

Bit and initial value (fresh start)

Bit	Mnemonic	Meaning when 1 and 0
15 (0)	spare	
14 (0)	FLVR	1 - Vertical rise (ascent guidance) 0 - Non-vertical rise
13 (0)	P7071FLG	1 - P70 or P71 using ascent guidance 0 - P12 using ascent guidance
12 (0)	FLPC	1 - No position control (ascent guidance) 0 - Position control
11 (0)	FLPI	1 - Pre-ignition phase (ascent guidance) 0 - Regular guidance
10 (0)	FLRCS	1 - RCS injection mode (ascent guidance) 0 - Main engine mode
9 (0)	LETABORT	1 - Abort programs are enabled 0 - Abort programs are not enabled
8 (0)	FLAP	1 - APS continued abort after DPS staging (ascent guidance) 0 - APS abort is not a continuation
7 (0)	ABTTGFLG	1 - Abort targeting to use J_2, K_2 0 - Abort targeting to use J_1, K_1
6 (0)	ROTFLAG	1 - P70 and P71 will force vehicle rotation in the preferred direction 0 - P70 and P71 will not force vehicle rotation in the preferred direction
5 (0)	QUITFLAG	1 - Discontinue orbital integration 0 - Continue integration
4 (0)	spare	
3 (0)	MID1FLAG	1 - Integrate to TDEC 0 - Integrate to TIMENOW
2 (0)	MIDAVFLG	1 - Integration entered from one of the drifting flight to powered flight handover routines 0 - Integration not entered as above
1 (0)	AVEMIDSW	1 - AVETOMID calling for W-matrix integration; do not write over $\underline{RN}$, $\underline{VN}$, PIPTIME 0 - AVETOMID without W-matrix integration; allow set up of RN, VN, PIPTIME

FLGWRD10

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
15	(0)	spare	
14	(0)	INTFLAG	1 - Integration in progress 0 - Integration not in progress
13		APSFLAG	1 - Ascent stage 0 - Descent stage (protected from fresh start)
12	(0)	spare	
11	(0)	spare	
10	(0)	spare	
9	(0)	spare	
8	(0)	spare	
7	(0)	REINTFLG	1 - Integration routine to be restarted 0 - Integration routine not to be restarted
6	(0)	spare	
5	(0)	spare	
4	(0)	spare	
3	(0)	spare	
2	(0)	spare	
1	(0)	spare	

FLGWRD11

Bit and initial value (fresh start)

Bit	Initial value	Mnemonic	Meaning when 1 and 0
15	(1)	LRBYPASS	1 - Bypass all landing radar updates 0 - Do not bypass
14	(0)	spare	
13	(0)	spare	
12	(0)	VXINH	1 - If Z velocity data unreasonable, bypass X velocity update on next pass 0 - Update X-axis velocity
11	(0)	PSTHIGAT	1 - Past higate 0 - Prehigate
10	(0)	NOLRREAD	1 - Landing radar reposition; bypass update 0 - Landing radar not repositioning
9	(0)	XORFLG	1 - Below limit inhibit X-axis override 0 - Above limit do not inhibit
8	(0)	LRINH	1 - Landing radar updates permitted by astronaut 0 - Landing radar updates inhibited by astronaut
7	(0)	VELDATA	1 - Landing radar velocity measurement made 0 - Landing radar velocity measurement not made
6	(0)	LPOS2FLG	1 - Position 2 transformation for LR data being used. 0 - Position 1 assumed.
5	(0)	READVEL	1 - Ok to read landing radar velocity data 0 - Do not read landing radar velocity data
4	(0)	RNGEDATA	1 - Landing radar altitude measurement made 0 - Landing radar altitude measurement not made
3	(0)	NO511FLG	1 - Do not test LR position in R12. 0 - OK to test LR position.
2	(0)	VFLSHFLG	1 - Landing radar velocity fail lamp should be flashing 0 - Landing radar velocity fail lamp should not be flashing

FLGWRD11 (Continued)

Bit and initial value (fresh start)

		Mnemonic	Meaning when 1 and 0
1	(0)	HFLSHFLG	1 - Landing radar altitude fail lamp should be flashing 0 - Landing radar altitude fail lamp should not be flashing

DAPBOOLS

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	PULSES	1 - Minimum impulse command mode 0 - Not minimum impulse
14	(1)	USEQRJTS	1 - Use of gimbal not allowed 0 - Gimbal may be used
13	(0)	CSMDOCKD	1 - CSM attached to LM 0 - CSM not attached
12	(0)	OURRCBIT	1 - Still in rate command mode 0 - Not in rate command mode
11	(0)	ACC4OR2X	1 - 4-jet P-axis translation 0 - 2-jet P-axis translation
10	(1)	AORBTRAN	1 - X translation B system 0 - X translation A system
9	(0)	XOVINHIB	1 - LPD phase; X-axis override disabled 0 - Not in Landing Point Designation Phase
8	(1)	DRIFTBIT	1 - Assume that offset acceleration is zero 0 - Offset acceleration likely
7	(1)	RHCSCALE	1 - Normal RHC scaling 0 - Fine RHC scaling
6	(0)	ULLAGER	1 - Internal ullage request 0 - No program ullage request
5	(1)	DBSLECT2	
4	(0)	DBSELECT	

N46 Digit "D" Load	DAP Deadband	BIT 5	BIT 4
0	$\pm 0.3^{\circ}$	0	0
1	$\pm 1.0^{\circ}$	0	1
2	$\pm 5.0^{\circ}$	1	0
3	$\pm 5.0^{\circ}$	1	1

Bit	Initial	Mnemonic	Meaning when 1 and 0
3	(0)	ACCSOKAY	1 - Computed accelerations probably correct 0 - Computed accelerations probably incorrect
2	(1)	AUTRATE2	Used together to determine index (RATEINDX)
1	(0)	AUTRATE1	which is used to select attitude maneuver rate

Mnemonic	Meaning when 1 and 0
00_2	0.2 degrees/second
01_2	0.5 degrees/second
10_2	2.0 degrees/second
11_2	10.0 degrees/second

RADMODES

Bit and initial value (fresh start)

Bit	Initial	Mnemonic	Meaning when 1 and 0
15	(0)	CDESFLAG	1 - LGC sends continuous designate commands to RR 0 - LGC checks for lock-on when designating
14	(0)	REMODFLG	1 - Remode of RR antenna is required 0 - No remode of RR antenna
13	(0)	RCDUOFLG	1 - RR CDU's are being zeroed 0 - RR CDU's are not being zeroed
12	(0)	ANTENFLG	1 - RR antenna is in mode 2 0 - RR antenna is in mode 1
11	(0)	REPOSMON	1 - RR antenna reposition taking place 0 - No RR antenna reposition taking place
10	(0)	DESIGFLG	1 - RR antenna designation taking place 0 - No RR antenna designation taking place
9	(0)	ALTSCALE	1 - LR altitude reading on high scale 0 - LR altitude reading on low scale
8	(0)	LRVELFLG	1 - LR velocity data fail 0 - LR velocity data good
7	(1)	RCDUFAIL	1 - RR CDU fail has not occurred 0 - RR CDU fail has occurred
6	*	LRPOSFLG	1 - LR antenna command to position #2 0 - LR antenna in position #1 * - State of B6 CH33
5	(0)	LRALTFLG	1 - LR altitude data fail 0 - LR altitude data good
4	(0)	RRDATAFL	1 - RR data fail 0 - RR data good
3	(0)	RRRSFLAG	1 - RR range data on high scale 0 - RR range data on low scale
2	(1)	AUTOMODE	1 - RR not in automatic mode 0 - RR is in automatic mode
1	(0)	TURNONFL	1 - RR turn-on sequence in progress 0 - No RR turn-on sequence in progress

List of References to Flagwords and Channels

The following is a listing of some of the routines in which the majority of the discrete bits of information are set, reset, and tested. Input channels can only be tested (though some flip-flops are automatically reset when tested, they will be set again immediately if the relevant hardware signal is still present). Output channels are usually just set and reset, but they can also be tested to assure that they are configured as required. Flagwords are set, reset, and tested by the programs. References refer to this document, not to the listing.

This list tries to include all references to each bit, but since flagwords and channels can be addressed in many different ways in the LGC program, there is no way to assure that all references are included here. (In some cases references are deliberately left out because the bit does not affect the material presented in this document.)

Channel 5

Bit	Routines
8	1 - WRITEU 0 - DOFSTRT1 WRITEU MOREIDLE test - TRYGTS
7	1 - WRITEU 0 - DOFSTRT1 WRITEU MOREIDLE test - TRYGTS
6	1 - WRITEV 0 - DOFSTRT1 WRITEV MOREIDLE test - TRYGTS
5	1 - WRITEV 0 - DOFSTRT1 WRITEV MOREIDLE test - TRYGTS
4	1 - WRITEU 0 - DOFSTRT1 WRITEU MOREIDLE test - TRYGTS
3	1 - WRITEU 0 - DOFSTRT1 WRITEU MOREIDLE test - TRYGTS

Bit	Routines
2	1 - WRITEV 0 - DOFSTRT1 WRITEV MOREIDLE test - TRYGTS
1	1 - WRITEV 0 - DOFSTRT1 WRITEV MOREIDLE test - TRYGTS

Channel 6

Bit	Routines
8	1 - WRITEP 0 - WRITEP MOREIDLE DOFSTRT1 test
7	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
6	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
5	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
4	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
3	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
2	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test
1	1 - WRITEP 0 - DOFSTRT1 MOREIDLE WRITEP test

Channel 11

Bit	Routines
14	1 - DOFSTART IMUMON ENGINOF3 GOPROG 0 - IGNITION ABRTJASK

Bit	Routines
13	1 - IGNITION GOPROG ABRTJASK 0 - IMUMON ENGINOF3 DOFSTART
10	1 - ERROR 0 - STARTSB2 DOFSTART
9	1 - READACCS 0 - STARTSB2 AVGEND DOFSTART
7	1 - V37 VBTSTLTS DSPALARM ALMCYCLE CHARALRM UPERROUT UPEND70 P20LEMB7 ALM/END V73UPDAT UPEND73 ABORTALM 0 - STARTSB2 ERROR TSTLTS3 DOFSTART
6	1 - FLASHSUB TESTNN VBTSTLTS REQDATZ REQMM 0 - NV50DSP STARTSB2 TSTLTS3 BLANKDSP ENTER GOLOADLV VBRESEQ DOFSTART
5	1 - CHARIN NV50DSP MONDO VBTSTLTS 0 - STARTSB2 WITCHONE RELDSP RELDSP1 DOFSTART
4	1 - IMUMON VBTSTLTS 0 - IMUMON STARTSB2 TSTLTS3 DOFSTART
3	1 - UPRUPT VBTSTLTS 0 - STARTSB2 TSTLTS3 V73UPDAT UPOUT4 ERROR VBRELDSP DOFSTART
2	1 - ADVAN 0 - STARTSB2 DUMMYJB2 DOFSTART
1	1 - SETISSW VBTSTLTS 0 - SETISSW TSTLTS3 DOFSTART test - ENDIMU

<u>Channel 12</u>

Bit	Routines
15	1 - ENDTNON 0 - STARTSB2 UNZ2 CAGESUB DOFSTRT1 test - IMUMON
14	1 - R23LEM LRS24.1 R04X DODES R29DODES 0 - R21LEM TRMTRACK RRGIMON STDESIG R04END R29 ENDRRD29 P63LM DOFSTRT1 test - R22LEM
13	1 - LRPOS2 0 - STARTSB2 LRPOSCAN DOFSTRT1 test -

Bit	Routines
12	1 - ACDT+C12 TRIMGIMB 0 - ACDT+C12 TRIMGIMB SUPERJOB MOREIDLE DOFSTRT1 NEGUSUM test - SPSCONT
11-9	same as bit 12, bit 10 reset to 0 by PITCHOFF
8	1 - LANDISP 0 - STARTSB2 IMUMON DISPRSET DOFSTRT1
6	1 - NEEDLER COARS IMUATTCK GOPROG CA+ECE DOFSTRT1 0 - NEEDLER IMUMON CAGESUB SETCOARS IMUZERO DOFSTRT1 test - NEEDLER
5	1 - IMUZERO ISSZERO CAGESUB 0 - IMUMON UNZ2 IMUZERO2 IMUFINE DOFSTRT1 test - IMUATTCK
4	1 - SETCOARS CAGESUB GOPROG DOFSTRT1 0 - IMUMON UNZ2 IMUZERO IMUFINE DOFSTRT1 test - TNONTEST GLOCKMON IFAILOK IMUATTCK SETCOARS 8192AUG TSTLTS3
2	1 - SETRRECR INTLZE 0 - STARTSB2 RRAUTCHK RRGIMON DORREPOS TRMTRACK STDESIG RESET22 IMUMON R24END R24LEM3 RRDESDUN RRDESEND POOH RR1AX2 RRDESNB R29DPAS2 DOFSTRT1 PROG20A R21LEM9 DISPRSET P12LM test - SETRRECR SPEEDRUN
1	1 - RRZEROSB NORRGMON 0 - STARTSB2 RRZEROSB DOFSTRT1

Channel 13

Bit	Routines
15	1 - JTLST T6JOBCHK 0 - STARTSB1 DOFSTRT1
14	1 - none 0 - DOFSTRT1
13	1 - none 0 - DOFSTRT1
12	1 - REDESMON STARTP64 STARTSB2 0 - DOFSTRT1

Bit	Routines
11	1 - P06
	0 - POSTAND STARTSB2 DOFSTRT1
10	1 - VBTSTLTS
	0 - ERROR TSTLTS3 STARTSB2 DOFSTRT1
9	1 - ZEROENBL
	0 - STARTSB2 DOFSTRT1
8	1 - ZEROENBL
	0 - STARTSB2 DOFSTRT1
7	1 - DODOWNTM
	0 - DOFSTRT1 WOZERO
	test - DODOWNTM
6-5	not set in LUMINARY
4	1 - RADSTART
	0 - INITREAD STARTSB2 DOFSTRT1
	test - C13STALL
3	1 - RADSTART
	0 - INITREAD STARTSB2 DOFSTRT1
	test - RADAREAD
2	1 - RADSTART
	0 - INITREAD STARTSB2 DOFSTRT1
	test - RADAREAD
1	1 - RADSTART
	0 - INITREAD STARTSB2 DOFSTRT1
	test - RADAREAD RENDRAD

<u>Channel 14</u>

15	1 - COARS2 ATTCK2 .... NEEDLES
	0 - DOFSTRT1 IMUMON STARTSB2
14	1 - COARS2 ATTCK2 . . . NEEDLES
	0 - DOFSTRT1 IMUMON STARTSB2
13	1 - COARS2 ATTCK2 . NEEDLES
	0 - DOFSTRT1 IMUMON STARTSB2
12	1 - RROUT SPEEDRUN
	0 - DOFSTRT1 IMUMON STARTSB2

Bit	Routines
11	1 - RROUT SPEEDRUN 0 - IMUMON STARTSB2 DOFSTRT1
10	1 - GYROEXIT 0 - STRTGYRO IMUMON SETCOARS STARTSB2 DOFSTRT1
9	1 - STRTGYR2 0 - STRTGYRO IMUMON STARTSB2 DOFSTRT1
8	1 - STRTGYR2 0 - STRTGYRO IMUMON STARTSB2 DOFSTRT1
7	1 - STRTGYR2 0 - STRTGYRO IMUMON STARTSB2 DOFSTRT1
6	1 - IMUPULSE 0 - IMUMON DOFSTRT1
4	1 - GOPROG DOIT P40ZOOM THROTUP ENGINOF3 0 - STARTSB2 DOFSTRT1
3	1 - ALTROUT1 ALTOUT1 0 - STARTSB2 DOFSTRT1
2	1 - ALTROUT1 0 - ALTOUT1 STARTSB2 DOFSTRT1

Channel 15

Tested in routines KEYRUPT1 and LIGHTSET

Channel 16

7	DESCBITS SOMEKEY MARKRUPT
6	SOMEKEY MARKRUPT
5	LIGHTSET MARKRUPT
4	MARKRUPT
3	MARKRUPT

Channel 30

Bit	Routines in which bit is tested
	Bits 15-11 are tested only in IMUMON
10	P40AUTO REDOMANC R61+LO2 FINDCDUW CHEKBITS
9	IMUMON
7	RRCDUCHK
6	LANDISP
5	P40AUTO
4	R10,R11
3	P66HZ
1	R10,R11

Channel 31

Bit	Routines in which bit is tested
15	DETENTCK CHEKSTIK
14	P40AUTO REDOMANC NEWDELHI IMUATTCK FINDCDUW R61+LO2 CHEKBITS
13	TSNEXTP TSNEXTS IMUATTCK DETENTCK QRAXIS CHEKBITS LUNLAND REDESMON
12-9	CHKVISFZ
8	RCS
7	RCS
6	TSNEXTS PITFALL REDESMON
5	TSNEXTS PITFALL REDESMON
4	TSNEXTP
3	TSNEXTP
2	TSNEXTS PITFALL REDESMON
1	TSNEXTS PITFALL REDESMON

Channel 32

Bit	Routines
14	PROCEEDE
9	DVMON
8-1	RCSMONIT

Channel 33

15	
14	
13	C33TEST
12	C33TEST
11	C33TEST
9	LRHEIGHT R04Z
8	INITREAD DGCHECK SCALCHNG R77CHECK
7	LRPOS2 LRPOSCAN P2CHK MUNRETRN LRPOSOUT
6	DORSAMP ASTNRET MUNRETRN R04Z P1CHK LRPOSOUT
5	DGCHECK INITREAD SCALCHNG R77CHECK
4	DODES INITREAD DGCHECK SCALCHNG DATGDCHK R29DPAS2 R61C+L01
3	RENDRAD LRS22.1 R04Z
2	RRAUTCHK R04X R22LEM P20LEMB7

FLAGWRDO

Bit	Mnemonic	Routines
15	NEED2FLG	1 - RATEDISP 0 - R60LEM DAPATTER TOTATTER DOFSTRT1 test - ALTDSPLY
14	JSWITCH	1 - ENDSTATE 0 - INTGRATE DOFSTRT1 test - NBRANCH DIFEQ+2 NEXTCOL INTGRATE
13	MIDFLAG	1 - TESTLOOP 0 - TESTLOOP DOFSTRT1 test - TIMESTEP ACCOMP DOW..
12	MOONFLAG	1 - P21CONT INTEGRV ORIGCHNG INTWAKEU INITVEL2 INTINT REV83 P76 USEPIOS ATTACHIT EXGSUB OTHINT 0 - P21CONT INTEGRV ORIGCHNG INTWAKEU INITVEL2 INTINT REV83 P76 ATTACHIT DOFSTRT1 OTHINT test - INTEGRV INTEGRVS RECTOUT TIMESTEP ORIGCHNG ACCOMP OBLATE ENDSTATE KEPPREP A-PCHK INTWAKEU P76 DOW.. ATTACHIT ORBCHGO
11	P21FLAG	1 - P21VSAVE 0 - GOPROG3 DOFSTRT1 test - PROG21
10	FSPASFLG	1 - R21LEM4 0 - 60TIMES DOFSTRT1 test - 60TIMES
9	P25FLAG	1 - PROG25 0 - TRMTRACK POOH RESET22 P63LM DOFSTRT1 test - TRMTRACK CANV37 RESET22 P25LEM1 V37RET
8	IMUSE	1 - IMUCHK RO2BOTH 0 - TRMTRACK IMUMON CANV37 POOH P06 ENDTEST1 DOFSTRT1 RESET22 SOMERR2 test - AGSVCALC TNONTEST ENDTNON IMUMON
7	RNDVZFLG	1 - PROG20A 0 - TRMTRACK IMUMON POOH P06 RESET22 P12LM DOFSTRT1 P63LM PROG25 test - CANV37 RESET22 RRCDUCHK P20LEMC P20LEMC1 TRMTRACK R22LEM CALLDGCH LUNDESCH V37RET CSMINT RELINUS
6	RRNBSW	1 - RRDESNB 0 - RRDESSM DOFSTRT1 CSMINT test - DODES

FLAGWRDO (Continued)

Bit	Mnemonic	Routines
5	LOKONSW	1 - VBCOARK R21LEM1 0 - VBCOARK DOFSTRT1 R21LEM R21LEM8 test - DODES
4	NEEDLFLG	1 - TOTATTER R60LEM 0 - DAPATTER DOFSTRT1 test - ALTDSPLY
3	FREEFLAG	1 - LSPOS CHKSDATA GVDETER 0 - LSPOS CHKSDATA GVDETER DOFSTRT1 test - P51C R51E SURFLINE GVDETER
2	R10FLAG	1 - P12LM ABRTJASK 0 - DOFSTRT1 test - DISPRSET SPEEDRUN LANDISP
1	OLDESFLG	1 - R29.LOS 0 - R29 DOFSTRT1 test - R29.LOS

FLAGWRD1

Bit	Mnemonic	Routines
15	NJETSFLG	1 - DPDAT1 0 - DPDAT1 DOFSTRT1 test - P41LM
14	DIDFLAG	1 - LANDISP 0 - GOPROG3 DISPRSET DOFSTRT1 test - LANDISP
13	ERADFLAG	1 - never set 0 - DOFSTRT1 P52LS P57POST P21VSAVE test - LAT-LONG LALOTORV
12	RODFLAG	1 - STRTP66A 0 - GOPROG3 DOFSTRT1 P66 test - LUNLAND
10	R61FLAG	1 - R61LEM 0 - R65LEM DOFSTRT1 test - R61C+L06
8	VEHUPFLG	1 - ORBCHGO CSMVEC 0 - LEMVEC PROG20 DOFSTRT1 test - UPPSV LSR22.3 RANGEBQ FAZC FAZAB3
7	UPDATFLG	1 - RESET22 PROG20A S34/35.5 P30 P32/P72C P33/P73B P75 P74 P72 P73 0 - TRMTRACK SEUDOPOO P30 CLUPDATE DOFSTRT1 test - R22LEM
6	NOUPFLAG	1 - UPDATOFF 0 - CSMVEC DOFSTRT1 test - R22LEM
5	TRACKFLG	1 - RESET22 PROG20A PROG25 P30 P72 P73 P74 P75 P76 0 - RNDREFDR TRMTRACK SEUDOPOO DOFSTRT1 test - TRMTRACK P20LEMC P20LEMD1 R22LEM P25LEM1 P20LEMC1 RDRUSECK P20LEMB RELINUS R61C+L02
3	SLOPESW	1 - LAMBERT 0 - ITERATOR DOFSTRT1 test - ITERATOR LAMBLOOP
2	GUESSW	1 - INITVEL 0 - INITVEL2 DOFSTRT1 test - LAMBERT

FLAGWRD2

Bit	Mnemonic	Routines
15	DRIFTFLG	1 - P51B CAL53A UNZ2 AVGEND GYCOARS INITBY 0 - RNDREFDR PREREAD GYCOARS DOFSTRT1 test - SVCT3 NBDONLY UNZ2
14	SRCHOPTN	1 - R24LEM 0 - P20LEMB3 PROG20A DOFSTRT1 TRMTRACK test - P20LEMB3 LRS24.1 DORROUT
13	ACMODFLG	1 - P20LEMB1 0 - P20LEMB3 PROG20A DOFSTRT1 test - P20LEMB3
12	LOSCMFLG	1 - R21LEM10 BEGDES29 R29.LOS 0 - RRDESDUN R29.LOS R21LEM1 R29DPAS2 PROG20A R24LEM R29 DOFSTRT1 R21LEM4 RRDESNB test - BEGDES29 R29.LOS DORROUT LPS20.1
11	STEERSW	1 - DVMON 0 - AVERAGEG DOFSTRT1 VGAIN* test - VGAIN* STEER?
9	IMPULSW	1 - S40.132 VGAIN* 0 - STEERING S40.13 P42IGN DOFSTRT1 test - P42IGN STEERING
8	XDELVFLG	1 - P30 ADVANCE 0 - INITVEL7 DOFSTRT1 test - S40.1 UPDATEVG S40.8
7	ETPIFLAG	1 - P74 0 - P74 DOFSTRT1 test - P34/P74C INTLOOP ELCALC
6	FINALFLG	1 - N45PROC P30 0 - SELECTMU DOFSTRT1 test - S34/35.5 P32/P72C P33/P73B VN1645 N45PROC

FLAGWRD2 (Continued)

Bit	Mnemonic	Routines
5	AVFLAG	1 - P42STAGE S40.9 P34 P35 P32 P33 0 - P74 P75 P72 P73 DOFSTRT1 test - PRECSET,
4	PFRATFLG	1 - S40.2,3 0 - R51E GYCOARS REGCOARS DOFSTRT1 test - PROG52
3	CALCMAN3	Not really functional: set in KALCMAN3; reset in DOFSTRT1
2	CALCMAN2	1 - WCALC 0 - NEWANGL DOFSTRT1 test - NEWANGL
1	NODOFLAG	1 - AGSVCALC STATINT1 P06 P76 0 - AGSVCALC STATINT1 POOH POSTAND DOFSTRT1 POODOO P76 test - V37

FLAGWRD3

Bit	Mnemonic	Routines
15	POOHFLAG	1 - STATINT1 0 - CANV37 DOFSTRT1 test - TESTLOOP
14	GLOKFAIL	1 - CALCGA 0 - REDO CANV37 DOFSTRT1 test - REDO
13	REFSMFLG	1 - P51C GYCOARS REGCOARS SURFDISP 0 - RNDREFDR GYCOARS GVDETER test - AGSINIT R02BOTH VN1645 PACKOPTN DSPOPTN R59
12	LUNAFLAG	1 - LANDJUNK P52LS P57POST P21VSAVE 0 - DOFSTRT1 P21VSAVE test - LAT-LONG LALOTORV
11	NOR29FLG	1 - AVGEND DOFSTRT1 0 - CMPONENT test - COPYCYC1 STARTSB2 R29RDJOB RDRUSECK
10	VFLAG	1 - R56 PIC3 0 - PIC3 DOFSTRT1 test - PIC3 PICEND
9	R04FLAG	1 - R04 R61C+L01 0 - R04END CANV37 PROG20A DOFSTRT1 R61C+L01 STARTSB2 test - R04Z RADAREAD RESAMPLE R77
	READRFLG	1 - R29DPAS2 0 - ENDR29D STARTSB2 DOFSTRT1 test - R29READ COPYCYC1
8	PRECIFLG	1 - STATINT1 CSMPREC INTEGRVS LEMPREC 0 - STATINT1 INTEXIT DOFSTRT1 test - TESTLOOP
7	CULTFLAG	1 - OCCULT 0 - OCCULT DOFSTRT1 test - PIC3
6	ORBWFLAG	1 - never set 0 - WMATEND INTWAKEU DOFSTRT1 test - AVETOMID
5	STATEFLG	1 - SETIFLGS WMATEND LSR22.3 LSR22.4 0 - ENDINT TESTLOOP DOFSTRT1 POODOO INTEXIT test - A-PCHK

Bit	Mnemonic	Routines
4	INTYPFLG	1 - OTHCONIC CSMCONIC LEMCONIC INTINT EXGSUB REV83 0 - MIDTOAV2 SETIFLGS CSMPREC LEMPREC MIDTOAV2 P76 LSR22.3 LSR22.4 INITVEL2 INTINT ORBCHGO DOFSTRT1 GETRVN PROG21 test - ALOADED
3	VINTFLAG	1 - STATINT1 CSMPREC CSMCONIC AVETOMID UPPSV LSR22.3 LSR22.4 ORBCHGO PROG21 0 - STATINT1 LEMPREC LEMCONIC AVETOMID MIDTOAV2 UPPSV LSR22.3 LSR22.4 DOFSTRT1 ORBCHGO PROG21 test - INTEGRV ENDSTATE A-PCHK
2	D6OR9FLG	1 - STATINT1 AVETOMID UPPSV LSR22.3 ORBCHGO 0 - ORBCHGO SETIFLGS LSR22.4 DOFSTRT1 test - ENDSTATE
1	DIMOFLAG	1 - STATINT1 AVETOMID UPPSV LSR22.3 LSR22.4 ORBCHGO 0 - SETIFLGS CSMPREC LEMPREC CSMCONIC LEMCONIC INTEGRVS WMATEND MIDTOAV2 LSR22.3 ORBCHGO DOFSTRT1 PROG21 P21CONT test - ACCOMP ENDSTATE

FLAGWRD4

Bit	Mnemonic	Routines
15	MRKIDFLG	1 - FLASHSUB 0 - JOBXCHS ENDRET STARTSB2 DOFSTRT1 test - OKTOPLAY MAKEPRIO FLASHSUB TERMATE PINBRNCH
14	PRIODFLG	1 - FLASHSUB 0 - ENDRET STARTSB2 DOFSTRT1 test - NORMBNCH MAKEMARK MAKEPRIO PINBRNCH ENDEXT OKTOPLAY TESTXACT TERMATE
13	NRMIDFLG	1 - FLASHSUB 0 - JOBXCHS ENDRET STARTSB2 DOFSTRT1 test - MAKEPLAY MAKEMARK MAKEPRIO PINBRNCH ENDEXT
12	PDSPFLAG	1 - R61C+LO2 RELINUS 0 - STARTSB2 R61C+LO2 DOFSTRT1 test - OKTOPLAY MAKEMARK CHKLINUS TESTXACT R61TEST
11	MWAITFLG	1 - MAKEMARK 0 - MARKWAKE STARTSB2 DOFSTRT1 test - OKTOPLAY MAKEMARK NORMRET
10	NWAITFLG	1 - OKTOPLAY 0 - NORMWAKE STARTSB2 DOFSTRT1 test - MAKEPLAY NORMRET
9	MRKNVFLG	1 - NV50DSP 0 - JOBXCHS NV50DSP STARTSB2 DOFSTRT1 test - OKTOPLAY MAKEMARK MAKEPRIO WITCHONE
8	NRMNVFLG	1 - NV50DSP 0 - JOBXCHS NV50DSP STARTSB2 DOFSTRT1 test - MAKEPLAY MAKEMARK MAKEPRIO WITCHONE
7	PRONVFLG	1 - NV50DSP 0 - NV50DSP STARTSB2 DOFSTRT1 test - OKTOPLAY MAKEMARK MAKEPRIO WITCHONE TESTXACT
6	PINBRFLG	1 - NORMBNCH 0 - ENDRET STARTSB2 DOFSTRT1 test - FLASHSUB
5	MRUPTFLG	1 - JOBXCHS 0 - MARKWAKE STARTSB2 DOFSTRT1 test - OKTOPLAY MAKEMARK NORMRET

FLAGWRD4 (Continued)

Bit	Mnemonic	Routines
4	NRUPTFLG	1 - JOBXCHS 0 - NORMWAKE STARTSB2 DOFSTRT1 test - MAKEPLAY NORMRET
3	MKOVFLAG	1 - MAKEMARK 0 - MARKPLAY MARKRET STARTSB2 DOFSTRT1 test - JOBXCHS
1	XDSPFLAG	1 - AGSVCALC MARKPLAY 0 - ENDEXT STARTSB2 DOFSTRT1 GOTOPOOH test - OKTOPLAY

FLAGWRD5

Bit	Mnemonic	Routines
15	DSKYFLAG	1 - KEYRUPT1 0 - DOFSTRT1 test - T4RUPT
13	SNUFFER	1 - SNUFFOUT 0 - OUTSNUFF DOFSTRT1 test - AFTERTJ
12	NOTHROTL	1 - S40.13 S40.13D 0 - P40LM S40.13D P63LM DOFSTRT1 test - P40IGN
11	R77FLAG	1 - R77 0 - R77END DOFSTRT1 STARTSB2 test - R77CHECK DORSAMP RDRUSECK
10	RNGSCFLG	1 - SCALCHNG 0 - LRS22.1 LRHJOB R29RANGE DOFSTRT1 test - READRDOT LRHJOB R29RANGE
9	DMENFLG	1 - LSR22.3 0 - LSR22.4 DOFSTRT1 test - INCORP1 INCORP2 FAZAB3 INCOR2-3
8	ZOOMFLAG	1 - P63ZOOM 0 - TIG-5 DOFSTRT1 test - LUNLAND
7	ENGONFLG	1 - IGNITION ABRTJASK 0 - ENGINOF3 IMUMON DOFSTRT1 test - GOPROG DCMCL
6	3AXISFLG	1 - R62DISP 0 - R52 ENDMANU1 V89RECL P40IN P41LM CANV37 DOFSTRT1 R61C+LO2 test - R60LEM REDOMANC
5	AORBSFLG	1 - PURGENCY PJETSLEC 0 - PURGENCY PJETSLEC DOFSTRT1 test - TSNEXTP PEGI PURGENCY

FLAGWRD5 (CONTINUED)

Bit	Mnemonic	Routines
4	NORRMON	1 - VBCOARK R23LEM R21LEM8 0 - RRDESEND STARTSB2 PROG20A R23LEM R23LEM2 DOFSTRT1 R21LEM1 test - RRGIMON
3	SOLNSW	1 - TIMERAD SUFFCHEK LAMBERT 0 - TIMERAD DOFSTRT1 LAMBERT test - none (telemetry)
2	MGLVFLAG	1 - GET.LVC 0 - GET+MGA DOFSTRT1 test - none (telemetry)
1	RENDWFLG	1 - WLINIT 0 - WMATRXNG WMATEND INTWAKEU V67CALL DOFSTRT1 ATMAG test - STATINT1 AVETOMID UPPSV ORBCHGO LSR22.3

FLAGWRD6

Bit	Mnemonic	Routines
15	S32.1F1	1 - CSI/B2 0 - CSI/A SCNDSOL DOFSTRT1 test - CSI/B2
14	S32.1F2	1 - CSI/A SCNDSOL 0 - FRSTPAS DOFSTRT1 test - CIRCL
13	S32.1F3A	1 - CIRCL FIFTYFPS 0 - CSI/A SCNDSOL DOFSTRT1 test - CSI/B2 CIRCL SCNDSOL
12	S32.1F3B	1 - CSI/A FIFTYFPS 0 - CIRCL SCNDSOL DOFSTRT1 test - CSI/B2 CIRCL SCNDSOL
10	GMBDRVSW	1 - PITCHOFF 0 - TRIMGIMB DOFSTRT1 test - PITCHOFF
8	MUNFLAG	1 - P63LM P12LM 0 - AVGEND DOFSTRT1 CANV37 test - NORMLIZE READACCS AVERAGEG RRGIMON P41SPOT SERVIDLE V83CALL GETRVN
6	REDFLAG	1 - P64DISPS 0 - P64DISPS STARTP64 P63LM DOFSTRT1 STRTP66A test - P64DISPS REDESIG
3	NTARGFLG	1 - NTARGCHK 0 - S34/35.5 DOFSTRT1 test - Not shown in document

FLAGWRD6 (Continued)

Bit	Mnemonic	Routines
2	AUXFLAG	1 - AVERAGEG 0 - AVERAGEG DOFSTRT1 test - AVERAGEG
1	ATTFLAG	1 - REFMF 0 - DOFSTRT1 test - PACKOPTN DSPOPTN ATTCHK

FLAGWRD7

Bit	Mnemonic	Routines
15	ITSWICH	1 - P34/P74C P33/P73B 0 - P34/P74C INTLOOP DOFSTRT1 test - INTLOOP ELCALC
13	IGNFLAG	1 - TIG-0 0 - TIG-5 P42IGN DOFSTRT1 test - *PROCEED
12	ASTNFLAG	1 - *PROCEED 0 - TIG-5 P42IGN DOFSTRT1 test - TIG-0
11	SWANDISP	1 - P63IGN ABRTIGN 0 - DOFSTRT1 AVGEND test - LANDISP
10	NORMSW	1 - INITVEL2 0 - PARAM DOFSTRT1 HAVEGUES test - S40.1B S40.9 GEOM UPDATEVG RASTEER1
9	RVSW	1 - INTLOOP CSI/B2 VN0611 0 - CDHMVR DOFSTRT1 ORBCHGO test - COMMNOUT
8	V67FLAG	1 - V67CALL 0 - DOFSTRT1 V67CALL test - V67CALL
7	IDLEFLAG	1 - STEERING ENGINOF1 COMFAIL DOFSTRT1 MAINENG SERVIDLE 0 - P42IGN GOABORT COMFAIL4 test - STEERING AVERAGE G MAINENG
6	V37FLAG	1 - PREREAD 0 - DOFSTRT1 AVGEND test - V37 POODOO RDRUSECK
5	AVEGFLAG	1 - PREREAD 0 - V37 DOFSTRT1 test - V82CALL READACCS REV83 RRGIMON P70 P71 R10,R11 V83CALL LRPOS2K
4	UPLOCKFL	1 - UPRUPT 0 - UPRUPT DOFSTRT1 test - UPRUPT

FLAGWRD7 (Continued)

Bit	Mnemonic	Routines
3	VERIFLAG	1 - UPSTORE 0 - UPSTORE DOFSTRT1 test - none (for telemetry)
2	V82EMFLG	1 - V82GOFF1 V82GON1 0 - V82GOFF1 V82GON1 DOFSTRT1 test - SR30.1
1	TFFSW	1 - CALCTPER 0 - CALCTFF DOFSTRT1 test - CALCTFF

FLAGWRD8

Bit	Mnemonic	Routines
15	RPQFLAG	1 - INTEGRV INTEGRVS GOBAQUE 0 - ACCOMP DOFSTRT1 test - TIMESTEP LUNSPH
13	NEWIFLG	1 - INTEGRV INTEGRVS 0 - TESTLOOP DOFSTRT1 test - TESTLOOP
12	CMOONFLG	1 - ENDSTATE A-PCHK INTWAKEU P76 ATTACHIT ORBCHGO 0 - ENDSTATE A-PCHK INTWAKEU P76 ATTACHIT ORBCHGO test - INTEGRV SELECTMU INTINT CSI/B2 P76
11	LMOONFLG	1 - ENDSTATE A-PCHK INTWAKEU 0 - ENDSTATE A-PCHK INTWAKEU test - V82GON1 AVETOMID INTEGRV FAZC ATTACHIT REV83 OTHINT
10	FLUNDISP	1 - COMFAIL 0 - GOABORT IGNITION GOCUTOFF COMFAIL4 DOFSTRT1 test - DISPEXIT ASCTERM1
		[illegible] [illegible] [illegible]
8	SURFFLAG	1 - LANDJUNK 0 - ATMAG test - CHANGEVB YMKRUPT RRDESSM ATTACHIT NBDONLY WLINIT V67CALL P20LEMA STATINT1 LSR22.3 R21LEM RANGEBQ R22LEM R24LEM3 SERVICER LUNDESCH LPS20.1 P20LEMB7 PROG20 AVESTAR VACSTOR SURFAGAN MARKRUPT UPPSV R22LEM42 INTEGRV LASTBIAS REV83 P21VSAVE UPPSV4 V83CALL R21LEM1 ATTACHED
7	INFINFLG	1 - INFINITY 0 - POLYCOEF DOFSTRT1 test - COMMNOUT LAMBLOOP
6	ORDERSW	1 - none 0 - DOFSTRT1 test - ITERATOR
5	APSESW	1 - TIMERAD 0 - TIMERAD DOFSTRT1 test - none (telemetry)

FLAGWRD8 (Continued)

Bit	Mnemonic	Routines
4	COGAFLAG	1 - TIMERAD TIMETHET 0 - COMMNOUT PARAM DOFSTRT1 test - none (telemetry)
2	INITALGN	1 - BYLMATT 0 - DOFSTRT1 ATTCHK test - SURFDISP INITBY SURFLINE
1	360SW	1 - GETX WLOOP 0 - GETX DOFSTRT1 test - POLYCOEF

FLAGWRD9

Bit	Mnemonic	Routines
14	FLVR	1 - P12LM INJTARG 0 - CMPONENT DOFSTRT1 test - CMPONENT
13	P7071FLG	1 - GOABORT 0 - DOFSTRT1 test - ASCENT CMPONENT
12	FLPC	1 - MAINENG 0 - DOFSTRT1 test - MAINENG
11	FLPI	1 - P12LM 0 - P12RET DOFSTRT1 test - CMPONENT
10	FLRCS	1 - CUTOFF 0 - GOABORT DOFSTRT1 test - ASCTERM1 ASCTERM ASCENT ATMAG
9	LETABORT	1 - P63IGN 0 - LANDJUNK TERMASC GOABORT DOFSTRT1 test - P70 P71 R10,R11
8	FLAP	1 - UPTHROT 0 - DOFSTRT1 test - GOABORT P12INIT
7	ABTTGFLG	1 - INJTARG 0 - CANV37 DOFSTRT1 test - for telemetry only
6	ROTFLAG	1 - INJTARG 0 - DOFSTRT1 CMPONENT test - CMPONENT
5	QUITFLAG	1 - VERB96 0 - STATINT1 DOFSTRT1 test - STATINT1 TESTLOOP
3	MID1FLAG	1 - MIDTOAV1 0 - MIDTOAV1 MIDTOAV2 CKMID2 DOFSTRT1 test - CKMID2
2	MIDAVFLG	1 - MIDTOAV2 0 - MIDTOAV2 DOFSTRT1 test - ENDSTATE
1	AVEMIDSW	1 - AVETOMID 0 - INTEXIT DOFSTRT1 test - SVDWN2

FLGWRD10

Bit	Mnemonic	Routines
14	INTFLAG	1 - INTSTALL 0 - INTWAKE1 GOPROG GOPROG2A DOFSTRT1 test - INTSTALL
13	APSFLAG	1 - LANDJUNK DPDAT1 ABRTJASK WANTAPS 0 - DPDAT1 test - P40LM P42LM S40.13 RCS 1/ACCS DAPDATA1 DAPDATA2 SERVICER P40AUTO DVMON AFTERTJ S40.13D
7	REINTFLG	1 - ENDSTATE A-PCHK P76 UPJOB INCORP2 0 - GOPROG2A INTWAKE1 DOFSTRT1 POODOO test - INTSTALL INTWAKE

FLGWRD11

Bit	Mnemonic	Routines
15	LRBYPASS	1 - SERVIDLE ABRTJASK CANV37 DOFSTRT1 0 - P63LM test - MUNRETRN R10,R11 RDRUSECK RADAREAD RESAMPLE STARTSB1
12	VXINH	1 - VMEASCHK 0 - ABRTJASK DOFSTRT1 VMEASCHK CANV37 SERVIDLE test - VMEASCHK
11	PSTHIGAT	1 - MUNRETRN 0 - ABRTJASK DOFSTRT1 CANV37 SERVIDLE test - MUNRETRN UPDATCHK
10	NOLRREAD	1 - MUNRETRN 0 - ABRTJASK POSGOOD DOFSTRT1 CANV37 SERVIDLE P1CHK test - UPDATCHK R10,R11 MUNRETRN
9	XORFLG	1 - MUNRETRN 0 - ABRTJASK DOFSTRT1 CANV37 SERVIDLE test - MUNRETRN
8	LRINH	1 - SET57 0 - NOREASON ABRTJASK LROFF DOFSTRT1 CANV37 SERVIDLE RESET57 test - NOREASON VMEASCHK
7	VELDATA	1 - LRVJOB 0 - ABRTJASK CONTSERV DOFSTRT1 CANV37 SERVIDLE test - VMEASCHK
6	LPOS2FLG	1 - POSGOOD 0 - ABRTJASK SERVIDLE DOFSTRT1 CANV37 test - for telemetry only
5	READVEL	1 - VALTCHK 0 - ABERJASK DOFSTRT1 CANV37 SERVIDLE test - VALTCHK
4	RNGEDATA	1 - LRHJOB 0 - CONTSERV ABRTJASK DOFSTRT1 CANV37 SERVIDLE test - UPDATCHK
3	NO511FLG	1 - P1CHK 0 - ABRTJASK SERVIDLE DOFSTRT1 CANV37 test - MUNRETRN

Bit	Mnemonic	Routines
2	VFLSHFLG	1 - VMEASCHK 0 - ABRTJASK VMEASCHK ONLITES DOFSTRT1 CANV37 SERVIDLE test - R10,R11 RADLITES
1	HFLSHFLG	1 - UPDATCHK 0 - ABRTJASK UPDATCHK ONLITES DOFSTRT1 CANV37 SERVIDLE test - R10,R11 RADLITES

RADMODES

Bit	Mnemonic	Routines
15	CDESFLAG	1 - VBCOARK LRS24.1 R21LEM8 0 - VBCOARK TRMTRACK ITURNON2 STARTSUB POOH RRAUTCHK P12LM R24END R24LEM3 RRDESEND RRDESNB RESET22 PROG20A R21LEM9 test - RRDESEND STDESIG DODES
14	REMODFLG	1 - RRDESSM RRDESNB R29 0 - ITURNON2 STARTSUB STARTSB2 RRAUTCHK REMODE test - RRGIMON RR1AX2 BEGDES LRS24.1 COPYCYC1
13	RCDUOFLG	1 - RRAUTCHK RRZERO 0 - ITURNON2 STARTSUB STARTSB2 RRZEROSB RRAUTCHK test - RRGIMON SETTRKF P20LEMB3 R22LEM COPYCYC1 RRCDUCHK
12	ANTENFLG	1 - RRZEROSB RMODINV 0 - ITURNON2 STARTSUB RRZEROSB RMODINV test - DORREPOS RRLIMCHK RRANGLES RMODINV REMODE DODES RRLIMNB R29 R21LEM
11	REPOSMON	1 - RRGIMON R29 0 - ITURNON2 STARTSUB STARTSB2 RRAUTCHK DORREPOS RR1AX2 PREPOS29 test - RRGIMON RR1AX2 STARTDES STDESIG DORROUT RRZERO RENDRAD COPYCYC1
10	DESIGFLG	1 - STARTDES R29 0 - VBCOARK RRDESEND TRMTRACK ITURNON2 STARTSUB POOH STDESIG STARTSB2 RRDESDUN R24END R24LEM3 RRDESNB RESET22 R29DPAS2 COPYCYC1 R29 P12LM PROG20A R21LEM9 test - DORREPOS RR1AX2 STDESIG R29 BEGDES29 REMODE
9	ALTSCALE	1 - SCALCHNG R04Z 0 - ITURNON2 STARTSUB SCALCHNG R04Z test - LRHEIGHT UPDATCHK
8	LRVELFLG	1 - RESAMPLE R77CHECK 0 - ERROR ITURNON2 STARTSUB GOODRAD TSTLTS3 R77CHECK test - RADLITES
7	RCDUFAIL	1 - ERROR ITURNON2 STARTSUB RRCDUCHK TSTLTS3 0 - RRCDUCHK test - RRCDUCHK SETTRKF ENDRADAR RENDRAD
6	LRPOSFLG	1 - STARTSUB LRPOS2 R04Z 0 - ITURNON2 STARTSUB R04Z test - DORSAMP

Bit	Mnemonic	Routines
5	LRALTFLG	1 - RESAMPLE R77CHECK 0 - ERROR ITURNON2 STARTSUB GOODRAD TSTLTS3 R77CHECK test - RADLITES
4	RRDATAFL	1 - RESAMPLE 0 - ERROR ITURNON2 STARTSUB GOODRAD TSTLTS3 test - SETTRKF
3	RRRSFLAG	1 - SCALCHNG LRS22.1 RO4Z 0 - ITURNON2 STARTSUB SCALCHNG LRS22.1 RO4Z test - RENDRAD RRANGOUT
2	AUTOMODE	1 - ITURNON2 STARTSUB RRAUTCHK 0 - RRAUTCHK test - RRAUTCHK RRCDUCHK RRGIMON SETTRKF RR1AX2 RRZERO COPYCYC1 R29RDJOB NORRGMON
1	TURNONFL	1 - RRAUTCHK 0 - ITURNON2 STARTSUB STARTSB2 RRTURNON RRAUTCHK test - RRZERO

DAPBOOLS

Bit	Mnemonic	Routines
15	PULSES	1 - MINIMP LANDJUNK 0 - NOMINIMP DOFSTRT1 IGNITION ABRTJASK TIGTASK test - TSNEXTP TSNEXTS
14	USEQRJTS	1 - ENGINOF3 AVERAGEG DVMON DOFSTRT1 0 - DVMON test - TJLAW4 TRYGTS SPSCONT
13	CSMDOCKD	1 - DPDAT1 0 - DPDAT1 DOFSTRT1 test - PURGENCY TJLAW4 1/ACCS DAPDATA2 DAPDATA1 DPDAT1 BACKP STIKLOAD FINDCDUW P40LM MINRTN
12	OURRCBIT	1 - DETENTCK 0 - DETENTCK DOFSTRT1 test - DETENTCK CHEKSTIK QRAXIS
11	ACC4OR2X	1 - DPDAT1 GOABORT P12LM 0 - DPDAT1 DOFSTRT1 test - DPDAT1 DAPDATA1 +XORULGE
10	AORBTRAN	1 - COMFAIL2 DPDAT1 DOFSTRT1 0 - COMFAIL2 DPDAT1 test - MINRTN DAPDATA1 +XORULGE
9	XOVINHIB	1 - CMPONENT MUNRETRN 0 - DOFSTRT1 P65START CMPONENT ABRTJASK CANV37 GOTOPOOH STRTP66A test - ~~TSNEXTP~~ FINDCDUW
8	DRIFTBIT	1 - ALLCOAST COMFAIL2 DOFSTRT1 0 - P42IGN ABRTJASK test - 1/ACCONT SPSRCS RCS BACKP AFTERTJ
7	RHCSCALE	1 - DPDAT1 DOFSTRT1 0 - DPDAT1 test - DAPDATA1 STIKLOAD
6	ULLAGER	1 - ULLGTASK COMFAIL2 0 - P42IGN ENGINOF1 GOPOST GOTOPOOH GOCUTOFF ABRTJASK DOFSTRT1 STOPCLOK test - RCS

Bit	Mnemonic	Routines
5	DBSL2FLG	1 - DPDAT1 0 - DPDAT1 test - RESTORDB DAPDATA1
4	DBSLECT2	1 - DPDAT1 DOFSTRT1 0 - DPDAT1 test - RESTORDB DAPDATA1
3	ACCSOKAY	1 - 1/ACCRET 0 - STARTSB1 DOFSTRT1 test - DAPIDLER
2	AUTRATE2	1 - DPDAT1 DOFSTRT1 0 - DPDAT1 test - DAPDATA1
1	AUTRATE1	1 - DPDAT1 0 - DPDAT1 DOFSTRT1 test - DAPDATA1

IMODES30

Bit	Routines	
15	1 - IMUMON 0 - IMUMON GOPROG TSTLTS3 DOFSTRT1 test - IMUMON	(IMU temp. out of limits)
14	1 - IMUMON GOPROG DOFSTRT1 0 - IMUMON test - IMUMON TNONTEST ENDTNON	(ISS turn-on delay initiate)
13	1 - IMUMON IFAILOK GOPROG TSTLTS3 DOFSTRT1 0 - IMUMON test - SETISSW	(IMU good)
12	1 - IMUMON GOPROG TSTLTS3 DOFSTRT1 0 - IMUMON test - SETISSW	(ICDU good)
11	1 - IMUMON GOPROG DOFSTRT1 0 - IMUMON test - IMUMON	(IMU not caged)
10	1 - ERROR C33TEST PFAILOK GOPROG TSTLTS3 DOFSTRT1 0 - C33TEST test - PIPFREE C33TEST SETISSW	(PIPA good)
9	1 - IMUMON DOFSTRT1 0 - IMUMON test - IMUCHK R02BOTH IMUMON TNONTEST C33TEST IMUZERO	(IMU not operating)
8	1 - TNONTEST 0 - TNONTEST GOPROG DOFSTRT1 test - TNONTEST C33TEST SETGLOCK	(turn-on delay incomplete)
7	1 - ITURNON2 0 - TNONTEST GOPROG DOFSTRT1 test TNONTEST C33TEST	(turn-on delay just started)
6	1 - CAGESUB2 0 - UNZ2 GOPROG DOFSTRT1 test - PIPUSE PFAILOK IMUPULSE STRTGYRO 8192AUG IMUZERO IMUZERO2 IMUCOARS COARS COARS2 IMUFINE IFAILOK V37 IMUFINED	(IMU caged)
5	1 - CAGESUB2 0 - PFAILOK DOFSTRT1 test - C33TEST	(Secondary PIPA fail monitor disabled)

IMODES30 (Continued)

Bit	Routines	
4	1 - CAGESUB2 IMUZERO SETCOARS DOFSTRT1 0 - UNZ2 IMUZERO2 IFAILOK test - SETISSW	(IMU fail monitor inhibit)
3	1 - CAGESUB2 IMUZERO 0 - UNZ2 IMUZERO2 DOFSTRT1 test - SETISSW	
2	1 - IMUMON 0 - ENDTNON GOPROG DOFSTRT1 test - IMUMON ENDTNON	(turn-on sequence failure)
1	1 - PIPFREE CAGESUB2 DOFSTRT1 0 - PIPUSE test - C33TEST SETISSW	(PIPA fail monitor disable)

<u>IMODES33</u>

Bit Routines

14 1 - PROCEEDE (proceed button)
0 - PROCEEDE STARTSB2 DOFSTRT1
test - PROCEEDE

13 1 - ERROR C33TEST PFAILOK STARTSB2 TSTLTS3 DOFSTRT1 (PIPA good)
0 - C33TEST
test - C33TEST

12 1 - ERROR C33TEST STARTSB2 TSTLTS3 DOFSTRT1 (downlink not too fast)
0 - C33TEST
test - DNTMFAST C33TEST

11 1 - ERROR C33TEST STARTSB2 TSTLTS3 DOFSTRT1 (uplink not too fast)
0 - C33TEST
test - UPTMFAST C33TEST

8 1 - INTLZE
0 - DISPRSET STARTSB2 DOFSTRT1
test - DISPRSET

7 1 - ALTROUT1
0 - ALTOUT1 DISPRSET STARTSB2 LANDISP DOFSTRT1
test - LANDISP

6 1 - IMUMON CAGESUB2 IMUZERO SETCOARS DOFSTRT1 (DAP disable)
0 - UNZ2 IMUZERO2 IMUFINE
test - CHEKBITS

5 1 - IMUZERO
0 - IMUZERO2 STARTSB2 DOFSTRT1 (zeroing in progress)
test - none (telemetry)

1 1 - VBTSTLTS
0 - TSTLTS3 STARTSB2 DOFSTRT1 (lamp test)
test - IMUMON SETISSW SETGLOCK SETTRKF

RCSFLAGS

Bit	Routines
15, 14	Not used
13	1 - DAPIDLER 0 - STARTSB1 DOFSTRT1 test - DAPIDLER
12	1 - STARTDAP SKIPPAXS 0 - PJETSLEC DOFSTRT1 test - SUPERJOB
11	1 - QRTIME 0 - STARTDAP DETENTCK TOPSEUDO DOFSTRT1 test - DETENTCK RHCACTIV
10	1 - PEGI 0 - STARTDAP DETENTCK PEGI DOFSTRT1 test - DETENTCK RATERROR
9	1 - DETENTCK 0 - DETENTCK CHEKSTIK DOFSTRT1 test - DETENTCK
8 7 6	Not used
5	1 - NEGUSUM 0 - STARTDAP ACDT+C12 DOFSTRT1 test - PAXFILT
4	1 - ALTDSPLY 0 - ALTDSPLY DOFSTRT1 test - ALTDSPLY
3	1 - NEEDLER DOFSTRT1 CHEKBITS 0 - NEEDLER test - NEEDLER
2	1 - NEEDLER 0 - NEEDLER DOFSTRT1 test - NEEDLER
1	1 - TRYUORV 0 - STARTDAP TRYUORV DOFSTRT1 test - TRYUORV

Bit	Routines	
9	1 - VBTSTLTS PROGLARM	(program check fail lamp)
	0 - ERROR TSTLTS3 SLAP1	
	test - none	
8	1 - SETTRKF VBTSTLTS	(tracker fail lamp)
	0 - ERROR GOPROG SETTRKF TSTLTS3 SLAP1	
	test - SETTRKF	
6	1 - SETGLOCK VBTSTLTS	(gimbal lock warning lamp)
	0 - SETGLOCK TSTLTS3	
	test - SETGLOCK IMUZERO DOFSTRT1	
5	1 - VBTSTLTS R10,R11 LITIT	(LR altitude fail)
	0 - R10,R11 ERROR SLAP1 TSTLTS3 GOPROG LITIT	
	test - LITIT	
4	1 - CAGESUB1 SETCOARS VBTSTLTS TSTLTS3	(no attitude lamp)
	0 - ISSZERO ENDTNON IMUZERO IMUFINE TSTLTS3	
	test - IMUZERO GOPROG DOFSTRT1	
3	1 - VBTSTLTS R10,R11 LITIT	(LR velocity fail)
	0 - R10,R11 ERROR SLAP1 TSTLTS3 GOPROG LITIT	
	test - LITIT	

ALIN

Alignment of the Inertial Sub-System

P51 Perform "IMUCHK" (assure that IMU is on)

Proceed to "GOPERF1" with TS = 00015_8
(Request that celestial body acquisition be performed)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "P51B"; if other response, continue at next step.)

THETAD = 0

Perform "GODSPRET" with TS = K:V06N22
(Display THETAD and continue at next step when display is up)

Perform "GODSPRET" with TS = K:V41N00 (Indicate coarse align in progress; continue at next step when display is up)

Perform "COARSE"

Proceed to second step of "P51"

P51B 1dPIPADT = less significant half of TIMENOW

PIPA = -0

GCOMP = 0

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

STARIND = 0

P51C Perform "AOTMARK"
(return after marks are averaged)
Perform "AOTSTALL"

If AOTGOOD = 0, perform "CURTAINS"

If STARIND = 0:

STARSAV1 = $STARAD_6$

TSt = TSIGHT

Perform "PLANET"

PLANVEC = TS

STARIND = 1

Proceed to "P51C"

TSt = TSIGHT

Perform "PLANET"

$\underline{TS}_{12} = \underline{TS}$

$\underline{TS}_6 = \underline{P}LANVEC$

$\underline{S}TARAD_0 = \underline{S}TARSAV1$

$\underline{S}TARAD_6 = \underline{S}TARSAV2$

Perform "CHKSDATA"

If FLAGWRD0 bit 3 (FREEFLAG) = 0: (error between actual separation and measured separation is not acceptable)

Proceed to second step of "P51"

Perform "AXISGEN"

[REFSMMAT] = [DCMAT]

Switch FLAGWRD3 bit 13 (REFSMFLG) to 1

Proceed to "GOTOPOOH"

COARSE If IMUCADR ≠ 0: (IMU in use)

Delay 1 second

Proceed to "COARSE"

Perform "IMUCOARS"

Perform "IMUSTALL"

If ISSGOOD = 0, perform "CURTAINS"

Perform "IMUFINE"

Perform "IMUSTALL"

If ISSGOOD = 0, perform "CURTAINS"

Return

AOTMARK Inhibit interrupts

If MARKSTAT ≠ 00000_8: (mark system already busy)

Proceed to "POODOO" with TS = 20105_8

If bits 2 and 3 of EXTVBACT are not both 0:

$TS1_{dp}$ = return address of routine calling "AOTMARK"

Proceed to "BAILOUT1" with TS = 31211_8

Switch bit 2 of EXTVBACT to 1

Establish a special additional working storage area for the mark system; store its address in bits 1-9 of MARKSTAT. It will be denoted hereafter as MARKVAC.

If no storage area is available for MARKVAC:

$TS1_{dp}$ = return address of routine calling "AOTMARK"

Proceed to "BAILOUT1" with TS = 31207_8

Establish "GETDAT" (pr15)

Release interrupt inhibit

Return

AOTSTALL Inhibit interrupts

If OPTCADR > 0 or if OPTCADR < -1:

$TS1_{dp}$ = return address of routine calling "AOTSTALL"

Proceed to "BAILOUT1" with TS = 31210_8

If OPTCADR = -1: (operation already complete and good)

OPTCADR = +0

AOTGOOD = 1

Release interrupt inhibit

Return

If OPTCADR = -0: (operation already complete and bad)

(If OPTCADR = -0:)

OPTCADR = +0

AOTGOOD = 0

Release interrupt inhibit

Return

(Otherwise, OPTCADR = +0)

OPTCADR = return address (to caller of "AOTSTALL")

Put present job to sleep

When awakened, return via LOC

MKRELEAS MARKSTAT = 00000_8

Release special working storage area MARKVAC

If OPTCADR = +0: ("AOTSTALL" not entered yet)

OPTCADR = -1

End task

LOC = OPTCADR

AOTGOOD = 1

Wake job put to sleep in "AOTSTALL"

OPTCADR = +0

End task

GETDAT Switch bit 12 of MARKSTAT to 1 (to inhibit processing of marks in "MARKRUPT")

Proceed to "GOXDSPF" with TS = K:V01N71 (AOTCODE)
(If terminate, proceed to "KILLAOT"; if proceed, continue at next step; if other response, proceed to "GETDAT")

(AOTCODE should be of the form 000 000 xxx xxx xxx_2)

XYMARK = bits 15-7 of AOTCODE shifted right 6 places to bit positions 9-1

If XYMARK $\leq$ 0, proceed to "GETDAT"

If XYMARK = 00007_8: (detent code 7 for COAS)

Proceed to "GOXDSPF" with TS = K:V06N87 (AZ, EL) (If terminate, proceed to "KILLAOT"; if proceed, continue at next step; if other response, repeat this step)

TSazm = AZ

TSelev = EL

TSsrot = 0

Proceed to "OPTAXIS"

(Otherwise, XYMARK is between 1 and 6 inclusive)

TSelev = $AOTEL_{XYMARK}$

TSazm = $AOTAZ_{XYMARK}$

TSsrot = $AOTAZ_2$ - TSazm

OPTAXIS Perform "OANB"

$\underline{U}YP$ = cosTSsrot $\underline{U}YP'$ - sinTSsrot $\underline{U}XP'$

$\underline{U}XP$ = cosTSsrot $\underline{U}XP'$ + sinTSsrot $\underline{U}YP'$

$\underline{S}TARAD_6$ = 0

Proceed to "GETMKS"

OANB TSelev = TSelev converted to one's complement form

TSazm = TSazm converted to one's complement form

$$\underline{S}CAXIS = \begin{pmatrix} \text{sinTSelev} \\ \text{cosTSelev sinTSazm} \\ \text{cosTSelev cosTSazm} \end{pmatrix}$$

$\underline{U}YP'$ = unit($\underline{S}CAXIS$ * $\underline{K}$:UNITX) (= (0, cosTSazm, -sinTSazm))

$\underline{U}XP'$ = unit($\underline{U}YP'$ * $\underline{S}CAXIS$)

Return

GETMKS XYMARK = 00000_8

MARKCNTR = 0

Switch bits 15 thru 10 of MARKSTAT to 0

TS = K:V54N71 ("mark X or Y" verb; star code noun)

PASTIT Proceed to "GOMARK4"
(If terminate, proceed to "KILIAOT"; if proceed, proceed to "MARKCHEX"; if other response, proceed to "GETDAT".)

MARKRUPT (Entered on program interrupt initiated by the mark or mark reject buttons or by a commanded change in descent rate.)

TScdu = CDU

TSt = TIMENOW

If bit 6 or 7 of channel 16 = 1:
(Commanded change in rate of descent)

Proceed to "SOMEKEY"

If bit 12 of MARKSTAT = 1, Resume
(Processing of marks inhibited)

If MARKSTAT = 00000_8: (mark program not operating)

Perform "ALARM" with TS = 00112_8

Resume

If bit 5 of channel 16 = 1: (mark reject)

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

If MARKCNTR > 0:

MARKCNTR = MARKCNTR - 1

Resume

Perform "ALARM" with TS = 00115_8

Resume

(If bit 5 of channel 16 = 1:)

If bits 10 and 11 of MARKSTAT both = 0: (no marks to reject)

Perform "ALARM" with TS = 00115_8

Resume

Proceed to "REJECT"

If bit 4 of channel 16 = 1, proceed to "YMKRUPT"

If bit 3 of channel 16 = 1, proceed to "XMKRUPT"

<u>SOMEKEY</u> If bit 6 or bit 7 of channel 16 = 1, proceed to "DESCBITS" (Commanded change in descent rate)

Perform "ALARM" with TS = 00113_8

Resume

<u>XMKRUPT</u> i = 0

XYMARK = 01000_8 (bit 10 = 1)

Skip next two steps

<u>YMKRUPT</u> i = 1

XYMARK = 02000_8 (bit 11 = 1)

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "SURFSTOR"

If bit 14 of MARKSTAT = 1: (mark pair just completed)

If MARKCNTR $\geq$ 4:

Perform "ALARM" with TS = 00107_8

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Proceed to "DSPV6N79"

Resume

MARKCNTR = MARKCNTR + 1

Switch bits 14, 11 and 10 of MARKSTAT to 0

(End of indented steps)

If bit which is 1 in XYMARK is also 1 in MARKSTAT:
(wrong mark)

Perform "ALARM" with TS = 00114_8

Resume

Proceed to "VACSTOR"

SURFSTOR i = 0

Switch bits 10 and 11 of MARKSTAT to 1
(Show surface mark for "MARKCHEX")

VACSTOR Get address of MARKVAC from low 9 bits of MARKSTAT

TSIGHT = TSt

i = i + 6 MARKCNTR (x: 0,6,12,18,24)
(y: 1,7,13,19,25)

MKDEX = i (store in case of surface mark)

$MARKVAC_i = TScdu_y$ (inner gimbal angle)

i = i + 2 (x: 2,8,14,20,26)
(y: 3,9,15,21,27)

$MARKVAC_i = TScdu_z$ (middle gimbal angle)

i = i + 2 (x: 4,10,16,22,28)
(y: 5,11,17,23,29)

$MARKVAC_i = TScdu_x$ (outer gimbal angle)

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Proceed to "REMARK" skipping first step

Switch bit 13 of MARKSTAT to 0 (enable selective mark reject)

MARKSTAT = MARKSTAT + XYMARK
(switching bit 10 or 11 of MARKSTAT to 1)

If bits 10 and 11 of MARKSTAT both = 1:

Switch bit 14 of MARKSTAT to 1 (indicate both marks taken)

Proceed to "REMARK"

REJECT If bit 13 of MARKSTAT = 0: (reject only latest mark)

If bit 10 of XYMARK = 1:

Switch bit 10 of MARKSTAT to 0

If bit 11 of XYMARK = 1:

Switch bit 11 of MARKSTAT to 0

If bit 13 of MARKSTAT = 1: (no mark since last reject)

Switch bits 10 and 11 of MARKSTAT to 0

Switch bit 14 of MARKSTAT to 0 (reject pair)

Switch bit 13 of MARKSTAT to 1 (see above)

REMARK MKDEX = bits 11 and 10 of MARKSTAT shifted right 9 places to bit positions 2 and 1 (MKDEX = 0,1,2 or 3)

Establish "CHANGEVB" (pr15)

Resume

CHANGEVB If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "DSPV6N79"

TS = K:V54N71 (request X or Y mark)

If MKDEX = 1, TS = K:V53N71 (request Y mark)

If MKDEX = 2, TS = K:V52N71 (request X mark)

Proceed to "PASTIT"

MARKCHEX Switch bit 12 of MARKSTAT to 1 (inhibit processing in "MARKRUPT")

Store address of MARKVAC in low 9 bits of XYMARK

MKDEX = 0

If bits 10 and 11 of MARKSTAT are not both 1: (last pair incomplete)

If MARKCNTR = 0:

(If MARKCNTR = 0:)

Perform "ALARM" with TS = 00111_8

Proceed to "GETMKS"

MARKCNTR = MARKCNTR - 1

<u>AVESTAR</u> MKDEX = MKDEX + 1

i = 6 MARKCNTR

Get address of MARKVAC from XYMARK

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "SURFSTAR"

$\underline{ANG} = (MARKVAC_{i+4}, MARKVAC_i, MARKVAC_{i+2})$

Perform "CD*TR*GS"

Perform "NBTOSM"

$\underline{TS} = [NBSMMAT]\ \underline{UXP}$

i = i + 1

$\underline{ANG} = (MARKVAC_{i+4}, MARKVAC_i, MARKVAC_{i+2})$

Perform "CD*TR*GS"

Perform "NBTOSM"

$\underline{TS} = ([NBSMMAT]\ \underline{UYP}) * \underline{TS}$

$\underline{TS}star = -unit\underline{TS}$

<u>AVEIT</u> n = MKDEX

$\underline{STARAD}_6 = \frac{n-1}{n} \underline{STARAD}_6 + \frac{1}{n} \underline{TS}star$

$\underline{STARSAV2} = \underline{STARAD}_6$

If MARKCNTR > 0:

MARKCNTR = MARKCNTR - 1

Proceed to "AVESTAR"

Call "MKRELEAS" in 0.05 seconds

Proceed to "ENDEXT"

DSPV6N79 Proceed to "GOXDSPF" with TS = V06N79 (CURSOR,SPIRAL,POSCODE)
(If terminate, proceed to "KILLAOT"; if proceed, proceed to "SURFEND"; if other response, continue at next step.)

If bit 6 of TSverb = 1, proceed to "SURFAGAN" (V32E)

Proceed to "DSPV6N79"

SURFEND Switch bit 14 of MARKSTAT to 1 (show mark end)

SURFAGAN Save VAC area for surface marking

$MARKVAC_{MKDEX+1}$ = CURSOR

$MARKVAC_{MKDEX+3}$ = SPIRAL

If bit 14 of MARKSTAT = 1, proceed to "MARKCHEX"

If MARKCNTR $\geq$ 4:

Perform "ALARM" with TS = 00107_8

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "DSPV6N79"

Resume

MARKCNTR = MARKCNTR + 1

Proceed to "GETMKS" skipping first two steps

SURFSTAR $\underline{ANG}$ = ($MARKVAC_{i+4}$,$MARKVAC_i$,$MARKVAC_{i+2}$) (Save CDU's at time of X-mark for use in "CD*TR*GS")

TSyrot = $MARKVAC_{i+1}$ (CURSOR angle converted to one's comp. form)

TSsrot = $MARKVAC_{i+3}$

If TSyrot = 0:

If TSsrot = 0:

$\underline{TS}$ = unit($\underline{S}$CAXIS)

Proceed to "JUSTOA"

$\underline{TS}$ = unit($\underline{U}$YP cosTSyrot - $\underline{U}$XP sinTSyrot)

$\underline{TS}$ = unit($\underline{TS}$ * $\underline{S}$CAXIS)

$TSsrot = MARKVAC_{i+3}$ (SPIRAL angle converted to one's comp. form)

TSsep = 1/12 (TSsrot - TSyrot + K:ABOUT1)

$\underline{TS}$ = unit(cosTSsep $\underline{S}$CAXIS + sinTSsep $\underline{TS}$)

JUSTOA Perform "CD*TR*GS"

Perform "NBTOSM"

$\underline{TS}$star = [NBSMMAT] $\underline{TS}$

Proceed to "AVEIT"

PLANET TSIGHT = TSt

i = low 6 bits of AOTCODE (0 to 40)

If STARIND = 0, BESTI = 6 i

If STARIND = 1, BESTJ = 6 i

If i = 0: (planet)

Proceed to "GOFLASH" with TS = K:V06N88 ($\underline{S}TARAD_0$)
(If terminate, repeat this step; if proceed, continue at next step; if other response, repeat this step.)

$\underline{TS}$ = unit$\underline{S}TARAD_0$

Return

If i < 38: (star)

$\underline{TS}$ = $\underline{K}$:$CATLOG_i$

Return

Perform "LOCSAM"

If i = 38, $\underline{TS}$ = $\underline{V}$SUN

If i = 39, $\underline{TS}$ = $\underline{V}$EARTH

If i = 40, $\underline{TS}$ = $\underline{V}$MOON

Return

LOCSAM QMIN = return address

TSIGHT = TSt

Perform "LSPOS"

TDEC1 = TSIGHT

Perform "LEMPREC"

If PBODY = 0: (earth centered)

$\underline{V}$MOON = unit(K:RSUBEM $\underline{V}$MOON - $\underline{R}$ATT)

$\underline{V}$EARTH = -unit$\underline{R}$ATT

CEARTH = cos(arcsin(K:RSUBE / |$\underline{R}$ATT|) + K:5DEGREES)

CMOON = K:CSS5

If PBODY = 2: (moon centered)

$\underline{V}$SUN = unit($\underline{V}$SUN - K:ROE $\underline{V}$MOON)

$\underline{V}$EARTH = - unit(K:RSUBEM $\underline{V}$MOON + $\underline{R}$ATT)

$\underline{V}$MOON = - unit$\underline{R}$ATT

CMOON = cos(arcsin(K:RSUBM / |$\underline{R}$ATT|) + K:5DEGREES)

CEARTH = K:CSS5

CSUN = K:CSSUN

Return via QMIN

CHKSDATA Switch FLAGWRDO bit 3 (FREEFLAG) to 1 (R54)

TSang = arccos($\underline{S}TARAD_0 \cdot \underline{S}TARAD_6$)

Switch FLAGWRDO bit 3 (FREEFLAG) to 0

THETA = arccos($\underline{T}S_6 \cdot \underline{T}S_{12}$) - TSang

$DSPTEM1_{dp}$ = THETA

Switch FLAGWRDO bit 3 (FREEFLAG) to 1

Proceed to "GOFLASH" with TS = K:V06N05 (DSPTEM1)
(If terminate, proceed to "GOTOPOOH"; if proceed, skip next step; if other response, continue at next step.)

Switch FLAGWRD0 bit 3 (FREEFLAG) to 0

Return

AXISGEN $\underline{TS1} = \underline{TS}_6$

$\underline{TS2} = \text{unit}(\underline{TS}_6 * \underline{TS}_{12})$

$\underline{TS3} = \underline{TS1} * \underline{TS2}$

$$[RFSTMAT] = \begin{bmatrix} TS1_x & TS1_y & TS1_z \\ TS2_x & TS2_y & TS2_z \\ TS3_x & TS3_y & TS3_z \end{bmatrix}$$

$\underline{TS1} = \underline{STARAD}_0$

$\underline{TS2} = \text{unit}(\underline{STARAD}_0 * \underline{STARAD}_6)$

$\underline{TS3} = \underline{TS1} * \underline{TS2}$

$$[SMSTMAT] = \begin{bmatrix} TS1_x & TS1_y & TS1_z \\ TS2_x & TS2_y & TS2_z \\ TS3_x & TS3_y & TS3_z \end{bmatrix}$$

$[DCMAT] = [SMSTMAT]^T [RFSTMAT]$

Unitize each of the three rows of [DCMAT]
(assure that it is orthogonal)

$\underline{STARAD}_0 = [DCMAT]^T \underline{K}:UNITX$

$\underline{STARAD}_6 = [DCMAT]^T \underline{K}:UNITY$

$\underline{STARAD}_{12} = [DCMAT]^T \underline{K}:UNITZ$

Return

PROG52 Perform "RO2BOTH"

If FLAGWRD2 bit 4 (PFRATFLG) = 1:

OPTION2 = 1 and skip next step

OPTION2 = 3 (REFSMMAT option)

P52B Perform "GOPERF4R" with OPTION1 = 1
(If terminate, proceed to "GOTOPOOH"; if proceed, skip next step; if other response, repeat this step.)

End job

If OPTION2 bit 2 = 0 and bit 1 = 1: (OPTION2 = 1,5,9,.....)

Proceed to "P52D" (Preferred)

If OPTION2 bits 2 and 1 both = 1: (OPTION2 = 3,7,11,.....)

Perform "R51" (REFSMMAT)

Proceed to "GOTOPOOH"

If OPTION2 bits 2 and 1 = 0: (OPTION2 = 0,4,12,.....)

TS = TLAND and skip next step (Landing site)

(Otherwise, OPTION2 = 2,6,....) (Nominal)

TS = -0

DSPTEM1 = TS

Proceed to "GOFLASH" with TS = K:V06N34 (DSPTEM1)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

If more significant half of DSPTEM1 $\leq$ 0:

TALIGN = TIMENOW and skip next step

TALIGN = DSPTEM1

If OPTION2 bit 2 = 1: (OPTION2 = 2,6,10,.....)

TSt = TALIGN

Perform "S52.3"

Proceed to "P52D"

P52LS Switch FLAGWRD1 bit 13 (ERADFLAG) to 0 (OPTION2 = 0,4,12,....)

Switch FLAGWRD3 bit 12 (LUNAFLAG)

$\underline{TS}$ = $\underline{RLS}$ (landing site vector in MF coordinates)

TSt = TALIGN

TLAND = TALIGN

Perform "MOONMX"

$\underline{ALPHAV} = \left[MOONMAT \right]^T (\underline{TS} + \underline{LM504} * \underline{TS})$

TSt = TALIGN

Perform "N89DISP"

$\underline{X}$SMDrf = unit$\underline{A}$LPHAV

Perform "LSORIENT" (compute landing site orientation)

Proceed to "P52D"

N89DISP TStime = TSt

Perform "LAT-LONG" (calculate and display landing site)

LANDLONG = LONG / 2

LANDALT = ALT

LANDLAT = LAT

Proceed to "GOFLASH" with TS = K:V06N89 (LANDLAT,LANDLONG,LANDALT)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

LONG = 2 LANDLONG

ALT = LANDALT

LAT = LANDLAT

TSt = TStime

Perform "LALOTORV"

Return

P52D Perform "S52.2" (compute gimbal angles)

Proceed to "GOFLASH" with TS = K:V06N22 ($\underline{THETAD}$)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "P52D".)

Proceed to "GOPERF1" with TS = 00013_8 (perform checklist #13)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "REGCOARS"; if other response, continue at next step.)

$\underline{X}DC$ = unit([REFSMMAT] $\underline{X}SMDrf$) (get desired SM wrt present SM)

$\underline{Y}DC$ = unit([REFSMMAT] $\underline{Y}SMDrf$)

$\underline{Z}DC$ = unit([REFSMMAT] $\underline{Z}SMDrf$)

Perform "GYCOARS"

Proceed to "GOTOPOOH"

GYCOARS QMAJ = return address

Perform "CALCGTA"

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 0

Switch FLAGWRD3 bit 13 (REFSMFLG) to 0

Perform "GODSPR" with TS = K:V16N20 (monitor gimbal angles)

Perform "IMUPULSE" with TS = address of OGC

Perform "IMUSTALL"

If ISSGOOD = 0, perform "CURTAINS" (bad return)

[REFSMMAT] = [XSMDMAT]

Switch FLAGWRD2 bit 4 (PFRATFLG) to 0

Switch FLAGWRD3 bit 13 (REFSMFLG) to 1

1dPIPADT = $TIMENOW_{1s}$

$\underline{P}IPA$ = -0

$\underline{G}COMP$ = 0

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

Proceed to "R51K"

S52.2 QMAJ = return address

Perform "CDUTRIG"

Perform "CALCSMSC"

$\underline{X}NBrf = unit([REFSMMAT]^T \underline{X}NBsm)$

$\underline{Y}NBrf = unit([REFSMMAT]^T \underline{Y}NBsm)$

$\underline{Z}NBrf = unit([REFSMMAT]^T \underline{Z}NBsm)$

$\underline{X}SMrf = \underline{X}SMDrf$

$\underline{Y}SMrf = \underline{Y}SMDrf$

$\underline{Z}SMrf = \underline{Z}SMDrf$

Perform "CALCGA"

Return via QMAJ

S52.3 QMAJ = return address

TDEC1 = TSt

Perform "LEMCONIC"

$\underline{X}SMDrf = unit\underline{R}ATT$

$\underline{Y}SMDrf = unit(\underline{V}ATT * \underline{R}ATT)$

$\underline{Z}SMDrf = unit(\underline{X}SMDrf * \underline{Y}SMDrf)$

Return via QMAJ

LSORIENT QMAJ = return address

$\underline{Z}SMDrf = unit[(\underline{R}RECTCSM * \underline{V}RECTCSM) * \underline{X}SMDrf]$

$\underline{Y}SMDrf = unit(\underline{Z}SMDrf * \underline{X}SMDrf)$

Return via QMAJ

CAL53A Perform "S52.2"

$\underline{T}S = |\underline{C}DU - \underline{T}HETAD|$ $(TS_x = |CDU_x - THETAD_x|$, etc.)

If any of the three components of $\underline{TS}$ is $\geq$ $\underline{K}$:DEGREE1 and < K:DEG359:

Perform "COARSE"

1dPIPADT = $TIMENOW_{1s}$

$\underline{P}$IPA = -0

$\underline{G}$COMP = 0

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

$$[REFSMMAT] = \begin{bmatrix} XSMrf_x & XSMrf_y & XSMrf_z \\ YSMrf_x & YSMrf_y & YSMrf_z \\ ZSMrf_x & ZSMrf_y & ZSMrf_z \end{bmatrix}$$

Return

R51 QMAJ = return address

Proceed to "GOPERF1" with TS = 00015_8 (perform checklist #15)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "R51E".)

TSt = TIMENOW + K:TSIGHT1

Perform "LOCSAM"

Perform "R56" (select best star pair)

If HAVEPAIR = 0:

Perform "ALARM" with TS = 00405_8

Proceed to "GOFLASH" with TS = K:V05N09 (display alarm)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to second step of "R51".)

R51E STARIND = 0 (start with star farther from line-of-sight)

Perform "R52" (set AOTCODE and maneuver LM)

Perform "AOTMARK"

Perform "AOTSTALL"

If AOTGOOD = 0, perform "CURTAINS" (bad return)

If STARIND = 0:

    $\underline{S}$TARSAV1 = $\underline{S}TARAD_6$

    TSt = TSIGHT

    Perform "PLANET"

    $\underline{P}$LANVEC = $\underline{T}$S

    STARIND = 1

    Proceed to second step of "R51E"

$\underline{S}$TARSAV2 = $\underline{S}TARAD_6$

TSt = TSIGHT

Perform "PLANET"

$\underline{S}TARAD_6$ = unit([REFSMMAT] $\underline{T}$S)

$\underline{S}TARAD_0$ = unit([REFSMMAT] $\underline{P}$LANVEC)

$\underline{T}S_6$ = $\underline{S}$TARSAV1

$\underline{T}S_{12}$ = $\underline{S}$TARSAV2

Perform "CHKSDATA" (R54)

If FLAGWRD0 bit 3 (FREEFLAG) = 1:

    Perform "AXISGEN"

    Perform "R55"

    Switch FLAGWRD2 bit 4 (PFRATFLG) to 0

R51K Proceed to "GOPERF1" with TS = 00014_8 (checklist #14)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to second step of "R51"; if other response, continue at next step.)

Return via QMAJ

R52 AOTCODE = 00200_8 + BESTI / 6 (setting detent code to 2)

If STARIND = 1, AOTCODE = 00200_8 + BESTJ / 6

Proceed to "GOFLASH" with TS = K:V01N70 (AOTCODE)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

TS1 = bits 15-7 of AOTCODE shifted right to bit positions 9-1

If TS1 $\leq$ 0, proceed to "GETAZEL" (COAS calibration)

If TS1 = 7, proceed to "GETAZEL" (COAS sighting)

i = TS1 (detent position between 1 and 6 inclusive)

TSazm = $AOTAZ_i$

TSelev = K:r52el

Proceed to "AZEL"

GETAZEL Proceed to "GOFLASH" with TS = K:V06N87 (AZ, EL)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

TSazm = AZ

TSelev = EL

AZEL Perform "OANB" (get SCAXIS)

TSt = TIMENOW

Perform "PLANET"

POINTVSM = unit([REFSMMAT] TS)

Perform "R60LEM" (attitude maneuver)

If bits 15-7 of AOTCODE = 0: (COAS calibration)

Proceed to "R52" skipping first two steps

Return

R55 QMIN = return address

Perform "CALCGTA"

Proceed to "GOFLASH" with TS = K:V06N93 (OGC,IGC,MGC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, return via QMIN.)

TS = address of (OGC, IGC, MGC)

Perform "IMUPULSE"

Perform "IMUSTALL"

If ISSGOOD = 0, perform "CURTAINS" (bad return)

Return via QMIN

R56 QMIN = return address

Perform "CDUTRIG"

Perform "CALCSMSC"

Switch FLAGWRD3 bit 10 (VFLAG) to 1

BESTI = 0

BESTJ = 0

$\underline{SAX} = \text{unit}([\text{REFSMMAT}]^T (\frac{1}{2}\, \underline{XNBsm} + \frac{1}{2}\, \underline{ZNBsm}))$

i = 38

PIC1 i = i - 1

If i = 0, proceed to "PICEND"

If $\underline{K}\text{:CATLOG}_i \cdot \underline{SAX} < \text{K:CSS33}$, proceed to "PIC1"

j = i

PIC3 j = j - 1

If j = 0, proceed to "PIC1"

If $\underline{K}$:$CATLOG_j \cdot \underline{S}AX$ < K:CSS33, proceed to "PIC3"

If $\underline{K}$:$CATLOG_i \cdot \underline{K}$:$CATLOG_j \geq$ K:CSS40, proceed to "PIC3"

$\underline{T}Sstar = \underline{K}$:$CATLOG_i$

Perform "OCCULT"   (see if first star is occulted)

If FLAGWRD3 bit 7 (CULTFLAG) = 1, proceed to "PIC1"

$\underline{T}Sstar = \underline{K}$:$CATLOG_j$

Perform "OCCULT"   (see if second star is occulted)

If FLAGWRD3 bit 7 (CULTFLAG) = 1, proceed to "PIC3"

If FLAGWRD3 bit 10 (VFLAG) = 1:

  Switch FLAGWRD3 bit 10 (VFLAG) to 0

  BESTI = 6 i

  BESTJ = 6 j

  Proceed to "PIC3"

BESTI = BESTI / 6

BESTJ = BESTJ / 6

TSa = $\underline{K}$:$CATLOG_{BESTI} \cdot \underline{K}$:$CATLOG_{BESTJ}$

BESTI = 6 BESTI

BESTJ = 6 BESTJ

Switch FLAGWRD3 bit 10 (VFLAG) to 1

TSb = $\underline{K}$:$CATLOG_i \cdot \underline{K}$:$CATLOG_j$

Switch FLAGWRD3 bit 10 (VFLAG) to 0

If TSa > TSb: (new pair has better separation)

BESTI = 6 i

BESTJ = 6 j

Proceed to "PIC3"

<u>PICEND</u> If FLAGWRD3 bit 10 (VFLAG) = 1: (no pairs found)

HAVEPAIR = 0

Return via QMIN

i = BESTI / 6

j = BESTJ / 6

If $\underline{S}AX \cdot \underline{K}:CATLOG_j < \underline{S}AX \cdot \underline{K}:CATLOG_i$:

BESTI = 6 j (farther)

BESTJ = 6 i (closer)

HAVEPAIR = 1

Return via QMIN

<u>OCCULT</u>

$$\left[TSmat\right] = \begin{bmatrix} VEARTH_x & VEARTH_y & VEARTH_z \\ VSUN_x & VSUN_y & VSUN_z \\ VMOON_x & VMOON_y & VMOON_z \end{bmatrix}$$

$$\underline{T}S = \begin{pmatrix} CEARTH \\ CSUN \\ CMOON \end{pmatrix} - \left[TSmat\right] \underline{T}Sstar$$

Switch FLAGWRD3 bit 7 (CULTFLAG) to 0

If $TS_x \leq 0$, or if TS_y or $TS_z < 0$:

Switch FLAGWRD3 bit 7 (CULTFLAG) to 1

Return

P57 Perform "IMUCHK" (assure that IMU is on)

OPTION2 = 00003_8 (REFSMMAT orientation)

P57OPT Perform "GOPERF4R" with OPTION1 = 00001_8
(If terminate, proceed to "GOTOPOOH"; if proceed, skip next step; if other response, repeat this step.)

End job

If OPTION2 bit 2 = 0 and bit 1 = 1: (OPTION2 = 1,5,9,...)

Proceed to "PACKOPTN" (Preferred orientation)

If OPTION2 bit 2 = 1 and bit 1 = 0: (OPTION2 = 2,6,10,...)

Proceed to "P57OPT" (recycle; invalid in P57)

If OPTION2 bits 2 and 1 = 1: (OPTION2 = 3,7,11,...)

$\lfloor$XSMD$\rfloor$ = $\lfloor$REFSMMAT$\rfloor$ (REFSMMAT orientation)

Proceed to "PACKOPTN"

(Otherwise, OPTION2 = 0,4,8,12,....) (Landing site orientation)

DSPTEM1 = TIG

Perform "GOFLASH" with TS = K:V06N34 (DSPTEM1)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

If DSPTEM1 = 0: (key-in time = 0)

TALIGN = TIMENOW

TDEC1 = TIMENOW

Proceed to "P57D"

If TIMENOW $\geq$ DSPTEM1: (key-in time $\leq$ present time)

TALIGN = DSPTEM1

TDEC1 = DSPTEM1

Proceed to "P57D"

TIG = DSPTEM1 (key-in time > present time)

TALIGN = DSPTEM1

TDEC1 = DSPTEM1

P57D Perform "LEMPREC"

XSMDrf = unitRATT

Perform "LSORIENT" (compute desired IMU orientation)

PACKOPTN OPTION2 = 00000_8 (zero alignment option)

OPTION3 = 00000_8 (zero flag bit configuration)

If FLAGWRD3 bit 13 (REFSMFLG) = 1, OPTION3 = 00100_8

If FLAGWRD6 bit 1 (ATTFLAG) = 1, OPTION3 = OPTION3 + 00010_8

OPTION1 = 00010_8

DSPOPTN Proceed to "GOFLASH" with TS = K:V05N06 (OPTION1,OPTION2,OPTION3)
(If terminate,proceed to "GOTOPOOH"; if proceed,continue at next step; if other response, repeat this step.)

If FLAGWRD3 bit 13 (REFSMFLG) = 1, proceed to "GETLMATT"

If FLAGWRD6 bit 1 (ATTFLAG) = 1, proceed to "BYLMATT"

If OPTION2 bit 2 = 1: (OPTION2 = 2,3,6,7,...)

Proceed to "BYLMATT" (don't have attitude)

Perform "ALARM" with TS = 00701_8 (option inconsistent with flags)

Proceed to "GOFLASH" with TS = K:V05N09 (display alarm)
(If terminate,proceed to "GOTOPOOH"; if proceed,proceed to "DSPOPTN"; if other response, proceed to "DSPOPTN".)

GETLMATT Perform "REFMF"

BYLMATT Switch FLAGWRD8 bit 2 (INITALGN) to 1

If OPTION2 bit 1 = 1: (OPTION2 = 1,3,5,7,....)

Proceed to "GVDETER"

ATTCHK If FLAGWRD6 bit 1 (ATTFLAG) = 1, proceed to "P57OPTO"

Switch FLAGWRD8 bit 2 (INITALGN) to 0

If OPTION2 bits 2 and 1 both = 0: (OPTION2 = 0,4,8,....)

Proceed to "P57OPTO"

If OPTION2 bit 2 = 0 and bit 1 = 1: (OPTION2 = 1,5,9,...)

Proceed to "P57OPT1"

If OPTION2 bit 2 = 1 and bit 1 = 0: (OPTION2 = 2,6,10,....)

Proceed to "P57OPT2"

If OPTION2 bits 2 and 1 both = 1: (OPTION2 = 3,7,11,....)

Proceed to "P57OPT3"

P57OPT0 VEC1 = YNBSAV

VEC2 = ZNBSAV

Perform "CDUTRIG"

Perform "CALCSMSC"

TS = YNBsm

SAMETYP STARSAV1 = TS

STARSAV2 = ZNBsm

Perform "MFREF" (VEC1,2 to reference coordinates)

Proceed to "SURFLINE"

P57OPT1 VEC1 = unitRLS

VEC2 = ZNBSAV

Perform "CDUTRIG"

Perform "CALCSMSC"

Perform "CDUTRIG"

Perform "NBTOSM"

TS = [NBSMMAT] GSAV

Proceed to "SAMETYP"

P57OPT2 Proceed to "2STARS"

P57OPT3 VEC1 = unitRLS

VEC2 = unitRLS

Perform "CDUTRIG"

Perform "NBTOSM"

STARSAV1 = [NBSMMAT] GSAV

Perform "MFREF"

Proceed to "1STAR"

SURFLINE $STARAD_0$ = unit([XSMDMAT] VEC1)

TS_6 = STARSAV1

$STARAD_6$ = unit([XSMDMAT] VEC2)

TS_{12} = STARSAV2

If FLAGWRD8 bit 2 (INITALGN) = 1, proceed to "INITBY"

Perform "CHKSDATA" (R54)

If FLAGWRD0 bit 3 (FREEFLAG) = 0, proceed to "P57POST"

INITBY Perform "AXISGEN"

Perform "CALCGTA"

If FLAGWRD8 bit 2 (INITALGN) = 1, skip next step

Proceed to "GOFLASH" with TS = K:V06N93 (gyro angles)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "P57POST".)

TS_{ogc} = (OGC, IGC, MGC)

TS = TS_{ogc} / K:5DEGREES

If overflow (any component of $\underline{TS} \geq 1$):

$\underline{X}SMrf = [DCMAT]^T \underline{K}:UNITX$

$\underline{Y}SMrf = [DCMAT]^T \underline{K}:UNITY$

$\underline{Z}SMrf = [DCMAT]^T \underline{K}:UNITZ$

Perform "CDUTRIG"

Perform "CALCSMSC"

Perform "CALCGA"

If FLAGWRD8 bit 2 (INITALGN) = 0, skip next step

Proceed to "GOFLASH" with TS = K:V06N22 ($\underline{T}$HETAD)
(If terminate,proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

Perform "COARSE"

1dPIPADT = $TIMENOW_{1s}$

$\underline{P}IPA$ = -0

$\underline{G}COMP$ = 0

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

$\underline{A}NG = \underline{T}HETAD$

Perform "CD*TR*GS"

Perform "NBTOSM"

$\underline{S}TARAD_0 = [NBSMMAT]\ \underline{K}:UNITX$

$\underline{S}TARAD_6 = [NBSMMAT]\ \underline{K}:UNITY$

Perform "CDUTRIG"

Perform "CALCSMSC"

$\underline{T}S_6 = \underline{X}NBsm$

$\underline{T}S_{12} = \underline{Y}NBsm$

Perform "AXISGEN"

Perform "CALCGTA"

(End of indented steps)

TS = address of (OGC,IGC,MGC)

Perform "IMUPULSE"

Perform "IMUSTALL"

If ISSGOOD = 0, perform "CURTAINS"

SURFDISP Switch FLAGWRD3 bit 13 (REFSMFLG) to 1

[REFSMMAT] = [XSMDMAT]

If OPTION2 = 00000_8, proceed to "P57POST"

If FLAGWRD8 bit 2 (INITALGN) = 1:

Proceed to "ATTCHK" skipping first step

Perform "REFMF"

Proceed to "P57POST"

2STARS STARIND = 0 (first star)

Skip next step

1STAR STARIND = 1 (second star)

R59 If FLAGWRD3 bit 13 (REFSMFLG) = 0, proceed to "R59OUT"

Proceed to "GOFLASH" with TS = K:V01N70 (detent and star code)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "R59".)

i = low 6 bits of AOTCODE

If STARIND = 0, BESTI = 6 i

If STARIND = 1, BESTJ = 6 i

If i = 0, proceed to "R59OUT" (planet, not star code)

If 38 ≤ i, proceed to "R59OUT" (sun, earth or moon)

TS = unit([REFSMMAT] K:CATLOG$_i$)

Perform "CDUTRIG"

Perform "SMTONB"

STAR = [SMNBMAT] TS (star vector in NB coor)

POSCODE = 1

INCAZ QMIN = AOTAZ$_{POSCODE}$

TSazm = QMIN

TSelev = K:r52el (elevation = 45 degrees)

Perform "OANB"

TS1 = arccos(STAR • SCAXIS)

TS2 = TS1 - K:DEG30

If TS2 ≥ 0: (star not in field of view; try next position)

POSCODE = POSCODE + 1

If POSCODE ≥ 7, proceed to "R59ALM"

Proceed to "INCAZ"

TS = TS1 - K:DEG.5

If TS < 0:

CURSOR = 0

SPIRAL = 0

(If TS < 0:)

Proceed to "79DISP"

TS1 = 12 TS1

TS2 = unit(SCAXIS * K:UNITX)

TS3 = unit(-TS2 * SCAXIS)

TS4 = unit(SCAXIS * STAR)

TS5 = arccos(TS4 · TS2)

TS = TS3 · TS4

If TS ≥ 0, skip next step

TS5 = K:ABOUT1 - TS5

TS5 = TS5 + QMIN (one's complement form, scaled revs)

TS6 = TS5 in two's complement form scaled B-1 in revs

CURSOR = TS6

TS = TS5 + TS1 (one's complement form, scaled revs)

TS = TS in two's complement form scaled B-1 in revs

SPIRAL = TS (two's complement form, scaled ½ revs)

79DISP Proceed to "GOFLASH" with TS = K:V06N79 (CURSOR,SPIRAL,POSCODE)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "R59".)

TS = bits 3-1 of POSCODE shifted left to bits 9-7

TS1 = bits 6-1 of AOTCODE

AOTCODE = TS + TS1 (star code in bits 6-1; detent code (derived from POSCODE) in bits 9-7)

R59OUT Perform "AOTMARK"

Perform "AOTSTALL"

If AOTGOOD = 0, perform "CURTAINS"

Proceed to "R59RET"

R59ALM Perform "ALARM" with TS = 00404_8

Proceed to "GOFLASH" with TS = K:V05N09 (display alarm)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "R59OUT"; if other response, proceed to "R59".)

R59RET If STARIND = 0, proceed to "ASTAR"

TSt = TSIGHT (time of second mark)

Perform "PLANET"

VEC2 = TS

Proceed to "SURFLINE"

ASTAR STARSAV1 = $STARAD_6$ (first star marked)

TSt = TSIGHT (time of first mark)

Perform "PLANET"

VEC1 = TS

Proceed to "1STAR" (get second star sighting)

GVDETER $THETAD_x$ = K:42DEG

$THETAD_y$ = -K:42DEG

$THETAD_z$ = K:35DEG

Switch FLAGWRD3 bit 13 (REFSMFLG) to 0

Perform "LUNG" (align to THETAD and get gravity vector)

Perform "NBTOSM"

XNBsm = [NBSMMAT] K:UNITX

XSMsm = 2 $STAR_x$ STAR - K:UNITX

YNBsm = [NBSMMAT] K:UNITY

YSMsm = 2 $STAR_y$ STAR - K:UNITY

ZNBsm = [NBSMMAT] K:UNITZ

ZSMsm = 2 $STAR_z$ STAR - K:UNITZ

Perform "CALCGA" (get new THETAD)

$STARAD_{12}$ = ½ GOUT

Perform "LUNG"

STARSAV1 = unit(½ GOUT + $STARAD_{12}$)

DSPTEM1 = arccos(STARSAV1 • GSAV)

Switch FLAGWRD0 bit 3 (FREEFLAG) to 0

Proceed to "GOFLASH" with TS = K:V06N04 (DSPTEM1)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, switch FLAGWRD0 bit 3 (FREEFLAG) to 1 and continue at next step.)

GSAV = STARSAV1

If FLAGWRD0 bit 3 (FREEFLAG) = 1, proceed to "GVDETER"

Proceed to "ATTCHK"

LUNG QMIN = return address

GACC = 0

Perform "COARSE"

GCTR = -20

1dPIPADT = K:PRIO31

GCOMPSW = 0

GCOMP = 0

Perform "PIPASR" skipping first step (don't load PIPTIME1)

GREED Call "GRABGRAV" in two seconds

End job

GRABGRAV Perform "PIPASR" skipping first step

Establish "ADDGRAV" (pr13)

End task

ADDGRAV Perform "1/PIPA"

GCTR = GCTR + 1

GACC = GACC + K:lungt DELV

If GCTR < 0, proceed to "GREED"

STAR = unitGACC

Perform "CDUTRIG"

Perform "SMTONB"

GOUT = [SMNBMAT] STAR

Return via QMIN

REGCOARS Perform "CAL53A"

Switch FLAGWRD3 bit 13 (REFSMFLG) to 1

Switch FLAGWRD2 bit 4 (PFRATFLG) to 0

Perform "R51"

Proceed to "GOTOPOOH"

P57POST Proceed to "GOPERF1" with TS = 00014_8 (checklist # 14)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "ATTCHK" skipping first step; if other response, continue at next step.)

If OPTION2 $\neq 00002_8$, proceed to "GOTOPOOH"

(OPTION2 = 00002_8 so calculate landing site)

Perform "CDUTRIG"

Perform "NBTOSM"

$\underline{T}Sgref = [REFSMMAT]^T$ NBSMMAT $\underline{G}SAV$

Switch FLAGWRD3 bit 12 (LUNAFLAG) to 1

$\underline{A}LPHAV = |\underline{R}LS|\ \underline{T}Sgref$

Switch FLAGWRD1 bit 13 (ERADFLAG) to 0

TSt = TIMENOW

Perform "N89DISP"

$\underline{R}N = \underline{A}LPHAV$ (scaled B29)

PIPTIME = TSt

Perform "MOONMX"

$\underline{R}LS = [MOONMAT](\underline{R}N - ([MOONMAT]^T\ \underline{L}M504) * \underline{R}N)$

Proceed to "GOTOPOOH"

KILLAOT Switch EXTVBACT to zero (allow extended verbs)

Proceed to "GOTOPOOH"

IMUCHK If bit 9 of IMODES30 = 1: (IMU off)

Perform "ALARM" with TS = 00210_8

Proceed to "GOTOPOOH"

Switch FLAGWRD0 bit 8 (IMUSE) to 1

Return

1dPIPADT: See IMUC section.

ALPHAV: See COOR section.

ALT: See COOR section.

ANG: See COOR section.

$AOTAZ_i$, $AOTEL_i$ (i = 1,2,3,4,5,6): Twelve single precision scalars stored in units of revolutions, scaled B-1 in two's complement form. The AOT has six working positions, fixed by detents, defined by angles of rotation around the -X spacecraft axis, measured from the +Z spacecraft axis (AOTAZ) and by angles of elevation above the Y-Z plane (AOTEL). These quantities are stored in erasable memory because their exact values vary among spacecraft, but the nominal values are;

i		$AOTAZ_i$	$AOTEL_i$
1	(left forward)	-60°	45°
2	(forward)	0°	45°
3	(right forward)	60°	45°
4	(right rear)	120°	45°
5	(rear)	180°	45°
6	(left rear)	-120°	45°

AOTCODE: A single precision scalar containing the star selection code in bits 6-1 (an octal number from 1 to 45_8 for stars, 0 for a planet, and 46_8-50_8 for sun, earth and moon), and the AOT detent code in bits 9-7 (1,2,3,4,5 or 6 for AOT detents; 0 for COAS calibration; 7 for COAS position to be specified.)

AOTGOOD: A flag indicating that the AOT marking procedure was successfully executed. (Implemented as a variable return address in the listing.)

AZ, EL: Single precision angles of azimuth and elevation of the COAS stored in units of revolutions, scaled B-1 in two's complement form.

BESTI: Single precision value, scaled B14, of the index parameter for star #1 of the "best" star pair as determined by "R56". It is the star farthest from the AOT center detent position and will be zero if no star pairs are found that are satisfactory. In "R59" it is the value of the index parameter of the first celestial body used for marking (if two bodies are to be used). It is equal to six times the decimal equivalent of the "star selection code" (see AOTCODE definition and the Star Table).

BESTJ: See BESTI. In "R56" it is the index parameter for star #2 which is the closest star to the AOT center detent position. In "R59" it is the index parameter of the second celestial body used for marking (if two are to be used) or the index parameter of the single body being used (Technique 3 alignment).

CDU: See COOR section.

CEARTH, CMOON, CSUN: Three double precision cosines defining the areas around the earth, moon and sun within which an object to be marked is considered to be occulted; scaled B2 and unitless.

CURSOR: Single precision angle through which the reticle must be rotated to place the cursor on a star for a given viewing position. Used only on the lunar surface, scaled B-1 in units of revolutions.

[DCMAT]: Double precision direction cosine matrix, scaled B1 and unitless. When multiplied by K:UNITX, K:UNITY and K:UNITZ it is equivalent to the program notation of XDC, YDC and ZDC respectively.

DELV: See SERV section.

DSPTEM1: See DATA section.

EXTVBACT: See EXVB section.

GACC: Double precision sum of measured gravity vectors, scaled B13 in units of centimeters per second squared and expressed in stable member coordinates.

GCOMP, GCOMPSW: See IMUC section.

GCTR: Single precision counter scaled B14 and unitless. Set to -20 in "LUNG" and incremented by one each two seconds until it becomes zero, thus allowing 40 seconds of PIPA readings for determination of the gravity vector.

GOUT: Double precision unit gravity vector, scaled B1 and expressed in navigation base coordinates.

GSAV: Double precision storage for unit gravity vector determined in previous pass through "P57", scaled B1 and expressed in navigation base coordinates.

HAVEPAIR: Single precision flag to indicate whether "R51" was successful in finding a star pair adequately separated for marking.

IMODES30, IMUCADR, ISSGOOD: See IMUC section.

K:35DEG: Single precision constant stored as 06211_8, scaled B-1 in units of revolutions. Equation value: 0.09793. (Equivalent to 35.255 degrees.)

K:42DEG: Single precision constant stored as 07357_8, scaled B-1 in units of revolutions. Equation value 0.11667 (Equivalent to 42.001 degrees.)

K:5DEGREES: Double precision constant stored as 0.013888889, scaled B0 in units of revolutions. Equation value: 0.013888889. (Equivalent to 5 degrees.)

K:ABOUT1: Double precision constant, scaled B0 in units of revolutions. Equation value: 0.99999999.

K:CATLOG$_i$: A list of thirty-seven unit vectors defining the position of stars to be used as references against which star sighting measurements can be compared, scaled B1 and unitless. See table below.

K:CSS33: Double precision constant stored as 0.16070, scaled B2 and unitless. Equation value: 0.64280. (Equivalent to the cosine of 50 degrees)

K:CSS40: Double precision constant stored as 0.16070, scaled B2 and unitless. Equation value: 0.64280. (Equivalent to the cosine of 50 degrees)

K:CSS5: Double precision constant stored as 0.2490475, scaled B2 and unitless. Equation value: 0.99619. (Equivalent to the cosine of 5 degrees)

K:CSSUN: Double precision constant stored as 0.125, scaled B2 and unitless. Equation value: 0.5. (Equivalent to the cosine of 60 degrees)

K:DEG.5: Double precision constant stored as 0.00138888, scaled B0 in units of revolutions. Equation value: 0.00138888.

K:DEG30: Double precision constant stored as 0.083333333, scaled B0 in units of revolutions. Equation value: 0.083333333. (Equivalent to 30 degrees)

K:DEG359: Single precision constant stored as 16338×2^{-14}, scaled B0 in units of revolutions. Equation value: 0.99719. (Equivalent to 359 degrees)

K:DEGREE1: Single precision constant stored as 46×2^{-14}, scaled B0 in units of revolutions. Equation value: 0.0028. (Equivalent to 1 degree)

K:lungt: Implied constant, scaled B-1 in units of seconds^{-1}. Equation value: $\frac{1}{2}$.

K:PRIO31: See SERV section.

K:r52el: Single precision constant stored as 10000_8, program notation "BIT13", scaled B-1 in units of revolutions. Equation value: 0.125. (Equivalent to 45 degrees)

K:ROE: Double precision constant stored as 0.00257125, scaled B0 and unitless. Equation value: 0.00257125. (Equivalent to the ratio of the mean Earth to Moon distance to the mean Earth to Sun distance)

K:RSUBE: Double precision constant stored as 6378166×2^{-29}, scaled B29 in units of meters. Equation value: 6378166. (Equivalent to 6378.166 km; the equatorial radius of the Earth)

K:RSUBEM: Double precision constant stored as 384402000×2^{-29}, scaled B29 in units of meters. Equation value: 384402000. (Equivalent to 384,402.0 km; the mean distance between the Earth and Moon)

K:RSUBM: Double precision constant stored as 1738090×2^{-29}, scaled B29 in units of meters. Equation value: 1738090. (Equivalent to 1738.09 km; the mean radius of the moon)

K:TSIGHT1: Double precision constant stored as 36000×2^{-28}, scaled B28 in units of centiseconds. Equation value: 36000. (Equivalent to 6 minutes)

K:UNITX, K:UNITY, K:UNITZ: See SERV section.

K:VxxNxx: See Major Variables section.

LANDLONG, LANDLAT, LANDALT: Double precision locations for display of LONG, LAT and ALT respectively in Noun 89.

LAT, LONG: See COOR section.

LM504: See COOR section.

LOC: Single precision octal storage for starting address of an awakened job.

MARKCNTR: Single precision counter to keep track of the number of valid marks made by the astronaut and to limit the number allowed, scaled B14 and unitless.

MARKSTAT: Single precision address storage and flagword. The address of the special working storage area assigned to the mark system is kept in bits 9-1 of MARKSTAT. Bit 15 is always 0. The other bits have the following significance:

Bit		Meaning
14	(1)	A set of marks is complete and the next mark is to be interpreted as part of a new set.
	(0)	A pair of marks is in the process of being executed and redundant marks will not be accepted.
13	(1)	A mark has been made since the last reject and it alone will be cleared if a reject is entered.
	(0)	If a reject is entered, both marks will be cleared.
12	(1)	The processing of marks and mark rejects in the "MARKRUPT" routine is inhibited because the parameters needed for interpretation of marks have not yet been initialized, or the astronaut has indicated that he is finished taking marks.
	(0)	The processing of marks is allowed.
11	(1)	The Y mark of a pair has been accepted.
	(0)	The Y mark of a pair has not yet been accepted.
10	(1)	The X mark of a pair has been accepted.
	(0)	The X mark of a pair has not yet been accepted.

MARKVAC: Single precision storage for two's complement values of gimbal angles from the CDU at the time of each of the X and Y marks taken, for use in constructing the line-of-sight vector at each mark and determining the average line-of-sight vector, scaled B-1 in units of revolutions.

$MARKVAC_{MKDEX+1,3}$: Single precision storage for CURSOR and SPIRAL angles respectively, indexed such that up to five sets of angles can be stored for use in determining the average line-of-sight vector. Storage locations actually used are those normally reserved for Y-mark CDU information. (Only X-mark information is used on the lunar surface.) Scaled B-1 in units of revolutions.

MKDEX: Single precision index scaled B14 and unitless, or B3 and unitless.

[MOONMAT]: See COOR section.

MPAC: See MATX section.

n: Single precision integer scaled B14.

[NBSMMAT]: See COOR section.

OGC, IGC, MGC: See COOR section.

OPTCADR: Single precision octal storage for address to return to program that has requested marking and is waiting for the marking to be completed.

OPTION1, OPTION2, OPTION3: See DATA section.

PBODY: See ORBI section.

PIPA: See IMUC section.

PIPTIME: See SERV section.

PLANVEC: Double precision unit vector in the direction of the first celestial body to be used for marking, scaled B1 and expressed in reference coordinates.

POINTVSM: See ATTM section.

POSCODE: Single precision counter to indicate the position of the AOT. Changed to the appropriate detent code in "79DISP"; scaled B14 and unitless.

QMAJ, QMIN: Single precision octal return address storage cells. QMIN is also used as temporary working storage.

RATT: See ORBI section.

[REFSMMAT]: See COOR section.

[RFSTMAT]: Double precision, 3x3 matrix such that $\underline{A}_{star} = [RFSTMAT]\ \underline{A}_{ref}$, where $\underline{A}$ is a vector expressed in "line-of-sight" and reference coordinates respectively; scaled B1 and unitless.

RLS: See DESC section.

RN: See SERV section.

RRECTCSM: See ORBI section.

SAX: Double precision unit vector in the direction of the line-of-sight of the AOT center position, scaled B1 and expressed in reference coordinates.

SCAXIS: See ATTM section.

[SMNBMAT]: See COOR section.

[SMSTMAT]: Double precision, 3x3 matrix defined such that $\underline{A}_{star} = [SMSTMAT]\ \underline{A}_{sm}$ where $\underline{A}$ is a vector expressed in "line-of-sight" and stable member coordinates respectively; scaled B1 and unitless.

SPIRAL: Single precision angle through which the reticle must be rotated to place the spiral on a given star. Used only on the lunar surface, scaled B-1 in units of revolutions.

STAR: Line-of-sight vector to a star, expressed in navigation base coordinates. Also temporary storage for GACC in "ADDGRAV".

$STARAD_0$, $STARAD_6$: Double precision unit vectors used primarily for storage of measured position vectors of the two celestial bodies being marked and expressed in stable member coordinates. Also used as working storage for other occasions.

$STARAD_{12}$: Double precision unit vector used for working storage.

STARIND: Single precision index scaled B14 and unitless.

STARSAV1, STARSAV2: Double precision vectors scaled B1 and unitless. Used to store the two "measurement" vectors for comparison with two "reference" vectors to determine IMU alignment. Expressed in stable member coordinates.

TALIGN: Double precision time for determination of IMU alignment, scaled B28 in units of centiseconds.

TDEC1: See ORBI section.

THETA: See COOR section.

THETAD: See IMUC section.

TIG: See BURN section.

TIMENOW: See EXVB section.

TLAND: See DESC section.

TSIGHT: Double precision time of latest sighting, scaled B28 in units of centiseconds.

UXP, UYP: Double precision vectors expressed in navigation base coordinates, scaled B1 and unitless. Unit vectors perpendicular to the planes whose images in the AOT eyepiece coincide with the horizontal (X) and vertical (Y) crosshairs. The intersection of these planes determines the line-of-sight vector to the star being marked.

UXP', UYP': Double precision vectors expressed in navigation base coordinates, scaled B1 and unitless. Unit vectors perpendicular to the AOT line-of-sight and to the horizontal (X) and vertical (Y) crosshairs in the AOT eyepiece reticle. Each defines one of two planes whose intersection determines the line-of-sight vector to the image of a star in the AOT eyepiece. Because of the construction of the AOT (a rotating shaft above a fixed mirror), the field of view rotates about the line-of-sight vector as the AOT shaft rotates about the -X spacecraft axis.

VATT: See ORBI section.

VEARTH, VMOON, VSUN: Double precision unit vectors in the direction of the earth, moon and sun with origin at the spacecraft, scaled B1 and expressed in reference coordinates. (Origin also at the earth or moon)

VEC1, VEC2: See COOR section.

VRECTCSM: See ORBI section.

XDC, YDC, ZDC: In "P52D" these are the desired stable member unit vectors in present stable member coordinates. They are also the x,y and z components of [DCMAT].

XNBrf, YNBrf, ZNBrf, XNBsm, YNBsm, ZNBsm: See COOR section.

XSMrf, YSMrf, ZSMrf: See COOR section.

XSMDrf, YSMDrf, ZSMDrf: See COOR section.

[XSMDMAT] or [XSMD]: Double precision, 3x3 matrix defined such that Asmd = [XSMDMAT] Aref, where A is a vector expressed in desired stable member coordinates and reference coordinates respectively; scaled B1 and unitless. The components are XSMDrf, YSMDrf and ZSMDrf.

XYMARK: Single precision octal storage for bit indicating whether mark being processed is an X or a Y mark, or working storage for other occasions.

YNBSAV, ZNBSAV: Double precision unit vectors in the directions of the Y and Z navigation base axes, scaled B1 and expressed in moon-fixed coordinates.

Values of $\underline{K}$:$CATLOG_i$ (Star Table)

Display	Index	X Component	Y Component	Z Component	Identification
01	1	0.8748658918	0.0260879174	0.4836621670	α Andromedae
02	2	0.9342640400	0.1735073142	-0.3115219339	β Ceti
03	3	0.4775639450	0.1166004340	0.8708254803	γ Cassiopeiae
04	4	0.4917678276	0.2204887125	-0.8423473935	α Eridani
05	5	0.0130968840	0.0078062795	0.9998837600	α Ursae Minoris
06	6	0.5450107404	0.5314955466	-0.6484410356	θ Eridani
07	7	0.7032235469	0.7075846047	0.0692868685	α Ceti
10	8	0.4105636020	0.4988110001	0.7632988371	α Persei
11	9	0.3507315038	0.8926333307	0.2831839492	α Tauri
12	10	0.2011399589	0.9690337941	-0.1432348512	β Orionis
13	11	0.1371725575	0.6813721061	0.7189685267	α Aurigae
14	12	-0.0614937230	0.6031563286	-0.7952489957	α Carinae
15	13	-0.1820751783	0.9404899869	-0.2869271926	α Canis Majoris
16	14	-0.4118589524	0.9065485360	0.0924226975	α Canis Minoris
17	15	-0.3612508532	0.5747270840	-0.7342932655	γ Velorum
20	16	-0.4657947941	0.4774785033	0.7450164351	ι Ursae Majoris
21	17	-0.7742591356	0.6152504197	-0.1482892839	α Hydrae
22	18	-0.8608205219	0.4636213989	0.2098647835	α Leonis
23	19	-0.9656605484	0.0525933156	0.2544280809	β Leonis
24	20	-0.9525211695	-0.0593434796	-0.2986331746	γ Corvi
25	21	-0.4523440203	-0.0493710140	-0.8904759346	α Crucis
26	22	-0.9170097662	-0.3502146628	-0.1908999176	α Virginis
27	23	-0.5812035376	-0.2909171294	0.7599800468	η Ursae Majoris
30	24	-0.6898393233	-0.4182330640	-0.5909338474	θ Centauri
31	25	-0.7861763936	-0.5217996305	0.3311371675	α Bootis
32	26	-0.5326876930	-0.7160644554	0.4511047742	α Coronae Borealis
33	27	-0.3516499609	-0.8240752703	-0.4441196390	α Scorpii
34	28	-0.1146237858	-0.3399692557	-0.9334250333	α Trianguli Australis
35	29	-0.1124304773	-0.9694934200	0.2178116072	α Ophiuchi
36	30	0.1217293692	-0.7702732847	0.6259880410	α Lyrae
37	31	0.2069525789	-0.8719885748	-0.4436288486	σ Sagittarii
40	32	0.4537196908	-0.8779508801	0.1527766153	α Aquilae
41	33	0.5520184464	-0.7933187400	-0.2567508745	β Capricorni
42	34	0.3201817378	-0.4436021946	-0.8370786986	α Pavonis
43	35	0.4541086270	-0.5392368197	0.7092312789	α Cygni
44	36	0.8139832631	-0.5557243189	0.1691204557	ε Pegasi
45	37	0.8342971408	-0.2392481515	-0.4966976975	α Piscis Austrini

The "Display" column gives the star number that is displayed by the program (as an octal quantity). **The "Index" column is the decimal equivalent of the "Display" column and when multiplied by six is equal to BESTI (or BESTJ).**

Stars in Star Table

Display	Name	Catlg.	Magnitude	Right Ascension	Declination
01	Alpheratz	α And	2.1	0 06 49.9	+28 55 29
02	Diphda	β Cet	2.2	0 42 05.0	-18 09 04
03	Navi	γ Cas	Var.	0 54 53.0	+60 33 17
04	Achernar	α Eri	0.6	1 36 35.9	-57 23 20
05	Polaris	α UMi	2.1	2 03 18.9	+89 07 34
06	Acamar	θ Eri	3.4	2 57 07.4	-40 25 27
07	Menkar	α Cet	2.8	3 00 42.5	+ 3 58 23
10	Mirfak	α Per	1.9	3 22 10.3	+49 45 21
11	Aldebaran	α Tau	1.1	4 34 11.8	+16 27 01
12	Rigel	β Ori	0.3	5 13 05.7	- 8 14 06
13	Capella	α Aur	0.2	5 14 28.2	+45 58 10
14	Canopus	α Car	-0.9	6 23 17.1	-52 40 44
15	Sirius	α CMa	-1.6	6 43 49.6	-16 40 25
16	Procyon	α CMi	0.5	7 37 43.9	+ 5 18 11
17	Regor	γ^2 Vel	1.9	8 08 36.4	-47 14 51
20	Dnoces	ι UMa	3.1	8 57 09.7	+48 09 38
21	Alphard	α Hya	2.2	9 26 06.8	- 8 31 40
22	Regulus	α Leo	1.3	10 06 46.5	+12 06 52
23	Denebola	β Leo	2.2	11 47 31.8	+14 44 23
24	Gienah	γ Crv	2.8	12 14 15.6	-17 22 32
25	Acrux	α^1 Cru	1.0	12 24 54.9	-62 55 59
26	Spica	α Vir	1.2	13 23 36.6	-11 00 19
27	Alkaid	η UMa	1.9	13 46 21.6	+49 27 45
30	Menkent	θ Cen	2.3	14 04 54.6	-36 13 23
31	Arcturus	α Boo	0.2	14 14 17.5	+19 20 16
32	Alphecca	α CrB	2.3	15 33 25.0	+26 48 53
33	Antares	α Sco	1.2	16 27 33.9	-26 22 01
34	Atria	α TrA	1.9	16 45 28.3	-68 58 31
35	Rasalhague	α Oph	2.1	17 33 32.4	+12 34 50
36	Vega	α Lyr	0.1	18 35 55.3	+38 45 17
37	Nunki	σ Sgr	2.1	18 53 24.3	-26 20 08
40	Altair	α Aql	0.9	19 49 19.1	+ 8 47 16
41	Dabih	β Cap	3.2	20 19 19.6	-14 52 38
42	Peacock	α Pav	2.1	20 23 17.0	-56 49 58
43	Deneb	α Cyg	1.3	20 40 24.4	+45 10 21
44	Enif	ε Peg	2.5	21 42 42.7	+ 9 44 12
45	Fomalhaut	α PsA	1.3	22 55 59.7	-29 46 54

"Display" gives the star number that is displayed by the program (as an octal quantity). "Catlg" refers to the name on pp. 282-292 of "The American Ephemeris and Nautical Almanac for the Year 1970," where the magnitude and coordinate information were obtained.

Right ascension is given in hours, minutes, and seconds; Declination is given in degrees, minutes, and seconds. Both are for January 0.767, 1970, the beginning of the Besselian year.

ASCT

Ascent Guidance

P12LM Perform "R02BOTH"

Switch RADMODES bit 10 (DESIGFLG) and bit 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0 (disable RRCDU error counters)

DVTHRUSH = K:THRESH2

DVCNTR = 4

TRKMKCNT = 0

Proceed to "GOFLASH" with TS = K:V06N33 (TIG)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue with next step; if other response, repeat this step.)

Switch FLAGWRD6 bit 8 (MUNFLAG) to 1

Switch DAPBOOLS bit 11 (ACC4OR2X) to 1

Switch FLAGWRD0 bit 2 (R10FLAG) to 1

Switch FLAGWRD0 bit 7 (RNDVZFLG) to 0

Switch FLAGWRD9 bit 11 (FLPI) to 1

Switch FLAGWRD9 bit 14 (FLVR) to 1

TSt = TIMENOW

Perform "MOONMX"

$\underline{TS} = [MOONMAT]^T (\underline{K}:UNITZ + \underline{LM}504 * \underline{K}:UNITZ)$

$\underline{WM} = K:MOONRATE \, [REFSMMAT] \, \underline{TS}$

$LANDMAG = |\underline{RLS}|$

Perform "P12INIT" (initialize APS parameters and ascent targets)

TGO = K:TGOA

TDEC1 = TIG

Perform "LEMPREC"

$\underline{V1S} = [REFSMMAT] \, \underline{VATT}$

$\underline{R} = [REFSMMAT] \, \underline{RATT}$

Perform "MUNGRAV" with TSr = R

UNITR = unitR

Y = RCO (UNITR · QAXIS)

XRANGE = - Y

ZDOTD = K:VINJNOM

RDOTD = K:RDOTDNOM

Proceed to "GOFLASH" with TS = K:V06N76 (ZDOTD, RDOTD, XRANGE)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue with next step; if other response, repeat this step.)

WHICH = "P12TABLE"

YCO = XRANGE + Y

V = K:49FPS UNITR + V1S

RDOT = V · UNITR

ZAXIS = unit(UNITR * QAXIS)

Proceed to "ASCENT" (calculate initial guidance quantities)

P12RET TS = $(ATP)^2 + (ATY)^2$

If TS = 0, skip next step

TS = $\arcsin(ATY / \sqrt{TS})$

YAW = TS

PITCH = - arccos(UNITR · unitUNFC)

Perform "PFLITEDB" with interrupts inhibited

Switch FLAGWRD9 bit 11 (FLPI) to 0

Proceed to "BURNBABY"
(Standard pre-ignition sequence; initializes average-g navigation at TIG-30 seconds and calls "P12IGN" at time of ignition which sets AVEGEXIT to "ATMAG" establishing the two second guidance loop. See BURN section for details.)

P70 — If MODREG = MMNUMBER or FLAGWRD9 bit 9 (LETABORT) = 0 or FLAGWRD7 bit 5 (AVEGFLAG) = 0:

Proceed to "ABORTALM"

P70A — TS = 0

Proceed to the second step of "P71A"

P71 — If MODREG = MMNUMBER or FLAGWRD9 bit 9 (LETABORT) = 0 or FLAGWRD7 bit 5 (AVEGFLAG) = 0:

Proceed to "ABORTALM"

P71A — TS = 2

Inhibit interrupts

Cause the "Resume" instruction to resume operations at "ABRTJASK"

Resume

(The purpose of the above manipulation of the "Resume" instruction is to cause the instructions beginning at "ABRTJASK" to be performed immediately. "ABRTJASK" will appear as a task to all other jobs, i.e. "ABRTJASK" will be performed prior to the performance of any other job. Note also that "ABRTJASK" is performed under interrupt inhibit so that no tasks which are scheduled on program interrupts will be performed until after "ABRTJASK" is completed.)

ABRTJASK — TSa = 70

If TS ≠ 0:

TSa = 71

Switch FLGWRD10 bit 13 (APSFLAG) to 1 (tell DAP we are on ascent stage)

MODREG = TSa

DISPDEX = TSa (positive to kill "CLOKTASK")

Switch DAPBOOLS bit 6 (ULLAGER) to 0

Switch DAPBOOLS bit 8 (DRIFTBIT) to 0

Switch DAPBOOLS bit 9 (XOVINHIB) to 0 (allow X-axis override)

Switch DAPBOOLS bit 15 (PULSES) to 0

DB = K:1DEGDB

Switch FLAGWRD5 bit 7 (ENGONFLG) to 1

Switch bits 14 and 13 of channel 11 to 01_2 (ensure engine on)

Switch FLGWRD11 to 40000_8 (bypass LR updates)

Switch FLAGWRD0 bit 2 (R10FLAG) to 1

TEVENT = TIMENOW

AVEGEXIT = "SERVEXIT"

Establish "GOABORT" in restart logic with priority 25_8

Cause "ENEMA" to maintain "SERVICER" and "R10,R11"

Clear all other restart logic

Proceed to "ENEMA"

<u>GOABORT</u> DVCNTR = 4

WHICH = "ABRTABLE"

Switch FLAGWRD9 bit 10 (FLRCS) to 0

Switch FLAGWRD8 bit 10 (FLUNDISP) to 0

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 0

Switch DAPBOOLS bit 11 (ACC4OR2X) to 1

Switch FLAGWRD9 bit 13 (P7071FLG) to 1

Perform "INITCDUW"

If MODREG = 70: (P70)

    TGO = TIMENOW - TIG

    TBUP = MASS / K:MDOTDPS

    DV1 = (MASS / K:DVD) / K:2SEC

    DV2 = DV1

    DV3 = DV1

    AT = K:ATD / DV1

    TTO = K:100PCTTO

    VE = - K:DPSVEX

(If MODREG = 70:)

Perform "COMMINIT" (initialize ascent targets)

Proceed to "INJTARG"

Switch FLAGWRD9 bit 9 (LETABORT) to 0

DVTHRUSH = K:THRESH2

Perform "P12INIT"

If FLAGWRD9 bit 8 (FLAP) = 1:

TGO1 = 2 TGO

TGO = TGO1

Proceed to the third step of "UPTHROT"

TGO = TIMENOW - TIG

INJTARG RDOTD = ABTRDOT

Y = RCO (UNITR · QAXIS)

TS = |Y| - YLIM

If TS ≥ 0, YCO = TS signY

XRANGE = YCO - Y

Switch FLAGWRD9 bit 14 (FLVR) to 1

TS = (unitRCSM * unitR) · WM

TS1 = signTS arccos(unitRCSM · unitR)

If TS1 ≥ THETCRIT:

Switch FLAGWRD9 bit 7 (ABTTGFLG) to 1

JPARM = J2PARM

KPARM = K2PARM

RP = THETCRIT (2^{24} meters/revolution) (this step included only because of coding efficiency)

If TS1 < THETCRIT:

JPARM = J1PARM

KPARM = K1PARM

RP = J2PARM (this step included only because of coding efficiency)

RP = RCO

Switch FLAGWRD9 bit 6 (ROTFLAG) to 1

UPTHROT Perform "THROTUP"

Switch FLAGWRD9 bit 8 (FLAP) to 1

Perform "P40AUTO"

Perform "THROTUP"

Change job priority to 17 (pr17)

AVEGEXIT = "ATMAG"

End job

THROTUP THRUST = K:MAXTHRUST

Switch bit 4 of channel 14 to 1

Return

P12INIT DV3 = K:DVA

DV2 = K:DVA

DV1 = K:DVA

AT = K:ATA

TBUP = K:TBUPA

TTO = - K:ATDECAY

VE = - K:APSVEX

If FLAGWRD9 bit 8 (FLAP) = 1, return

COMMINIT RCO = K:HINJECT + LANDMAG

TXO = 0

YCO = 0

YDOTD = 0

QAXIS = unit([REFSMMAT] (VRECTCSM * RRECTCSM))

Return

ABORTALM Switch bit 7 of channel 11 to 1 (turn on Operator Error lamp)

Perform "RELDSP"

Proceed to "PINBRNCH"

ATMAG (Entered via AVEGEXIT at the end of each "SERVICER" cycle)

RDOT = HDOTDISP

ZAXIS = UHZP

QAXIS = UHYP

(documentation convenience; RDOT and HDOTDISP are the same cells in the computer; as are ZAXIS and UHZP and QAXIS and UHYP)

If FLAGWRD9 bit 10 (FLRCS) = 1:

Proceed to "ASCENT"

If ABDVCONV < K:MINABDV:

Perform "STOPRATE" with interrupts inhibited

Proceed to "ASCTERM1"

Switch FLAGWRD8 bit 8 (SURFFLAG) to 0

Switch FLAGWRD5 bit 1 (RENDWFLG) to 0

DV0 = DV1

DV1 = DV2

DV2 = DV3

DV3 = K:ONE / ABDVCONV

TS = (DV0 + DV1 + DV2 + DV3) VE K:2SEC9 / 4

TBUP = (TS + TBUP - K:6SEC18) / 2

AT = VE / TBUP

ASCENT RMAG = |R|

ZDOT = ZAXIS · V

LAXIS = ZAXIS * UNITR

YDOT = LAXIS · V

Y = RCO (UNITR · QAXIS)

GEFF = (GDT1 / K:2SEC18) · UNITR + (|UNITR * V|^2 / RMAG)

If FLAGWRD9 bit 13 (P7071FLG) = 1, perform "ZDOTDCMP"

DZDOT = ZDOTD - ZDOT

DYDOT = YDOTD - YDOT

DRDOT = RDOTD - RDOT

$\underline{V}$GVECT = DRDOT $\underline{U}$NITR + DYDOT $\underline{L}$AXIS + DZDOT $\underline{Z}$AXIS

$\underline{V}$GVECT = $\underline{V}$GVECT - $\frac{1}{2}$ TGO GEFF $\underline{U}$NITR

$\underline{V}$GBODY = [XNBPIP] $\underline{V}$GVECT

If FLAGWRD9 bit 10 (FLRCS) = 1:

TGO = |$\underline{V}$GVECT| / K:ATRCS

PCONS = 0

PRATE = 0

Perform "RPCOMP2"

End job

MAINENG TS1 = |$\underline{V}$GVECT| / VE

TGO = TBUP TS1 (1 - 0.5 TS1) - TTO

TTOGO = - TGO

If FLAGWRD7 bit 7 (IDLEFLAG) = 0:

If TGO < K:4SEC17:

TS = - (TIMENOW - PIPTIME + TTOGO)

If TS ≤ 0, TS = 1

ENGOFFDT = TS

Call "ENGOFF1" in ENGOFFDT centiseconds

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 1

If TGO < K:T2A:

Proceed to "CMPONENT"

TS = (TBUP - TGO) / TBUP

TSa = - $\log_e$(TS)

D12 = TBUP - (TGO / TSa)

If FLAGWRD9 bit 12 (FLPC) = 1:

PRATE = 0

YRATE = 0

Proceed to "CONST"

If TGO $<$ K:T3:

Switch FLAGWRD9 bit 12 (FLPC) to 1

PRATE = 0

YRATE = 0

Proceed to "CONST"

D21 = TGO - D12

TSe = $\frac{1}{2}$ TGO - D21

PRATE = (DRDOT D21 + TGO RDOT + RMAG - RCO) / (TSe TGO)

YRATE = (DYDOT D21 + TGO YDOT + Y - YCO) / (TSe TGO)

If PRATE $\geq$ 0:

PRATE = 0

Proceed to "CONST"

If PRATE / TBUP $<$ K:PRLIMIT:

PRATE = K:PRLIMIT TBUP

CONST PCONS = (DRDOT / TSa) - PRATE D12

YCONS = (DYDOT / TSa) - YRATE D12

CMPONENT If **FLAGWRD9** bit 13 (P7071FLG) = 1, perform "RPCOMP2"

ATR = {(K:100CS PRATE + PCONS) / TBUP} - GEFF

ATY = (K:100CS YRATE + YCONS) / TBUP

AH = ATY LAXIS + ATR UNITR

AHMAG = |AH|

ATPSQ = $(AT)^2 - (AHMAG)^2$

If ATPSQ < 0:

  $\underline{AH}$ = (AT / AHMAG) $\underline{AH}$

  ATP = 0

  Skip next step

ATP = $\sqrt{ATPSQ}$ sign(DZDOT)

$\underline{UNFC}$ = ATP $\underline{ZAXIS}$ + $\underline{AH}$

If FLAGWRD9 bit 11 (FLPI) = 1:

  Proceed to "P12RET"

If FLAGWRD9 bit 14 (FLVR) = 1:

  If RMAG - LANDMAG < K:25KFT:

    Switch DAPBOOLS bit 9 (XOVINHIB) to 1

    $\underline{TS}$ = unit(ATY $\underline{LAXIS}$ + ATP $\underline{ZAXIS}$)

    If RDOT < K:40FPS:

      $\underline{UNWC}$ = $\underline{TS}$

      $\underline{UNFC}$ = $\underline{UNITR}$

      Proceed to "ASCTERM"

    Switch FLAGWRD9 bit 6 (ROTFLAG) to 0

  Switch FLAGWRD9 bit 14 (FLVR) to 0

  If FLAGWRD9 bit 6 (ROTFLAG) = 0:

    TXO = PIPTIME + K:10SECS

$\underline{UNWC}$ = - $\underline{UNITR}$

If TXO ≥ PIPTIME:

  Proceed to "ASCTERM"

```
            If FLAGWRD9 bit 6 (ROTFLAG) = 1:
                TS = (unitUNFC · XNBPIP) - COSTHET1
                If TS < 0:
                    TS = (XNBPIP · UNITR) - COSTHET2
                    If TS < 0:
                        UNFC = UNITR
                        Proceed to "ASCTERM"
                Switch FLAGWRD9 bit 6 (ROTFLAG) to 0
            Switch DAPBOOLS bit 9 (XOVINHIB) to 0   (allow x-axis override)
            If FLAGWRD9 bit 13 (P7071FLG) = 0:
                Switch FLAGWRD3 bit 11 (NOR29FLG) to 0
ASCTERM     If FLAGWRD9 bit 10 (FLRCS) = 1:
                End job
            Perform "FINDCDUW"
ASCTERM1    If FLAGWRD9 bit 10 (FLRCS) = 1:
                End job
            If FLAGWRD8 bit 10 (FLUNDISP) = 1:
                End job
            Proceed to "GODSP" with TS = K:V06N63   (ABVEL, HDOTDISP, HCALC1)

ENGOFF1     Perform "ENGINOF2"
            Establish "CUTOFF"                         (pr17)
            End task
```

CUTOFF Switch FLAGWRD9 bit 10 (FLRCS) to 1

Proceed to "GOFLASH" with TS = K:V16N63 (ABVEL, HDOTDISP, HCALC1)
(If terminate, proceed to "TERMASC"; if proceed, continue with next step; if other response, repeat this step.)

Inhibit interrupts

Perform "ZATTEROR"

Perform "SETMINDB"

Release interrupt inhibit

Proceed to "GOFLASH" with TS = K:V16N85 (VGBODY)
(If terminate, proceed to "TERMASC"; if proceed, proceed to "TERMASC"; if other response, repeat this step.)

TERMASC Inhibit interrupts

Perform "RESTORDB"

Switch FLAGWRD9 bit 9 (LETABORT) to 0

Release interrupt inhibit

Proceed to "GOTOPOOH"

RPCOMP2 $RP = RMAG + RDOT\ TGO + \frac{PCONS\ TGO^2}{2\ TBUP} + \frac{PRATE\ TGO^3}{6\ TBUP}$

Return

ZDOTDCMP $TS = (unit\underline{R}CSM * unit\underline{R}) \cdot \underline{WM}$

$TS1 = signTS\ \arccos(unit\underline{R}CSM \cdot unit\underline{R})$

$RA = JPARM + KPARM\ TS1 - RP$

If $RA < RAMIN$, $RA = RAMIN$

$ZDOTD = \sqrt{2\ K{:}MUMm37\ RA\ /\ (RA + RP)\ RP}$

Return

ABDVCONV: Double precision magnitude of sensed change in velocity converted to units of meters per centisecond and scaled B5.

ABRTABLE: see WHICH of the BURN section.

ABTRDOT: Double precision erasable memory constant representing the radial rate required at insertion for aborts from powered descent, scaled B7 in units of meters per centisecond.

ABVEL: see SERV section.

AH: Double precision intermediate computation, scaled B-9 in units of meters per centisecond squared.

AHMAG: Double precision intermediate computation, scaled B-9 in units of meters per centisecond squared.

AT: Double precision LM thrust acceleration magnitude, scaled B-9 in units of meters per centisecond squared.

ATP: Double precision intermediate computation, scaled B-9 in units of meters per centisecond squared.

ATPSQ: Double precision intermediate computation, scaled B-18 in units of meters squared per centiseconds to the fourth power.

ATR: Double precision required radial acceleration, scaled B-9 in units of meters per centisecond squared.

ATY: Double precision required crossrange acceleration, scaled B-9 in units of meters per centisecond squared.

AVEGEXIT: see SERV section.

COSTHET1: Double precision erasable memory constant, scaled B2 and unitless.

COSTHET2: Double precision erasable memory constant, scaled B2 and unitless.

DAPBOOLS: see DAPA section.

DB: see DAPB section.

DISPDEX: see BURN section.

DRDOT, DYDOT, DZDOT: Double precision velocity-to-be-gained components in the radial, crossrange, and downrange directions respectively, scaled B7 in units of meters per centisecond.

DVCNTR: see SERV section.

DVTHRUSH: see SERV section.

DVO, DV1, DV2, DV3: Double precision quantities representing the reciprocal of successive PIPA readings, scaled B7 in units of centiseconds per meter; program notation 1/DVO, 1/DV1, 1/DV2, 1/DV3.

D12: Double precision intermediate computation, scaled B17 in units of centiseconds.

D21: Double precision intermediate computation, scaled B17 in units of centiseconds.

ENGOFFDT: Single precision delta time for engine cutoff, scaled B14 in units of centiseconds.

GDT1: see SERV section.

GEFF: Double precision effective gravity, scaled B-9 in units of meters per centiseconds squared.

HCALC1: See DESC section.

HDOTDISP: See SERV section.

JPARM: Double precision parameter used in the calculation of ZDOTD for aborts from the powered descent, scaled B24 in units of meters (see note following K2PARM). JPARM contains J1PARM or J2PARM.

J1PARM: Double precision parameter used in the calculation of ZDOTD for aborts where the LM to CSM phase angle is less than THETCRIT, scaled B24 in units of meters (see note following K2PARM); part of the erasable load.

J2PARM: Double precision parameter used in the calculation of ZDOTD for aborts where the LM to CSM phase angle is greater than or equal to THETCRIT, scaled B24 in units of meters (see note following K2PARM); part of the erasable load.

KPARM: Double precision parameter used in the calculation of ZDOTD for aborts from the powered descent, scaled B24 in units of meters per revolution (see note following K2PARM). KPARM contains K1PARM or K2PARM.

K1PARM: Double precision parameter used in the calculation of ZDOTD for aborts where the LM to CSM phase angle is less than THETCRIT, scaled B24 in units of meters per revolution (see note following K2PARM); part of the erasable load.

K2PARM: Double precision parameter used in the calculation of ZDOTD for aborts where the LM to CSM phase angle is greater than or equal to THETCRIT, scaled B24 in units of meters per revolution (see note below); part of the erasable load.

Note: JPARM and KPARM are considered in this document to be scaled B24; thus the erasable parameters J1PARM, J2PARM, K1PARM, K2PARM are expected to be multiplied by 2 and then scaled B24 (of course this is the same as scaling by B23) in order to introduce a factor of 2 into the equation which calculates RA.

K:APSVEX: Single precision constant stored as - 30.3 X 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: - 30.3

K:ATA: Double precision constant stored as 3.2883 E-4 X 2^{9}, scaled B-9 in units of meters per centisecond squared; program notation (AT)A. Equation value: 3.2883 E-4

K:ATD: Double precision constant stored as 0.02, scaled B-2 in units of reciprocal centiseconds; program notation K(AT). Equation value: 0.005

K:ATDECAY: Double precision constant stored as - 18. X 2^{-28}, scaled B28 in units of centiseconds. Equation value: - 18.

K:ATRCS: Double precision constant stored as 0.785 E-4 X 2^{10}, scaled B-10 in units of meters per centisecond squared; program notation AT/RCS. Equation value: 0.785 E-4

K:DPSVEX: Single precision constant stored as - 29.5588868 X 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: - 29.5588868.

K:DVA: Double precision constant stored as 15.2 X 2^{-7}, scaled B7 in units of centiseconds per meter; program notation (1/DV)A. Equation value: 15.2

K:DVD: Double precision constant stored as 436.7 X 2^{-9}, scaled B9 in units of kilogram-meters per centisecond-second; program notation K(1/DV). Equation value: 436.7

K:HINJECT: Double precision constant stored as 18288. X 2^{-24}, scaled B24 in units of meters. Equation value: 18288. (equivalent to 60,000 feet)

K:MAXTHRUST: Single precision constant stored as 10000_8, scaled B14 in units of DPS throttle pulses; program notation BI^8T13. Equation value: 4096. (enough to oversaturate the throttle -- see THRUST)

K:MDOTDPS: Double precision constant stored as 0.148 X 2^{-3}, scaled B3 in units of kilograms per centisecond. Equation value: 0.148 (equivalent to 32.62 pounds mass per second.)

K:MINABDV: Double precision constant stored as 0.0356 X 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: 0.0356

K:MOONRATE: Double precision constant stored as 0.2661699489 E-7 X 2^{19}, scaled B-19 in units of radians per centisecond. Equation value: 0.2661699489 E-7

K:MUMm37: Double precision constant stored as 4.902778 E 8 X 2^{-37}, scaled B37 in units of meters cubed per centisecond squared; program notation MUM(-37). Equation value: 4.902778 E 8

K:ONE: Single precision constant stored as 00004_8, scaled B12 and unitless; program notation BIT3H. Equation value: 1.0.

K:PRLIMIT: Double precision constant stored as - 0.0639, scaled B-21 in units of meters per centisecond cubed. Equation value: -0.3048 E-7.

K:RDOTDNOM: Double precision constant stored as 0.059436 X 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.059436. (Corresponds to 19.5 feet per second.)

K:TBUPA: Double precision constant stored as 91902. X 2^{-17}, scaled B17 in units of centiseconds; program notation (TBUP)A. Equation value: 91902.

K:TGOA: Double precision constant stored as 3.7 E 4 X 2^{-17}, scaled B17 in units of centiseconds; program notation (TGO)A. Equation value: 3.7 E 4

K:THRESH2: Double precision constant stored as 308. X 2^{-14}, scaled B14 in units of centimeters per second. Equation value: 308.

K:T2A: Double precision constant stored as 200. X 2^{-17}, scaled B17 in units of centiseconds. Equation value: 200.

K:T3: Double precision constant stored as 1000. X 2^{-17}, scaled B17 in units of centiseconds. Equation value: 1000.

K:UNITZ: Double precision constant vector stored as (0, 0, 0.5), scaled B1 and unitless. Equation value: (0, 0, 1)

K:VINJNOM: Double precision constant stored as 16.7924 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 16.7924. (Equivalent to 5509.31758 feet per second.)

K:10SECS: Double precision constant stored as 1000 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 1000.

K:100PCTTO: Double precision constant stored as 24 x 2^{-17}, scaled B17 in units of centiseconds. Equation value: 24.

K:100CS: Double precision constant stored as 200 x 2^{-18}, scaled B17 in units of centiseconds. Equation value: 100.

K:1DEGDB: Single precision constant stored as 00554_8, scaled B-3 in units of revolutions. Equation value: 0.00277. (Equivalent to 1 degree.)

K:2SEC: Implicit program constant equal to two (2) seconds.

K:2SEC18: Double precision constant stored as 200. X 2^{-18}, scaled B18 in units of centiseconds; program notation 2SEC(18). Equation value: 200.

K:2SEC9: Double precision constant stored as 200. X 2^{-9}, scaled B9 in units of centiseconds; program notation 2SEC(9). Equation value: 200.

K:25KFT: Double precision constant stored as 7620. X 2^{-24}, scaled B24 in units of meters. Equation value: 7620.

K:4SEC17: Double precision constant stored as 400. X 2^{-17}, scaled B17 in units of centiseconds; program notation 4SEC(17). Equation value: 400.

K:49FPS: Double precision constant representing the expected LM RDOT at the end of the vertical rise phase, stored as 0.149352 X 2^{-6}, scaled B6 in units of meters per centisecond. Equation value: 0.149352

K:40FPS: Double precision constant stored as 0.12192 X 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.12192

K:6SEC18: Double precision constant stored as 600. X 2^{-18}, scaled B18 in units of centiseconds; program notation 6SEC(18). Equation value: 600.

LANDMAG: Double precision magnitude of lunar landing (or launch) site radius, scaled B24 in units of meters; program notation /LAND/.

LAXIS: Double precision crossrange unit vector, scaled B1, unitless, and expressed in the Platform Coordinate system.

LM504: Double precision libration vector of the moon, scaled B0 in units of radians and expressed in moon-centered, moon-fixed coordinates.

MASS: see SERV section.

MMNUMBER: see PGSR section.

MODREG: see DATA section.

[MOONMAT]: Double precision, 3X3 orthogonal transformation matrix, scaled B1 and unitless. Defined such that Asg = [MOONMAT] Aref, where A is a vector expressed in selenographic and reference coordinates respectively.

PCONS: Double precision pitch guidance coefficient, scaled B9 in units of meters per centisecond .

PIPTIME: see SERV section.

PITCH: Double precision predicted FDAI pitch angle at the end of the pitch over maneuver, scaled B0 in units of revolutions.

PRATE: Double precision pitch rate guidance coefficient, scaled B-8 in units of meters per centisecond squared.

P12TABLE: see WHICH of the BURN section.

QAXIS: Double precision unit vector, scaled B1, unitless, and expressed in the Platform Coordinate system.

R: Double precision present navigated vector position of the LM, measured from the center of the moon, scaled B24 in units of meters and expressed in the Platform Coordinate system.

RA: Double precision distance from the center of the moon to the apogee of the desired insertion orbit, scaled B24 in units of meters.

RADMODES: see page 74.

RAMIN: Double precision parameter which is the minimum value allowed for RA, scaled B24 in units of meters; part of the erasable load.

RATT, VATT: see ORBI section.

RCO: Double precision desired insertion radius magnitude, scaled B24 in units of meters.

RCSM: see SERV section.

RDOT, YDOT, ZDOT: Double precision velocity components in the radial, crossrange, and downrange directions respectively, scaled B7 in units of meters per centisecond.

RDOTD, YDOTD, ZDOTD: Double precision desired velocity components in the radial, crossrange, and downrange direction respectively, scaled B7 in units of meters per centisecond.

[REFSMMAT]: see COOR section.

RLS: Double precision lunar landing (or launch) site vector, measured from the center of the moon, scaled B27 in units of meters and expressed in moon-fixed coordinates.

RMAG: Double precision magnitude of the LM position vector, scaled B24 in units of meters. Program notation (/R/MAG).

RP: Double precision predicted insertion radius magnitude measured from the center of the moon, scaled B24 in units of meters.

RRECTCSM, VRECTCSM: see ORBI section.

TBUP: Double precision ratio of mass to mass flow rate, scaled B17 in units of centiseconds.

TDEC1: see ORBI section.

TEVENT: Double precision time-of-event for downlink information (as used in this section time-of-abort) , scaled B28 in units of centiseconds.

TGO, TGO1: Double precision predicted length of burn, scaled B17 in units of centiseconds.

THETCRIT: Double precision LM to CSM phase angle at which abort targets are switched, scaled B0 in units of revolutions; part of the erasable load.

TIG: Double precision time of engine ignition, scaled B28 in units of centiseconds.

THRUST: Cell used to provide DPS throttle commands when Bit 4 of channel 14 is set, scaled B14 in units of DPS throttle pulses. One pulse corresponds to about 2.8 pounds of thrust. The maximum command recognized by the throttle is 3428 pulses.

TIMENOW: Current time scaled B28 in units of centiseconds, incremented every centisecond.

TRKMKCNT: see RNAV section.

TTO: Double precision time delay from the issuance of the engine OFF signal to actual thrust decay, scaled B17 in units of centiseconds.

TTOGO: Double precision negative of time-to-go for display purposes, scaled B28 in units of centiseconds.

TXO: Double precision time at which X-axis override is permitted in ascent guidance, scaled B28 in units of centiseconds.

UHZP: see SERV section.

UNITR: Double precision unit vector in the radial direction, scaled B1, unitless, and expressed in the Platform coordinate system; program notation UNIT/R/.

UNFC: see BURN section.

UNWC: see BURN section.

V: Double precision present navigated velocity vector of the LM, scaled B7 in units of meters per centisecond and expressed in the Platform coordinate system.

VE: Double precision engine exhaust velocity, scaled B7 in units of meters per centisecond.

VGBODY: Double precision velocity-to-be-gained vector in body coordinates, scaled B7 in units of meters per centisecond.

VGVECT: Double precision velocity-to-be-gained vector in Platform coordinates, scaled B7 in units of meters per centisecond.

V1S: Double precision LM velocity vector at TIG, scaled B7 in units of meters per centisecond.

WHICH: see BURN section.

WM: Double precision lunar rotation rate vector, i.e. lunar rotation rate times the lunar rotation axis vector, scaled B-17 in units of radians per centisecond.

[XNBPIP] : Double precision matrix with the first row equal to the components of XNBPIP, the second row equal to the components of YNBPIP, and the third row equal to the components of ZNBPIP, where XNBPIP, YNBPIP, ZNBPIP are unit vectors along the X, Y, and Z spacecraft axes, scaled B1 and expressed in the Platform coordinate system at PIPTIME.

XRANGE: Double precision magnitude of the crossrange distance to be removed during the ascent maneuver, scaled B29 in units of meters.

Y: Double precision magnitude of the out-of-CSM-plane position, scaled B24 in units of meters.

YAW: Double precision predicted FDAI yaw angle at the end of the vertical rise phase, scaled B0 in units of revolutions.

YCO: Double precision desired crossrange position at orbit insertion, scaled B24 in units of meters.

YCONS: Double precision yaw guidance coefficient, scaled B9 in units of meters per centisecond.

YDOT: see RDOT.

YDOTD: see RDOTD.

YLIM: Double precision erasable memory constant representing the maximum cross-range distance to be removed during an abort from the powered descent, scaled B24 in units of meters.

YRATE: Double precision yaw rate guidance coefficient, scaled B-8 in units of meters per centisecond squared.

ZAXIS: Double precision unit vector in the downrange direction, scaled B1 and unitless; program notation ZAXIS1.

ZDOT: see RDOT.

ZDOTD: see RDOTD.

ATTM

R60LEM TEMPR60 = return address

If FLAGWRD5 bit 6 (3AXISFLG) = 0:

Perform "VECPOINT"

THETAD = TS

Switch FLAGWRD0 bit 4 (NEEDLFLG) to 1

Switch FLAGWRD0 bit 15 (NEED2FLG) to 0

Perform "BALLANGS"

TOBALLA Perform "GOPERF2R" with TS = K:V06N18 (display FDAI angles)
(If terminate, proceed to "R61TEST"; if proceed, proceed to "REDOMANC"; if other response, proceed to "ENDMANU1".)

Perform "CHKLINUS" (make display priority if necessary)

End job

REDOMANC If FLAGWRD5 bit 6 (3AXISFLG) = 0:

Perform "VECPOINT"

THETAD = TS

Perform "BALLANGS"

If bit 10 of channel 30 = 1 (not PGNCS control), or if bit 14 of channel 31 = 1 (not AUTO control mode):

Proceed to "TOBALLA" (not AUTO)

Perform "GODSPR" with TS = K:V06N18 (display FDAI angles)

Perform "CHKLINUS" (make display priority if necessary)

Perform "GOMANUR"

Proceed to "ENDMANUV"

GOMANUR If ATTCADR ≠ 0: (single precision check)

$TS1_{dp}$ = calling address +1 in 2CADR format

Proceed to "BAILOUT1" with TS = 31210_8

ATTCADR = calling address +1, in 2CADR format

ATTPRIO = bits 14-10 of PRIORITY

Proceed to "KALCMAN3"

ENDMANUV Proceed to "TOBALLA"

ENDMANU1 Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Return via TEMPR60

R61TEST If MODREG = 0, proceed to "ENDMANU1"

If FLAGWRD4 bit 12 (PDSPFLAG) = 1, proceed to "TRMTRACK"

Proceed to "GOTOPOOH"

BALLANGS BALLEXIT = return address

Perform "CD*TR*GS" with ANG = THETAD

TS_z = arcsin(-SINOGA COSMGA)

SINTH = SINMGA

COSTH = COSOGA COSMGA

Perform "ARCTAN"

TS_x = THETA

COSTH = COSOGA COSIGA - SINMGA SINOGA SINIGA

SINTH = SINIGA COSOGA + SINMGA SINOGA COSIGA

Perform "ARCTAN"

TS_y = THETA

FDAI = TS converted to two's complement form

Return via BALLEXIT

VECPNT1 VECQTEMP = return address (input here from "R61C+L02" with desired CDU's in $\underline{TScdu}$)

Skip next two steps

VECPOINT VECQTEMP = return address

$\underline{TScdu} = \underline{CDU}$

Perform "CDUTODCM"

$[MIS] = [TSmat]$

$\underline{TSfinal} = \text{unit}(\underline{POINTVSM}\ [MIS])$

$\underline{COF} = -\ \text{unit}(\underline{TSfinal} * \underline{SCAXIS})$

$TSang = \arccos(\underline{SCAXIS} \cdot \underline{TSfinal})$

If overflow or if $|\underline{TSfinal} * \underline{SCAXIS}| < 2^{-12}$:

> If $\underline{TSfinal} \cdot \underline{SCAXIS} \geq 0$:
>
> > $\underline{TS} = \underline{TScdu}$ (specify zero maneuver)
> >
> > Return via VECQTEMP
>
> $\underline{YSM}_{sc} = \underline{K}:UNITY\ [MIS]$
>
> $\underline{TS} = \text{unit}(\underline{YSM}_{sc} * \underline{K}:UNITX) * \underline{SCAXIS}$
>
> $\underline{COF} = \text{unit}\underline{TS}$
>
> If overflow or if $|\underline{TS}| < 2^{-12}$:
>
> > $\underline{COF} = \underline{K}:UNITX$
>
> $TSang = \frac{1}{2}$

Perform "DELCOMP"

$[TSmat] = [MIS]\ [DELMAT]$

$\underline{YSM}_{scd} = \underline{K}:UNITY\ [TSmat]$

If $|\underline{YSM}_{scd} \cdot \underline{K}:UNITX| <$ K:SINGIMLC, proceed to "FINDGIMB"

If $|SCAXIS_x| \geq$ K:SINVEC1, proceed to "FINDGIMB"

(Otherwise, the vector being pointed is something other than the thrust vector and an effort will be made to avoid gimbal lock with a preliminary roll.)

$[MIS] = [TSmat]$

$\underline{Y}SM_{sc} = \underline{K}$:UNITY [MIS]

$\underline{T}S = (\underline{Y}SM_{sc}$ signYSM_{sc_x}) * $\underline{S}$CAXIS

$\underline{C}OF = - \underline{S}$CAXIS sign$TS_x$

If $|SCAXIS_x| \geq$ K:SINVEC2, TSang = K:VECANG1 (AOT)

If $|SCAXIS_x| <$ K:SINVEC2, TSang = K:VECANG2 (Radar, Y or Z)

Perform "DELCOMP"

[TSmat] = [MIS] [DELMAT]

FINDGIMB Perform "DCMTOCDU"

$\underline{T}S = \underline{T}$Scdud converted to two's complement form

Return via VECQTEMP

KALCMAN3 $\underline{B}CDU = \underline{C}DU$

If $|THETAD_z| \geq$ K:LOCKANGL:

Perform "ALARM" with TS = 00401_8

Proceed to "NOGO"

$\underline{T}Scdu = \underline{B}CDU$

Perform "CDUTODCM"

[MIS] = [TSmat]

$\underline{T}Scdu = \underline{T}HETAD$

Perform "CDUTODCM"

[MFS] = [TSmat]

$[TMIS] = [MIS]^T$

[MFI] = [TMIS] [MFS]

$[TMFI] = [MFI]^T$

$COFSKEW_z = \frac{1}{2}(TMFI_{12} - MFI_{12})$

$COFSKEW_y = \frac{1}{2} (MFI_{13} - TMFI_{13})$

$COFSKEW_x = \frac{1}{2} (TMFI_{23} - MFI_{23})$

$CAM = \frac{1}{2} (MFI_{11} + MFI_{22} + MFI_{33} - 1)$

AM = arccosCAM

If AM < K:MINANG: (No need for rate limited maneuver)

$\underline{C}DUD = \underline{T}HETAD$

Proceed to "NOGO"

If AM < K:MAXANG:

$\underline{C}OF = unit\underline{C}OFSKEW$ (normal path)

Switch FLAGWRD2 bit 3 (CALCMAN3) to 1

Proceed to "WCALC"

$$[MFISYM] = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} ([MFI] + [TMFI])$$

TS = 1 - CAM

If $|TS| \geq 2$, TS = K:posmaxdp signTS

$COF_z = \sqrt{2 (MFISYM_{33} - CAM) / TS}$

$COF_y = \sqrt{2 (MFISYM_{22} - CAM) / TS}$

$COF_x = \sqrt{2 (MFISYM_{11} - CAM) / TS}$

$\underline{C}OF = unit\underline{C}OF$ (eliminates common factors)

If $COF_x \geq COF_y$ and $COF_x \geq COF_z$: (method 1)

$COF_x = COF_x \; signCOFSKEW_x$

$COF_y = COF_y \; signCOFSKEW_x \; signMFISYM_{12}$

$COF_z = COF_z \; signCOFSKEW_x \; signMFISYM_{13}$

Switch FLAGWRD2 bit 3 (CALCMAN3) to 1

Proceed to "WCALC"

If $COF_y > COF_x$ and $COF_y \geq COF_z$: (method 2)

$COF_x = COF_x$ $signCOFSKEW_y$ $signMFISYM_{12}$

$COF_y = COF_y$ $signCOFSKEW_y$

$COF_z = COF_z$ $signCOFSKEW_y$ $signMFISYM_{23}$

Switch FLAGWRD2 bit 3 (CALCMAN3) to 1

Proceed to "WCALC"

(Otherwise, $COF_z > COF_y$ and $COF_z > COF_x$)

$COF_x = COF_x$ $signCOFSKEW_z$ $signMFISYM_{13}$ (method 3)

$COF_y = COF_y$ $signCOFSKEW_z$ $signMFISYM_{23}$

$COF_z = COF_z$ $signCOFSKEW_z$

Switch FLAGWRD2 bit 3 (CALCMAN3) to 1

<u>WCALC</u> Perform "DELCOMP" with TSang = $K{:}ARATE_{RATEINDX}$

$\underline{B}RATE = K{:}ARATE_{RATEINDX}\ \underline{C}OF$

TM = AM K:ANGLTIME / $K{:}ARATE_{RATEINDX}$

Switch FLAGWRD2 bit 2 (CALCMAN2) to 1

<u>NEWANGL</u> [MIS] = [MIS] [DELMAT]

Perform "DCMTOCDU" with [TSmat] = [MIS]

$\underline{N}CDU = \underline{T}Scdud$ converted from one's to two's complement form

If FLAGWRD2 bit 2 (CALCMAN2) = 1:

Switch FLAGWRD2 bit 2 (CALCMAN2) to 0

TM = TM + TIMENOW - K:ONESEK

Inhibit interrupts

OMEGARD = $BRATE_z$

DELRERОR = |OMEGARD| K:BIASCALE OMEGARD / 1JACCR

OMEGAQD = $BRATE_y$

```
               (If FLAGWRD2 bit 2 = 1:)
                    DELQEROR = |OMEGAQD| K:BIASCALE OMEGAQD / 1JACCQ
                    OMEGAPD = BRATE_x
                    DELPEROR = |OMEGAPD| K:BIASCALE OMEGAPD / 1JACCP
                    NEXTIME = TIMENOW + K:ONESEK     (less significant halves only)
               DELCDU = K:DTdTAU (BCDU - NCDU)       (two's complement difference )
               CDUD = BCDU
               BCDU = NCDU
               Release interrupt inhibit
               TS = TM - TIMENOW
               If TS > 0, proceed to "CONTMANU"
               If TS = 0:
                    Call "MANUSTOP" in 1 second
                    End job
               TS = TS + K:ONESEK + 1
               If TS ≤ 0, TS = 1
               Call "MANUSTOP" in TS centiseconds
               End job
CONTMANU       TS = NEXTIME - TIMENOW     (less significant halves only)
               If TS < 0, TS = 2^14 + TS
               Call "UPDTCALL" in TS centiseconds
               NEXTIME = NEXTIME + K:ONESEK     (less significant halves only)
               End job
```

(pr26)

UPDTCALL Establish "NEWDELHI"

End task

NEWDELHI If bit 14 of channel 31 = 1 (not AUTO control mode):

Perform "ZATTEROR"

Proceed to "NOGO"

Proceed to "NEWANGL"

NOGO Perform "STOPRATE"

Call "GOODMANU" in 0.02 seconds

End job

MANUSTOP DELCDU = 0

OMEGARD = 0

DELREROR = 0

OMEGAQD = 0

DELQEROR = 0

CDUD = THETAD

OMEGAPD = 0

DELPEROR = 0

GOODMANU TS = ATTCADR

ATTCADR = +0

Establish a job starting at the address specified in TS with priority equal to that it had before entering "KALCMAN3" (saved in ATTPRIO in "GOMANUR") Essentially equivalent to "returning" to that job.

End task

CDUTODCM ϕ = $TScdu_x$ converted to one's complement form (outer gimbal)

θ = $TScdu_y$ converted to one's complement form (inner gimbal)

ψ = $TScdu_z$ converted to one's complement form (middle gimbal)

$$[TSmat] = \begin{bmatrix} \cos\theta\cos\psi & \sin\theta\sin\phi - \cos\theta\cos\phi\sin\psi & \sin\theta\cos\phi + \cos\theta\sin\phi\sin\psi \\ \sin\psi & \cos\psi\cos\phi & -\cos\psi\sin\phi \\ -\sin\theta\cos\psi & \cos\theta\sin\phi + \sin\theta\cos\phi\sin\psi & \cos\theta\cos\phi - \sin\theta\sin\phi\sin\psi \end{bmatrix}$$

Return

DCMTOCDU $\psi = \arcsin TSmat_{21}$ (limited to within $\pm \frac{1}{4}$)

$TScos = \cos\psi$

If $|TScos| \geq 1$, $TScos = (1 - 2^{-28})\ sign TScos$

$\theta = \arcsin(-TSmat_{31} / TScos)$

If $TSmat_{11} < 0$, $\theta = \frac{1}{2}\ sign\theta - \theta$

$\phi = \arcsin(-TSmat_{23} / TScos)$

If $TSmat_{22} < 0$, $\phi = \frac{1}{2}\ sign\phi - \phi$

$TScdud = (\phi, \theta, \psi)$

Return

DELCOMP $TSs = \sin TSang$

$TSc = \cos TSang$

$TSd = 1 - \cos TSang$

If $|TSd| \geq 2$ (overflow), TSd = K:posmaxdp signTSd

$TS = COF_x^2\ TSd + TSc$

If $|TS| \geq 1$, TS = K:posmaxdp signTS

$DELMAT_{11} = TS$

$TS = COF_y^2 \ TSd + TSc$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{22} = TS$

$TS = COF_z^2 \ TSd + TSc$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{33} = TS$

$TS = COF_x \ COF_y \ TSd + COF_z \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{21} = TS$

$TS = COF_x \ COF_y \ TSd - COF_z \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{12} = TS$

$TS = COF_x \ COF_z \ TSd + COF_y \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{13} = TS$

$TS = COF_x \ COF_z \ TSd - COF_y \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{31} = TS$

$TS = COF_y \ COF_z \ TSd + COF_x \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{32} = TS$

$TS = COF_y \ COF_z \ TSd - COF_x \ TSs$

If $|TS| \geq 1$, $TS = K{:}posmaxdp \ signTS$

$DELMAT_{23} = TS$

Return

R62DISP Proceed to "GOFLASH" with TS = K:V06N22 (THETAD)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, repeat this step.)

Switch FLAGWRD5 bit 6 (3AXISFLG) to 1

Perform "R60LEM"

Proceed to "ENDEXT"

V89CALL Perform "R02BOTH"

$OPTIONX_0 = 3$

$OPTIONX_1 = 1$

Proceed to "GOFLASH" with TS = K:V04N12 ($OPTIONX_0$, $OPTIONX_1$)
(If terminate, proceed to "ENDEXT", if proceed, continue at next step; if other response, repeat this step.)

V89RECL TSt = TIMENOW + K:DP1MIN

TDEC1 = TSt

Perform "CSMCONIC"

TSr = RATT

TDEC1 = TSt

Perform "LEMCONIC"

TS = [REFSMMAT](TSr - RATT) (adjusted to prevent overflow in unit operation)

POINTVSM = unitTS

If $OPTIONX_1 = 1$, SCAXIS = K:UNITZ and skip next step

SCAXIS = K:UNITX

Perform "VECPOINT"

THETAD = TS

Perform "BALLANGS"

Proceed to "GOFLASH" with TS = K:V06N18 (FDAI)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, proceed to "V89RECL".)

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Perform "R60LEM"

Proceed to "ENDEXT"

CHKLINUS If FLAGWRD4 bit 12 (PDSPFLAG) = 0, return

TBASE2 = return address of last display (prior to call of "CHKLINUS")

Set restart group 2 to phase 13
(causes "RELINUS" to be established with priority 10_8 if restart.)

Perform "BLANKET" with TS = 00100_8

Return

RELINUS (Entered for restart group 2.13, due to "CHKLINUS")

Change job priority to 26 (pr26)

If FLAGWRD1 bit 5 (TRACKFLG) = 1:

Switch FLAGWRD4 bit 12 (PDSPFLAG) to 1

Return to address specified by TBASE2 (to display generated prior to calling of "CHKLINUS")

If FLAGWRD0 bit 7 (RNDVZFLG) = 1: (P20 running)

Set restart group 2 to phase 7 (this will cause "P20LEMC1" to be started in 15 seconds if there is a restart.)

End job

Set restart group 2 to phase 11 and TBASE2 = - TIME1
(causes "P25LEM1" to be established with priority 14_8 if restart.)

End job

1JACCP, 1JACCQ, 1JACCR: See DAPB section.

A: See MATX section.

AM: Angle of rotation, a double precision angle between 0 and $\frac{1}{2}$ (0° and 180°), scaled B0 in units of revolutions.

ANG: See COOR section.

ATTCADR: Double precision address storage set to +0 (single precision) when the attitude maneuver routine (KALCMAN3) is not in use.

ATTPRIO: Priority of job calling "GOMANUR".

BALLEXIT: Single precision octal return address storage.

BCDU: Single precision vector storage for the three "present" gimbal angles throughout an attitude maneuver. The X, Y and Z components contain the outer, inner and middle gimbal angles respectively, scaled B-1 in units of revolutions and stored in two's complement form.

BRATE: Double precision vector containing the desired maneuver rates about each of the three principal axes, scaled B-3 in units of revolutions per second.

CAM: Double precision cosine of the total maneuver angle, scaled B1 and unitless.

CDU: See COOR section.

CDUD: See DAPA section.

COF: A double precision unit vector defining the axis of rotation of the calculated maneuver in terms of present spacecraft coordinates, scaled B1. Extracted from the transformation matrix [MFI] by one of two procedures depending on the magnitude of the rotation angle.

COFSKEW: Double precision calculation of COF from the off-diagonal terms of [MFI]; equivalent to COF or used to supplement a calculation of COF from the diagonal terms of [MFI] by supplying sign information; scaled B1 and unitless.

COSOGA, COSIGA, COSMGA: See COOR section.

COSTH: See COOR section.

DELCDU: See DAPA section.

[DELMAT]: Double precision, 3x3 transformation matrix describing a rotation about COF through a particular angle. Used in "NEWANGL" to update [MIS] at one-second intervals; scaled B0 and unitless.

DELPEROR, DELQEROR, DELREROR: See DAPA section.

FDAI: Single precision vector containing the values expected to appear on the FDAI at the completion of an attitude maneuver (astronaut roll, pitch and yaw in that order), scaled B-1 in units of revolutions.

K:ANGLTIME: Double precision constant stored as 0.0001907349, scaled B19 in units of centiseconds per second. Equation value: 100. (Used with a right shift of 5 to convert from seconds scaled B4 to centiseconds scaled B28.)

K:ARATE$_i$: Table of four double precision constants, scaled B-4 in units of revolutions per second.

i	Stored Value	Equation Value	Equivalent
0	0.0088888888	0.0005555555	0.2 degrees/second
1	0.0222222222	0.0013888888	0.5 degrees/second
2	0.0888888888	0.0055555550	2.0 degrees/second
3	0.4444444444	0.0277777778	10.0 degrees/second

K:BIASCALE: Single precision constant stored as 75777_8, scaled B2 and unitless. Equation value: $\frac{1}{4}$.

K:DP1MIN: Double precision constant stored as 0.0000223517 (6000×2^{-28}), scaled B28 in units of centiseconds. Equation value: one minute.

K:DTdTAU: Single precision constant stored as 0.1, scaled B0 in units of seconds per DAP cycle. Equation value: 0.1.

K:LOCKANGL: Single precision constant stored as 0.388889, scaled B-1 in units of revolutions. Equation value: 0.194445. (Equivalent to 70 degrees.)

K:MAXANG: Double precision constant stored as 0.472222222, scaled B0 in units of revolutions. Equation value: 0.472222222. (Equivalent to 170 degrees.)

K:MINANG: Double precision constant stored as 0.00069375, scaled B0 in units of revolutions. Equation value: 0.00069375. (Equivalent to 0.25 degrees.)

K:ONESEK: Double precision constant stored as 100 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 100.

K:posmaxdp: See list of Major Variables.

K:SINGIMLC: Double precision constant stored as 0.4285836003, scaled B1 and unitless. Equation value: 0.8571672. (Corresponds to the sine of 59 degrees.)

K:SINVEC1: Double precision constant stored as 0.3796356537, scaled B1 and unitless. Equation value: 0.7592713074. (Corresponds to the sine of 49.4 degrees.)

K:SINVEC2: Double precision constant stored as 0.2462117800, scaled B1 and unitless. Equation value: 0.49242356. (Corresponds to the sine of 29.5 degrees.)

K:UNITX, K:UNITY, K:UNITZ: Double precision constant vectors scaled B1 and unitless. Equation value: (1,0,0), (0,1,0) and (0,0,1).

K:VECANG1: Double precision constant stored as 0.1388888889, scaled B0 in units of revolutions. Equation value: 0.1388888889. (Equivalent to 50 degrees.)

K:VECANG2: Double precision constant stored as 0.09722222222, scaled B0 in units of revolutions. Equation value: 0.09722222222. (Equivalent to 35 degrees.)

K:VxxNxx: See list of Major Variables.

[MFI]: Double precision, 3x3 transformation matrix, scaled B2 and defined such that Asc = [MFI] Ascd where A is a vector expressed in terms of "present spacecraft" and "desired spacecraft" coordinate systems, respectively.

[MFISYM]: A double precision, 3x3 matrix synthesized from [MFI] and its transpose to enable an accurate computation of COF from the diagonal terms; scaled B2.

[MFS]: A double precision, 3x3 transformation matrix, scaled B1 and defined such that Asm = [MFS] Ascd where A is a vector expressed in terms of "stable member" and "desired spacecraft" coordinate systems, respectively.

[MIS]: A double precision, 3x3 transformation matrix scaled B1 and defined such that $\underline{A}sm$ = [MIS] $\underline{A}sc$ where $\underline{A}$ is a vector expressed in terms of "stable member" and "present spacecraft" coordinate systems, respectively.

$$[MIS] = \begin{bmatrix} \underline{XSM}_{sc}^{T} \\ \underline{YSM}_{sc}^{T} \\ \underline{ZSM}_{sc}^{T} \end{bmatrix} = \left[\underline{XSC}_{sm}, \underline{YSC}_{sm}, \underline{ZSC}_{sm}\right]$$

$$[MIS]^{t} = \begin{bmatrix} \underline{XSC}_{sm}^{T} \\ \underline{YSC}_{sm}^{T} \\ \underline{ZSC}_{sm}^{T} \end{bmatrix} = \left[\underline{XSM}_{sc}, \underline{YSM}_{sc}, \underline{ZSM}_{sc}\right]$$

MODREG: See DATA section.

$\underline{N}$CDU: Single precision vector storage for the gimbal angles desired after the next one-second period, scaled B-1 in units of revolutions and stored in two's complement form.

NEXTIME: Single precision scheduled time of the next "UPDTCALL" cycle, scaled B14 in units of centiseconds.

OMEGAPD, OMEGAQD, OMEGARD: See DAPA section.

$OPTIONX_0$, $OPTIONX_1$: See EXVB section.

$\underline{P}$OINTVSM: Double precision "desired direction" vector, a unit vector scaled B1 and expressed in terms of "stable member" coordinates.

PRIORITY: See MATX section.

RATEINDX: See DAPB section.

$\underline{R}$ATT: See ORBI section.

[REFSMMAT]: See COOR section.

$\underline{S}$CAXIS: Double precision unit vector defining the spacecraft axis that is to be pointed in the "desired direction", scaled B1 and expressed in "present spacecraft" coordinates.

SINOGA, SINIGA, SINMGA: See COOR section.

SINTH: See COOR section.

TBASE2: Cell used for address storage purposes in "CHKLINUS" to permit restoration of program display after a restart. The cell is single precision and is normally used to contain waitlist restart information (as it is in "RELINUS".)

TDEC1: See ORBI section.

TEMPR60: Single precision octal return address storage.

THETA: See COOR section.

THETAD: Single precision vector containing the gimbal angles that define the desired orientation to which the attitude maneuver routines are to maneuver; scaled B-1 in units of revolutions and stored in two's complement form. Also called CPHI, CTHETA, CPSI in program, (x,y,z components respectively).

TIME1: The least signifigant half of TIMENOW. See EXVB section.

TIMENOW: See EXVB section.

TM: Double precision time of maneuver end, calculated on the first pass through "NEWANGL" and scaled B28 in units of centiseconds.

[TMFI]: Double precision, 3x3 matrix equal to the transpose of [MFI]; scaled B2.

[TMIS]: Double precision, 3x3 matrix equal to the transpose of [MIS]; scaled B1.

VECQTEMP: Single precision octal return address storage.

$\underline{Y}SM_{sc}$, $\underline{Y}SM_{scd}$: Double precision unit vector along the Y-axis of the "stable member" coordinate system expressed in terms of "present spacecraft" and "desired spacecraft" coordinate systems, respectively; scaled B1.

BURN

P40LM	WHICH = "P40TABLE"
	If FLGWRD10 bit 13 (APSFLAG) = 1, proceed to "P40ALM"
	Perform "R02BOTH"
	DVTHRUSH = K:THRESH1 + K:THRESH3
	If DAPBOOLS bit 13 (CSMDOCKD) = 1, DVTHRUSH = K:THRESH3
	DVCNTR = 4
	Switch FLAGWRD5 bit 12 (NOTHROTL) to 0
	F = K:FDPS
	MDOT = K:MDOTDPS
	TDECAY = K:DTDECAY
	VEX = - 2 K:DPSVEX
	Proceed to "P40IN"
P42LM	WHICH = "P42TABLE"
	If FLGWRD10 bit 13 (APSFLAG) = 0, proceed to "P40ALM"
P42STAGE	Perform "R02BOTH"
	DVTHRUSH = K:THRESH2
	DVCNTR = 4
	Switch FLAGWRD2 bit 5 (AVFLAG) to 1 (LM active)
	F = K:FAPS
	MDOT = K:MDOTAPS
	TDECAY = K:ATDECAY
	VEX = - 2 K:APSVEX
P40IN	Perform "S40.1" (get initial target vectors)
	Perform "S40.2,3" (get initial attitude)

Perform "PFLITEDB" with interrupts inhibited

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Perform "R60LEM"

Proceed to "BURNBABY"

<u>P41LM</u> WHICH = "P41TABLE"

Perform "R02BOTH"

If FLAGWRD1 bit 15 (NJETSFLG) = 0, F = K:FRCS4

If FLAGWRD1 bit 15 (NJETSFLG) = 1, F = K:FRCS2

Perform "S40.1" (get initial target vectors)

Perform "S40.2,3" (get initial attitude)

Inhibit interrupts

Perform "ZATTEROR"

Perform "SETMINDB"

Release interrupt inhibit

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Perform "R60LEM" (return after attitude maneuver complete)

Perform "S41.1" with $\underline{T}$Sref = $\underline{V}$GPREV

$\underline{V}$GBODY = $\underline{T}$Sbody

Perform "GODSPRET" with TS = K:V16N85 ($\underline{V}$GBODY)

DISPDEX = 05000_8 (positive to enable "DYNMDISP")

Establish "DYNMDISP" (pr05)

Proceed to the third step of "BURNBABY"

<u>P40ALM</u> Perform "ALARM" with TS = 01706_8

<u>REP40ALM</u> Proceed to "GOFLASH" with TS = K:V05N09
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

Proceed according to fourteenth entry in WHICH table
(-----, REP40ALM, -----, P42STAGE, -----, -----)

```
BURNBABY  DVTOTAL = 0

          Perform "P40AUTO"      (assure proper mode switching)

          GOBLTIME = TIG

          Perform "ENGINOF3" with interrupts inhibited

          Proceed according to the fifth entry in WHICH table
          (P40SPOT, P40SPOT, P41SPOT, P40SPOT, P41SPOT, ---)

P40SPOT   DISPDEX = - 15   ("CLOKTASK" controlled otherwise by P41, P63)

          Perform "STCLOK3"    (start computation of TTOGO)

P41SPOT   TDEC1 = TIG - K:D29.9SEC     ("CLOKTASK" already running if P63)

          Perform "INITCDUW"      (initialize steering)

          If FLAGWRD6 bit 8 (MUNFLAG) = 1:

               Perform "CSMPREC"

               VCSM = [REFSMMAT] VATT

               RCSM = [REFSMMAT] RATT

               Perform "MUNGRAV" with TSr = RCSM

               GCSM = GDT1

               TDEC1 = TAT

          Perform "MIDTOAV1"

          If TSerror = 1:          (did not finish in time)

               TIG = PIPTIME1 + K:D29.9SEC

          SAVET = TSt - K:5SECDP

          Call "TIG-35" in SAVET centiseconds

          If MODREG = 63:

               DISPDEX = -15       (enable display of TTOGO)

               ABVEL = |VN1|

          End job

TIG-35    Call "TIG-30" in 5 seconds
```

DISPDEX = - 2 (cause "CLOKTASK" to blank display)

If the 6th entry in WHICH table < 0: (P41)

Call "TIG-30.1" in 4.9 seconds

Establish "P41BLANK" (pr17)

End task

P41BLANK Perform "CLEANDSP"

End job

TIG-30.1 Establish "TIG-30A" (pr17)

End task

TIG-30A Proceed to "REGODSP" with TS = K:V16N85 (VGBODY)

TIG-30 (Entered at TIG - 29.9 seconds)

Call "TIG-5" in 24.9 seconds

DISPDEX = - 15 (enable TTOGO display)

TSt = 6th entry in WHICH table (0, 2240, -1, 2640, 2240, -----)

If TSt ≤ 0, proceed to "ULLGNOT"

Call "ULLGTASK" in TSt centiseconds

Proceed according to first entry in WHICH table
(ULLGNOT, ULLGNOT,-----, WANTAPS, ULLGNOT, ULLGNOT)

WANTAPS Switch FLGWRD10 bit 13 (APSFLAG) to 1

ULLGNOT AVEGEXIT = 7th entry in WHICH table
(SERVEXIT, STEERING, CALCN85, STEERING, SERVEXIT,-----)

Proceed to "PREREAD"

ULLGTASK Switch DAPBOOLS bit 6 (ULLAGER) to 1

End task

TIG-5 Call "TIG-0" in 5 seconds

Switch FLAGWRD7 bits 13 (IGNFLAG) and 12 (ASTNFLAG) to 0

Switch FLAGWRD5 bit 8 (ZOOMFLAG) to 0

Proceed according to 11th entry in WHICH table
(DISPCHNG, P40SJUNK, End task, P40SJUNK, DISPCHNG, DISPCHNG)

P40SJUNK If "S40.13" is running, proceed to "DISPCHNG" (indicated by restart tables)

Establish "S40.13" (P40's burn initialization) (pr20)

DISPCHNG DISPDEX = - 11 (initiate astronaut branching capability)

End task

*PROCEED Switch FLAGWRD7 bit 12 (ASTNFLAG) to 1

If FLAGWRD7 bit 13 (IGNFLAG) = 1:

Call "IGNITION" in 0.01 seconds

DISPDEX = - 15 (display only)

End job

TIG-0 Switch FLAGWRD7 bit 13 (IGNFLAG) to 1

If MODREG = 63, call "ZOOM" in ZOOMTIME centiseconds

If FLAGWRD7 bit 12 (ASTNFLAG) = 1, proceed to "IGNITION"

Proceed according to the 12th entry in WHICH table
(End task, End task, TIGTASK, End task, End task, End task)

TIGTASK Establish "TIGNOW" (pr16)

Switch DAPBOOLS bit 15 (PULSES) to 0

End task

IGNITION Switch FLAGWRD5 bit 7 (ENGONFLG) to 1

Switch bit 14 of channel 11 to 0

Switch bit 13 of channel 11 to 1 (send ignition command)

TEVENT = TIMENOW

Switch DAPBOOLS bit 15 (PULSES) to 0

TIG = TGO + TIMENOW (now contains cutoff time)

Switch FLAGWRD8 bit 10 (FLUNDISP) to 0

Proceed according to the 13th entry in WHICH table
(P12IGN, P40IGN, ---, P42IGN, P63IGN, ABRTIGN)

P12IGN AOSQ = IGNAOSQ (Initialize DAP bias acceleration estimates)

AOSR = IGNAOSR

ABRTIGN DISPDEX = Z (positive to kill "CLOKTASK")

AVEGEXIT = "ATMAG"

Switch FLAGWRD7 bit 11 (SWANDISP) to 1

Proceed to "P42IGN"

P63IGN AVEGEXIT = "LUNLAND"

DISPDEX = Z (positive to kill "CLOKTASK")

Switch FLAGWRD9 bit 9 (LETABORT) to 1

Switch FLAGWRD7 bit 11 (SWANDISP) to 1

TIG = TIMENOW

WCHPHASE = 0

WCHPHOLD = 0

FLPASS0 = 2

Proceed to "P42IGN"

P40IGN If FLAGWRD5 bit 12 (NOTHROTL) = 0:

Call "ZOOM" in ZOOMTIME centiseconds

P42IGN Switch DAPBOOLS bit 8 (DRIFTBIT) to 0

If FLAGWRD2 bit 9 (IMPULSW) = 1:

If TGO $\leq$ 0, TGO = 1

TGO = 2^{14} (fractional part of TGO / 2^{14}) (more significant half zeroed)

Call "ENGOFTSK" in TGO centiseconds

Switch FLAGWRD7 bits 13 (IGNFLAG) and 12 (ASTNFLAG) to 0

Switch FLAGWRD2 bit 9 (IMPULSW) to 0

Delay 0.5 second

Switch DAPBOOLS bit 6 (ULLAGER) to 0

End task

Switch FLAGWRD7 bits 13 (IGNFLAG) and 12 (ASTNFLAG) to 0

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 0

Delay 0.5 second

Switch DAPBOOLS bit 6(ULLAGER) to 0

End task

ZOOM If MODREG = 63: (If P63 running)

Proceed to "P63ZOOM" (otherwise it's P40)

P40ZOOM THRUST = K:MAXTHRUST

Switch bit 4 of channel 14 to 1

End task

P63ZOOM Switch FLAGWRD5 bit 8 (ZOOMFLAG) to 1

Perform "FLATOUT"

End task

STEERING Perform "UPDATEVG"

If FLAGWRD2 bit 9 (IMPULSW) = 0, proceed to "SERVEXIT"

If FLAGWRD7 bit 7 (IDLEFLAG) = 1, proceed to "SERVEXIT"

Perform "STOPRATE"

Switch FLAGWRD2 bit 9 (IMPULSW) to 0

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 1

Inhibit interrupts

TSt = TIG - TIMENOW

If TSt $\leq$ 0, TSt = 1

TGO = 2^{14} (fractional part of TSt / 2^{14}) (more significant half zeroed)

Call "ENGOFTSK" in TGO centiseconds

Release interrupt inhibit

End job

ENGOFTSK Perform "ENGINOFF"

End task

ENGINOFF Establish "POSTBURN" (pr12)

ENGINOF2 Call "COASTSET" in 0.01 second

ENGINOF1 Switch FLAGWRD7 bit 7 (IDLEFLAG) to 1

Switch DAPBOOLS bit 6 (ULLAGER) to 0

ENGINOF4 TEVENT = TIMENOW

ENGINOF3 Switch FLAGWRD5 bit 7 (ENGONFLG) to 0

Switch bit 13 of channel 11 to 0

Switch bit 14 of channel 11 to 1

Switch DAPBOOLS bit 14 (USEQRJTS) to 1

THRUST = - K:MAXTHRUST (wipe out any throttle setting above that specified manually)

Switch bit 4 of channel 14 to 1

Return

COASTSET Perform "ALLCOAST"

End task

POSTBURN DISPDEX = Z (positive to kill "CLOKTASK")

AVEGEXIT = "CALCN85"

Perform "GOFLASHR" with TS = K:V16N40 (TTOGO, DELVSAB, DVTOTAL)
(If terminate, proceed to "TERM40"; if proceed, proceed to "TIGNOW"; if other response, proceed to "POSTBURN".)

End job

TIGNOW Inhibit interrupts

Perform "ZATTEROR"

Perform "SETMINDB"

Release interrupt inhibit

Perform "REFLASHR" with TS = K:V16N85 (VGBODY)
(If terminate, proceed to "TERM40"; if proceed, proceed to "TERM40"; if other response, repeat this step.)

End job

TERM40 AVEGEXIT = "SERVEXIT"

TRKMKCNT = 0

DISPDEX = Z (positive to kill "CLOKTASK")

Perform "RESTORDB" with interrupts inhibited

Proceed to "GOTOPOOH"

STCLOK3 TSt = TIG - TIMENOW (modulo 2^{19} centiseconds)

TSt = 2 + 100 (fractional part of TSt / 100)

(If TIG - TIMENOW were 4723 centiseconds, TSt would be 25 centiseconds, starting "CLOKTASK" at TIG - 46.98 seconds.)

Call "CLOKTASK" in TSt centiseconds

Return

COMPTGO DISPDEX = 0 (compute TTOGO but do not display)

Call "CLOKTASK" in 0.02 second

Return

CLOKTASK If DISPDEX > 0, end task

Establish "CLOKJOB" (pr27)

Delay 1 second

Proceed to "CLOKTASK"

CLOKJOB TTOGO = TIMENOW - TIG

If DISPDEX = - 29:

NVWORD1 = -0 (to specify a verb 97 paste)

Proceed to "CLOCPLAY" with TS = $NVWORD_2$

(If terminate, proceed to "STOPCLOK"; if proceed, proceed according to the second entry in WHICH table: COMFAIL3, COMFAIL4,-----, COMFAIL4, COMFAIL3, COMFAIL3; if other response, proceed to "COMFAIL2".)

If DISPDEX = - 21:

Proceed to "REFLASH" with TS = K:V06N61
(If terminate, proceed to "STOPCLOK"; if proceed, continue at next step; if other response, repeat this step.)

DISPDEX = 0

Establish "ASTNRET" (pr13)

End job

If DISPDEX = - 15:

Proceed to "REGODSP" with TS = zero entry in WHICH table (V06N74, V06N40, ---, V06N40, V06N62, V06N63)

If DISPDEX = - 11:

V99RECYC NVWORD1 = 77377_8 (to specify a verb 99 paste)

Proceed to "CLOCPLAY" with TS = zero entry in WHICH table (V06N74, V06N40, ----, V06N40, V06N62, V06N63)
(If terminate, proceed to "STOPCLOK"; if proceed proceed to "*PROCEED"; if other response, proceed to "*ENTER".)

If DISPDEX = -2:

Perform "CLEANDSP"

End job

STOPCLOK Switch DAPBOOLS bit 6 (ULLAGER) to 0

Remove "ULLGTASK" from the waitlist if it is there

DISPDEX = Z (positive to kill "CLOKTASK")

Proceed to "GOTOPOOH"

*ENTER Inhibit interrupts

Proceed according to the 3rd entry in WHICH table (GOCUTOFF, GOPOST, ---, GOPOST, V99RECYC, GOCUTOFF)

GOPOST Establish "POSTBURN" (pr12)

Inhibit interrupts

Perform "ALLCOAST"

Switch DAPBOOLS bit 6 (ULLAGER) to 0

Remove "ULLGTASK" from the waitlist if it is there

DISPDEX = Z (positive to kill "CLOKTASK")

Release interrupt inhibit

End job

GOCUTOFF Establish "CUTOFF" (pr17)

Switch FLAGWRD8 bit 10 (FLUNDISP) to 0

Inhibit interrupts

Perform "ALLCOAST"

Switch DAPBOOLS bit 6 (ULLAGER) to 0

Remove "ULLGTASK" from the waitlist if it is there

DISPDEX = Z (positive to kill "CLOKTASK")

Release interrupt inhibit

End job

COMFAIL Switch FLAGWRD7 bit 7 (IDLEFLAG) to 1

Switch FLAGWRD8 bit 10 (FLUNDISP) to 1

DVCNTR = 4

If DISPDEX > 0:

    DISPDEX = 0

    Perform "STCLOK3"

DISPDEX = - 29 (initiate fail branch display)

End job

COMFAIL2 Inhibit interrupts

Remove "ZOOM" from the waitlist if it is there

Perform "ENGINOF4"

Switch DAPBOOLS bit 8 (DRIFTBIT) to 1

Invert DAPBOOLS bit 10 (AORBTRAN)

Switch DAPBOOLS bit 6 (ULLAGER) to 1

Call "TIG-5" in 0.01 second

End job

COMFAIL3 DISPDEX = Z (positive to kill "CLOKTASK")

Skip next step

COMFAIL4 DISPDEX = - 15

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 0

Switch FLAGWRD8 bit 10 (FLUNDISP) to 0

End job

DYNMDISP If DISPDEX ≤ 0, end job ("CLOKTASK" has started)

Perform "S41.1" with TSref = VGPREV

VGBODY = TSbody

Delay 1 second

Proceed to "DYNMDISP"

CALCN85 Perform "UPDATEVG"

Perform "S41.1" with TSref = VGPREV

VGBODY = TSbody

Proceed to "SERVEXIT"

S41.1 Perform "CDUTRIG"

Perform "SMTONB"

TSbody = [SMNBMAT] [REFSMMAT] TSref

Return

P40AUTO If FLGWRD10 bit 13 (APSFLAG) = 1 and bit 10 of channel 30 = 0 (PGNCS) and bit 14 of channel 31 = 0 (DAP in Auto mode) or if FLGWRD10 bit 13 (APSFLAG) = 0 and bit 10 of channel 30 = 0 and bit 14 of channel 31 = 0 and bit 5 of channel 30 = 0 (Auto throttle mode also required if descent):

Return (switch configuration proper)

Proceed to "GOPERF1" with TS = 00203_8 (request proper moding)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "P40AUTO"; if other response, continue at next step.)

Return

P47LM Perform "R02BOTH"

Perform "MIDTOAV2"

Call "STARTP47" in TSt centiseconds (TSt modulo 2^{14})

End job

STARTP47 AVEGEXIT = "CALCN83"

Establish "P47BODY" (pr20)

Proceed to "PREREAD"

P47BODY DELVIMU = 0

DELVCTL = 0

Perform "GOFLASHR" with TS = K:V16N83 (DELVIMU)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "GOTOPOOH"; if other response, proceed to "P47BODY".)

End job

CALCN83 DELVSIN = DELVCTL + DELVREF

Perform "S41.1" with TSref = DELVSIN

DELVIMU = TSbody

DELVCTL = DELVSIN

Proceed to "SERVEXIT"

UPDATEVG QTEMP1 = return address

Perform "S40.8"

If FLAGWRD2 bit 8 (XDELVFLG) = 1, return via QTEMP1

If FLAGWRD7 bit 10 (NORMSW) = 0:

  TS = PIPTIME - TIGSAVE - TNEWA

  If TS < 0, proceed to "GETRANS"

  TStp = TIGSAVE + TNEWA

If "S40.9" is still active: (as indicated by restart tables)

  Return via QTEMP1

Establish "S40.9" (pr10)

TIGSAVE = TStp

RINIT = RN

VINIT = VN

GETRANS DELLT4 = TPASS4 - PIPTIME

Return via QTEMP1

S40.8 If FLAGWRD2 bit 8 (XDELVFLG) = 0, proceed to "RASTEER1"

TS1 = VGPREV - DELVREF

VGAIN* VG = TS1

UNFC = [REFSMMAT] VG

DELVSAB = |VG|

VGPREV = VG

If FLAGWRD2 bit 11 (STEERSW) = 0, return

TS = unit(-DELVREF) • VG

If TS ≥ 0:

Perform "ALARM" with TS = 01407_8

Perform "FINDCDUW"

Return (to caller of "S40.8")

TS1 = 1 + (TS / VEX)

TGO = (K:mFOURDT TS TS1 / |$\underline{D}$ELVREF|) + TDECAY

TIG = TGO + PIPTIME

If TGO ≥ K:FOURSECS:

Perform "FINDCDUW"

Return (to caller of "S40.8")

Switch FLAGWRD2 bit 9 (IMPULSW) to 1

Switch FLAGWRD2 bit 11 (STEERSW) to 0

Return (to caller of "S40.8")

S40.1 QTEMP = return address

TIGSAVE = TIG

If FLAGWRD2 bit 8 (XDELVFLG) = 0, proceed to "S40.1B"

$\underline{V}$INIT = $\underline{V}$TIG

TS = |$\underline{V}$TIG * $\underline{R}$TIG|

$\underline{U}$T = unit($\underline{V}$TIG * $\underline{R}$TIG)

$\underline{R}$INIT = $\underline{R}$TIG

TSang = (TS K:THETACON / |$\underline{R}$TIG|^2) (DELVSAB MASS / F)

$\underline{T}$Sa = ($\underline{D}$ELVSIN · $\underline{U}$T) $\underline{U}$T

$\underline{T}$Sb = $\underline{D}$ELVSIN - $\underline{T}$Sa

$\underline{V}$GPREV = $\underline{T}$Sa + |$\underline{T}$Sb| (sinTSang unit($\underline{T}$Sb * $\underline{U}$T) + cosTSang unit$\underline{T}$Sb)

$\underline{U}$T = unit$\underline{V}$GPREV

Perform "GET.LVC" with TS = VGPREV

Return via QTEMP

S40.1B TDEC1 = TIG

DELLT4 = TPASS4 - TIG

Perform "LEMPREC"

RTIG = RATT and RINIT = RATT

UNITR = unitRATT

VTIG = VATT

VINIT = VATT

VTARGTAG = 0

CNANGL = K:EPS1

If FLAGWRD7 bit 10 (NORMSW) = 1, CNANGL = K:EPS1 + K:EPS2

RTX1 = MUDEX

RTX2 = PBODY

Perform "INITVEL"

VGPREV = DELVEET3

UT = unitVGPREV

DELVSAB = | VGPREV |

Perform "GET.LVC" with TS = DELVEET3

Return via QTEMP

S40.2.3 POINTVSM = [REFSMMAT] UT

SCAXIS = K:UNITX

XSMDrf = UT

TSa = unit(XSMDrf * RTIG)

$TS_{dp} = |\underline{X}SMDrf * \underline{R}TIG|$

If the most significant half of TS_{dp} = 0:

$\underline{T}Sa = unit(\underline{X}SMDrf * \underline{V}TIG)$

$\underline{Y}SMDrf = \underline{T}Sa$

$\underline{Z}SMDrf = - \underline{Y}SMDrf * \underline{X}SMDrf$

Switch FLAGWRD2 bit 4 (PFRATFLG) to 1

Return

S40.9 Switch FLAGWRD2 bit 5 (AVFLAG) to 1

CNANGL = K:EPS1

If FLAGWRD7 bit 10 (NORMSW) = 1, CNANGL = K:EPS1 + K:EPS2

Perform "HAVEGUES" with VTARGTAG = 0

End job

RASTEER1 $RMAG = |\underline{R}N|$

$\underline{T}Suc = unit(\underline{R}TARG - \underline{R}N)$

$TSc = |\underline{R}TARG - \underline{R}N|$ (quasi floating point)

TSr1c = TSc RMAG (quasi floating point)

TSss = RMAG + RTMAG + TSc) / 2

TS1 = (MUASTEER - (TSss - TSc) MUdA / 2

$TSa = \left[2\ (TSss - RMAG)\ TS1\ /\ TSr1c\right]^{\frac{1}{2}}$ signGEOMSGN

TS1 = MUASTEER - TSss MUdA / 2

$TSb = \left[2\ (TSss - RTMAG)\ TS1\ /\ TSr1c\right]^{\frac{1}{2}}$

$TS1 = \left[(TSss - TSc)\ /\ TSss\right]^{\frac{1}{2}}$

$TS1 = K{:}2PI{+}3\ \arcsin TS1 - (TSc\ /\ TSss)^{\frac{1}{2}}\ (TS1)$

TS1 = TS1 signGEOMSGN

$TS = \left[TSss^3\ /\ (2\ MUASTEER)\right]^{\frac{1}{2}}\ (K{:}2pi{+}1 - TS1) - TPASS4 + PIPTIME$

TS = TSb signTS

If FLAGWRD7 bit 10 (NORMSW) = 1:

If TSuc • UNITR < 0, proceed to "NEGPROD"

TS1 = unit(UNITR + TSuc)

TS2 = unit((- TS1 * UN) signGEOMSGN) TS

VIPRIME = TS1 TSa + TS2

DELVEET3 = VIPRIME - VN1

Proceed to "FIRSTTME"

TS1 = unit(TSuc - UNITR) TS

VIPRIME = unit(TSuc + UNITR) TSa + TS1

DELVEET3 = VIPRIME - VN1

Proceed to "FIRSTTME"

NEGPROD TS1 = unit(TSuc - UNITR)

TS2 = unit((TS1 * UN) signGEOMSGN) TSa

VIPRIME = TS1 TS + TS2

DELVEET3 = VIPRIME - VN1

FIRSTTME If RTX2 = 0: (earth centered orbit)

TS = UNITGOBL (PIPTIME - GOBLTIME) K:EARTHMU / $|RN|^2$

DELVEET3 = DELVEET3 + TS

TS1 = DELVEET3 and skip next step

TS1 = DELVEET3

Proceed to "VGAIN*"

S40.13 Switch FLAGWRD2 bit 9 (IMPULSW) to 0

TS = |VGPREV| - K:4SEC K:FRCS2 / MASS

If FLGWRD10 bit 13 (APSFLAG) = 0, proceed to "S40.13D"

Switch FLAGWRD5 bit 12 (NOTHROTL) to 1

TSa = TS - (K:K1VAL / MASS)

If TSa < 0:

    TSt = (TS MASS + K:K2VAL) / K:K3VAL

    Proceed to "S40.132"

TSb = F K:5SECS / (MASS - MDOT K:3.5SEC)

TSc = TSa - TSb

If TSc ≥ 0, proceed to "S40.13D" (TGO ≥ 6 seconds)

TSt = K:1SEC2D + K:5SECS TSa / TSb

Proceed to "S40.132"

S40.13D TS = TS MASS

If FLGWRD10 bit 13 (APSFLAG) = 1:

    TSt = TS / K:FAPS

    Proceed to second step of "S40.132"

TSt = TS / K:S40.136

Switch FLAGWRD5 bit 12 (NOTHROTL) to 0

If overflow ($TSt \geq 2^{14}$):

    TGO = TS / K:S40.136*

    End job

If TSt < K:6SEC: (TGO < 6 seconds)

    Proceed to "S40.132"

If TSt < (K:6SEC + K:89SECS): (TGO < 95 seconds)

    Switch FLAGWRD5 bit 12 (NOTHROTL) to 1

Proceed to the second step of "S40.132"

S40.132 Switch FLAGWRD2 bit 9 (IMPULSW) to 1

TGO = (0, TSt_{ms})

End job

INITCDUW OGABIAS = 0

UNFV = K:UNITX

UNWC = K:UNITX

Return

FINDCDUW TSnewthrust = UNFC

QCDUWUSR = return address

NDXCDUW = bit 13 of DAPBOOLS (CSMDOCKD) (1 or 0)

FLPAUTNO = 1

FLAGOODW = bit 9 of DAPBOOLS (XOVINHIB) (1 or 0)

Inhibit interrupts

ANG = CDU

If bit 10 of channel 30 = 0 and bit 14 of channel 31 = 0:
(PGNCS control; DAP in Auto mode)

FLPAUTNO = 0

ANG = CDUD

Release interrupt inhibit

UNX = unitTSnewthrust

UNZ = unitUNWC

(argument of unit operation adjusted to reduce the possibility of overflow)

Perform "QUICTRIG"

If overflow (in either unit operation above), proceed to "NOATTCNT"

TS = unitDELV

If no overflow ($|DELV| \geq 2^{-7}$ cm/sec):

Perform "SMTONB"

TSdv = [SMNBMAT] TS

TS = $(TSdv_y - UNFV_y)$ $K{:}GAINFLTR_{NDXCDUW}$

If $|TS| >$ K:DUNFVLIM, TS = K:DUNFVLIM signTS

(If no overflow:)

$UNFV_y = UNFV_y + TS$

If $|UNFV_y| > K{:}UNFVLIM$, $UNFV_y = K{:}UNFVLIM \; signUNFV_y$

$TS = (TSdv_z - UNFV_z) \; K{:}GAINFLTR_{NDXCDUW}$

If $|TS| > K{:}DUNFVLIM$, $TS = K{:}DUNFVLIM \; signTS$

$UNFV_z = UNFV_z + TS$

If $|UNFV_z| > K{:}UNFVLIM$, $UNFV_z = K{:}UNFVLIM \; signUNFV_z$

If FLAGOODW = 1:

If $(\underline{UNZ} \cdot \underline{UNX})^2 < K{:}DOTSWFMX$:

Proceed to "DCMCL"

FLAGOODW = 0

$\underline{UNZ} = \underline{ZNBPIP}$

If $(\underline{UNZ} \cdot \underline{UNX})^2 < K{:}DOTSWFMX$:

Proceed to "DCMCL"

FLAGOODW = 0

$\underline{UNZ} = - \underline{XNBPIP}$

DCMCL $\underline{UNY} = unit(\underline{UNZ} * \underline{UNX})$

$\underline{UNZ} = \underline{UNY} * \underline{UNX}$

$\underline{UNX} = unit(\underline{UNX} + UNFV_z \, \underline{UNZ} - UNFV_y \, \underline{UNY})$

$\underline{UNY} = \underline{UNX} * \underline{UNZ}$

$\underline{UNZ} = - \underline{UNY} * \underline{UNX}$

Perform "NB2CDUSP"

$TScdu_x = TScdu_x + OGABIAS$

If $|TScdu_z| > K{:}CDUZDLIM$:

$TScdu_z = K{:}CDUZDLIM \; signTScdu_z$

Perform "ALARM" with TS = 00401_8

Inhibit interrupts

$\underline{THETAD} = \underline{TScdu}$

$\underline{mDELGMB} = -(\underline{TScdu} - \underline{CDUD})$

If $mDELGMB_y{}^2 + K{:}HI5 > 0$, FLAGOODW = 0

If $mDELGMB_z{}^2 + K{:}HI5 > 0$, FLAGOODW = 0

If FLPAUTNO > 0 or if FLAGWRD5 bit 7 (ENGONFLG) = 0:

Proceed to the second step of "NOATTCNT"

i = NDXCDUW

If $|mDELGMB_z| > K{:}DAZMAX_i$, $mDELGMB_z = K{:}DAZMAX_i \; sign(mDELGMB_z)$

$TS = mDELGMB_y \; COSMGA$

If $|TS| > K{:}DAYd2MAX_i$, $TS = K{:}DAYd2MAX_i \; signTS$

$TSa = mDELGMB_y$

$mDELGMB_y = TS / COSMGA$

$TS = - SINMGA \; TSa - mDELGMB_x$

If $|TS| > K{:}DAXMAX_i$, $TS = K{:}DAXMAX_i \; signTS$

$mDELGMB_x = - TS$

If FLAGOODW = 0, $mDELGMB_x = 0$

$mDELGMB_x = mDELGMB_x - SINMGA \; mDELGMB_y$

$OMEGAPD = K{:}dvtoacc \; (- mDELGMB_x - SINMGA \; mDELGMB_y)$

$OMEGAQD = K{:}dvtoacc \; (- COSOGA \; COSMGA \; mDELGMB_y - SINOGA \; mDELGMB_z)$

$OMEGARD = K{:}dvtoacc \; (SINOGA \; COSMGA \; mDELGMB_y - COSOGA \; mDELGMB_z)$

$\underline{DELCDU} = K{:}DTdDELT \; \underline{mDELGMB}$ (converted to two's comp. form)

$TS = |OMEGARD| \; OMEGARD \; K{:}biascale / 1JACCR$

If $|TS| > K{:}DELERLIM$, $TS = K{:}DELERLIM \; signTS$

DELREROR = TS

TS = $|OMEGAQD|$ OMEGAQD K:biascale / 1JACCQ

If $|TS| >$ K:DELERLIM, TS = K:DELERLIM signTS

DELQEROR = TS

TS = $|OMEGAPD|$ OMEGAPD K:biascale / 1JACCP

If $|TS| >$ K:DELERLIM, TS = K:DELERLIM signTS

DELPEROR = TS

Release interrupt inhibit

Return via QCDUWUSR

NOATTCNT Perform "ALARM" with TS = 00402_8

Perform "STOPRATE" with interrupts inhibited

Return via QCDUWUSR

QUICTRIG Inhibit interrupts

(single precision without converting first to one's complement form; error is tolerated for speed.)

SINMGA = $\sin_{sp} ANG_z$

COSMGA = $\cos_{sp} ANG_z$

SINIGA = $\sin_{sp} ANG_y$

COSIGA = $\cos_{sp} ANG_y$

SINOGA = $\sin_{sp} ANG_x$

COSOGA = $\cos_{sp} ANG_x$

Release interrupt inhibit

Return

NB2CDUSP TS = $1 - UNX_y^2$

If TS < 0, TS = 0

TScosmga = $\sqrt{TS}$

If TScosmga ≥ 1, TScosmga = K:posmaxsp

TScos = TScosmga

$TSsin = UNX_y$

Perform "ARCTRGSP"

$TScdu_z = TSang$

$TS = TScosmga - |UNX_x|$

If $TS > 0$, $TS = UNX_x / TScosmga$

$TScos = TS$

$TS = TScosmga - |UNX_z|$

If $TS > 0$, $TS = UNX_z / TScosmga$

$TSsin = - TS$

Perform "ARCTRGSP"

$TScdu_y = TSang$

$TS = TScosmga - |UNY_y|$

If $TS > 0$, $TS = UNY_y / TScosmga$

$TScos = TS$

$TS = TScosmga - |UNZ_y|$

If $TS > 0$, $TS = UNZ_y / TScosmga$

$TSsin = - TS$

Perform "ARCTRGSP"

$TScdu_x = TSang$

Return

<u>ARCTRGSP</u> If $TSsin = 0$:

  If $TScos \geq 0$, $TSang = 0$

  If $TScos < 0$, $TSang = - K{:}posmaxsp$

  Return

$TSsec = TScos / TSsin$

If $|TSsec| \geq 1$:

    If TScos $\geq$ +0, TSang = 0

    If TScos $\leq$ -0:

        TSsin = - TSsin

        TSang = - K:posmaxsp

    TSang = $\arcsin_{sp}$(TSsin) - TSang   (converted to two's comp. form)

    Return

TSang = $\frac{1}{2}$ + $\arcsin_{sp}$(- TScos)   (converted to two's comp. form)

If TSsin < 0, TSang = - TSang

Return

1JACCP, 1JACCQ, 1JACCR: See DAPB section.

ABVEL: See SERV section. (Displayed by nouns 62 and 63 in "CLOKJOB".)

ANG: See COOR section.

ACSQ, AOSR: See DAPA section.

AVEGEXIT: See SERV section.

CDU: See IMUC section.

CDUD: See DAPA section.

CNANGL: See TRGL section.

COSIGA, COSMGA, COSOGA: See COOR section.

DAPBOOLS, DELCDU: See DAPA section.

DELLT4: See TRGL section.

DELPEROR, DELQEROR, DELREROR: See DAPA section.

DELV: See SERV section.

DELVCTL: Double precision vector sum of velocity gained since the initiation of the Delta-v monitor program, P47, scaled B7 in units of meters per centisecond and expressed in the Reference coordinate system.

DELVEET3: See TRGL section.

DELVIMU: Double precision vector equivalent to DELVCTL, but expressed in the Body coordinate system for display.

DELVREF: Double precision sensed-change-in-velocity vector, scaled B7 in units of meters per centisecond and expressed in the Reference coordinate system.

DELVSAB: Double precision magnitude of velocity to be gained, program notation also VGDISP, scaled B7 in units of meters per centisecond; generated in the External Delta-V targeting routines and in the Burn routines.

DELVSIN: Double precision vector scaled B7 in units of meters per centisecond and expressed in the Reference coordinate system. Represents the velocity-to-be-gained vector generated by the External Delta-V targeting routines during P40, P41 and P42. Represents the total accumulated change-in-velocity during P47.

DISPDEX: Single precision index controlling the function of "CLOKJOB" and "CLOKTASK", scaled B14 and unitless. "CLOKJOB" and "CLOKTASK" operate semi-independently of the guidance programs and the primary interface between them and guidance is DISPDEX.

DVCNTR, DVTHRUSH: See SERV section.

DVTOTAL: See SERV section. (Displayed by nouns 40 and 62 in "CLOKJOB".)

F: Double precision thrust expected during the burn, scaled B7 in units of kilogram meters per centisecond squared.

FLAGOODW: Single precision flag set or reset on every pass through "FINDCDUW" to indicate whether steering is or is not based on the desired window pointing vector; scaled B6 and unitless.

FLPASSO: See DESC section.

FLPAUTNO: Single precision flag set to indicate that the burn is not under automatic control and reset to indicate that the DAP control quantities are to be calculated, scaled B6 and unitless.

$\underline{G}$CSM, $\underline{G}$DT1: See SERV section.

GEOMSGN: See TRGL section.

GOBLTIME: Double precision storage for TIG, scaled B28 in units of centiseconds; used to bias the velocity-to-be-gained vector to offset the effect of gravity during an extended Lambert burn.

IGNAOSQ, IGNAOSR: Single precision initial DAP bias acceleration estimates, scaled B-2 in units of revolutions per second squared; a pad loaded quantity.

K:1SEC2D: Double precision constant stored as 100×2^{-14}, scaled B14 in units of centiseconds. Equation value: 100.

K:2pi+1: Double precision constant stored as $3.141592653 \times 2^{-2}$, scaled B1 in units of radians. Equation value: $\pi / 2$. Program notation: 2PI+3

K:2PI+3: Double precision constant stored as $3.141592653 \times 2^{-2}$, scaled B3 in units of radians per revolution. Equation value: 2π.

K:3.5SEC: Double precision constant stored as 350×2^{-13}, scaled B13 in units of centiseconds. Equation value: 350.

K:4SEC: Double precision constant stored as 400×2^{-17}, scaled B17 in units of centiseconds. Equation value: 400.

K:5SECDP: Double precision constant stored as 500×2^{-28}, scaled B28 in units of centiseconds. Equation value: 500.

K:5SECS: Double precision constant stored as 500×2^{-14}, scaled B14 in units of centiseconds. Equation value: 500.

K:6SEC: Double precision constant stored as 600 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 600.

K:89SECS: Double precision constant stored as 8900 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 8900.

K:APSVEX: Single precision constant stored as - 30.30 X 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: - 30.30

K:ATDECAY: Double precision constant stored as - 18. X 2^{-28}, scaled B28 in units of centiseconds. Equation value: - 18.

K:biascale: Single precision constant stored as 02000_8, scaled B2 and unitless; program notation BIT11. Equation value: 0.25

K:CDUZDLIM: Single precision constant stored as 0.3888888888, scaled B-1 in units of revolutions. Equation value: 0.1944444444 (Equivalent to 70 degrees.)

K:D29.9SEC: Double precision constant stored as 2990. X 2^{-28}, scaled B28 in units of centiseconds. Equation value: 2990.

K:DAXMAX$_0$: Single precision constant stored as 0.11111111111, scaled B-1 in units of revolutions. Equation value: 0.05555555555 (Equivalent to 20 degrees.)

K:DAXMAX$_1$: Single precision constant stored as 0.0111111111 , scaled B-1 in units of revolutions. Equation value: 0.00555555555 (Equivalent to 2 degrees.)

K:DAYd2MAX$_0$: Single precision constant stored as 0.05555555555, scaled B0 in units of revolutions. Equation value: 0.05555555555 (Equivalent to 20 degrees.)

K:DAYd2MAX$_1$: Single precision constant stored as 0.0055555555, scaled B0 in units of revolutions. Equation value: 0.0055555555 (Equivalent to 2 degrees.)

K:DAZMAX$_0$: Single precision constant identical to K:DAXMAX$_0$.

K:DAZMAX$_1$: Single precision constant identical to K:DAXMAX$_1$.

K:DELERLIM: Single precision constant stored as 0.05555555555, scaled B-1 in units of revolutions. Equation value: 0.02777777777 (Equivalent to 10 degrees.)

K:DOTSWFMX: Single precision constant stored as 0.93302 X 2^{-4}, scaled B4 and unitless. Equation value: 0.93302 (Equivalent to the square of the cosine of 15 degrees.)

K:DPSVEX: Single precision constant stored as - 29.5588868 x 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: - 29.5588868.

K:DTdDELT: Single precision constant stored as 0.05, scaled B0 in units of guidance cycles per DAP cycle. Equation value: 0.05

K:DTDECAY: Double precision constant stored as - 38. X 2^{-28}, scaled B28 in units of centiseconds. Equation value: - 38.

K:DUNFVLIM: Single precision constant stored as 0.007 x 2^{-1}, scaled B1 and unitless. Equation value: 0.007.

K:dvtoacc: Constant implicit in the 2-second navigation cycle, scaled B-1 in units of seconds to the minus one power. Equation value: $\frac{1}{2}$.

K:EARTHMU: Double precision constant stored as -3.986032 E10 x 2^{-36}, scaled B36 in units of meters cubed per centisecond squared. Equation value: -3.986032 E10.

K:EPS1: Double precision constant stored as 2.777777778 E-2, scaled B0 in units of revolutions. Equation value: 2.777777778 E-2. (Equivalent to 10 degrees.)

K:EPS2: Double precision constant stored as 9.722222222 E-2, scaled B0 in units of revolutions. Equation value: 9.722222222 E-2. (Equivalent to 35 degrees.)

K:FAPS: Double precision constant stored as 1.5569 x 2^{-7}, scaled B7 in units of kilogram meters per centisecond squared. Equation value: 1.5569 (Equivalent to 3500 pounds force.)

K:FDPS: Double precision constant stored as 4.3670 x 2^{-7} , scaled B7 in units of kilogram meters per centisecond squared. Equation value: 4.3670. (Equivalent to 9817.5 pounds force.)

K:FOURSECS: Double precision constant stored as 400 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 400.

K:FRCS2: Double precision constant stored as 0.08896 x 2^{-7}, scaled B7 in units of kilogram meters per centisecond squared. Equation value: 0.08896 . (Equivalent to 200 pounds force.)

K:FRCS4: Double precision constant stored as 0.17792 x 2^{-7}, scaled B7 in units of kilogram meters per centisecond squared. Equation value: 0.17792 . (Equivalent to 400 pounds force.)

K:$GAINFLTR_0$: Single precision constant stored as 0.2, scaled B0 and unitless. Equation value: 0.2

K:$GAINFLTR_1$: Single precision constant stored as 0.1, scaled B0 and unitless. Equation value: 0.1

K:HI5: Single precision constant stored as 76000_8, scaled B-2 in units of revolutions squared. Equation value: -0.015625. (Equivalent to about minus the square of 45 degrees.)

K:K1VAL: Double precision constant stored as 124.55 x 2^{-23}, scaled B23 in units of kilogram meters per centisecond. Equation value: 124.55.

K:K2VAL: Double precision constant stored as 31.138 x 2^{-24}, scaled B24 in units of kilogram meters per centisecond. Equation value: 31.138.

K:K3VAL: Double precision constant stored as 1.5569×2^{-10}, scaled B10 in units of kilogram meters per centisecond squared. Equation value: 1.5569

K:MAXTHRUST: Single precision constant stored as 10000_8, scaled B14 in units of DPS throttle pulses. Equation value: 4096. (See THRUST).

K:MDOTAPS: Double precision constant stored as 0.05135×2^{-3}, scaled B3 in units of kilograms per centisecond. Equation value: 0.05135 (Equivalent to 11.32 pounds mass per second.)

K:MDOTDPS: Double precision constant stored as 0.148×2^{-3}, scaled B3 in units of kilograms per centisecond. Equation value: 0.148

K:mFOURDT: Double precision constant stored as $- 800. \times 2^{-18}$, scaled B16 in units of centiseconds. Equation value: - 200.

K:posmaxsp: see Major Variables.

K:S40.136: Double precision constant stored as 0.4671×2^{-9}, scaled B9 in units of kilogram-meters per centisecond squared. Equation value: 0.4671 (Equivalent to 1050. pounds force).

K:S40.136*: Double precision constant stored as 0.4671×2^{1}, scaled B-1 in units of kilogram-meters per centisecond squared. Equation value: 0.4671

K:THETACON: Double precision constant stored as 0.31830989×2^{-8}, scaled B6 in units of revolutions per radian. Equation value: 1/4 .

K:THRESH1: Single precision constant stored as $24. \times 2^{-14}$, scaled B14 in units of centimeters per second. Equation value: 24.

K:THRESH2: Single precision constant stored as $308. \times 2^{-14}$, scaled B14 in units of centimeters per second. Equation value: 308.

K:THRESH3: Single precision constant stored as $12. \times 2^{-14}$, scaled B14 in units of centimeters per second. Equation value: 12.

K:UNFVLIM: Single precision constant stored as 0.129×2^{-1}, scaled B1 and unitless. Equation value: 0.129

K:UNITX: Single precision constant vector stored as (0.5, 0, 0), scaled B1 and unitless. Equation value: (1,0,0)

MASS: see SERV section.

mDELGMB: Single precision vector containing the complement of the proposed additions to the desired gimbal angle command to be issued to the DAP, scaled B-1 in units of revolutions.

MDOT: Double precision nominal mass flow rate during thrust, scaled B3 in units of kilograms per centisecond.

MODREG: see DATA section.

MUdA, MUASTEER: see TRGL section.

MUDEX: see CONC section.

NDXCDUW: Single precision index (0 or 1) to select the proper steering constants for LM alone or CSM-LM configuration, scaled B14 and unitless.

NVWORD1: Single precision cell used to specify either a V97 or V99 display.

NVWORD$_2$: See DINT section.

OGABIAS: Single precision quantity representing the outer gimbal angle bias for window pointing commands to account for window bending due to cabin pressurization. Set to zero in "INITCDUW" and changed to AZBIAS in P64 ("EXNORM"). Scaled B-1 in units of revolutions.

OMEGAPD, OMEGAQD, OMEGARD: See DAPA section.

PBODY: See ORBI section.

PIPTIME, PIPTIME1: See SERV section.

POINTVSM: See ATTM section.

QCDUWUSR: Single precision octal return address storage.

QTEMP, QTEMP1: Single precision octal return address storage.

RATT, VATT, TAT: See ORBI section.

RCSM, VCSM: See SERV section.

[REFSMMAT]: See COOR section.

RINIT, VINIT: See TRGL section.

RMAG: See ASCT section. Scaled B29 (earth) or B27 (moon) here.

RN, VN: See SERV section.

RTARG: See TRGL section.

RTIG, VTIG: see TRGX section.

RTMAG: see TRGL section.

RTX1, RTX2: see ORBI section.

SAVET: Double precision temporary storage cell for time information, scaled B28 in units of centiseconds.

SCAXIS: See ATTM section.

SINIGA, SINMGA, SINOGA: See COOR section.

[SMNBMAT]: See COOR section.

TDEC1: See ORBI section.

TDECAY: Double precision thrust decay time added to TGO, scaled B28 in units of centiseconds.

TEVENT: Double precision time-of-event for downlink information, scaled B28 in units of centiseconds.

TGO: Double precision predicted length of burn, scaled B28 in units of centiseconds.

THETAD: See IMUC section.

THRUST: See DESC section.

TIG: Double precision predicted time of ignition input to the burn routines, or predicted cutoff time, scaled B28 in units of centiseconds.

TIGSAVE: Double precision storage for the effective time of the last performance of the Lambert routine; scaled B28 in units of centiseconds.

TIMENOW: See EXVB section.

TNEWA: Double precision pad loaded quantity giving the Lambert cycle period; scaled B28 in units of centiseconds.

TPASS4: See TRGL section.

TRKMKCNT: See RNAV section

TTOGO: Double precision time until engine ignition (or cutoff), scaled B28 in units of centiseconds.

UN: see TRGL section.

UNFC: Double precision desired thrust acceleration vector, with variable scaling in units of meters per centisecond squared and expressed in the Platform coordinate system.

UNFV: Double precision filtered value of the sensed thrust direction vector, scaled B1 and unitless, and expressed in what might best be called the "theoretical" body coordinate system. The X component is not used, but the Y and Z components are used to bias the desired thrust vector with respect to the spacecraft so that the desired direction of thrust passes through the center of gravity of the spacecraft.

UNITGOBL: Double precision vector used to bias the velocity-to-be-gained vector to offset the effect of gravity during an extended Lambert burn, scaled B1 and expressed in the Reference coordinate system.

UNITR: See SERV section.

UNWC: Double precision vector along the desired pointing direction of the landing window, scaling and units variable, expressed in the Platform coordinate system.

UT: Double precision unit vector in the direction of velocity to be gained, used to determine initial attitude for burns, scaled B1 and expressed in the Reference coordinate system.

UNX, UNY, UNZ: Double precision unit vectors along the desired directions of the three body axes, scaled B1 and expressed in the Platform coordinate system.

VEX: Double precision engine exhaust velocity, scaled B7 in units of meters per centisecond.

VG: Double precision velocity-to-be-gained vector, scaled B7 in units of meters per centisecond and expressed in the reference coordinate system.

VGBODY: Double precision velocity-to-be-gained vector, scaled B7 in units of meters per centisecond and expressed in the Body coordinate system.

VGPREV: Double precision previous value of VG, program notation also VGTIG, scaled B7 in units of meters per centisecond and expressed in the reference coordinate system.

VIPRIME: See TRGL section.

VN1: See SERV section.

VTARGTAG: See TRGL section.

WCHPHOLD, WCHPHASE: See DESC section.

WHICH: Single precision octal address of one of the following tables:

Table entry number	Tables					
	P12TABLE	P40TABLE	P41TABLE	P42TABLE	P63TABLE	ABRTABLE
0	K:V06N74	K:V06N40	--------	K:V06N40	K:V06N62	K:V06N63
1	ULLGNOT	ULLGNOT	--------	WANTAPS	ULLGNOT	ULLGNOT
2	COMFAIL3	COMFAIL4	--------	COMFAIL4	COMFAIL3	COMFAIL3
3	GOCUTOFF	GOPOST	--------	GOPOST	V99RECYC	GOCUTOFF
4	End task	End task	--------	End task	End task	End task
5	P40SPOT	P40SPOT	P41SPOT	P40SPOT	P41SPOT	--------
6	0	2240	-1	2640	2240	--------
7	SERVEXIT	STEERING	CALCN85	STEERING	SERVEXIT	--------
11	DISPCHNG	P40SJUNK	End task	P40SJUNK	DISPCHNG	DISPCHNG
12	End task	End task	TIGTASK	End task	End task	End task
13	P12IGN	P40IGN	-------	P42IGN	P63IGN	ABRTIGN
14	-------	REP40ALM	-------	P42STAGE	-------	-------

XNBPIP, YNBPIP, ZNBPIP: See SERV section.

XSMDrf, YSMDrf, ZSMDrf: See COOR section.

ZOOMTIME: Single precision time after ignition at which the DPS is to be commanded to full throttle, scaled B14 in units of centiseconds; part of the erasable load.

Z: Z register, or program counter. Cantains the address of the next step. The contents of Z are always a positive number.

TIMERAD RTNCONC = return address

Perform "PARAM" with $\underline{TSr} = \underline{RVEC}$ and $\underline{TSv} = \underline{VVEC}$

If overflow (in "PARAM"):

Switch FLAGWRD8 bit 4 (COGAFLAG) to 1

Proceed to "POODOO" with $TS = 20607_8$

$\underline{TSe} = (1 - RdA)\ \underline{UR1} - \sqrt{P\ (2 - RdA)}\ COGA\ \underline{U2}$

If overflow (or if $|\underline{TSe}| < 2^{-18}$):

Switch FLAGWRD5 bit 3 (SOLNSW) to 1

Proceed to "POODOO" with $TS = 20607_8$

$\underline{TSue} = \text{unit}\,\underline{TSe}$

$COSF = \left[(R1\ P\ /\ RDESIRED) - 1\right] / |\underline{TSe}|$

If $(1 - COSF^2) < 0$:

$COSF = 1\ \text{sign}COSF$

$TSs = 0$

Switch FLAGWRD8 bit 5 (APSESW) to 1

If $(1 - COSF^2) \geq 0$:

$TSs = \sqrt{1 - COSF^2}\ \text{sign}SGNRDOT$

Switch FLAGWRD8 bit 5 (APSESW) to 0

$\underline{TSu2} = TSs\ \underline{UN} * \underline{TSue} + COSF\ \underline{TSue}$

$CSTH = \underline{TSu2} \cdot \underline{UR1}$ (magnitude limited to less than 1)

$SNTH = \underline{UR1} * \underline{TSu2} \cdot \underline{UN}$

$TSp = P$

Perform "GETX"

Switch FLAGWRD5 bit 3 (SOLNSW) to 0

Proceed to "COMMNOUT"

APSIDES RTNCONC = return address

$\underline{T}$Sr = $\underline{R}$VEC

$\underline{T}$Sv = $\underline{V}$VEC

Perform "PARAM"

(Ignore any overflow)

$ECC = \sqrt{1 - P\ RdA}$

TSrp = R1 P / (1 + ECC)

TSra = (2 R1 / RdA) - TSrp

If TSra < 0 or if overflow, TSra = K:posmaxdp

Return via RTNCONC

TIMETHET RTNCONC = return address

$\underline{T}$Sr = $\underline{R}$VEC

$\underline{T}$Sv = $\underline{V}$VEC

Perform "PARAM"

If overflow (anywhere above):

Switch FLAGWRD8 bit 4 (COGAFLAG) to 1

Proceed to "POODOO" with TS = 20607_8

TSp = P

Perform "GETX"

COMMNOUT If FLAGWRD8 bit 7 (INFINFLG) = 1, proceed to "POODOO" with TS = 20607_8

Switch FLAGWRD8 bit 4 (COGAFLAG) to 0

Perform "DELTIME"

If FLAGWRD7 bit 9 (RVSW) = 0, perform "NEWSTATE"

Return via RTNCONC

KEPLERN $i = \text{MUDEX} + 2$ (always entered from "KEPPREP")

$\text{ldMU} = \text{K:MUTABLE}_{i}$

$\text{ROOTMU} = \text{K:MUTABLE}_{i+2}$

$\text{ldROOTMU} = \text{K:MUTABLE}_{i+4}$

$\text{ITERCTR} = 20$

$\underline{\text{U}}\text{RRECT} = \text{unit}\underline{\text{R}}\text{RECT}$

$\text{R1} = \left| \underline{\text{R}}\text{RECT} \right|$

$\text{KEPC1} = \underline{\text{R}}\text{RECT} \cdot \underline{\text{V}}\text{RECT ldROOTMU}$

$\text{KEPC2} = \underline{\text{V}}\text{RECT} \cdot \underline{\text{V}}\text{RECT ldMU R1} - 1$

$\text{ALPHA} = (1 - \text{KEPC2}) / \text{R1}$

If ALPHA < 0:

$\text{TS} = \text{K:m50SC} / \text{ALPHA}$

$\text{TS} = \sqrt{\text{TS}}$

If overflow, TS = K:posmaxdp

If ALPHA ≥ 0:

$\text{TS} = \text{K:2PISC} / \sqrt{\text{ALPHA}}$

If overflow, TS = K:posmaxdp

$\text{XMAX} = \text{TS}$

$\text{TSperiod} = \text{XMAX ldROOTMU} / \text{ALPHA}$

PERIODCH If $0 \leq \text{TSperiod} < 2^{28}$ and if $\left|\text{TAU}\right| \geq \text{TSperiod}$:

$\text{TAU} = \left[\left|\text{TAU}\right| - \text{TSperiod} \right] \text{signTAU}$

Proceed to "PERIODCH"

X = XKEPNEW

If X signTAU $\leq$ 0 or if $|X\ \text{signTAU}| \geq$ XMAX, X = (XMAX/2)signTAU

If TAU $<$ 0:

XMIN = -XMAX

XMAX = 0

Proceed to "DXCOMP"

XMIN = 0

Proceed to "DXCOMP"

DXCOMP EPSILONT = $|\text{TAU K:BEE22}|$

DELX = X - XPREV

KEPLOOP TSx2 = X^2

XI = X^2 ALPHA

Perform "DELTIME"

If overflow, (somewhere above):

If X $<$ 0, XMIN = X

If X $\geq$ 0, XMAX = X

DELX = DELX / 2

If $|\text{DELX}| < 2^{b-28}$ (b is the scale factor of DELX):

Return via KEPRTN

X = X - DELX

(If overflow)

T = TC

Proceed to "BRNCHCTR"

DELT = TAU - T

If |DELT| ≤ EPSILONT, proceed to "KEPCONVG"

TS = DELX DELT / (T - TC)

If TS < 0:

XMAX = X

DELX = TS

If TS ≤ XMIN - XMAX or if (XMIN - XMAX - TS) overflows:

DELX = (XMIN - X) K:DP9d10

If TS ≥ 0:

XMIN = X

DELX = TS

If TS > XMAX - XMIN or if (XMAX - XMIN - TS) overflows:

DELX = (XMAX - X) K:DP9d10

If |DELX| < 2^{b-28} (b is the scale factor of DELX):

Proceed to "KEPCONVG"

X = X + DELX

TC = T

Proceed to "BRNCHCTR"

BRNCHCTR ITERCTR = ITERCTR - 1

If ITERCTR = 0, proceed to "KEPCONVG"

Proceed to "KEPLOOP"

KEPCONVG RCV = (R1 - XSQCxi) URRECT + (T - X^3 1dROOTMU Sxi) VRECT

TSv = [(XI Sxi - 1) ROOTMU X /|RCV|] URRECT

$\underline{\text{V}}$CV = (1 - XSQCxi / $|\underline{\text{R}}\text{CV}|$) $\underline{\text{V}}$RECT + $\underline{\text{T}}$Sv

TC = T

XPREV = X

Return via KEPRTN (to caller of "KEPPREP")

LAMBERT RTNCONC = return address

Switch FLAGWRD5 bit 3 (SOLNSW) to 0

i = MUDEX + 2

ldMU = K:MUTABLE_i

ROOTMU = K:MUTABLE_{i+2}

ldROOTMU = K:MUTABLE_{i+4}

EPSILONL = TDESIRED K:BEE19

Switch FLAGWRD1 bit 3 (SLOPESW) to 1

$\underline{\text{T}}$Sr1 = $\underline{\text{R}}$1VEC

$\underline{\text{T}}$S2 = $\underline{\text{R}}$2VEC

Perform "GEOM"

SNTH = TSsin

TSlam = R1 / MAGVEC2

CSTH = TScos

lmCSTH = 1 - CSTH

If $|\text{lmCSTH}| < 2^{-27}$:

  Switch FLAGWRD5 bit 3 (SOLNSW) to 1

  Return via RTNCONC

TS = $\sqrt{2\ \text{TSlam} / \text{lmCSTH}}$ + SNTH / lmCSTH

If overflow or if TS $\geq$ K:COGUPLIM, TS = K:COGUPLIM

COGAMAX = TS

CSTHmRHO = CSTH - TSlam

TS = CSTHmRHO / SNTH

If overflow ($|TS| \geq 2^5$) or if GEOMSGN < 0, TS = K:COGLOLIM

COGAMIN = TS

If FLAGWRD1 bit 2 (GUESSW) = 0, TWEEKIT = 2^{-14}

If FLAGWRD1 bit 2 (GUESSW) = 1:

    TWEEKIT = 2^{-2}

    COGA = ½ (COGAMIN + COGAMAX)

    DCOGA = COGA

LAMBLOOP   TS = 1mCSTH / (COGA SNTH - CSTHmRHO)

If TS ≤ 0:

    If DCOGA ≥ 0, proceed to "LOENERGY"

    Proceed to "HIENERGY"

P = TS

RdA = 2 - P (1 + $COGA^2$)

If overflow (P or RdA), proceed to "HIENERGY"

TSp = P

Perform "GETX"

TPREV = T

If FLAGWRD8 bit 7 (INFINFLG) = 1:

    If DCOGA ≥ 0, proceed to "LOENERGY"

    Proceed to "HIENERGY"

Perform "DELTIME"

If overflow:

    T = TPREV

    Proceed to "LOENERGY"

TERRLAMB = TDESIRED - T

If $|TERRLAMB| \leq$ EPSILONL, proceed to "INITV"

ITERCTR = ITERCTR - 1

If ITERCTR = 0, proceed to "SUFFCHEK"

If FLAGWRD1 bit 3 (SLOPESW) = 0:

If T - TPREV = 0, proceed to "SUFFCHEK"

Perform "ITERATOR" with INDEP = COGA, DELINDEP = DCOGA, DEP = T, DEPREV = TPREV, DELDEP = TERRLAMB, MAX = COGAMAX and MIN = COGAMIN

DCOGA = DELINDEP

COGAMAX = MAX

COGAMIN = MIN

If $|DCOGA| < 2^{-23}$, proceed to "SUFFCHEK"

COGA = COGA + DCOGA

Proceed to "LAMBLOOP"

LOENERGY COGAMAX = COGA

Skip next step

HIENERGY COGAMIN = COGA

DCOGA = DCOGA / 2

If $|DCOGA| < 2^{-23}$, proceed to "SUFFCHEK"

COGA = COGA - DCOGA

Proceed to "LAMBLOOP"

SUFFCHEK If $|TERRLAMB| >$ K:BEE17 TDESIRED + 1:

Switch FLAGWRD5 bit 3 (SOLNSW) to 1

Proceed to "INITV"

Proceed to "INITV"

INITV TS = $\sqrt{P / R1}$ ROOTMU

$\underline{V}$VEC = TS COGA $\underline{UR}$1 + TS $\underline{UN}$ * $\underline{UR}$1

If VTARGTAG = 0:

R2 = MAGVEC2

Perform "LAMENTER"

$\underline{V}$TARGET = $\underline{T}$Sv

Return via RTNCONC

PARAM — RTNPRM = return address

Switch FLAGWRD7 bit 10 (NORMSW) to 0

Switch FLAGWRD8 bit 4 (COGAFLAG) to 0

GEOMSGN = 27777_8 (positive)

$\underline{T}$Srl = $\underline{T}$Sr

$\underline{T}$S2 = $\underline{T}$Sv

Perform "GEOM"

COGA = TScos / TSsin

i = MUDEX + 2

$ldMU = K:MUTABLE_i$

$ROOTMU = K:MUTABLE_{i+2}$

$ldROOTMU = K:MUTABLE_{i+4}$

TS = $MAGVEC2^2$ ldMU R1

RdA = 2 - TS

P = TS $TSsin^2$

Return via RTNPRM

GEOM — $\underline{U}$2 = unit$\underline{T}$S2

MAGVEC2 = |$\underline{T}$S2|

$\underline{U}$R1 = unit$\underline{T}$Srl

TScos = $\underline{U}$R1 · $\underline{U}$2

R1 = |$\underline{T}$Srl|

$\underline{TS} = \underline{UR1} * \underline{U2}$

If FLAGWRD7 bit 10 (NORMSW) = 0:

$\underline{UN}$ = unit$\underline{TS}$ signGEOMSGN

If overflow (unit vector poorly defined), $\underline{UN} = \underline{UN} / 2$

TSsin = $|\underline{TS}|$ signGEOMSGN

Return

GETX   i = 1

Switch FLAGWRD8 bit 1 (360SW) to 0

TSsqp = $\sqrt{\text{TSp}}$

TS = SNTH / (1 - CSTH)

If overflow ($|TS| \geq 2^5$):

If TS < 0, Switch FLAGWRD8 bit 1 (360SW) to 1

Proceed to "INVRSEQN"

TS = (TS - COGA) TSsqp

If overflow ($|TS| \geq 2^5$):

If TS < 0, Switch FLAGWRD8 bit 1 (360SW) to 1

Proceed to "INVRSEQN"

WLOOP   TSw = TS

TS = RdA + TSw^2

If TS < 0, proceed to "INFINITY"

TS = $\sqrt{\text{TS}}$ + TSw

If overflow ($|TS| \geq 2^5$):

i = 1

If TS < 0, Switch FLAGWRD8 bit 1 (360SW) to 1

Proceed to "INVRSEQN"

If i $<$ 3:

    i = i + 1

    Proceed to "WLOOP"

TS = 1 / TS

If overflow ($|TS| \geq 2^2$), proceed to "INFINITY"

Proceed to "POLYCOEF"

INVRSEQN $TSw2 = \left| SNTH / (1 + CSTH - SNTH\ COGA) \sqrt{P} \right|$

TSw3 = 1

1/WLOOP $TS = RdA\ TSw2^2 + TSw3^2$

If TS $<$ 0, proceed to "INFINITY"

$TSw3 = \sqrt{TS} + TSw3$

If i $<$ 3:

    i = i + 1

    Proceed to "1/WLOOP"

TS = TSw2 / TSw3

POLYCOEF If TS $<$ 0, proceed to "INFINITY"

$x = RdA\ TS^2$

$TS = 16\ TS\ (K{:}unia + K{:}unib\ x + K{:}unic\ x^2 + \ldots + K{:}unig\ x^6)$

If FLAGWRD8 bit 1 (360SW) = 1:

    If RdA $<$ 0, proceed to "INFINITY"

    $TS = (K{:}2PISC / \sqrt{RdA}) - TS$

$XI = RdA\ TS^2$

$X = \sqrt{R1}\ TS$

$TSx2 = X^2$

$KEPC1 = \sqrt{P\ R1}\ COGA$

KEPC2 = 1 - RdA

Switch FLAGWRD8 bit 7 (INFINFLG) to 0

Return

INFINITY (Clear overflow indicator if set)

Switch FLAGWRD8 bit 7 (INFINFLG) to 1

Return

DELTIME $Sxi = K{:}S0 + K{:}S1\ XI + K{:}S2\ XI^2 + \ldots + K{:}S9\ XI^9$

$Cxi = K{:}C0 + K{:}C1\ XI + K{:}C2\ XI^2 + \ldots + K{:}C9\ XI^9$

XSQCxi = TSx2 Cxi

T = ldROOTMU (X (R1 + TSx2 Sxi KEPC2) + KEPC1 XSQCxi)

Return

NEWSTATE $\underline{T}Sr = (R1 - XSQCxi)\ \underline{U}R1 + (T - X^3\ ldROOTMU\ Sxi)\ \underline{V}VEC$

$R2 = |\underline{T}Sr|$

LAMENTER $\underline{T}S$ = (ROOTMU (XI Sxi - 1) X / R2) $\underline{U}R1$

$\underline{T}Sv = \underline{T}S + (1 - XSQCxi / R2)\ \underline{V}VEC$

Return

ITERATOR If FLAGWRD1 bit 3 (SLOPESW) = 0:

TS = DELDEP DELINDEP / (DEP - DEPREV)

If FLAGWRD8 bit 6 (ORDERSW) = 1, TS = |TS| signDELDEP

If FLAGWRD1 bit 3 (SLOPESW) = 1:

Switch FLAGWRD1 bit 3 to 0

TS = (MAX TWEEKIT - MIN TWEEKIT) signDELDEP

If TS < 0:

If FLAGWRD8 bit 6 (ORDERSW) = 0, MAX = INDEP

(If TS < 0)

If INDEP + TS ≤ MIN or if overflow:

TS = K:DP9d10 (MIN - INDEP)

DELINDEP = TS

Return

If FLAGWRD8 bit 6 (ORDERSW) = 0, MIN = INDEP

If INDEP + TS > MAX or if overflow:

TS = K:DP9d10 (MAX - INDEP)

DELINDEP = TS

Return

PERIAPO1 VVEC = TSv (rescaled for lunar orbit computations)

RVEC = TSr (rescaled for lunar orbit computations)

PERIAPO NORMEX = return address

TSo = K:RPAD

If Mudex ≠ 0, TSo = |RLS|

XXXALT = TSo

Perform "APSIDES"

TSha = TSra - XXXALT

TShp = TSrp - XXXALT

Return via NORMEX

1dMU: Double precision storage register for the gravitational constant for the moon or the earth, whichever is the central body; scaled B-34 (earth) or B-28 (moon) in units of centiseconds squared / meters cubed.

1dROOTMU: Double precision square root of 1dMU, scaled B-17 (earth) or B-14 (moon) in units of centiseconds / meters $^{3/2}$.

1mCSTH: Double precision storage for (1 - CSTH), scaled B2 and unitless.

ALPHA: Double precision inverse of the semi-major axis for the universal form of Kepler's equation, scaled B-22 (earth) or B-20 (moon) in units of meters^{-1}.

COGA: Double precision cotangent of flight path angle (measured from vertical), scaled B5 and unitless.

COGAMAX, COGAMIN: Upper and lower bounds on COGA, scaled B5 and unitless.

COSF: Double precision cosine of the true anomaly at the desired radius in the time-radius problem, scaled B1 and unitless.

CSTH: Double precision cosine of the true anomaly difference or of the angle between present and desired position vectors, scaled B1 and unitless.

CSTHmRHO: Double precision intermediate quantity used in the calculation of P and COGAMIN, scaled B7 and unitless.

C[illegible]: One of the two special functions used in the universal formulation of the conic equation; double precision, scaled B4 and unitless.

DCOGA: Double precision change in COGA in Lambert iteration step, scaled B5 and unitless.

DELDEP: Double precision change in the dependent variable for the "ITERATOR" subroutine, variable scaling and units.

DELINDEP: Double precision change in the independent variable for the "ITERATOR" subroutine, variable scaling and units.

DELT: Double precision difference between the desired time interval and the computed approximation to it during the "KEPLERN" iteration, scaled B28 in units of centiseconds.

DELX: Difference between successive values of the universal conic parameter X, scaled B17 (earth) or B16 (moon) in units of meters$^{1/2}$.

DEP, DEPREV: Double precision storage for two successive values of the independent variable to be used in the "ITERATOR" subroutine; scaling and units variable.

ECC: Double precision eccentricity computed in the "APSIDES" routine, scaled B3 and unitless.

EPSILONL, EPSILONT: Double precision definitions of convergence in the Lambert and Kepler iteration loops respectively, scaled B28 in units of centiseconds.

GEOMSGN: Single precision sign for the sine of the true anomaly difference, scaled B0 and unitless.

i: Single precision index, scaled B14, and unitless.

INDEP: Double precision independent variable for the "ITERATOR" subroutine; scaling and units variable.

ITERCTR: Single precision iteration counter, scaled B14 and unitless.

KEPC1: Double precision coefficient in the Kepler equation, scaled B17 (earth) or B16 (moon) in units of meters.

KEPC2: Double precision coefficient in the Kepler equation, scaled B6 and unitless.

KEPRTN: Single precision octal return address storage.

K:2PISC: Double precision constant, stored as 6.2831853 x 2^{-6}, scaled B6 and unitless. Equation value: 6.2831853.

K:BEE17: Double precision constant stored as 2^{-17}, scaled B0 and unitless. Equation value: 2^{-17}.

K:BEE19: Double precision constant stored as 2^{-19}, scaled B0 and unitless. Equation value: 2^{-19}.

K:BEE22: Double precision constant stored as 2^{-22}, scaled B0 and unitless. Equation value: 2^{-22}.

K:C0,...K:C9: Ten double precision constants defining the special function C(x), all unitless.

	Stored Value	Scale Factor	Theoretical Value	Equation Value
K:C0	0.031250001	B4	0.5	0.500000016
K:C1	-0.166666719	B-2	-0.041666667	-0.041666680
K:C2	0.355555413	B-8	1.38888889 E-3	1.38888833 E-3
K:C3	-0.406347410	B-14	-2.48015873 E-5	-2.48014777 E-5
K:C4	0.288962094	B-20	2.75573192 E-7	2.75575727 E-7
K:C5	-0.140117894	B-26	-2.08767570 E-9	-2.08791932 E-9
K:C6	0.049247387	B-32	1.14707456 E-11	1.14663008 E-11
K:C7	-0.013081923	B-38	-4.77947733 E-14	-4.75917586 E-14
K:C8	0.002806389	B-44	1.56192070 E-16	1.59524745 E-16
K:C9	-0.000529414	B-50	-4.11031762 E-19	-4.70214090 E-19

K:COGLOLIM: Double precision constant stored as -0.999511597, scaled B5 and unitless. Equation value: -31.9843711. (Cot 1° 48'.)

K:COGUPLIM: Double precision constant stored as 0.999511597, scaled B5 and unitless. Equation value: 31.9843711.

K:DP9d10: Double precision constant, stored as 0.9, scaled B0 and unitless. Equation value: 0.9.

K:M50SC: Double precision constant, stored as -50.0×2^{-12}, scaled B12 and unitless. Equation value: -50.0.

K:MUTABLE$_i$: A table of constants containing four gravitational constants for the earth and four for the moon.

i	K:MUTABLE	Scale Factor	Units	Equation Value	Significance
0	3.986032 E10 B-36	B36	meters^3/cs^2	3.986032 E10	μ_e
2	0.25087606 E-10 B34	B-34	cs^2/m^3	0.25087606 E-10	$1/\mu_e$
4	1.99650495 E5 B-18	B18	m$^{3/2}$/cs	1.99650495 E5	$\sqrt{\mu_e}$
6	0.50087529 E-5 B17	B-17	cs/m$^{3/2}$	0.50087529 E-5	$\sqrt{1/\mu_e}$
8	4.902778 E8 B-30	B30	m^3/cs^2	4.902778 E8	μ_m
10	0.203966 E-8 B28	B-28	cs^2/m^3	0.203966 E-8	$1/\mu_m$
12	2.21422176 E4 B-15	B15	m$^{3/2}$cs	2.21422176 E4	$\sqrt{\mu_m}$
14	0.45162595 E-4 B14	B-14	cs/m$^{3/2}$	0.45162595 E-4	$\sqrt{1/\mu_m}$

K:RPAD: Double precision stored as 6373338×2^{-29}, scaled B29 in units of meters. Equation value: 6,373,338. (Equivalent to 20,909,901.57 feet.)

K:SO, ...K:S9: Ten double precision constants defining the special function S(x), all unitless.

	Stored Value	Scale Factor	Theoretical Value		Equation Value
K:SO	0.083333334	B1	0.166666667		1.66666668 E-1
K:S1	-0.266666684	B-5	-8.33333333	E-3	-8.33333387 E-3
K:S2	0.406349155	B-11	1.98412698	E-4	1.98412673 E-4
K:S3	-0.361198675	B-17	-2.75573192	E-6	-2.75572720 E-6
K:S4	0.210153242	B-23	2.50521084	E-8	2.50522187 E-8
K:S5	-0.086221951	B-29	-1.60590438	E-10	-1.60600899 E-10
K:S6	0.026268812	B-35	7.64716373	E-13	7.64523051 E-13
K:S7	-0.006163316	B-41	-2.81145725	E-15	-2.80275162 E-15
K:S8	0.001177342	B-47	8.22063525	E-18	8.36551806 E-18
K:S9	-0.000199055	B-53	-1.95729411	E-20	-2.20995444 E-20

K:unia,...K:unig: Seven double precision constants used in the definition of the independent variable for the universal formulation of Kepler's equation, scaled B1 and unitless

	Stored Value	Equation Value
K:unia	0.5	1.0
K:unib	-0.166666770	-0.33333354
K:unic	0.100000392	0.200000784
K:unid	-0.071401086	-0.142802172
K:unie	0.055503292	0.111006584
K:unif	-0.047264098	-0.094528196
K:unig	0.040694204	0.081388408

MAGVEC2: Magnitude of the second vector input to the "GEOM" routine, double precision with variable scaling and units.

MAX, MIN: Double precision maximum and minimum bounds for the "ITERATOR" subroutine; variable scaling and units.

MUDEX: Single precision index indicating whether the gravitational constant for the earth (0) or the moon (8) should be used in the conic equations, scaled B14 and unitless. MUDEX = -X1 - 2 where X1 is index register one in the listing.

NORMEX: Single precision octal return address storage.

P: Double precision ratio of semi-latus rectum and magnitude of present position, scaled B4 and unitless.

R1, R2: Double precision magnitudes of present and desired position vectors, respectively, scaled B29 (earth) or B27 (moon) in units of meters.

R1VEC, R2VEC: Double precision vector inputs to the "LAMBERT" routine; present and desired position, respectively, scaled B29 (earth) or B27 (moon) in units of meters.

RCV: Double precision vector output of the "KEPLERN" routine; conic position vector at the specified time, scaled B29 (earth) or B27 (moon) in units of meters.

RdA: Double precision ratio of present radius to semi-major axis, scaled B6 and unitless.

RDESIRED: Double precision input to the "TIMERAD" routine scaled B29 (earth) or B27 (moon) in units of meters.

RLS: Double precision position vector at the lunar surface, scaled B27 in units of meters and expressed in selenographic coordinates.

ROOTMU: Double precision square root of the relevant gravitational constant, scaled B18 (earth) or B15 (moon) in units of meters $^{3/2}$ /cs.

RRECT: Double precision vector input to the "KEPLERN" routine; the position vector to be advanced through the specified time, scaled B29 (earth) or B27 (moon), in units of meters.

RTNCONC, RTNPRM: Single precision, octal return address storage cells.

RVEC: Double precision vector input to the majority of the conic routines, a position scaled B29 (earth) or B27 (moon) in units of meters.

SGNRDOT: Single precision flag indicating the sign of the radial velocity desired at the desired radius.

SNTH: Double precision sine of true anomaly difference or of the angle between present and desired position vectors, scaled B1 and unitless.

SS1: One of the two special functions used in the universal formulation of the conic equation; double precision, scaled B1 and unitless.

T: Double precision time (computed in "DELTIME") to go from present position to desired position, scaled B28 in units of centiseconds.

TAU: Desired transfer time input to the "KEPLERN" routine; double precision scaled B28 in units of centiseconds.

TC: Double precision time since latest rectification, scaled B28 in units of centiseconds.

TDESIRED: Double precision input to the "LAMBERT" routine; desired transfer time, scaled B28 in units of centiseconds.

TERRLAMB: Double precision difference between desired and computed transfer time during the "LAMBERT" iterations, scaled B28 in units of centiseconds.

TPREV: Previous value of T, scaled B28 in units of centiseconds and double precision.

TWEEKIT: Single precision factor used in the initiation of the "ITERATOR" routine, set large or small depending on whether or not a good first approximation of COGA is available.

U2, UR1: Double precision unit vectors in the directions of the two vectors input into the "GEOM" routine, scaled B1 and unitless.

UN: Unit normal vector, scaled B1 and unitless.

URRECT: Unit vector in the direction of RRECT, scaled B1 and unitless.

VCV: Double precision vector output of the "KEPLERN" routine; the velocity vector at the specified time, scaled B7 (earth) or B5 (moon) in units of meters/centisecond.

VRECT: Double precision vector input to the "KEPLERN" routine; the velocity vector to be advanced through the specified time, scaled B7 (earth) or B5 (moon) in units of meters/centisecond.

VTARGET: Double precision velocity vector output of the "LAMBERT" routine; velocity at R2VEC, scaled B7 (earth) or B5 (moon) in units of meters/centisecond.

VTARGTAG: Single precision flag input to "LAMBERT" to indicate that VTARGET is to be computed (if VTARGTAG = 0).

VVEC: Double precision vector input to the majority of the conic routines; velocity, scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

x: Double precision temporary storage cell used in "POLYCOEF", scaled B0 and unitless.

X: Double precision universal conic parameter equal to the product of semi-major axis and eccentric anomaly difference (for the ellipse) or to the product of the hyperbolic analogs of eccentric anomaly difference and semi-major axis (for the hyperbola), scaled B17 (earth) or B16 (moon) in units of meters to the one-half power.

XI: Double precision square of the eccentric anomaly difference for an ellipse, or the negative of the square of its hyperbolic analog for a hyperbola, scaled B6 in units of radians squared.

XKEPNEW: Double precision value of X at entry to "KEPLERN", scaled B17 (earth) or B16 (moon) in units of meters to the one-half power.

XMAX, XMIN: Double precision upper and lower bounds on X, scaled B17 (earth) or B16 (moon) in units of meters to the one-half power.

XPREV: Previous value of X; same units and scaling as X.

XSQCxi: Double precision product of X^2 and Cxi, scaled B33 (earth) or B31 (moon) in units of meters.

XXXALT: Value of base altitude for computing apogee and perigee information, scale factor B29 (earth) or B27 (moon), units meters.

COOR

CDUTRIG $\underline{ANG} = \underline{CDU}$

CD*TR*GS $\underline{ANG} = \underline{ANG}$ converted to one's complement form

$SINOGA = \sin ANG_x$

$COSOGA = \cos ANG_x$

$SINIGA = \sin ANG_y$

$COSIGA = \cos ANG_y$

$SINMGA = \sin ANG_z$

$COSMGA = \cos ANG_z$

Return

NBTOSM

$$[TS1] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & COSOGA & -SINOGA \\ 0 & SINOGA & COSOGA \end{bmatrix}$$

$$[TS2] = \begin{bmatrix} COSMGA & -SINMGA & 0 \\ SINMGA & COSMGA & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[TS3] = \begin{bmatrix} COSIGA & 0 & SINIGA \\ 0 & 1 & 0 \\ -SINIGA & 0 & COSIGA \end{bmatrix}$$

$$[NBSMMAT] = [TS3]\,[TS2]\,[TS1]$$

Return

CALCSMSC $\underline{X}NBsm = (COSIGA\ COSMGA\ ,\ SINMGA\ ,\ -SINIGA\ COSMGA)$

$TS = SINOGA\ SINMGA$

$ZNBsm_z$ = COSIGA COSOGA − SINIGA TS

$ZNBsm_y$ = − SINOGA COSMGA

$ZNBsm_x$ = COSOGA SINIGA + COSIGA TS

$\underline{Y}NBsm = \underline{Z}NBsm * \underline{X}NBsm$

Return

SMTONB

$$[TS1] = \begin{bmatrix} COSIGA & 0 & -SINIGA \\ 0 & 1 & 0 \\ SINIGA & 0 & COSIGA \end{bmatrix}$$

$$[TS2] = \begin{bmatrix} COSMGA & SINMGA & 0 \\ -SINMGA & COSMGA & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$[TS3] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & COSOGA & SINOGA \\ 0 & -SINOGA & COSOGA \end{bmatrix}$$

$$[SMNBMAT] = [TS3]\ [TS2]\ [TS1]$$

Return

EARTHMX

AZ504 = AZO + K:WEARTH (TEPHEM + TSt)

If overflow ($|AZ504| \geq 1$), AZ504 = fractional part of AZ504

$$[EARTHMAT] = \begin{bmatrix} \cos AZ504 & \sin AZ504 & 0 \\ -\sin AZ504 & \cos AZ504 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$\underline{L}E504 = (-UNITW_y, UNITW_x, 0)$

Return

MOONMX $EI = K{:}BSUBO + K{:}BDOT\ (TEPHEM + TSt)$

$MR = K{:}FSUBO + K{:}FDOT\ (TEPHEM + TSt)$

$MN = K{:}NODIO + K{:}NODDOT\ (TEPHEM + TSt)$

$\underline{T}Sb = (-\sin MN\ ,\ \cos MN \cos EI\ ,\ \cos MN \sin EI)$

$\underline{T}Sa = (\cos MN\ ,\ \sin MN \cos EI\ ,\ \sin MN \sin EI)$

$\underline{T}Sc = (0\ ,\ -\sin EI\ ,\ \cos EI)$

$$\begin{bmatrix} MNMAT1 \end{bmatrix} = \begin{bmatrix} TSa_x & TSa_y & TSa_z \\ TSb_x & TSb_y & TSb_z \\ TSc_x & TSc_y & TSc_z \end{bmatrix} = \begin{bmatrix} \cos MN & \sin MN & 0 \\ -\sin MN & \cos MN & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos EI & \sin EI \\ 0 & -\sin EI & \cos EI \end{bmatrix}$$

$$\begin{bmatrix} MNMAT2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & K{:}COSI & -K{:}SINI \\ 0 & K{:}SINI & K{:}COSI \end{bmatrix}$$

$$\begin{bmatrix} MNMAT3 \end{bmatrix} = \begin{bmatrix} -\cos MR & -\sin MR & 0 \\ \sin MR & -\cos MR & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\frac{1}{2} + MR) & \sin(\frac{1}{2} + MR) & 0 \\ -\sin(\frac{1}{2} + MR) & \cos(\frac{1}{2} + MR) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} MOONMAT \end{bmatrix} = \begin{bmatrix} MNMAT3 \end{bmatrix} \begin{bmatrix} MNMAT2 \end{bmatrix} \begin{bmatrix} MNMAT1 \end{bmatrix}$$

Return

LAT-LONG $ALPHAM = |\underline{A}LPHAV|$

If FLAGWRD3 bit 12 (LUNAFLAG) = 1:

Perform "MOONMX"

$\underline{T}S = \begin{bmatrix} MOONMAT \end{bmatrix} (\underline{A}LPHAV - (\begin{bmatrix} MOONMAT \end{bmatrix}^T \underline{L}M504) * \underline{A}LPHAV)$

$\underline{A}LPHAV = \text{unit}\underline{T}S$

GAMRP = K:1B1

If FLAGWRD1 bit 13 (ERADFLAG) = 1, ERADM = K:504RM

If FLAGWRD1 bit 13 (ERADFLAG) = 0, ERADM = $|\underline{R}LS|$

If FLAGWRD3 bit 12 (LUNAFLAG) = 0:

Perform "EARTHMX"

$\underline{TS} = [EARTHMAT] (\underline{ALPHAV} - \underline{LE504} * \underline{ALPHAV})$

$\underline{ALPHAV} = unit\underline{TS}$

GAMRP = K:B2dA2

If FLAGWRD1 bit 13 (ERADFLAG) = 1:

$TS = 1 - ALPHAV_z^{\ 2}$

$ERADM = \sqrt{K{:}B2XSC / (1 - K{:}EE\ TS)}$

If FLAGWRD1 bit 13 (ERADFLAG) = 0, ERADM = K:ERAD

$COSTH = GAMRP \sqrt{ALPHAV_x^{\ 2} + ALPHAV_y^{\ 2}}$

$SINTH = ALPHAV_z$

Perform "ARCTAN"

LAT = THETA

$COSTH = ALPHAV_x$

$SINTH = ALPHAV_y$

Perform "ARCTAN"

LONG = THETA

ALT = ALPHAM - ERADM

Return

<u>LALOTORV</u> If FLAGWRD3 bit 12 (LUNAFLAG) = 1, GAMRP = K:1B1

If FLAGWRD3 bit 12 (LUNAFLAG) = 0, GAMRP = K:B2dA2

$TS_z = GAMRP\ sinLAT$

$TS_y = sinLONG\ cosLAT$

$TS_x = cosLONG\ cosLAT$

$\underline{ALPHAV} = unit\underline{TS}$

If FLAGWRD3 bit 12 (LUNAFLAG) = 1:

If FLAGWRD1 bit 13 (ERADFLAG) = 1, ERADM = K:504RM

If FLAGWRD1 bit 13 (ERADFLAG) = 0, ERADM = $\left| \underline{R}LS \right|$

Perform "MOONMX"

$\underline{T}S = \left[MOONMAT \right]^T (\underline{A}LPHAV + \underline{L}M504 * \underline{A}LPHAV)$

If FLAGWRD3 bit 12 (LUNAFLAG) = 0:

If FLAGWRD1 bit 13 (ERADFLAG) = 1:

$TS = 1 - ALPHAV_z^2$

$ERADM = \sqrt{K:B2XSC / (1 - K:EE\ TS)}$

If FLAGWRD1 bit 13 (ERADFLAG) = 0, ERADM = K:ERAD

Perform "EARTHMX"

$\underline{T}S = \left[EARTHMAT \right]^T (\underline{A}LPHAV + (\left[EARTHMAT \right] \underline{L}E504) * \underline{A}LPHAV)$

$\underline{A}LPHAV = \underline{T}S\ (ERADM + ALT)$

Return

<u>ARCTAN</u> $TS = SINTH^2 + COSTH^2$

If TS = 0:

THETA = 0

Return

$TS = SINTH / \sqrt{TS}$

If $|TS| \geq 1$:

THETA = $\frac{1}{4}$ signSINTH

Return

THETA = arcsinTS

If COSTH < 0, THETA = - THETA + $\frac{1}{2}$ signTHETA

Return

CALCGA (entered with XNBrf and XSMrf etc., XNBsm and XSMnb etc., etc.)

TS = unit(XNB * YSM)

COSTH = TS · ZNB

SINTH = TS · YNB

Perform "ARCTRIG"

OGC = THETA

COSTH = TS * XNB · YSM

SINTH = YSM · XNB

Perform "ARCTRIG"

MGC = THETA

If |MGC| ≥ K:gloktest:

- Perform "ALARM" with TS = 00401_8
- Switch FLAGWRD3 bit 14 (GLOKFAIL) to 1

COSTH = ZSM · TS

SINTH = XSM · TS

Perform "ARCTRIG"

IGC = THETA

TS = (OGC, IGC, MGC) converted to two's complement form

THETAD = TS

Return

ARCTRIG If |SINTH| ≥ K:QTSN45, THETA = arccosCOSTH signSINTH

If |SINTH| < K:QTSN45:

- THETA = arcsinSINTH
- If COSTH < 0, THETA = $\frac{1}{2}$ signSINTH − THETA

Return

CALCGTA $\underline{TS}$ = unit(− $DCMAT_{13}$, 0 , $DCMAT_{11}$)

$SINTH = TS_x$

$COSTH = TS_z$

Perform "ARCTRIG"

IGC = THETA

$SINTH = DCMAT_{12}$

$COSTH = TS_z\ DCMAT_{11} - TS_x\ DCMAT_{13}$

Perform "ARCTRIG"

MGC = THETA

$COSTH = \underline{TS} \cdot (DCMAT_{31}, DCMAT_{32}, DCMAT_{33})$

$SINTH = \underline{TS} \cdot (DCMAT_{21}, DCMAT_{22}, DCMAT_{23})$

Perform "ARCTRIG"

OGC = THETA

Return

LSPOS TSt = (TSt + TEPHEM) / K:CSTODAY

Switch FLAGWRD0 bit 3 (FREEFLAG) to 0

GTMP = K:amod sin(K:1d27 TSt + K:aarg)

Switch FLAGWRD0 bit 3 (FREEFLAG) to 1

GTMP = GTMP + K:bmod sin(K:1d32 TSt + K:barg)

$STMP_0$ = K:lomo + K:lomr TSt − GTMP

GTMP = K:cmod sin(K:1d365 TSt + K:carg)

$STMP_2$ = K:loso + K:losr TSt − GTMP

$STMP_4$ = K:lono + K:lonr TSt

$$\underline{TS} = \left[K:KONMAT\right]\begin{pmatrix}\cos STMP_0 \\ \sin STMP_0 \\ \sin(STMP_0 - STMP_4)\end{pmatrix}$$

$\underline{VMOON} = \text{unit}\,\underline{TS}$

$$\underline{TS} = \left[K:KONMAT\right]\begin{pmatrix}\cos STMP_2 \\ \sin STMP_2 \\ 0\end{pmatrix}$$

$\underline{VSUN} = \text{unit}\,\underline{TS}$

$\underline{TS}sun = \text{unit}\,\underline{TS}$

Return

MFREF $\underline{TS}_1 = \underline{VEC1}$

TSt = TIMENOW

Perform "MOONMX"

$\underline{VEC1} = (\underline{TS}_1 * \underline{LM504} + \underline{TS}_1)\left[MOONMAT\right]$

$\underline{TS}_1 = \underline{VEC2}$

Perform "MOONMX"

$\underline{VEC2} = (\underline{TS}_1 * \underline{LM504} + \underline{TS}_1)\left[MOONMAT\right]$

Return

REFMF Perform "CDUTRIG"

TSt = TIMENOW

Perform "CALCSMSC"

$\underline{TS}_1 = \text{unit}(\underline{YNBsm}\left[REFSMMAT\right])$

Perform "MOONMX"

$\underline{TS} = \underline{LM504}\left[MOONMAT\right]$

$\underline{YNBSAV} = \left[MOONMAT\right](\underline{TS}_1 - \underline{TS} * \underline{TS}_1)$

$\underline{TS}_1 = \text{unit}(\underline{ZNBsm}\left[REFSMMAT\right])$

Perform "MOONMX"

$\underline{TS} = \underline{L}M504 \left[MOONMAT \right]$

$\underline{Z}NBSAV = \left[MOONMAT \right] (\underline{TS}_1 - \underline{TS} * \underline{TS}_1)$

Switch FLAGWRD6 bit 1 (ATTFLAG) to 1

Return

ALPHAM: Magnitude of position vector input to "LAT-LONG" routine.

ALPHAV: Working storage for the position vector or unit position vector in reference, selenographic, or geographic coordinates.

ALT: Double precision altitude, scaled B29 in units of meters.

ANG: Single precision vector containing the outer, inner, and middle gimbal angles in its X, Y, and Z components, respectively, stored in units of revolutions in two's complement form scaled B-1 or in one's complement form scaled B0.

AZ504: Double precision angle of rotation of the earth around its polar axis, scaled B0 in units of revolutions. Program notation "504AZ".

AZO: Double precision position angle of the earth at the time when TEPHEM equals zero, scaled B0 in units of revolutions; included in the erasable load.

CDU (CDU_x, CDU_y, CDU_z): Single precision vector containing the measured values of the IMU gimbal angles (outer, inner and middle gimbal in X, Y and Z components respectively), scaled B-1 in units of revolutions and stored in two's complement form. Each component is an LGC input counter incremented directly from the Coupling Data Unit in response to changes in the IMU gimbal angles.

COSIGA, COSMGA, COSOGA: Double precision cosines of the inner, middle and outer gimbal angles respectively, scaled B1 and unitless. Program notation "$COSCDU_0$, $COSCDU_2$, $COSCDU_4$."

COSTH: Double precision cosine scaled B1 in "ARCTAN" and B2 in "ARCTRIG."

[DCMAT]: See ALIN section.

[EARTHMAT]: Double precision, 3x3, orthogonal transformation matrix, scaled B1 and unitless. Agd = [EARTHMAT] Aref, where A is a vector expressed in geodetic and reference coordinates respectively.

EI: Double precision angle of inclination of the equatorial plane measured from the ecliptic plane around the earth-to-sun vector at the vernal equinox, scaled B0 in units of revolutions. Used to transform from reference coordinates to a right-handed, orthogonal system whose X-axis is along the earth-to-sun vector at the vernal equinox and whose Z-axis is perpendicular to the ecliptic.

ERADM: Double precision radius of earth or moon, scaled B29 in units of meters.

GAMRP: Double precision square of the ratio of polar radius to equatorial radius, scaled B1 and unitless.

GTMP: Working storage in "LSPOS" scaled B0 in units of revolutions.

K:1B1: Double precision constant stored as 2^{-1}, scaled B1 and unitless. Equation value: 1. (Corresponds to the square of the ratio of polar radius to equatorial radius for the moon.)

K:1d27: Double precision constant stored as 0.036291712 x 2, program notation VAL67+4, scaled B-1 and unitless. Equation value: 0.036291712. (Equivalent to 1 / 27.5545 and used in the extension of the circular approximation to the moon's orbit to account for eccentricity and rotation of the line of apsides.)

K:1d32: Double precision constant stored as 0.03125 x 2, program notation VAL67+10, scaled B-1 and unitless. Equation value: 0.03125.

K:1d365: Double precision constant stored as 0.002737925 x 2, program notation VAL67+16, scaled B-1 and unitless. Equation value: 0.002737925. (Equivalent to 1 / 365.2401.)

K:504RM: Double precision constant stored as 1738090 x 2^{-29}, scaled B29 in units of meters. Equation value: 1738090.

K:aarg: Double precision constant stored as 0.530784445, program notation VAL67+2, scaled B0 in units of revolutions. Equation value: 0.530784445.

K:amod: Double precision constant stored as 0.01726666666 x 2, program notation VAL67, scaled B-1 in units of revolutions. Equation value: 0.017266666. (Corresponds to 2 e / 2π where e is the mean eccentricity of the moon's orbit = 0.054245.)

K:B2dA2: Double precision constant stored as 0.9933064884 x 2^{-1}, scaled B1 and unitless. Equation value: 0.9933064884. (corresponds to the square of the ratio of polar radius to equatorial radius for the earth.)

K:B2XSC: Double precision constant stored as 0.0179450689, scaled B51 in units of meters squared. Equation value: 6356784 squared. (Corresponds to the square of the polar radius of the earth.)

K:barg: Double precision constant stored as 0.585365625, program notation VAL67+8, scaled B0 in units of revolutions. Equation value: 0.585365625.

K:BDOT: Double precision constant stored as -1.145529388 E-16 x 2^{28}, scaled B-28 in units of revolutions per centisecond. Equation value: -1.145529388 E-16. (Equivalent to 7.197573418 E-14 radians per second or 2,766,240 years per revolution.)

K:bmod: Double precision constant stored as 0.003505277 x 2, program notation VAL67+6, scaled B-1 in units of revolutions. Equation value 0.003505277.

K:BSUBO: Double precision constant stored as 6.512013939 E-2, scaled B0 in units of revolutions. Equation value 6.512013939 E-2. (Equivalent to 4.09161903 E-1 radians.)

K:carg: Double precision constant, stored as -0.01106341036, program notation VAL67+14, scaled B0 and unitless. Equation value: -0.01106341036.

K:cmod: Double precision constant stored as 0.005325277 x 2, program notation VAL67+12, scaled B-1 in units of revolutions. Equation value: 0.005325277. (Corresponds to 2e / 2π where e is the mean eccentricity of the geocentric solar orbit = 0.01674.)

K:COSI: Double precision constant stored as 0.999641732 x 2^{-1}, scaled B1 and unitless. Equation value 0.999641732.

K:CSTODAY: Double precision constant stored as 8640000 x 2^{-33}, scaled B33 in units of centiseconds. Equation value: 8640000.

K:EE: Double precision constant stored as 6.6935116 E-3, scaled B0 and unitless. Equation value: 6.6935116 E-3. (Corresponds to the square of the eccentricity of the Fischer ellipsoid.)

K:ERAD: Double precision constant stored as 6373338 x 2^{-29}, scaled B29 in units of meters. Equation value: 6373338. (Corresponds to the pad radius on the earth.)

K:FDOT: Double precision constant stored as 4.253263473 E-9 x 2^{27}, scaled B-27 in units of revolutions per centisecond. Equation value: 4.253263473 E-9. (Equivalent to 2.672404256 E-6 radians per second or 27.21 days per revolution.)

K:FSUBO: Double precision constant stored as 8.290901511 E-1, scaled B0 in units of revolutions. Equation value: 8.290901511 E-1. (Equivalent to 5.209327056 radians.)

K:gloktest: Double precision constant stored as 0.1666666667, scaled B0 in units of revolutions. Equation value: 0.1666666667. (Equivalent to 60 degrees.) Program notation ".166...".

K:KONMAT : Double precision 3x3 matrix, scaled B1 and unitless. Used to transform from ecliptic to equatorial, earth-centered coordinates. Equation value:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & a & b \\ 0 & c & d \end{bmatrix}$$

where a = 0.91745, b = -0.03571 (-0.39784 x 0.08976), c = 0.39784 and d = 0.082350 (0.91745 x 0.08976). The quantities 0.39784 and 0.91745 are the sine and cosine respectively of 23.444 degrees; 0.08976 is the sine of 5.150 degrees. Note that the factor 0.08976 has no effect on the transformation of the sun's position vector and is actually a parameter of the lunar orbit.

K:lomo: Double precision constant stored as 0.815282336, program notation RATESP+6, scaled B0 in units of revolutions. Equation value: 0.815282336. (Corresponds to the position of the moon in it's orbit at July 1, 1969.)

K:lomr: Double precision constant stored as 0.03660098×2^4, program notation RATESP, scaled B-4 in units of revolutions per day. Equation value: 0.03660098. (Equivalent to 1 revolution per 27.32167 days. Corresponds to the sidereal period of the moon.)

K:lono: A double precision constant stored as 0.986209499, program notation RATESP+10, scaled B0 in units of revolutions. Equation value: 0.986209499. (Corresponds to the position of the mean ascending node of the lunar orbit on the ecliptic at July 1, 1969.)

K:lonr: Double precision constant stored as -0.00014719×2^4, program notation RATESP+4, scaled B-4 in units of revolutions per day. Equation value: -0.00014719.

K:loso: Double precision constant stored as 0.274674910, program notation RATESP+8, scaled B0 in units of revolutions. Equation value: 0.274674910. (Corresponds to the position of the sun at July 1, 1969.)

K:losr: Double precision constant stored as 0.00273779×2^4, program notation RATESP+2, scaled B-4 in units of revolutions per day. Equation value: 0.00273779. (Equivalent to 1 revolution per 365.2581 days. Corresponds as closely to the sidereal year as to the anomalistic year.)

K:NODDOT: Double precision constant stored as -1.703706190 E-11 x 2^{28}, scaled B-28 in units of revolutions per centisecond. Equation value: -1.703706190 E-11. (Equivalent to-1.070470170 E-8 radians per second or 18.600 years per revolution.)

K:NODIO: Double precision constant stored as 9.862094363 E-1, scaled BO in units of revolutions. Equation value: 9.862094363 E-1. (Equivalent to 6.196536640 radians.)

K:QTSN45: Double precision constant stored as 0.1768, scaled B2 and unitless. Equation value: 0.7072. (Equivalent to the sine of 45°.)

K:SINI: Double precision constant stored as 2.676579050 E-2 x 2^{-1}, scaled B1 and unitless. Equation value: 2.676579050 E-2.

K:WEARTH: Double precision constant stored as 1.160576171 E-7 x 2^{23}, scaled B-23 in units of revolutions per centisecond. Equation value: 1.160576171 E-7. (Equivalent to 7.292115147 E-5 radians per second.)

LAT: Double precision geodetic or selenographic latitude, scaled BO in units of revolutions.

LE504: Double precision vector to account for precession and nutation of the earth's polar axis (the deviation of the true pole from the mean pole), scaled BO, unitless and expressed in reference coordinates; used in an approximate transformation from reference to true equatorial coordinates.

LM504: Double precision vector to account for precession and nutation of the moon's polar axis (the deviation of the true pole from the mean pole), scaled BO in units of radians and expressed in selenographic coordinates; an approximation most accurate at the nominal midpoint of a mission. Program notation "504LM" (in erasable load).

LONG: Double precision geodetic or selenographic longitude, scaled BO in units of revolutions.

MN: Double precision angle in the ecliptic plane, measured from the earth-to-sun vector at the vernal equinox to the moon-to-sun vector at the mean descending node on the ecliptic of the moon's orbit around the earth. Used to rotate the X-axis in the ecliptic plane.

[MNMAT1], [MNMAT2], [MNMAT3]: Three double precision, 3x3, orthogonal transformation matrices, scaled B1 and unitless. [MNMAT1] incorporates a rotation around the X reference axis through the angle of inclination of the earth's polar axis followed by a rotation around the Z axis (now perpendicular to the ecliptic) through the angle to the descending node on the ecliptic of the moon's orbit around the earth. [MNMAT2] rotates the system around the new X axis through the angle of inclination of the moon's true polar axis. [MNMAT3] completes the transformation to selenographic coordinates by rotating around the moon's polar axis from the descending node to the present position relative to that node.

[MOONMAT]: Double precision, 3x3, orthogonal transformation matrix, scaled B1 and unitless. $\underline{A}sg$ = [MOONMAT] $\underline{A}ref$ where A is a vector expressed in selenographic and reference coordinates respectively.

MR: Double precision angle of rotation of the moon around its true polar axis, scaled B0 in units of revolutions.

[NBSMMAT]: Double precision, 3x3, orthogonal transformation matrix, scaled B1 and unitless. $\underline{A}sm$ = [NBSMMAT] $\underline{A}nb$ where A is a vector expressed in stable member and navigation base (body) coordinates respectively.

OGC,IGC,MGC: Double precision commanded gimbal angles scaled B0 in units of revolutions or (equivalently) scaled B21 in units of gyro torque pulses of 2^{-21} revolutions each.

[REFSMMAT]: Double precision, 3x3 transformation matrix, scaled B1 and unitless. Defined such that $\underline{A}sm$ = [REFSMMAT] $\underline{A}rf$ where A is a vector expressed in stable member and reference coordinates respectively.

$\underline{R}$LS: See CONC section.

SINIGA, SINMGA, SINOGA: Double precision sines of the inner, middle and outer gimbal angles respectively, scaled B1 and unitless. Program notation $SINCDU_0$, $SINCDU_2$, $SINCDU_4$.

SINTH: Double precision sine, scaled B1 in the "ARCTAN" routine and B2 in the "ARCTRIG" routine.

[SMNBMAT]: Double precision, 3x3, orthogonal transformation matrix, scaled B1 and unitless. $\underline{A}nb$ = [SMNBMAT] $\underline{A}sm$ where A is a vector expressed in navigation base (body) and stable member coordinates respectively.

$STMP_i$ (i=0,2,4): Three double precision working storage registers in "LSPOS" scaled B0 in units of revolutions.

TEPHEM: Triple precision elapsed time between July 1.0 universal time and the time the LGC clock is zeroed, scaled B42 in units of centiseconds; included in the erasable load.

THETA: Double precision angle computed from SINTH and COSTH, scaled B0 in units of revolutions.

$\underline{T}$HETAD: See ATTM section.

TSsun: Double precision vector contents of the MPAC when return from "LSPOS", representing the unit position vector of the sun, scaled B1 and unitless.

UNITW: Double precision vector, scaled B0, which gives the polar axis in the reference coordinate system. The x component (program notation "-AYO") gives the "true to mean pole rotation about the -Y axis"; the y component (program notation "AXO") gives the "true to mean pole rotation about the +X axis"; included in the erasable load.

VEC1, VEC2: Working storages for the position vectors or unit position vectors in reference, selenographic, or geographic coordinates.

VMOON, VSUN: Double precision unit position vectors of the moon and sun, scaled B1 and unitless.

XNB, YNB, ZNB (XNBrf, YNBrf, ZNBrf; XNBsm, YNBsm, ZNBsm): Double precision unit vectors along the X, Y and Z navigation base axes (body axis) respectively, scaled B1, unitless, and expressed in reference or stable member coordinates.

XSM, YSM, ZSM (XSMrf, YSMrf, ZSMrf; XSMnb, YSMnb, ZSMnb): Double precision unit vectors along the X, Y and Z stable member axes respectively, scaled B1, unitless, and expressed in reference or navigation base coordinates.

YNBSAV, ZNBSAV: Working storages for the YNB and ZNB unit vectors in moon fixed coordinates.

DAPA

Digital Autopilot Control Routines

T5RUPT (Entered on program interrupt #2)

Proceed to address specified in T5ADR

DOT6RUPT (Entered on program interrupt #1)

Perform "T6JOBCHK"

Resume

T6JOBCHK If TIME6 $<$ 0 or TIME6 = +0, proceed to "CCSHOLE"

If TIME6 $>$ 0, return

i = NXT6AXIS

TIME6 = $T6NEXTTM_0$

NXT6AXIS = $T6NEXTAX_1$

$T6NEXTTM_0 = T6NEXTTM_2$

$T6NEXTAX_1 = T6NEXTAX_3$

$T6NEXTTM_2$ = K:posmaxsp

$T6NEXTAX_3$ = 0

If TIME6 $\geq$ K:T6lim, TIME6 = K:posmaxsp

If TIME6 $<$ K:T6lim:

- Perform "C13STALL" protecting the L and Q registers
- Switch bit 15 of channel 13 to 1 (enable TIME6 counter)
- If $T6NEXTTM_0 \geq$ K:T6lim, $T6NEXTTM_0$ = K:posmaxsp

If i = 0, perform "WRITEP" with TS = NEXTP

If i = 4, perform "WRITEU" with TS = NEXTU

If i = 13, perform "WRITEV" with TS = NEXTV

Return

<u>JTLST</u> TS = TIME6

If $T6NEXTTM_2$ < TS: (new jet time shorter than smallest remaining jet-on time)

TSt = TS - $T6NEXTTM_2$

TSa = NXT6AXIS

TIME6 = $T6NEXTTM_2$

NXT6AXIS = $T6NEXTAX_3$

$T6NEXTTM_2$ = $T6NEXTTM_0$

$T6NEXTAX_3$ = $T6NEXTAX_1$

$T6NEXTTM_0$ = TSt

$T6NEXTAX_1$ = TSa

Perform "C13STALL"

Switch bit 15 of channel 13 to 1 (Enable TIME6 counter)

Return

TS = TS + $T6NEXTTM_0$

If $T6NEXTTM_2$ < TS: (New jet time shorter than second remaining jet-on time in list)

TSt = TS - $T6NEXTTM_2$

TSa = $T6NEXTAX_1$

$T6NEXTTM_0$ = $T6NEXTTM_2$ - TIME6

$T6NEXTAX_1$ = $T6NEXTAX_3$

$T6NEXTTM_2$ = TSt

$T6NEXTAX_3$ = TSa

Return

$T6NEXTTM_2$ = $T6NEXTTM_2$ - TS (New time is longest)

Return

DAPIDLER If RCSFLAGS bit 13 = 0:

    Switch RCSFLAGS bit 13 to 1

    Establish "1/ACCSET" (pr 27)

  Perform "CHEKBITS"

  If DAPBOOLS bit 3 (ACCSOKAY) = 1:

    Proceed to "STARTDAP"

MOREIDLE Perform "QERRCALC"

  Perform "CALCPERR"

  T5ADR = "DAPIDLER"

  NEXTP, NEXTU and NEXTV = 00000_8

  Switch channels 5 and 6 to 00000_8 (all jets off)

  Switch bits 12-9 of channel 12 to 0 (gimbal drive bits)

  Set TIME5 to cause program interrupt #2 in 100 milliseconds

  Resume

CHEKBITS If bits 13 and 14 of channel 31 both = 1:

    Proceed to "MOREIDLE" (No longer in Held or Auto mode)

  If IMODES33 bit 6 = 1: (Internal DAP disable)

    Switch RCSFLAGS bit 3 (DSTEPONE) to 1

    Proceed to third step of "MOREIDLE"

  Perform "ALTDSPLY"

  If channel 30 bit 10 = 1, Proceed to "MOREIDLE"

  Return

ALTDSPLY Invert RCSFLAGS bit 4 (DSPLYALT)

  If RCSFLAGS bit 4 (DSPLYALT) = 1, proceed to "NEEDLER"

  If FLAGWRDO bit 15 (NEED2FLG) = 1:

    $\underline{AK}$ = - (OMEGAP, OMEGAQ, OMEGAR)

    Return

If FLAGWRDO bit 4 (NEEDLFLG) = 1:

$TStheta = THETAD_y - CDU_y$ (converted to one's comp form)

$TSpsi = THETAD_z - CDU_z$ (similarly converted)

$AK_y = M21\ TStheta + M22\ TSpsi$ (limited within $\pm \frac{1}{2}$)

$AK_z = M31\ TStheta + M32\ TSpsi$ (limited within $\pm \frac{1}{2}$)

$TSphi = THETAD_x - CDU_x$ (converted to one's comp form)

$AK_x = M11\ TStheta + TSphi$ (limited to within $\pm \frac{1}{2}$)

If FLAGWRDO bit 4 (NEEDLFLG) = 0, $\underline{AK}$ = -(PERROR, QERROR, RERROR)

Return

NEEDLER If RCSFLAGS bit 3 (DSTEPONE) = 1:

Switch bit 6 of channel 12 to 0 (Reset ICDU Error Counter enable discrete)

$\underline{AK} = 0$ (-0)

$\underline{EDRIVE} = 0$ (-0)

$CDU_iCMD = 0$ for i = x, y, z (-0)

Switch RCSFLAGS bit 3 (DSTEPONE) to 0

Switch RCSFLAGS bit 2 (DSTEPTWO) to 1

Return

If RCSFLAGS bit 2 (DSTEPTWO) = 1:

Switch bit 6 of channel 12 to 1

Switch RCSFLAGS bits 2 (DSTEPTWO) and 3 (DSTEPONE) to 0

Return

If bit 6 of channel 12 = 0: (ICDU Error Counters have been disabled)

Switch RCSFLAGS bit 3 (DSTEPONE) to 1

Return

NEEDLES Perform the indented steps for $i = z$, then y, then x

$TS = -AK_i$ K:ONETENTH

If $|TS| \geq$ K:eclim, TS = K:eclim signTS

$CDU_iCMD = CDU_iCMD +$ K:trvtoc $(TS - EDRIVE_i)$

$EDRIVE_i = TS$

Switch bits 13, 14 and 15 of channel 14 to 1 (send CDU_iCMD's)

Return

STARTDAP Perform "ZATTEROR"

TJ_0, TJ_1, and $TJ_2 = 0$

OMEGAP, OMEGAQ, and OMEGAR = 0

TRAPEDP, TRAPEDQ, and TRAPEDR = 0

AOSQ and AOSR = 0

ALPHAQ and ALPHAR = 0

$NEGU_0$ and $NEGU_2 = 0$

AOSQTERM and AOSRTERM = 0

QACCDOT and RACCDOT = 0

ALLOWGTS = 0

COTROLER = 0

INGTS = 0

QGIMTIMR and RGIMTIMR = 0

OLDPMIN and OLDQRMIN = 0

$PJETCTR_i = 0$ $(i = 1,2,3)$

Switch RCSFLAGS bits 1,5 (CALLGMBL),10 (PBIT),11 (QRBIT) to 0

OLDXFORP = CDU_x

OLDYFORP = CDU_y

OLDZFORQ = CDU_z

Switch RCSFLAGS bit 12 to 1

$SKIPUV_0$ and $SKIPUV_1$ = 4

TIME6 = K:posmaxsp

$T6NEXTTM_0$ and $T6NEXTTM_2$ = K:posmaxsp

$T6NEXTAX_1$ and $T6NEXTAX_3$ = 0

NXT6AXIS = 0

NEXTP, NEXTU, and NEXTV = 00000_8

DAPZRUPT = -10

NPTRAPS, NQTRAPS and NRTRAPS = 2

T5ADR = "PAXIS"

Set TIME5 to cause program interrupt #2 in 100 milliseconds

Resume

PAXIS — Set TIME5 to cause program interrupt #2 in (100 - TIME5) milliseconds

If DAPZRUPT > 0, proceed to "BAILOUT" with TS = 32000_8
(previous DAP cycle still in progress)

Perform "CHEKBITS"

CDUiTMP = CDUi (i = x,y,z)

$\underline{C}DUD = \underline{C}DUD - \underline{D}ELCDU$

TCP = TCP - 1

TCQR = TCQR - 1

Proceed to "PAXFILT"

RATELOOP — i = 2

TSq = 2 i

If $TJ_i = 0$, TSt = 0

If $TJ_i \neq 0$:

If $|TJ_i| \leq$ K:100msT6:

TSt = K:T6tosec TJ_i

$TJ_i = 0$

(If $TJ_i \neq 0$)

If $|TJ_i| >$ K:100msT6:

$TJ_i = TJ_i$ - K:100msT6 signTJ_i

TSt = K:0.1secB0 signTJ_i

TS_i = TSt NUMJETS$_i$

TSdnln = K:BIT10 TS_i

If TSdnln $\leq$ 0:

TSdnln = -TSdnln

TSq = TSq + 1

DOWNTORK$_{TSq}$ = DOWNTORK$_{TSq}$ + TSdnln

If i > 0:

i = i - 1

Proceed to 2nd step of "RATELOOP"

JETRATER = 1JACCR $(TS_1 + TS_2)$

JETRATEQ = 1JACCQ $(TS_1 - TS_2)$

BACKP JETRATEP = 1JACCP TS_0

TS = CDUxTMP

TSx = TS - OLDXFORP (converted to one's complement form)

OLDXFORP = TS

TRAPEDP = TRAPEDP - $\frac{1}{2}$ JETRATEP

TRAPEDQ = TRAPEDQ - $\frac{1}{2}$ (JETRATEQ + AOSQTERM)

TRAPEDR = TRAPEDR - $\frac{1}{2}$ (JETRATER + AOSRTERM)

TS = CDUyTMP

TSy = TS - OLDYFORP (converted to one's complement form)

OLDYFORP = TS

MEASRATE = (TSx + M11 TSy) / K:1d40 (limited)

TRAPEDP = TRAPEDP + MEASRATE - OMEGAP (limited)

DXERROR = DXERROR + (M11 TSy + TSx) - K:1d40 PLAST

TS = CDUzTMP

TSz = TS - OLDZFORQ (converted to one's complement form)

OLDZFORQ = TS

MEASRATE = (M21 TSy + M22 TSz) / K:1d40 (limited)

TRAPEDQ = TRAPEDQ + MEASRATE - OMEGAQ (limited)

DYERROR = DYERROR + (M21 TSy + M22 TSz) - K:1d40 QLAST

MEASRATE = (M31 TSy + M32 TSz) / K:1d40 (limited)

TRAPEDR = TRAPEDR + MEASRATE - OMEGAR (limited)

DZERROR = DZERROR + (M31 TSy + M32 TSz) - K:1d40 RLAST

If DAPBOOLS bit 13 (CSMDOCKD) = 1:

- n = DKOMEGAN
- na = DKKAOSN
- TRAPSIZE = DKTRAP

If DAPBOOLS bit 13 (CSMDOCKD) = 0:

- n = LMOMEGAN
- na = LMKAOSN
- TRAPSIZE = LMTRAP

If |TRAPEDP| > - TRAPSIZE:

- OMEGAP = OMEGAP + TRAPEDP / NPTRAPS (limited)
- TRAPEDP = 0
- NPTRAPS = n

NPTRAPS = NPTRAPS + 1

```
OMEGAP = OMEGAP + JETRATEP                                  (limited)
If |TRAPEDQ| > - TRAPSIZE:
     QKALERR = TRAPEDQ / NQTRAPS
     TRAPEDQ = 0
     OMEGAQ = OMEGAQ + QKALERR                              (limited)
     AOSQ = AOSQ + K:1d100ms QKALERR / (NQTRAPS + na)
     NQTRAPS = n
NQTRAPS = NQTRAPS + 1
OMEGAQ = OMEGAQ + JETRATEQ + AOSQTERM                       (limited)
If |TRAPEDR| > - TRAPSIZE:
     RKALERR = TRAPEDR / NRTRAPS
     TRAPEDR = 0
     OMEGAR = OMEGAR + RKALERR                              (limited)
     AOSR = AOSR + K:1d100ms RKALERR / (NRTRAPS + na)
     NRTRAPS = n
NRTRAPS = NRTRAPS + 1
OMEGAR = OMEGAR + JETRATER + AOSRTERM                       (limited)
If DAPBOOLS bit 8 (DRIFTBIT) = 1:
     ALPHAQ and ALPHAR = 0
     AOSQTERM and AOSRTERM = 0
     AOSQ and AOSR = 0                                      (sp)
If DAPBOOLS bit 8 (DRIFTBIT) = 0 :
     AOSQ = AOSQ + K:CALLCODE QACCDOT
     ALPHAQ = AOSQ
     AOSQTERM = K:aosint AOSQ
```

(If DAPBOOLS bit 8 (DRIFTBIT) = 0)

AOSR = AOSR + K:CALLCODE RACCDOT

ALPHAR = AOSR

AOSRTERM = K:aosint AOSR

Proceed to 2nd line of "SUPERJOB"

PAXFILT The following coding causes the "Resume" instruction to resume operations at "SUPERJOB" instead of at the job that was interrupted

If RCSFLAGS bit 5 (CALLGMBL) = 1:

Perform "ACDT+C12"

DAPARUPT = $ARUPT_{dp}$

DAPBQRPT = BRUPT

DAPBQRPT +1 = QRUPT

$DAPZRUPT_{dp}$ = $ZRUPT_{dp}$

BRUPT = Instruction stored at location SUPERJOB

ZRUPT = Address of SUPERJOB + 1

Resume

The purpose of this unusual manipulation of the "Resume" instruction is to establish "SUPERJOB" on a time-critical basis--immediately--while still allowing it to be interrupted by tasks and other interrupts.

SUPERJOB Proceed to "RATELOOP"

If QGIMTIMR = 0:

$NEGU_0$ = 0

QACCDOT = 0

Switch bits 9 and 10 of channel 12 to 0 (Q GTS drives)

QGIMTIMR = – K:posmaxsp

If QGIMTIMR > 0:

QGIMTIMR = QGIMTIMR – 1

If RGIMTIMR = 0:

$NEGU_2 = 0$

RACCDOT = 0

Switch bits 11 and 12 of channel 12 to 0 (R GTS drives)

RGIMTIMR = - K:posmaxsp

If RGIMTIMR > 0:

RGIMTIMR = RGIMTIMR - 1

$PJETCTR_i = PJETCTR_i - 1 \text{ sign} PJETCTR_i$ (i = 1,2,3) (zero unchanged)

If RCSFLAGS bit 12 = 1, proceed to "CHKVISFZ"

SKIPPAXS Switch RCSFLAGS bit 12 to 1

Proceed to "QRAXIS"

CHKVISFZ TS = - contents of channel 31 (all bits complemented)

If bits 9-12 of TS all = 0:

$TS = 00000_8$

Proceed to "TSNEXTP"

i = bits 9-12 of TS shifted right 8 to bit positions 1-4

$ROTINDEX = K:INDXYZ_i$ (if somehow i is illegal, proceed to two steps before "TSNEXTP")

TRYUORV NUMBERT = 6

Perform "SELECTP" with i = NUMBERT

If NUMBERT = 6: (required jets are all available)

TS = POLYTEMP

Proceed to "TSNEXTP"

If ROTINDEX ≤ 5: (Principal axis translation cannot be accomplished because of jet failure; try tacking along an appropriate U or V axis)

$TS = 00000_8$

Invert RCSFLAGS bit 1

If RCSFLAGS bit 1 = 1, $TS = 00001_8$

(If ROTINDEX $\leq$ 5)

ROTINDEX = ROTINDEX + TS + 4

Proceed to "TRYUORV"

If NUMBERT $\geq$ 4: (One combination of jets is available to accomplish a U or V axis translation)

TS = POLYTEMP

Proceed to "TSNEXTP"

Perform "ALARM" with TS = 02001_8

Invert RCSFLAGS bit 1

TS = 00000_8

TSNEXTP NEXTP = TS

If bit 13 of channel 31 = 1 and DAPBOOLS bit 9 (XOVINHIB) = 1:

Proceed to "PURGENCY" (Auto with X-axis override disabled)

If bit 13 of channel 31 = 1 or DAPBOOLS bit 15 (PULSES) = 0:

Proceed to "DETENTCK" (Minimum impulse not allowed or not specified by DAPBOOLS)

(otherwise, minimum impulse mode)

PERROR = 0

$CDUD_x = CDU_x$

If OLDPMIN > 0: (not returned to detent since jets fired)

TS = - contents of channel 31 (all bits complemented)

OLDPMIN = bits 3 and 4 of TS

Proceed to "JETSOFF"

(Otherwise, OLDPMIN = 0, indication that no yaw commands were present during last DAP cycle)

If bits 3 and 4 of channel 31 both = 1, proceed to "JETSOFF"

If bit 4 of channel 31 = 0, TJ_0 = - K:minimptj (-P)

If bit 3 of channel 31 = 0, TJ_0 = K:minimptj (+P)

OLDPMIN = 1

NUMBERT = 4

If FLAGWRD5 bit 5 (AORBSFLG), NUMBERT = 5

Proceed to "PJETSLEC"

ZEROENBL $SAVEHAND_0$ = RHCQ

$SAVEHAND_1$ = RHCR

RHCP, RHCQ, and RHCR = 0

Perform "C13STALL" with interrupts inhibited

Switch bits 8 and 9 of channel 13 to 1
(Start RHC read and enable RHC counters)

Return

DETENTCK TS_{ch31} = channel 31

If TS_{ch31} bit 15 = 1 and DAPBOOLS bit 12 (OURRCBIT) = 0:

Proceed to "PURGENCY"

If TS_{ch31} bit 15 = 0 and DAPBOOLS bit 12 (OURRCBIT) = 1:

Switch RCSFLAGS bit 9 (JUSTIN) to 1

Proceed to "RATERROR"

If TS_{ch31} bit 15 = 0 and DAPBOOLS bit 12 (OURRCBIT) = 0:

Switch RCSFLAGS bit 9 (JUSTIN) to 1

PERROR = 0

Switch DAPBOOLS bit 12 (OURRCBIT) to 1

$DXERROR_{dp}$ = 0

$DYERROR_{dp}$ = 0

$DZERROR_{dp}$ = 0

PLAST = 0

QLAST = 0

RLAST = 0

(If TS_{ch31} bit 15 = 0 and DAPBOOLS bit 12 (OURRCBIT) = 0)

RHCQ = 0

RHCR = 0

Switch RCSFLAGS bits 10 (PBIT) and 11(QRBIT) to 0

Perform "ZEROENBL"

Proceed to "JETSOFF"

If TS_{ch31} bit 15 = 1 and DAPBOOLS bit 12 (OURRCBIT) = 1:

If RCSFLAGS bit 9 (JUSTIN) = 1:

If channel 31 bit 13 = 0, proceed to "RATEDAMP"

Switch RCSFLAGS bits 9 & 11 (JUSTIN & QRBIT) to 0

Proceed to "RATEDAMP"

If RCSFLAGS bit 10 (PBIT) = 1, proceed to "RATEDAMP"

If RCSFLAGS bit 11 (QRBIT) = 1, proceed to "RATEDAMP"

Switch DAPBOOLS bit 12 (OURRCBIT) to 0

If channel 31 bit 13 = 1:

$CDUD_x = CDU_x$

Proceed to "PURGENCY"

Perform "ZATTEROR"

Proceed to "PURGENCY"

<u>RATERROR</u> $CDUD_x = CDU_x$

TSp = PLAST

PLAST = STIKSENS RHCP ((|RHCP| - 1) + K:LINRAT)

TS1 = PLAST - TSp

Perform "ZEROENBL"

EDOT = OMEGAP - PLAST

If $|TS1| >$ RATEDB:

TCP = K:40cyc

Proceed to "PEGI"

If RCSFLAGS bit 10 (PBIT) = 1, proceed to "PEGI"

E = DXERROR

PERROR = DXERROR

Proceed to third line of "PURGENCY"

RATEDAMP RHCP = 0

Proceed to "RATERROR"

PEGI $CDUD_x = CDU_x$

$DXERROR_{dp} = 0$

PERROR = 0

ABSEDOTP = |EDOT|

If ABSEDOTP > RATEDB and if TCP > 0:

Switch RCSFLAGS bit 10 (PBIT) to 1

Skip next step

Switch RCSFLAGS bit 10 (PBIT) to 0

$TJ_0 = -2$ K:25B5 EDOT 1dANETP (limited)

If ABSEDOTP > 2JETLIM:

NUMBERT = 6

Proceed to "PJETSLEC"

$TJ_0 = 2\ TJ_0$

NUMBERT = 4

If FLAGWRD5 bit 5 (AORBSFLG) = 1, NUMBERT = 5

Proceed to "PJETSLEC"

CALCPERR $E = M11 (CDU_y - CDUD_y)$

$E = E + CDU_x - CDUD_x + DELPEROR$

PERROR = E

Return

PURGENCY Perform "CALCPERR"

EDOT = OMEGAP - OMEGAPD

AXISCTR = -1

If DAPBOOLS bit 13 (CSMDOCKD) = 1:

    Perform "SPSRCS" with interrupts inhibited

    If $TJ_0 = 0$:

        Invert FLAGWRD5 bit 5 (AORBSFLG)

        Proceed to "JETSOFF"

    NUMBERT = 4

    If FLAGWRD5 bit 5 (AORBSFLG) = 1, NUMBERT = 5

    Proceed to "PJETSLEC"

SENSETYP = 0

Perform "TJETLAW" with interrupts inhibited

NUMBERT = 6

If FIREFCT $\geq$ K:mFOURDEG or if $|TJ_0| \leq$ K:160msT6:

    NUMBERT = 4

    If FLAGWRD5 bit 5 (AORBSFLG) = 1, NUMBERT = 5

PJETSLEC TS = 1

If $TJ_0 = 0$, proceed to "JETSOFF"

If $TJ_0 < 0$, TS = 0

ABSTJ = $|TJ_0|$

ROTINDEX = TS

Perform "SELECTP" with i = 6

If NUMBERT = 6, TS = 4 (jets all available for 4-jet rotation)

If NUMBERT ≠ 6, TS = 2

$NUMJETS_0$ = TS

Perform "WRITEP" with TS = POLYTEMP (turn on rotation)

If ABSTJ ≥ K:150msT6, proceed to "QRAXIS"

If ABSTJ < K:150msT6 - K:136msT6:

  ABSTJ = K:150msT6 - K:136msT6

  TJ_0 = K:MINTIMES signTJ_0

Inhibit interrupts

$T6NEXTTM_2$ = ABSTJ

$T6NEXTAX_3$ = 0 (0 indicates P-axis)

Perform "JTLST"

Switch RCSFLAGS bit 12 to 0

Invert FLAGWRD5 bit 5 (AORBSFLG)

Release interrupt inhibit

Proceed to "QRAXIS"

<u>JETSOFF</u> Perform "WRITEP" with TS = NEXTP

TJ_0 = 0

Proceed to "QRAXIS"

<u>WRITEP</u> Set bits 1-8 of channel 6 = bits 1-8 of TS

Return

<u>SELECTP</u> TSa = $K{:}quadsP_{NUMBERT}$

TSb = K:typmanP$_{ROTINDEX}$

POLYTEMP = TSa $\wedge$ TSb (logic "and" function)

If any of the binary bits that are 1 in POLYTEMP are also 1 in CH6MASK (at least one of the required jets has been failed)

If i = 0: (i cannot be zero in selection of translation jets)

Perform "ALARM" with TS = 02003_8 (rotation failure)

Proceed to "JETSOFF"

i = i - 1

NUMBERT = i

Proceed to "SELECTP"

Return

QRAXIS EDOTR = OMEGAR - OMEGARD (limited)

EDOTQ = OMEGAQ - OMEGAQD (limited)

If channel 31 bit 13 = 0:

If DAPBOOLS bit 12 (OURRCBIT) = 1, skip next step

Perform "QERRCALC"

If COTROLER = 0, proceed to "TRYGTS"

If COTROLER > 0, proceed to "GTS"

Proceed to "RCS"

QERRCALC $TSy = CDU_y - CDUD_y$
$TSz = CDU_z - CDUD_z$ (converted to one's comp. form)

QERROR = M21 TSy + M22 TSz + DELQEROR

RERROR = M31 TSy + M32 TSz + DELREROR

Return

RCS COTROLER = 0

OMEGAU = - COEFFQ EDOTQ + COEFFR EDOTR (limited)

OMEGAV = COEFFQ EDOTQ + COEFFR EDOTR (limited)

If channel 31 bit 7 = 0:

    TS = 5

    Proceed to "+XORULGE"

If channel 31 bit 8 = 0:

    TS = 4

    Proceed to "+XORULGE"

If DAPBOOLS bit 6 (ULLAGER) = 1:

    TS = 5

    Proceed to "+XORULGE"

NEXTU = 0

NEXTV = 0

If DAPBOOLS bit 8 (DRIFTBIT) = 1:

    SENSETYP = 0

    Proceed to 3rd step of "TSNEXTS"

SENSETYP = 0

If FLGWRD10 bit 13 (APSFLAG) = 1, SENSETYP = 2

Proceed to 3rd step of "TSNEXTS"

+XORULGE ROTINDEX = TS

SENSETYP = ROTINDEX - 3

If DAPBOOLS bit 11 (ACC4OR2X) = 1:

    TS1 = 4 and skip next 3 steps

If DAPBOOLS bit 10 (AORBTRAN) = 1:

    TS1 = 3 and skip next step

TS1 = 2

NUMBERT = TS1

Perform "SELCTSUB"

If POLYTEMP > 0, proceed to "TSNEXTS"

Perform "ALARM" with TS = 02002_8

TSNEXTS NEXTU = bits 8,7, 4 and 3 of POLYTEMP

NEXTV = bits 6,5,2 and 1 of POLYTEMP

(Note that translation codes in NEXTU and NEXTV may not be implemented at the same time, but each cell contains codes for a jet pair on diagonally opposite quads.)

If channel 31 bit 13 = 1, proceed to "ATTSTEER"

If DAPBOOLS bit 15 (PULSES) = 0, proceed to "CHEKSTIK"

(Otherwise, minimum impulse)

Perform "ZATTEROR" with interrupts inhibited

QERROR = 0

RERROR = 0

TS = - contents of channel 31 (all bits complemented)

If OLDQRMIN > 0: (not returned to detent since jets fired)

OLDQRMIN = bits 1,2,5 and 6 of TS (+Q, -Q, +R, -R)

Proceed to "XTRANS"

(Otherwise, OLDQRMIN = 0, indication that no Q or R commands were present during the last DAP cycle)

If bits 1,2,5 and 6 of TS all = 0, proceed to "XTRANS"

If bit 1 of TS = 1: (+Q)

TJ_1 = K:pTJMINT6 (U)

TJ_2 = - K:pTJMINT6 (V)

Proceed to "MINQR"

If bit 2 of TS = 1: (-Q)

TJ_1 = - K:pTJMINT6 (U)
TJ_2 = K:pTJMINT6 (V)
Proceed to "MINQR"

If bit 5 of TS = 1: (+R)

TJ_1 = K:pTJMINT6 (U)
TJ_2 = K:pTJMINT6 (V)
Proceed to "MINQR"

If bit 6 of TS = 1: (-R)

TJ_1 = - K:pTJMINT6 (U)
TJ_2 = - K:pTJMINT6 (V)

MINQR RETJADR = "MINRTN

OLDQRMIN = 1

AXISCTR = 1

MINRTN If DAPBOOLS bit 13 (CSMDOCKD) = 1:

$TJ_{AXISCTR}$ = sign$TJ_{AXISCTR}$ K:60msT6

NUMBERT = 2

If DAPBOOLS bit 10 (AORBTRAN) = 1, NUMBERT = 3

Proceed to "AFTERTJ"

CHEKSTIK INGTS = 0

COTROLER = -1

If TS_{ch31} bit 15 = 0, proceed to "RHCACTIV" (TS_{ch31} was loaded in "DETENTCK")

If DAPBOOLS bit 12 (OURRCBIT) = 0:

Proceed to "ATTSTEER"

Switch RCSFLAGS bit 9 (JUSTIN) to 0

$SAVEHAND_0$ = 0

$SAVEHAND_1$ = 0

<u>RHCACTIV</u> TSq = QLAST

QLAST = STIKSENS $SAVEHAND_0$ (($|SAVEHAND_0|$ - 1) + K:LINRAT)

TS3 = QLAST - TSq

TSr = RLAST

RLAST = STIKSENS $SAVEHAND_1$ (($|SAVEHAND_1|$ - 1) + K:LINRAT)

TS4 = RLAST - TSr

QRATEDIF = OMEGAQ - QLAST

RRATEDIF = OMEGAR - RLAST

URATEDIF = - COEFFQ QRATEDIF + COEFFR RRATEDIF (limited)

VRATEDIF = COEFFQ QRATEDIF + COEFFR RRATEDIF (limited)

If $|TS3|$ > RATEDB, proceed to "ENTERUV"

If $|TS4|$ > RATEDB, proceed to "ENTERUV"

If RCSFLAGS bit 11 (QRBIT) = 1:

    Proceed to 2nd step of "ENTERUV"

Proceed to "ATTSTEER"

<u>ENTERUV</u> TCQR = K:40cyc

Inhibit interrupts

Perform "ZATTEROR"

Release interrupt inhibit

$DYERROR_{dp}$ = 0

$DZERROR_{dp}$ = 0

If $|URATEDIF|$ < RATEDB:

    If $|VRATEDIF|$ < RATEDB:

        Proceed to "TOPSEUDO"

    URATEDIF = 0

    Proceed to "QRTIME"

If |VRATEDIF| < RATEDB:

VRATEDIF = 0

QRTIME If TCQR > 0:

Switch RCSFLAGS bit 11 (QRBIT) to 1

Skip next step

TOPSEUDO Switch RCSFLAGS bit 11 (QRBIT) to 0

RETJADR = "BACKHAND"

AXISCTR = 1

BACKHAND NUMBERT = 4

If $SKIPUV_{AXISCTR} = 0$:

$SKIPUV_{AXISCTR} = 4$

If AXISCTR = 0, proceed to "CLOSEOUT"

AXISCTR = AXISCTR - 1

Proceed to location stored in RETJADR

TS = URATEDIF

If AXISCTR = 1, TS = VRATEDIF

i = 16 AXISCTR + 2 (2 or 18)

If TS < 0, i = i + 1 (index the proper 2-jet acceleration in the direction of desired acceleration)

$TSt = - K{:}bksel\ TS\ 1dANET_i$

If |TSt| ≥ K:bklim, TSt = TSt / 3 (still > K:150msQR)

i = AXISCTR + 1

$TJ_i = TSt$

Proceed to "AFTERTJ"

ATTSTEER UERROR = - COEFFQ QERROR + COEFFR RERROR (limited)

VERROR = COEFFQ QERROR + COEFFR RERROR (limited)

```
TJLAW    RETJADR = "TJLAW4"

         AXISCTR = 1

TJLAW4   If SKIPUV_AXISCTR = 0:

              SKIPUV_AXISCTR = 4

              If AXISCTR = 0, proceed to "CLOSEOUT"

              AXISCTR = AXISCTR - 1

              Proceed to location stored in RETJADR

         If AXISCTR = 1:

              E = VERROR

              EDOT = OMEGAV

         If AXISCTR = 0:

              E = UERROR

              EDOT = OMEGAU

         If DAPBOOLS bit 13 (CSMDOCKD) = 1:

              If DAPBOOLS bit 14 (USEQRJTS) = 0, COTROLER = 8191

              Perform "SPSRCS"  with interrupts inhibited

              NUMBERT = 4

              Proceed to "AFTERTJ"

         Perform "TJETLAW"

AFTERTJ  If FLAGWRD5 bit 13 (SNUFFER) = 0, proceed to "DOROTAT"

         If FLGWRD10 bit 13 (APSFLAG) = 1, proceed to "DOROTAT"

         If DAPBOOLS bit 8 (DRIFTBIT) = 0, proceed to "XTRANS"

DOROTAT  i = AXISCTR + 1

         If TJ_i = 0:      (no rotation command; execute trans command)

              If AXISCTR = 1:

                   Perform "WRITEV" with TS = NEXTV
```

(If AXISCTR = 1:)

AXISCTR = 0

Proceed to address specified by RETJADR

Perform "WRITEU" with TS = NEXTU

Proceed to "CLOSEOUT"

TS = 2

If $TJ_i < 0$, TS = 0

ABSTJ = $|TJ_i|$

ROTINDEX = AXISCTR + TS (0,1,2,3)

If ABSTJ > K:150msQR

Perform "SELCTSUB"

If AXISCTR = 1, perform "WRITEV" with TS = POLYTEMP

If AXISCTR = 0, perform "WRITEU" with TS = POLYTEMP

Proceed to "FEEDBACK"

If ABSTJ < K:pTJMINT6:

ABSTJ = K:pTJMINT6

i = AXISCTR + 1

TJ_i = K:pTJMINT6 signTJ_i

NUMBERT = 0

If bit 1 of channel 4 = 1, NUMBERT = 1

(Bit 1 of channel 4 is used here as sort of a random number generator; it is part of the computer clock and oscillates at a frequency of 3200 pps.)

If SENSETYP > 0, NUMBERT = SENSETYP - 1

Perform "SELCTSUB"

If AXISCTR = 1, TS = 13

If AXISCTR = 0, TS = 4

Inhibit interrupts

$T6NEXTAX_3$ = TS

If $T6NEXTAX_3$ = 13, perform "WRITEV" with TS = POLYTEMP

If $T6NEXTAX_3$ = 4, perform "WRITEU" with TS = POLYTEMP

$T6NEXTTM_2$ = ABSTJ

Perform "JTLST"

Release interrupt inhibit

$SKIPUV_{AXISCTR}$ = 0 (cause this axis to be skipped next cycle)

FEEDBACK i = AXISCTR + 1

If NUMBERT > 3, $NUMJETS_i$ = 2

If NUMBERT ≤ 3, $NUMJETS_i$ = 1

If AXISCTR = 0, proceed to "CLOSEOUT"

AXISCTR = AXISCTR - 1

Proceed to address specified in RETJADR

XTRANS TJ_1 = 0

TJ_2 = 0

Inhibit interrupts

If $SKIPUV_0 \neq 0$, perform "WRITEU" with TS = NEXTU

$SKIPUV_0$ = 4

If $SKIPUV_1 \neq 0$, perform "WRITEV" with TS = NEXTV

$SKIPUV_1$ = 4

Release interrupt inhibit

Proceed to "CLOSEOUT"

WRITEU Set bits 3,4,7 and 8 of channel 5 = bits 3,4,7 and 8 of TS

Return

WRITEV Set bits 1,2,5 and 6 of channel 5 = bits 1,2,5 and 6 of TS

Return

SELCTSUB TSa = K:quadsQR$_{NUMBERT}$

TSb = K:typmanQR$_{ROTINDEX}$

POLYTEMP = TSa $\wedge$ TSb (logic "and" function)

If any of the binary bits that are 1 in POLYTEMP are also 1 in CH5MASK (at least one of the required jets is flagged as failed):

NUMBERT = 3

Proceed to "FAILOOP"

Return

FAILOOP TSa = K:quadsQR$_{NUMBERT}$

TSb = K:typmanQR$_{ROTINDEX}$

POLYTEMP = TSa $\wedge$ TSb (logic "and" function)

If POLYTEMP $\wedge$ CH5MASK $\neq 00000_8$:

If NUMBERT = 0:

Perform "ALARM" with TS = 02004_8

If AXISCTR = 0:

$TJ_1 = 0$

Perform "WRITEU" with TS = NEXTU

Proceed to "CLOSEOUT"

$TJ_2 = 0$

Perform "WRITEV" with TS = NEXTV

AXISCTR = 0

Proceed to address specified by RETJADR

NUMBERT = NUMBERT - 1

Proceed to "FAILOOP"

Return (to routine that called "SELCTSUB")

TRYGTS If DAPBOOLS bit 14 (USEQRJTS) = 1, proceed to "RCS"

If ALLOWGTS = 0, proceed to "RCS"

If channel 5 = 00000_8, proceed to "GTS"

If INGTS = 0, proceed to "RCS"

Perform "TIMEGMBL" with interrupts inhibited

INGTS = 0

Proceed to "RCS"

GTS COTROLER = -1

$SKIPUV_0$ = 4

$SKIPUV_1$ = 4

INGTS = 2

QGIMTIMR = 2

RGIMTIMR = 2

QRCNTR = 2

TS_L = AOSR (rescaled to B-3) (limited)

WCENTRAL = EDOTR

ACENTRAL = TS_L

KCENTRAL = RDAPK

If KCENTRAL = 0:

  K2THETA = 0

  Proceed to "NEGUSUM"

TS = RERROR

ALGORTHM K2THETA = KCENTRAL TS

A2CNTRAL = $ACENTRAL^2$ / (2 KCENTRAL) (limited)

K2CNTRAL = WCENTRAL (rescaled to B3)

FUNCTION = K2CNTRAL + A2CNTRAL signACENTRAL

DEL = 1 signFUNCTION

If $|\text{FUNCTION}| < 2^{-25}$, DEL = 0

K2CNTRAL = DEL K2CNTRAL + A2CNTRAL

A2CNTRAL = K2CNTRAL - A2CNTRAL / 3

K2THETA = K2THETA + ACENTRAL A2CNTRAL

FUNCTION = KCENTRAL K2CNTRAL

K2CNTRAL = DEL K2CNTRAL

If DEL = 0, proceed to "NEGUSUM"

RSTOFGTS Perform "GTSQRT"

K2CNTRAL = TS_{sqrt} K2CNTRAL

SHFTFLAG = ININDEX / 2 + SHFTFLAG

TS = $2^{-SHFTFLAG}$ K2CNTRAL

K2THETA = K2THETA + TS (values less than 2^{-28} are considered to be zero)

NEGUSUM TS_1 = $NEGU_{QRCNTR}$

$NEGU_{QRCNTR}$ = 1 · signK2THETA

If K2THETA < 2^{-28}, $NEGU_{QRCNTR}$ = 0

TS_2 = TS_1 $NEGU_{QRCNTR}$ (old NEGU value times new NEGU value)

If TS_2 < 0: (If a reversal of gimbal drive direction is called for)

$QACCDOT_{QRCNTR}$ = 0

If QRCNTR > 0:

Set bits 11 & 12 of channel 12 = 0

Skip next step

Set bits 9 & 10 of channel 12 = 0

If $TS_2 \leq 0$, set bit 5 (CALLGMBL) of RCSFLAGS = 1

If QRCNTR = 2:

QRCNTR = 0

WCENTRAL = EDOTQ

ACENTRAL = AOSQ (rescaled to B-3) (limited)

KCENTRAL = QDAPK

If KCENTRAL = 0:

K2THETA = 0

Proceed to "NEGUSUM"

TS = QERROR

Proceed to "ALGORTHM"

<u>CLOSEOUT</u> This routine returns processing to Task status via the EDRUPT instruction and then uses the following equations to resume operation at the job whose address is in DAPZRUPT (thus ending a prolonged semi-interrupt of that job)

$ARUPT_{dp}$ = $DAPARUPT_{dp}$

BRUPT = DAPBQRPT

Q = DAPBQRPT +1

$ZRUPT_{dp}$ = $DAPZRUPT_{dp}$

DAPZRUPT = - K:posmaxsp

Resume

ACDT+C12 QACCDOT = - $NEGU_0$ ACCDOTQ

RACCDOT = - $NEGU_2$ ACCDOTR

TS = 00000_8

If $NEGU_0$ = 1, switch bit 10 of TS to 1

If $NEGU_0$ = -1, switch bit 9 of TS to 1

If $NEGU_2$ = 1, switch bit 12 of TS to 1

If $NEGU_2$ = -1, switch bit 11 of TS to 1

Set bits 9-12 of channel 12 = bits 9-12 of TS

Switch bit 5 of RCSFLAGS (CALLGMBL) to 0

Return

TIMEGMBL ALLOWGTS = 1

$NEGU_2$ = 0

If ACCDOTR ≤ 0 or if AOSR = 0, proceed to "TIMQGMBL"

TS = - K:0.4gts AOSR

$NEGU_2$ = -1 signTS (If TS = 0, NEGU will be zeroed below)

If |TS| ≥ 2 ACCDOTR:

RGIMTIMR = K:OCT31

ALLOWGTS = 0

Proceed to "TIMQGMBL"

TSt = |TS| K:OCT00240 / ACCDOTR (units of 100 milliseconds)

If TSt < K:gtstmin, $NEGU_2$ = 0

If TSt ≥ K:gtstmin, RGIMTIMR = TSt

TIMQGMBL $NEGU_0$ = 0

If ACCDOTQ ≤ 0 or if AOSQ = 0, proceed to "DONEYET2"

TS = - K:0.4gts AOSQ

$NEGU_0$ = -1 signTS

If |TS| ≥ 2 ACCDOTQ:

QGIMTIMR = K:OCT31

ALLOWGTS = 0

Proceed to "DONEYET2"

TSt = |TS| K:OCT00240 / ACCDOTQ (units of 100 milliseconds)

If TSt < K:gtstmin, $NEGU_0$ = 0

If TSt ≥ K:gtstmin, QGIMTIMR = TSt

DONEYET2 Perform "ACDT+C12"

Return

ALLCOAST Perform "STOPRATE"

AOSQ and AOSR = 0

ALPHAQ and ALPHAR = 0

AOSQTERM and AOSRTERM = 0

Switch DAPBOOLS bit 8 (DRIFTBIT) to 1

Perform "RESTORDB"

Return

ZATTEROR CDUD = CDU

STOPRATE OMEGAPD, OMEGAQD and OMEGARD = 0

DELCDU = 0

DELPEROR, DELQEROR, and DELREROR = 0

Return

DAPT4S (Entered every 240 milliseconds from "T4RUPT"; also called GPMATRIX. This calculates the gimbal rate to body rate matrix)

M11 = $\sin_{sp} CDU_z$

COSMG = $\cos_{sp} CDU_z$

$M22 = \sin_{sp} CDU_x$

$M31 = - COSMG\ M22$

$M32 = \cos_{sp} CDU_x$

$M21 = COSMG\ M32$

Return

RCSMONIT (Entered every 480 milliseconds from "T4RUPT", also called RCSMON)

TS = - contents of channel 32 (all bits complemented)

TSq = bits 1-8 of TS (RCS thruster fail discretes)

$TS = 00000_8$

For i = 1,2,3,4,5,6, 7 and 8: If bit i of TSq ≠ bit i of PVALVEST, switch bit i of TS to 1

If $TS = 00000_8$, return (no change)

For i = 8 through 1, in that order, examine bit i of TS; upon finding the first bit that is a "1", continue at next step with i = that bit number.

If bit i of PVALVEST = 1, proceed to "VOPENED"

Switch bit of CH5MASK indicated by $K{:}5FAILTAB_i$ to 1

Switch bit of CH6MASK indicated by $K{:}6FAILTAB_i$ to 1

Switch bit i of PVALVEST to 1

Establish "1/ACCJOB" (pr 27)

Return

VOPENED Switch bit of CH5MASK indicated by $K{:}5FAILTAB_i$ to 0

Switch bit of CH6MASK indicated by $K{:}6FAILTAB_i$ to 0

Switch bit i of PVALVEST to 0

Establish "1/ACCJOB" (pr27)

Return

GTSQRT If FUNCTION ≤ 0: (bad argument for square root)

SHFTFLAG = 0

$TS_{sqrt} = 0$

Return

SHFTFLAG = 0

If FUNCTION $< 2^{-20}$: (most significant half = 0)

SHFTFLAG = 7

FUNCTION = FUNCTION 2^{14} (operate on least significant half)

ININDEX = 12

SCALLOOP If $2^{-ININDEX - 6}$ - FUNCTION ≤ 0:

ININDEX = ININDEX - 2

If ININDEX = 0, Skip next step

Proceed to "SCALLOOP"

TS = FUNCTION / $2^{-6 - ININDEX}$ (rescaled for square root accuracy)

HALFARG = TS / 2

TS_{sqrt} = K:ROOTHALF

If HALFARG $\geq \frac{1}{4}$, $TS_{sqrt} = 1$

$TS_{sqrt} = \frac{1}{2} TS_{sqrt} + HALFARG / TS_{sqrt}$ (Newton algorithm)

$TS_{sqrt} = \frac{1}{2} TS_{sqrt} + HALFARG / TS_{sqrt}$

$TS_{sqrt} = \frac{1}{2} TS_{sqrt} + HALFARG / TS_{sqrt}$

TSsqrt = TSsqrt 2^{-3}

Return

Quantities In Computations

1dANET$_i$ (i = 2,3,18,19): See DAPB section.

1dANETP: Single precision inverse of the acceleration expected from the simultaneous firing of two P-axis RCS jets scaled B8 in units of seconds squared per revolution, (also called 1dANET$_{-14}$).

1JACCP, 1JACCQ, 1JACCR: See DAPB section.

2JETLIM: Single precision rate limit used in "PEGI" to decide if two or four jets should be used for a P-axis rotation, scaled B-3 in units of revolutions per second. Actually stored as a negative quantity with the program notation -2JETLIM, but interpreted in this document as positive.

A2CNTRAL: Double precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it first contains $\alpha^2/2K$ scaled B3 with units rev/sec; when next used it contains $-\Delta\omega + \alpha^2/3K$ scaled B3 with units revs/sec.

ABSEDOTP: Temporary single precision storage for the magnitude of EDOT in "PEGI" scaled B-3 in units of revolutions per second. Actually ABSEDOTP = the magnitude of EDOT minus one least increment (not compensated for CCS instruction)

ABSTJ: Temporary storage for the magnitude of TJ$_i$, scaled B10 in units of centiseconds.

ACCDOTQ, ACCDOTR: See DAPB section.

ACENTRAL: Single precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it contains α, the disturbing acceleration scaled B-3 with units of revs/sec^2.

AK: Single precision vector containing the desired setting of the FDAI error needles, scaled B-1 in units of revolutions for attitude errors and B-3 in units of revs/sec for rate errors.

ALLOWGTS: A single precision, binary switch set to allow entry into the Gimbal Trim System attitude control law if other conditions are satisfied, scaled B14 and unitless.

ALPHAQ, ALPHAR: Single precision storage for the most significant halves of AOSQ and AOSR for down telemetry, scaled B-2 in units of revolutions per second squared.

AOSQ, AOSR: Double precision disturbing acceleration due to thrust vector/c.g. offset or other external torques, scaled B-2 in units of revolutions per second squared.

AOSQTERM, AOSRTERM: Single precision addition to vehicle rate that would be added during one 100 millisecond period as a result of disturbing accelerations, scaled B-3 in units of revolutions per second.

ARUPT, BRUPT, ZRUPT, and Q: Special cells used with the interrupt and resume instructions. Q is also the return address register.

AXISCTR: Single precision index used to differentiate among the three axes, scaled B14 and unitless. A value of 1 corresponds to the V axis, 0 to the U axis and -1 to the P axis.

CDU (CDU_x, CDU_y, CDU_z): Single precision vector containing the measured values of the IMU gimbal angles (outer, inner and middle gimbal in X,Y and Z components, respectively), scaled B-1 in units of revolutions and stored in two's complement form. Each component is an LGC input counter incremented directly from the Coupling Data Unit in response to changes in the IMU gimbal angles.

CDU_xCMD, CDU_yCMD, CDU_zCMD: See IMUC section.

CDUiTMP (i = x,y,z): Single precision storage locations for values of CDU_x, CDU_y and CDU_z respectively. Used to obtain a synchronous sample of the CDU values for rate estimation.

CDUD: Single precision vector interface with steering and attitude maneuver routines containing the desired values for the IMU gimbal angles (outer, inner and middle gimbal angles in x,y and z components, respectively), scaled B-1 in units of revolutions and stored in two's complement form.

CH5MASK, CH6MASK: Single precision octal flagwords whose individual bits (1 through 8 only) are set to indicate jet failures (in "RCSMONIT"). See description of K:5FAILTAB and K:6FAILTAB.

COEFFQ: Single precision negative of the quantity used for the first column of the matrix taking a vector expressed in Q,R coordinates to one expressed in the non-orthogonal U', V' coordinates. Scaled B0 and unitless.

COEFFR: Single precision quantity used for the second column of the matrix taking a vector expressed in Q, R coordinates to one expressed in the non-orthogonal U', V' coordinates. Scaled B0 and unitless.

COSMG: Single precision cosine of middle gimbal angle, scaled B0 and unitless.

COTROLER: Single precision variable cell scaled B14 and unitless controlling access to the Q, R-axis gimbal trim system.

DAPBOOLS: Single precision flagword whose individual bits have the following meanings:

Bit	Mnemonic	Meaning when set (1)	Meaning when clear (0)
15	PULSES	Minimum impulse command mode	Not minimum impulse
14	USEQRJTS	GTS not allowed	GTS allowed
13	CSMDOCKD	CSM attached to LM Backup SPS DAP	CSM not attached Normal LM DAP
12	OURRCBIT	Still in Rate Command Mode	Not in Rate Command
11	ACC4OR2X	4-jet P-axis translation requested	2-jet P-axis translation
10	AORBTRAN	X-trans B system	X-trans A system
9	XOVINHIB	LPD phase; X-axis overide disabled	Not in Landing Point Designation Phase
8	DRIFTBIT	Assume that offset acceleration is zero	Offset acceleration likely
7	RHCSCALE	Normal RHC scaling	Fine RHC scaling
6	ULLAGER	Internal ullage request	No program ullage request
5	DBSLECT2	Bits 4 and 5 are used together to select attitude deadbands. The meanings are:	
4	DBSELECT		

bit 5	bit 4	Deadband
1	1	5°
1	0	5°
0	1	1°
0	0	0.3°

Bit	Mnemonic	Meaning when set (1)	Meaning when clear (0)
3	ACCSOKAY	Computed accelerations probably correct	Computed accelerations probably incorrect
2	AUTRATE2	Used together to determine index (RATEINDX) which is used to select attitude maneuver rate	
1	AUTRATE1		

DAPARUPT, DAPBQRPT, DAPZRUPT: Double precision storage locations for the accumulator, L register, Q, B, Z and BBANK registers for the job interrupted by "SUPERJOB".

DEL: Single precision switch which is described in MIT's Luminary GSOP, Section 3, as a capital delta (-Δ); scaled B14 and unitless.

DELCDU: Interface with steering and attitude maneuver routines, minus desired change in gimbal angles per 100 millisecond period, scaled B-1 in units of revolutions, stored in two's complement form.

DELPEROR, DELQEROR, DELREROR: Single precision smoothing terms calculated during attitude maneuver and steering routines, used un automatic control portions of the DAP. Scaled B-1 in units of revolutions.

DKOMEGAN, DKKAOSN: Single precision Kalman filter gains for the docked configuration, scaled B14 and unitless. See discussion of the "Recursive State Estimator" in Section 3 of the Luminary GSOP. Part of the erasable load.

DKTRAP: Single precision deadband for the state estimator in the docked configuration, scaled B-3 in units of revolutions per second. Part of the erasable load.

$DOWNTORK_i$ (i = 0-5): Single precision table of quantities for downlink which give comulative jet on times for the various axes; the correspondence is (0,+P; 1,-P; 2,+U; 3,-U; 4,+V; 5,-V); Scaled B5 in units of seconds. Part of erasable load.

DXERROR, DYERROR, DZERROR: Double precision cumulative error between the actual rate and the rate requested through the hand controller. Scaled B-1 with units of revolutions.

E, EDOT: See DAPB section.

EDOTQ, EDOTR: Single precision biased rate estimates, scaled B-3 in units of revolutions per second.

EDRIVE: Single precision vector containing the present settings of the FDAI error needles, scaled B-1 in units of ten revolutions for attitude errors or B-3 in units of 10 revs/sec for rate errors.

FIREFCT: See DAPB section.

FUNCTION: Double precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it contains first $\omega + \alpha|\alpha|/2K$ scaled B3 with units of revs/sec; second it contains $K(-\Delta\omega + \alpha^2/2K)$ scaled B-6 with units of $revs^2/sec^4$.

HALFARG: One half of the argument j in single precision, for the square root iteration in "SCALLOOP", scaling variable and units "revolutions per second squared" squared.

IMODES33: See INTR section.

INDEX: Single precision variable used in the same way as SHFTFLAG to count multiplications by four in the square-root routine. Scaled B14 and unitless.

INGTS: Single precision two-valued switch set to indicate that the GTS attitude control law was operating during the previous cycle, scaled B14 and unitless.

JETRATEP, JETRATEQ, JETRATER: Single precision addition to vehicle rate expected to have been contributed by the RCS jets during the last period, scaled B-3 in units of revolutions per second. Alternate program notations, JETRATE+0, JETRATE+1, JETRATE+2.

K:0.1secB0: Single precision constant, program notation -100MS, stored as -0.1, scaled B0 in units of seconds. Equation value: 0.1. (Equivalent to +100 milliseconds).

K:0.4gts: Single precision constant actually equal to 1 + 0.6 (stored value 0.6, program notation OCT23146, added to 1) but scaled B-2 and unitless. Equation value: 0.4

K:100msT6: Single precision constant, program notation -100MST6, stored as -160×2^{-14}, scaled B10 in units of centiseconds. Equation value: +10. (Equivalent to +100 milliseconds)

K:136msT6: Single precision constant, program notation -136MST6, stored as 77445_8, scaled B10 in units of centiseconds. Equation value: +13.625 (Equivalent to +136.25 milliseconds)

K:150msQR: Single precision constant, program notation -150MS, stored as 77417_8, scaled B10 in units of centiseconds. Equation value: +15 (Equivalent to +150 milliseconds)

K:150msT6: Single precision constant, program notation +150MST6, stored as 00360_8, scaled B10 in units of centiseconds. Equation value: +15 (Equivalent to +150 milliseconds)

K:160msT6: Single precision constant, program notation -160MST6, stored as 77377_8, scaled B10 in units of centiseconds. Equation value: +16. (Equivalent to +160 milliseconds)

K:1d100ms: Single precision constant stored as 00005_8, program notation FIVE, scaled B15 in units of $seconds^{-1}$. Equation value: 10 (Equivalent to 1/0.100)

K:1d40: Single precision constant, program notation 1/40, stored as 00632_8, scaled B2 in units of seconds. Used to convert sensed vehicle attitude change data, scaled B-1 in units of revolutions, to vehicle rate data, scaled B-3 in units of revolutions per second. Equation value: 0.10 (Equivalent to 100 milliseconds)

K:25B5: Single precision constant, program notation 25/32, stored as 31000_8, scaled B5 in units of centiseconds per second. Equation value: 25.

K:40cyc: Single precision constant, program notation 40CYCL or 40CYC, stored as 00050_8, scaled B14 with units of deci-second. Used as the initial setting for the timing cell for the "direct" manual control mode. Equation value: 40

$K{:}5FAILTAB_i$: Table of eight single precision octal constants indicating which bit of channel 5 is to be disabled by each one of bits 1-8 of channel 32.

i	$K{:}5FAILTAB_i$	jet #	Channel 5 code and bit #	
8	00040	10	2D	6
7	00020	9	2U	5
6	00100	13	1U	7
5	00200	14	1D	8
4	00010	6	3D	4
3	00001	1	4U	1
2	00004	5	3U	3
1	00002	2	4D	2

K:60msT6: Single precision constant stored as 00140_8, scaled B10 in units of centiseconds. Equation value 6. Used as minimum impulse jet on time for the docked configuration for the Q, R axes.

$K{:}6FAILTAB_i$: Table of eight single precision octal constants indicating which bit of channel 6 is to be disabled by each one of bits 1-8 of channel 32.

i	$K{:}6FAILTAB_i$	jet #	Channel 6 code and bit #	
8	00010	11	2F	4
7	00020	12	2S	5
6	00004	15	1F	3
5	00200	16	1S	8
4	00001	7	3F	1
3	00002	3	4F	2
2	00040	8	3S	6
1	00100	4	4S	7

K:aosint: Sinble precision constant, program notation 200MS, stored as 06315_8, scaled B-1 in units of seconds. Equation value: 0.1 (Equivalent to 100 milliseconds.)

K:BIT10: Single precision constant, stored as 01000_8, scaled B5 and unitless. Used to rescale the jet on times for the downlink. Equation value 1.0.

K:bklim: Value of overflow in a quantity scaled B10 in units of centiseconds. Equation value: 1024

K:bkscl: Actually not a constant as such, rather a multiplication by three implemented as a double and an add; effective units, centiseconds per second; effective scale factor, B5. Equation value: 100 x 0.96 (0.96 is the error introduced because the method is approximate).

K:CALLCODE: Single precision constant, stored as 00032_8, scaled B6 in units of seconds. Equation value: 0.1016 (Equivalent to 102 milliseconds).

K:eclim: Single precision constant stored as 00600_8, program notation DACLIMIT +1, scaled B-1 for attitude errors in units of ten revolutions or B-3 for rate errors in units of ten revolutions / second. Equation values are: attitude errors, 0.01171875 (equivalent to 42.1875 degrees); rate errors, 0.00292968, (equivalent to 10.5469 deg/sec). Note however that the FDAI error needles are pinned by attitude commands of 5.06250 deg. and rate commands of 1.265625 deg/sec.

K:gtstmin: Value of least significant bit in a single precision quantity scaled B14 and unitless. Equation value: 1

K:INDXYZ$_i$: Table of eight single precision constants to translate any of the eight possible combinations of inputs from the Q and R axes of the translational hand controller (bits 9-12 of channel 31) into an index to select the proper jets from the table of K:typmanP; scaled B14, unitless and stored as follows:

i	function	K:INDXYZ$_i$
1	+Q	4
2	-Q	2
3	error	
4	+R	5
5	+U	9
6	+V	10
7	error	
8	-R	3
9	-V	8
10	-U	7

K:LINRAT: Single precision constant stored as 00056_8, scaled B12 in units of RHC counts. Used as the coefficient of the linear term in the quadratic expression for hand controller response. Equation value: 11.5

K:mFOURDEG: Single precision constant, program notation -FOURDEG, stored as 75117_8, scaled B-3 in units of revolutions. Equation value: -0.01111 (Equivalent to -3.9996 deg)

K:minimptj: Single precision constant stored as 00012_8, program notation TEN, scaled B10 in units of centiseconds. Equation value: 0.625 (corrected to true minimum impulse time in jet selection routine)

K:MINTIMES: Single precision constant stored as 77751_8, scaled B10 in units of centiseconds. Equation value: 1.375.

K:OCT00240: Single precision constant stored as 00240_8, scaled B10 in units of 1/seconds (actually computation cycles per second). Equation value: 10.

K:OCT31: Single precision constant stored as 00031_8, scaled B14 and unitless. Equation value: 25. (Corresponds to an interval of 2.5 seconds, 25 cycles at 100 ms per cycle.)

K:ONETENTH: Single precision constant stored as 03146_8, scaled B0 and unitless. Equation value: 0.10.

K:pTJMINT6: Single precision constant, program notation +TJMINT6, stored as 00026_8, scaled B10 in units of centiseconds. Equation value: 1.375 (Equivalent to 13.75 milliseconds).

K:quadsP$_i$: Table of seven single precision octal constants, program notation TYPEP, containing the binary codes for various jet pairs that can be used to accomplish a given maneuver, stored in order of their desirability ($i \geq 4$, more desirable; $i \leq 3$, less desirable). The constants and their significance is indicated below.

i	K:quadsP$_i$	Bits 8-1	Use
6	00377	1111 1111	All quads; translation or rotation
5	00245	1010 0101	Quads 1 and 3; jets 7,8,15,16
4	00132	0101 1010	Quads 2 and 4; jets 3,4,11,12
			2-jet rotations above use diagonal quads
			2-jet rotations below use adjacent quads
3	00151	0110 1001	Rotation using jets 4,7 or 8,11
2	00231	1001 1001	Rotation using jets 7,12 or 11,16
1	00226	1001 0110	Rotation using jets 12,15 or 3,16
0	00146	0110 0110	Rotation using jets 4,15 or 3,8

$K:quadsQR_i$: Table of five single precision octal constants, program notation TYPEPOLY, containing the binary codes for primary and secondary jet combinations that can be used to accomplish rotations around an axis in the Q-R plane and translations perpendicular to the Q-R plane.

i	$K:quadsQR_i$	Bits 8-1	Use
4	00377	1111 1111	All quads; 2-jet rotation, 4-jet trans
3	00231	1001 1001	1-jet rotation, B-system (1,6,9,14) +X jets 6,14 (quads 1,3); -X jets 1,9 (quads 2,4)
2	00146	0110 0110	1-jet rotation, A-system (2,5,10,13) +X jets 2,10 (quads 2,4); -X jets 5,13 (quads 1,3)
1	00252	1010 1010	1-jet rotation using only +X jets
0	00125	0101 0101	1-jet rotation using only -X jets

K:ROOTHALF: Single precision constant, stored as 26501_8, scaled B0 and unitless. Used as a starting value of the Newton algorithm in the GTS law square root routine. Equation value 0.70710.

K:T6lim: Single precision constant, program notation 1 - PRIO37, stored as $(40000_8 - 37000_8)$, scaled B10 in units of centiseconds. Equation value: 32. (equivalent to 320 milliseconds)

K:T6tosec: Single precision constant, program notation ELEVEN, stored as 00013_8, used in such a way (L register retained after multiplication) that effective scaling is B-10 in units of seconds per centisecond. Equation value 11/1024 or 0.010742.

K:trvtoc: Constant implicit in the FDAI error counter, equation interface, scaled B15 or B17 in units of ICDU error counter increments per ten revolutions for attitude errors or ten revs per second for rate errors. Equation value 32768 or 131072. (one least increment to the error counter represents about 0.11 degrees on the FDAI error needles)

$K:typmanP_i$: Table of eleven single precision octal constants, program notation JETSALL, each containing the binary codes for all the jet pairs that can be used to accomplish a particular maneuver. The constants and their significance is indicated below.

i	$K:typmanP_i$	Bits 8-1	Maneuver (jet numbers)
0	00252	1010 1010	-P rotation (3,8,11,16)
1	00125	0101 0101	+P rotation (4,7,12,15)
2	00140	0110 0000	-Y translation (4,8)
3	00006	0000 0110	-Z translation (3,15)
4	00220	1001 0000	+Y translation (12,16)
5	00011	0000 1001	+Z translation (7,11)
6	00151	0110 1001	+V translation (4,11 and 7,8)
7	00146	0110 0110	-U translation (8,15 and 3,4)
8	00226	1001 0110	-V translation (3,12 and 15, 16)
9	00231	1001 1001	+U translation (7,16 and 11, 12)
10	00151	0110 1001	+V translation (4,11 and 7,8)

K:typmanQR$_i$: Table of six single precision octal constants, program notation ALLJETS, each containing the binary codes for all the jets that can be used to accomplish rotations around an axis in the Q-R plane and translations perpendicular to the Q-R plane.

i	K:typmanQR$_i$	Bits 8-1	Maneuver (jet numbers)
0	00110	0100 1000	-U rotation (6,13)
1	00022	0001 0010	-V rotation (2,9)
2	00204	1000 0100	+U rotation (5,14)
3	00041	0010 0001	+V rotation (1,10)
4	00125	0101 0101	-X translation (1,5,9,13)
5	00252	1010 1010	+X translation (2,6,10,14)

K2CNTRAL: Double precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it first contains ω scaled B3 with units of revs / sec; second it contains $-\Delta\omega + \alpha^2/2K$ scaled B3 with units of revs / sec ; third it contains $-\Delta(-\Delta\omega + \alpha^2/2K)$ scaled B3 with units of revs / sec; fourth it contains $(K(-\Delta\omega + \alpha^2/2K))^{\frac{1}{2}}(-\Delta(-\Delta\omega + \alpha^2/2K))$ with scaling undetermined at this point because of variable scale return from square root routine and with units of rev^2/ sec^3.

K2THETA: Double precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it first contains $K\theta$ scaled B0 in units of revs^2/ sec^3 ; second it contains $K\theta + \alpha(-\Delta\omega + \alpha^2/3K)$ scaled B0 with units of revs^2/ sec^3 ; third it contains $K\theta + \alpha(-\Delta\omega + \alpha^2/3K) + (K(-\Delta\omega + \alpha^2/2K))^{\frac{1}{2}}(-\Delta(-\Delta\omega + \alpha^2/2K))$, or U', scaled B0 with units of revs^2/ sec^3.

KCENTRAL: Single precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it contains K, three tenths of the "jerk" or time derivative of angular acceleration, scaled B-9 with units of rev / sec^3.

LMOMEGAN, LMKAOSN: Single precision Kalman filter gains for the LM alone configuration, scaled B14 and unitless. See discussion of the "Recursive State Estimator" in section 3 of the Luminary GSOP. Part of the erasable load.

LMTRAP: Single precision deadband for the state estimator in the LM alone configuration, scaled B-3 in units of revolutions per second. Part of the erasable load.

M11, M21, M22, M31, M32: Single precision factors used in transforming from gimbal angle differences into body axis rotations, scaled B0 and unitless.

MEASRATE: Single precision temporary storage for measured rate, program notation OMEGAU, scaled B-3 in units of revolutions per second. Carefully limited in case of overflow to $\pm$ 0.12499 ($\pm$ 44.997 degrees/ sec)

$NEGU_0$, $NEGU_2$: Single precision switches, program notations NEGUQ and NEGUR, indicating whether the DPS gimbal drives should be driven and whose signs indicate the complement of the direction in which each gimbal is to be driven; scaled B14 and unitless. See the discussion of the GTS control law in MIT's Luminary GSOP, Chp 3.

NEXTP, NEXTU, NEXTV: Single precision storage for the P,U and V-axis translation jet codes to be implemented as soon as rotation commands are completed. Stored as octal quantities and set to 00000_8 to indicate that all jets are to be turned off.

NPTRAPS, NQTRAPS, NRTRAPS: Single precision time varying portion of the Kalman filter gain, scaled B14 and unitless.

NUMBERT: Single precision index used to select the number of jets to be used for a particular maneuver and to specify which jet quads will be used, scaled B14 and unitless.

$NUMJETS_0$, $NUMJETS_1$, $NUMJETS_2$: Single precision number of jets used for P,U and V-axis rotation, respectively, scaled B14 and unitless. Program notations NO.PJETS, NO.UJETS, and NO.VJETS.

NXT6AXIS: Single precision quantity used to indicate which set of jets is to be turned off at the next TIME6 interrupt (0 for P, 4 for U, and 13 for V), program notation NXT6ADR, scaled B14 and unitless.

OLDPMIN, OLDQRMIN: Single precision flagwords set greater than zero when a minimum impulse command is sensed and reset to zero when no commands are present.

OLDXFORP, OLDYFORP, OLDZFORQ: Single precision storage for the value of the gimbal angles at the previous sample, used to calculate rate of change of gimbal angles, scaled B-1 in units of revolutions and stored in two's complement form.

OMEGAP, OMEGAQ, OMEGAR, OMEGAU, OMEGAV: Single precision estimated vehicle rate, calculated using commanded accelerations and times, scaled B-3 in units of revolutions per second. Limited to ± 0.12499 ($\pm$ 44.997 degrees/sec) by overflow checks

OMEGAPD, OMEGAQD, OMEGARD: Single precision rate biases generated in the attitude maneuver and steering routines, scaled B-3 in units of revs / sec.

PERROR: Single precision P-axis error, scaled B-1 in units of revolutions.

$PJETCTR_i$ (i = 1,2,3): Single precision timing counters used to separate RCS jet firings for the docked configuration. The index correspondence is 1 - P axis; 2 - U axis; 3 - V axis. Scaled B14 with units of deci-seconds. For i = 2 or 3 alternate program notation is UJETCTR or VJETCTR.

PLAST, QLAST, RLAST: Single precision quantities, giving the rate requested by the astronaut via the hand controller. Scaled B-3 with units of revolutions per second.

POLYTEMP: Single precision logical intersection of octal constants from the tables K:typman and K:quads. Bits 9 through 15 will be zero and bits 1 through 8 will contain, at most, four binary ones indicating four jets to be actuated to perform a maneuver. See tables below showing codes for various types of maneuvers.

PVALVEST: Single precision octal quantity reflecting the latest estimation of the state of the jet failure bits in CH5MASK and CH6MASK.

QACCDOT, RACCDOT: Actual rate of change of the rotational rates induced by the thrust vector/c.g. offset, single precision, scaled B-8 in units of revolutions per second cubed; equal to zero or plus or minus ACCDOTQ and ACCDOTR, respectively; These can be loaded by indexing as in the routine "NEGUSUM".

QDAPK, RDAPK: Single precision derivatives of angular acceleration about the positive Q and R axes, respectively, multiplied by a fractional gain (in "SPSCONT") and scaled B-9 in units of revs per second cubed. Program notations KQ and KRDAP.

QERROR, RERROR: Single precision Q and R-axis error, scaled B-1 in units of revolutions.

QGIMTIMR, RGIMTIMR: Single precision counters scaled B14 in units of 100 millisecond intervals.

QKALERR, RKALERR: Single precision filtered difference between calculated rate and measured rate, scaled B-3 in units of revolutions per second. Program notations DAPTEMP1 and DAPTEMP2.

QRATEDIF, RRATEDIF: Single precision storage for difference between desired rate and actual rate for the Q and R axes, respectively, used only in non-automatic modes, scaled B-3 in units of revolutions per second.

QRCNTR: Single precision index scaled B14 and unitless.

RATEDB: Single precision rate deadband, scaled B-3 in units of revolutions per second. RATEDB is called -RATEDB in the program and is stored as a negative quantity for convenience. In this document however, for convenience in interpretation, the sign has been changed and RATEDB is a positive quantity.

RCSFLAGS: Listed separately on next page.

RETJADR: Return address used to distinguish between manual and automatic modes during the Q,R-axis computations when the same routine must be performed twice. Equals "BACKHAND", "TJLAW4", or "MINRTN".

RHCP, RHCQ, RHCR: Three single precision counters, program notations P-RHCCTR, Q-RHCCTR and R-RHCCTR, scaled B14 in units of counts from the Rotational Hand Controller. The value of these counts in terms of commanded rate is variable and determined by the astronaut through the DAP Data Load Routine (03).

ROTINDEX: Single precision index indicating the type of maneuver for which jets are to be selected, scaled B14 and unitless. See detailed descriptions of K:INDXYZ.

$SAVEHAND_0$,$SAVEHAND_1$: Temporary storage for the Rotational Hand Controller inputs from the Q and R axes, necessary because all the RHC counters are reset at once during the P-axis routines.

SENSETYP: Single precision quantity scaled B14 and unitless. Used to indicate X-axis translational sense desired during U and V rotations. A value of 0 implies balanced couples, 1 implies -X, and 2 implies +X.

RCSFLAGS: Single precision flagword whose individual bits have the following meanings (note that not all bits are used at present):

Bit	Mnemonic	Meaning when set (1)	Meaning when reset (0)
15	-----	-----	-----
14	-----	-----	-----
13	-----	Job to calculate DAP parameters not needed at present	Set up job to calculate DAP parameters
12	-----	Perform P-axis calculations	Skip P-axis calculations
11	QRBIT	In "direct" rate command for Q,R axes	Not in "direct" rate command
10	PBIT	In "direct" rate command for P-axis	Not in "direct" rate command
9	JUSTIN	Hand-controller just sensed as out of detent	Hand-controller just sensed as in detent
8	-----	-----	-----
7	-----	-----	-----
6	-----	-----	-----
5	CALLGMBL	Perform "ACDT+C12" routine to set engine gimbal drive bits	"ACDT+C12" not being called
4	DSPLYALT	Output errors to FDAI	Calculate the errors (this bit controls the display-calculate cycle)
3	DSTEPONE	Initialize FDAI error drive	Drive already initialized
2	DSTEPTWO	Initialize the display-calculate cycle for the FDAI errors	Cycle already initialized
1	-----	Used to alternate in "tacking" translation policies	Used to alternate in "tacking" translation policies

SHFTFLAG: Single precision variable scaled B14 and unitless, used to count multiplications by four; used in the square-root routine to maintain accuracy.

$SKIPUV_0$, $SKIPUV_1$: Single precision flags set equal to zero when either U or V-axis computations are to be skipped because a short jet firing was calculated on the last DAP pass. Scaled B14 and unitless. Program notation SKIPU and SKIPV.

STIKSENS: A single precision conversion factor which converts a quadratic expression in hand-controller counts to a rate desired in revolutions per second. Scaled B-15 in units of revolutions per second per RHC - count squared.

T5ADR: Double precision variable starting address for the TIME5 interrupt. Set equal to "PAXIS" or "DAPIDLER".

$T6NEXTAX_1$, $T6NEXTAX_3$: Single precision quantities, program notations T6NEXT +1 and T6FURTHA +1, respectively, used to form a list of jets to be cut off at various intervals after the next TIME6 interrupt, scaled B14 and unitless. See description of NXT6AXIS.

$T6NEXTTM_0$, $T6NEXTTM_2$: Single precision quantities, program notations T6NEXT and T6FURTHA, respectively, used to store the time interval after the next TIME6 interrupt when the jets indicated in $T6NEXTAX_1$ and $T6NEXTAX_3$ are to be cut off, scaled B10 in units of centiseconds.

TCP, TCQR: A single precision timer used in the "direct" manual rate control. Scaled B14 with units of deciseconds.

THETAD: See list of major variables.

TIME5: Single precision counter incremented every 10 milliseconds (every centisecond) which causes the "T5RUPT" routine to be entered in the interrupt mode whenever it overflows.

TIME6: Single precision counter decremented every 0.625 milliseconds when enabled (channel 13, bit 15); causes the "DOT6RUPT" routine to be entered as an interrupt whenever it is reduced to zero (-0).

TJ_0, TJ_1, TJ_2: Single precision jet fire times for the P, U and V axes, respectively, scaled B10 in units of centiseconds. Program notations TJP, TJU, and TJV.

TRAPEDP, TRAPEDQ, TRAPEDR: Transient rate error measured as the difference between measured rate and calculated rate at every 100 millisecond interval, filtered and used to correct rate calculation parameters, scaled B-3 in units of revolutions per second. Carefully limited in case of overflow to ± 0.12499 (± 44.997 degrees).

TRAPSIZE: Single precision filter deadband, stored as a negative quantity. Scaled B-3 in units of revolutions per second. Program notation, DAPTREG6.

UERROR, VERROR: Single precision storage for the attitude errors around the U and V axes, scaled B-1 in units of revolutions.

URATEDIF, VRATEDIF: Single precision storage for attitude rate error in non-automatic modes of operation, scaled B-3 in units of revolutions per second.

WCENTRAL: Single precision working storage for trim gimbal control logic. Using the notation of section 3 of the Luminary GSOP it contains ω, the angular velocity, scaled B-3 with units of rev/sec.

Codes for jets with thrust perpendicular to the P-axis

Bits 1-8 of channel 6 (in the order 8765 4321) (Jets 16,4,8,12, 11,15,3,7)

NUMBERT	-P Rotation ROTINDEX = 0	+P Rotation ROTINDEX = 1	-Y Trans. ROTINDEX = 2	-Z Trans. ROTINDEX = 3
6	1010 1010	0101 0101	0110 0000	0000 0110
5	1010 0000	0000 0101	0010 0000*	0000 0100*
4	0000 1010	0101 0000	0100 0000*	0000 0010*
3	0010 1000	0100 0001	0110 0000*	0000 0000*
2	1000 1000	0001 0001	0000 0000*	0000 0000*
1	1000 0010	0001 0100	0000 0000*	0000 0110*
0	0010 0010	0100 0100	0110 0000*	0000 0110*

	+Y Trans. ROTINDEX = 4	+Z Trans. ROTINDEZ = 5	+V Trans. ROTINDEX = 6	
6	1001 0000	0000 1001	0110 1001	
5	1000 0000*	0000 0001*	0010 0001	
4	0001 0000*	0000 1000*	0100 1000	
3210	meaningless	meaningless	meaningless	(see above and below)

	-U Trans. ROTINDEX = 7	-V Trans. ROTINDEX = 8	+U Trans. ROTINDEX = 9	+V Trans. ROTINDEX = 10
6	0110 0110	1001 0110	1001 1001	0110 1001
5	0010 0100	1000 0100	1000 0001	0010 0001
4	0100 0010	0001 0010	0001 1000	0100 1000
3	0110 0000*	0000 0000*	0000 1001*	0110 1001*
2	0000 0000*	1001 0000*	1001 1001*	0000 1001*
1	0000 0110*	1001 0110*	1001 0000*	0000 0000*
0	0110 0110*	0000 0110*	0000 0000*	0110 0000*

* Those codes immediately followed by asterisks are meaningless because 1) they include less than two binary one's (less than two jets would be actuated), 2) they are identical to a code higher in the list, or 3) they contain one pair of binary one's but this pair does not accomplish the required maneuver.

Maneuver Codes

Codes for jets with thrust perpendicular to the Q-R plane

Channel 5 bits 1-8 (in the order 8765,4321) (jets 14,13,10,9, 6,5,2,1)

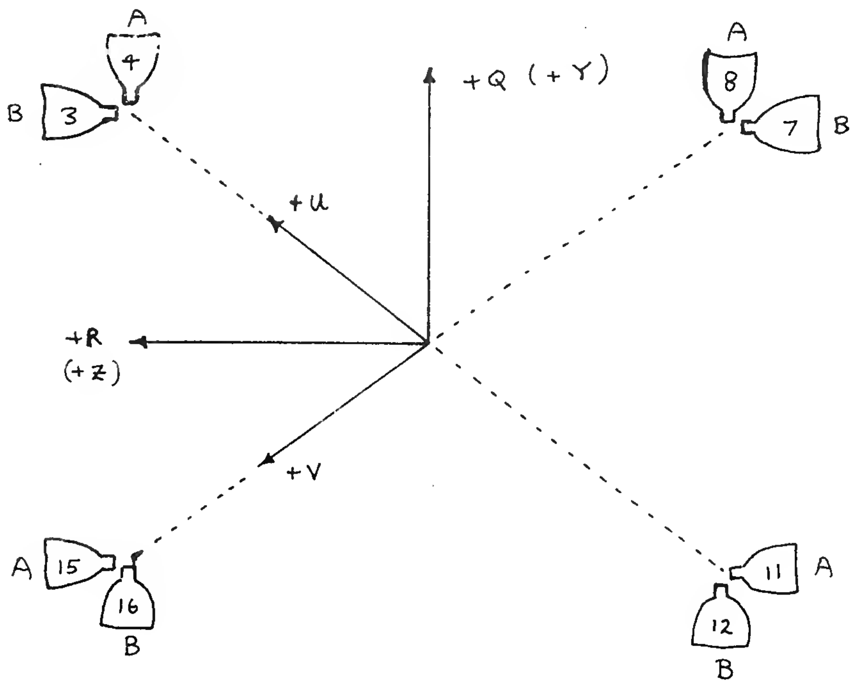
NUMBERT	-U Rotation ROTINDEX = 0	-V Rotation ROTINDEX = 1	+U Rotation ROTINDEX = 2	+V Rotation ROTINDEX = 3
4	0100 1000	0001 0010	1000 0100	0010 0001
3	0000 1000	0001 0000	1000 0000	0000 0001
2	0100 0000	0000 0010	0000 0100	0010 0000
1	0000 1000*	0000 0010*	1000 0000*	0010 0000*
0	0100 0000*	0001 0000*	0000 0100*	0000 0001*

	-X Trans. ROTINDEX = 4	+X Trans. ROTINDEX = 5
4	0101 0101	1010 1010
3	0001 0001	1000 1000
2	0100 0100	0010 0010
1	0000 0000*	1010 1010*
0	0101 0101*	0000 0000*

* Those codes immediately followed by asterisks are meaningless because, for the rotations, they are identical to a code higher in the list, and for the translations, they are identical to a code higher in the list or are zero. It should be noted that the rotation codes derived with NUMBERT = 0 or 1 are valid and are used to assure that 1-jet rotation commands will complement X translation commands in effect.

Jet Thrust Directions, Numbers and Channel 5 or 6 codes
(each jet indicated as belonging to either system A or B)

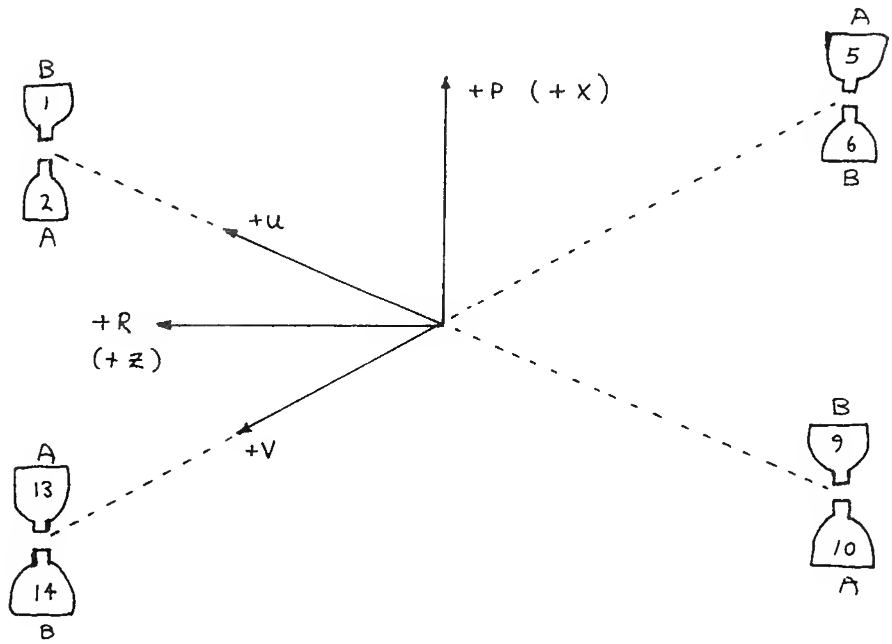
1. Horizontal jets, P-axis rotations and Y,Z translations



Channel 6

Jet	Bit
3	2
4	7
7	1
8	6
11	4
12	5
15	3
16	8

2. Vertical jets, U,V axis rotations and P translations



Channel 5

Jet	Bit
1	1
2	2
5	3
6	4
9	5
10	6
13	7
14	8

Digital Autopilot Phase Plane Logic

1/ACCSET AOSQ and AOSR = 0 (most significant halves of d.p. words)

ALPHAQ and ALPHAR = 0

1/ACCJOB Perform "1/ACCS"

End Job

1/ACCS DOCKTEMP = bit 13 of DAPBOOLS (CSMDOCKD)

LEMMASS = MASS

If DOCKTEMP = 1, LEMMASS = MASS - CSMMASS

Inhibit interrupts

If FLGWRD10 bit 13 (APSFLAG) = 1: (ascent or lunar surface)

    2JETLIM = K:nomaxjts

    i = 12

    If LEMMASS < K:LOASCENT, LEMMASS = K:LOASCENT

    If LEMMASS ≥ HIASCENT, LEMMASS = HIASCENT

If FLGWRD10 bit 13 (APSFLAG) = 0:

    2JETLIM = K:2jlimdwn

    i = 6

    If LEMMASS < K:LODESCNT + HIASCENT:

        LEMMASS = K:LODESCNT + HIASCENT

    If LEMMASS ≥ K:HIDESCNT, LEMMASS = K:HIDESCNT

MASS = LEMMASS

If DOCKTEMP = 1, MASS = LEMMASS + CSMMASS

Release interrupt inhibit

If DOCKTEMP = 1, proceed to "DOCKED"

i = i - 2

1JACCR = K:INERCONB$_i$ + K:INERCONA$_i$/(LEMMASS + K:INERCONC$_i$)

$i = i - 2$

$1JACCQ = K{:}INERCONB_i + K{:}INERCONA_i \ / \ (LEMMASS + K{:}INERCONC_i)$

$i = i - 2$

$1JACCP = K{:}INERCONB_i + K{:}INERCONA_i \ / \ (LEMMASS + K{:}INERCONC_i)$

Perform "COMMEQS" (see pg. DAPB - 22)

$1JACCU = - \ COEFFQ \ 1JACCQ + COEFFR \ 1JACCR$ (rescaled to B-2)

If i > 0: (ascent)

ALLOWGTS = 0

INGTS = 0

Proceed to "1/ACCONT"

$LPVTARM = K{:}LconB + K{:}LconA \ / \ (LEMMASS + K{:}LconC)$

$MPAC_0 = (K{:}dvtoacc \ ABDELV \ MASS \ / \ K{:}GFACTM) \ LPVTARM$ (limited)

Inhibit interrupts

$ACCDOTR = MPAC_0 \ 1JACCR \ / \ K{:}TORKJET1$ (limited)

$ACCDOTQ = MPAC_0 \ 1JACCQ \ / \ K{:}TORKJET1$ (limited)

Proceed to "SPSCONT"

<u>DOCKED</u> $MPAC_0 = K{:}inrtcofC \ LEMMASS \ CSMMASS + K{:}inrtcofF$

$MPAC_0 = MPAC_0 + (K{:}inrtcofA \ CSMMASS + K{:}inrtcofD) \ CSMMASS$

$MPAC_0 = MPAC_0 + (K{:}inrtcofB \ LEMMASS + K{:}inrtcofE) \ LEMMASS$

$MPAC_1 = MPAC_0$

$MPAC_0 = K{:}cgcoefC \ LEMMASS \ CSMMASS + K{:}cgcoefF$

$MPAC_0 = MPAC_0 + (K{:}cgcoefA \ CSMMASS + K{:}cgcoefD) \ CSMMASS$

$MPAC_0 = MPAC_0 + (K{:}cgcoefB \ LEMMASS + K{:}cgcoefE) \ LEMMASS$

$1JACCP = K{:}1JACCON \ / \ MASS$ (limited)

$1dANET_{-14} = K{:}posmax$

$1dANET_{+2} = K{:}posmax$

$1dANET_{+3}$ = K:posmax

$1dANET_{+18}$ = K:posmax

$1dANET_{+19}$ = K:posmax

Inhibit interrupts

1JACCQ = K:TORQCONS / $MPAC_1$

1JACCR = 1JACCQ

COEFFQ = - 0.70711

COEFFR = 0.70711

ACCDOTR = K:dvtoacc ABDELV MASS $MPAC_0$ / $MPAC_1$ (limited)

ACCDOTQ = ACCDOTR

<u>SPSCONT</u> QDAPK = ACCDOTQ K:DGBF

RDAPK = ACCDOTR K:DGBF

$MPAC_1$ = channel 12

TS = bits 12 (-R GTS) and 11 (+R GTS) of $MPAC_1$

If TS = 0, RACCDOT = 0

If TS ≠ 0:

- If bit 12 of $MPAC_1$ = 0, RACCDOT = ACCDOTR (+R GTS)
- If bit 12 of $MPAC_1$ = 1, RACCDOT = - ACCDOTR (-R GTS)

TS = bits 10 (-Q GTS) and 9 (+Q GTS) of $MPAC_1$

If TS = 0, QACCDOT = 0

If TS ≠ 0:

- If bit 10 of $MPAC_1$ = 0, QACCDOT = ACCDOTQ (+Q GTS)
- If bit 10 of $MPAC_1$ = 1, QACCDOT = - ACCDOTQ (-Q GTS)

If DAPBOOLS bit 14 (USEQRJTS) = 1:

- ALLOWGTS = 0
- INGTS = 0
- Proceed to "DOCKTEST"

If T5ADR ≠ "PAXIS":

ALLOWGTS = 0

INGTS = 0

Proceed to "DOCKTEST"

If INGTS = 0, perform "TIMEGMBL"

<u>DOCKTEST</u> If DOCKTEMP = 1, proceed to "1/ACCRET"

<u>1/ACCONT</u> DBVAL1 = DB

DBVAL2 = - 3 DB / 4

DBVAL3 = DB / 2

Inhibit interrupts

AOSU = -COEFFQ AOSQ + COEFFR AOSR (limited)

AOSV = COEFFQ AOSQ + COEFFR AOSR (limited)

Release interrupt inhibit

DRIFTER = bit 8 of DAPBOOLS (DRIFTBIT)

FLATEMP = 0

If ALLOWGTS = 1 or DRIFTER = 1:

FLATEMP = K:FLATVAL

Z3TEM = 0

If DRIFTER = 1:

Z3TEM = K:ZONE3MAX

TS = 1 + 2 1JACCP / K:acp

Inhibit interrupts

$1dANET_{-14}$ = 1 / (2 1JACCP)

$1dANET_{-13}$ = 1 / (2 1JACCP)

$ACCFCT_{-14}$ = (- $1dANET_{-14}$) / TS

$ACCFCT_{-13}$ = $ACCFCT_{-14}$

$1dACOAST_{-16}$ = K:1dp03

$1dACOAST_{-15}$ = K:1dp03

```
          Release interrupt inhibit
          If DRIFTER = 1, AOSU and AOSV = 0
          UV = 0
BOTHAXES  i = 0
          If UV = 0:
               If AOSU ≤ 0, i = 1
               ABSAOS = |AOSU|
          If UV = 1:
               If AOSV ≤ 0, i = 1
               ABSAOS = |AOSV|
          DBB1 = DBVAL1
          DBB2 = DBVAL1
          If ABSAOS ≤ K:miniacc, proceed to "NOTMUCH"
          If FLATEMP = 0:    (powered flight without fine GTS)
               If i = 0:
                    DBB2 = DBB2 + DBVAL1
                    DBB4 = DBB2
                    If ABSAOS ≤ K:m.1875
                         DBB1 = (1 - 32 ABSAOS) DB
                         Skip next step
                    DBB1 = - DBVAL3
                    DBB3 = DBVAL2
               If i = 1:
                    DBB1 = DBB1 + DBVAL1
                    DBB3 = DBB1
                    If ABSAOS ≤ K:m.1875
                         DBB2 = (1 - 32 ABSAOS) DB
                         Skip next step
                    DBB2 = - DBVAL3
                    DBB4 = DBVAL2
          j = |i - 1|
          1dACOSTT_j = 8/9 ( 1/ ABSAOS )
          1dACOSTT_i = K:1dp03
```

```
            TS = 1 + (2 1JACCU + ABSAOS) / K:acp

            If overflow (|TS| ≥ 2^6):        (ANET ≥ 88.6 degrees/second^2)

                 ANET = 1JACCU + ½ ABSAOS

                 1dATEMP = ½ / ANET

                 TSf = - ½ 1dATEMP / (½ + ANET / K:acp)

                 Proceed to "ACCTHERE"

            ANET = 2 1JACCU + ABSAOS

            1dATEMP = 1 / ANET

            TSf = - 1dATEMP / (1 + ANET / K:acp)

            Proceed to "ACCTHERE"

NOTMUCH     1dACOSTT_0 = K:1dp03

            1dACOSTT_1 = K:1dp03

            If FLATEMP = 0:

                 If ABSAOS > K:miniacc - K:tinyacc:

                      If i = 0:

                           DBB3 = DBVAL3

                           DBB4 = 2 DBVAL3

                      If i = 1:

                           DBB4 = DBVAL3

                           DBB3 = 2 DBVAL3

                 If ABSAOS ≤ K:miniacc - K:tinyacc:

                      DBB3 = DBVAL1

                      DBB4 = DBVAL1

            ANET = 2 1JACCU + ABSAOS

            1dATEMP = 1 / ANET

            TSf = - 1dATEMP / (1 + ANET / K:acp)
```

$j = |i - 1|$

ACCTHERE $ACFTEM_{j+2} = TSf$

$1dATEM1_{j+2} = 1dATEMP$

$TS = 1 + (1JACCU + ABSAOS) / K{:}acp$

If overflow ($|TS| \geq 2^6$):

  $TS = K{:}posmaxsp$

$ANET = K{:}acp\ (TS - 1)$

$1dATEMP = 1 / ANET$

$TSf = - 1dATEMP / (1 + ANET / K{:}acp)$

$ACFTEM_j = TSf$

$1dATEM1_j = 1dATEMP$

$ANET = 2\ 1JACCU - ABSAOS$

$ACCSWTEM = 0$

If $ANET \leq K{:}miniacc$:

  $ANET = K{:}miniacc$

$1dATEMP = 1 / ANET$

$TSf = - 1dATEMP / (1 + ANET\ 1dACOSTT_j)$

$ACFTEM_{i+2} = TSf$

$1dATEM1_{i+2} = 1dATEMP$

$ANET = 1JACCU - ABSAOS$

If $ANET \leq K{:}miniacc$: (replace 1-jet params with 2-jet params unless relevant jet failed.)

  $ACCSWTEM = 2\ i - 1$

  If UV = 0 and i = 0 and bits 4 and 7 of CH5MASK = 0:

    Proceed to "STMIN-" (no minus U jets failed)

(If ANET ≤ K:miniacc)

If UV = 0 and i = 1 and bits 3 and 8 of CH5MASK = 0:

Proceed to "STMIN-" (no plus U jets failed)

If UV = 1 and i = 0 and bits 2 and 5 of CH5MASK = 0:

Proceed to "STMIN-" (no minus V jets failed)

If UV = 1 and i = 1 and bits 1 and 6 of CH5MASK = 0:

Proceed to "STMIN-" (no plus V jets failed)

ANET = K:miniacc (jet failure; use K:miniacc instead of 2-jet parameters)

1dATEMP = 1 / ANET

TSf = - 1dATEMP / (1 + ANET $1dACOSTT_j$)

STMIN- $ACFTEM_i$ = TSf

$1dATEM1_i$ = 1dATEMP

If UV = 0 and bit 3 or 8 of CH5MASK = 1, or if UV = 1 and bit 1 or 6 of CH5MASK = 1: (jet failure - positive torque)

$1dATEM1_3$ = $1dATEM1_1$

$ACFTEM_3$ = $ACFTEM_1$

If UV = 0 and bit 4 or 7 of CH5MASK = 1, or if UV = 1 and bit 2 or 5 of CH5MASK = 1: (jet failure - negative torque)

$1dATEM1_2$ = $1dATEM1_0$

$ACFTEM_2$ = $ACFTEM_0$

$AXDSTEM_0$ = FLATEMP + DBB1 - DBB3

$AXDSTEM_1$ = FLATEMP + DBB2 - DBB4

Inhibit interrupts

$ACCSW_{UV}$ = ACCSWTEM

If UV = 0:

$1dANET_0$ = $1dATEM1_0$

(If UV = 0)

$1dANET_1 = 1dATEM1_1$

$1dANET_2 = 1dATEM1_2$

$1dANET_3 = 1dATEM1_3$

$1dACOAST_0 = 1dACOSTT_0$

$1dACOAST_1 = 1dACOSTT_1$

$ACCFCT_0 = ACFTEM_0$

$ACCFCT_1 = ACFTEM_1$

$ACCFCT_2 = ACFTEM_2$

$ACCFCT_3 = ACFTEM_3$

Release interrupt inhibit

UDB1 = DBB1

UDB2 = DBB2

UDB3 = DBB3

UDB4 = DBB4

$UAXDIST_0 = AXDSTEM_0$

$UAXDIST_1 = AXDSTEM_1$

UV = 1

Proceed to "BOTHAXES"

$1dANET_{16} = 1dATEM1_0$

$1dANET_{17} = 1dATEM1_1$

$1dANET_{18} = 1dATEM1_2$

$1dANET_{19} = 1dATEM1_3$

$1dACOAST_{16} = 1dACOSTT_0$

$1dACOAST_{17} = 1dACOSTT_1$

$ACCFCT_{16} = ACFTEM_0$

$ACCFCT_{17} = ACFTEM_1$

$ACCFCT_{18} = ACFTEM_2$

$ACCFCT_{19} = ACFTEM_3$

FLAT = FLATEMP

ZONE3LIM = Z3TEM

$FIREDB_{-16} = DBVAL1$

$FIREDB_{-15} = DBVAL1$

$COASTDB_{-16} = DBVAL1 + FLAT$

$COASTDB_{-15} = DBVAL1 + FLAT$

$AXISDIST_{-16} = 0$

$AXISDIST_{-15} = 0$

If FLAT > 0:

$FIREDB_0 = DBVAL1$

$FIREDB_1 = DBVAL1$

$FIREDB_{16} = DBVAL1$

$FIREDB_{17} = DBVAL1$

$COASTDB_0 = DBVAL1 + FLAT$

$COASTDB_1 = DBVAL1 + FLAT$

$COASTDB_{16} = DBVAL1 + FLAT$

$COASTDB_{17} = DBVAL1 + FLAT$

$AXISDIST_0 = 0$

$AXISDIST_1 = 0$

$AXISDIST_{16} = 0$

$AXISDIST_{17} = 0$

If FLAT = 0: (Powered flight without fine GTS)

$FIREDB_0$ = UDB1

$FIREDB_1$ = UDB2

$COASTDB_0$ = UDB4

$COASTDB_1$ = UDB3

$AXISDIST_0$ = $UAXDIST_0$

$AXISDIST_1$ = $UAXDIST_1$

$FIREDB_{16}$ = DBB1

$FIREDB_{17}$ = DBB2

$COASTDB_{16}$ = DBB4

$COASTDB_{17}$ = DBB3

$AXISDIST_{16}$ = $AXDSTEM_0$

$AXISDIST_{17}$ = $AXDSTEM_1$

1/ACCRET Switch DAPBOOLS bit 3 (ACCSOKAY) to 1

Release interrupt inhibit

Return

TJETLAW HOLDQ = return address

ADRSDIF1 = K:AXISDIFF$_{AXISCTR}$ (-16, 0 or 16; see description of storage sequence at end of glossary)

If EDOT $>$ 0:

ADRSDIF2 = ADRSDIF1

ROTSENSE = K:SENSOR

If EDOT $\leq$ 0:

E = - E

EDOT = - EDOT

ADRSDIF1 = ADRSDIF1 + 1

ADRSDIF2 = ADRSDIF1

ROTSENSE = - K:SENSOR

i = ADRSDIF1

If $|E| \geq 2^{-5}$: ($|E| \geq$ 11.25 degrees)

If E $<$ 0, proceed to "RUFLAW1"

If E $>$ 0, proceed to "RUFLAW2"

(E = E rescaled from B-1 to B-3)

If $|EDOT| \geq 2^{-6}$, proceed to "RUFLAW3" (EDOT $\geq$ 5.625 deg/sec)

(EDOT = EDOT rescaled from B-3 to B-6)

EDOTSQ = EDOT^2

If $|E| \leq$ FIREDB$_i$ - K:m3deg

NUMBERT = SENSETYP - 1

If $|E| >$ FIREDB$_i$ - K:m3deg or if SENSETYP = 0:

ADRSDIF2 = ADRSDIF2 + 2 (index 2-jet parameters rather than 1-jet parameters)

NUMBERT = 4

j = ADRSDIF2

```
          FIREFCT = FIREDB_i - ½ 1dANET_j EDOTSQ - E

          If FIREFCT ≤ 0:

               Perform "Z123COMP"

               If FIREFCT + FLAT ≤ 0:

                    TS = FIREFCT + FLAT

                    Proceed to "ZONE1"          (reverse rate error and
                                                 reach switch curve)
               If TTOAXIS > ZONE3LIM, proceed to "ZONE2"   (null rate error)

               Proceed to "ZONE3"          (minimum impulse zone)

          TS = COASTDB_i + ½ 1dACOAST_i EDOTSQ + E

          If TS > 0, proceed to "ZONE4"          (coast zone)

          Proceed to "ZONE5"          (increase error rate and reach
                                       switch curve)
Z123COMP  ROTSENSE = - ROTSENSE

          TTOAXIS = EDOT 1dANET_j

          If TTOAXIS > K:tjmax:

               TSt = K:250msB2

               Proceed to "RETURNTJ"

          Return

ZONE1     HH = 2 (TS - AXISDIST_i) ACCFCT_j          (ACCFCT_j is negative)

          TS = (TTOAXIS - K:tjmax)^2 - HH     (only high half of HH used)

          If TS ≤ 0:     (total time greater than K:tjmax)

               TSt = K:250msB2

               Proceed to "RETURNTJ"

          If HH > K:50mssq:     (H between 44 and 150 milliseconds)

               TSt = (HH / K:200msB2) + K:37.5msB2 + TTOAXIS
```

If HH $\leq$ K:50mssq: (H between 0 and 44 milliseconds)

TSt = (HH / K:50msB2) + TTOAXIS

If TSt $\leq$ K:tjmin, TSt = 0

Proceed to "RETURNTJ"

ZONE2 TSt = TTOAXIS

Proceed to "RETURNTJ"

ZONE3 TSt = K:minimpt (illegally small jet on time changed to minimum impulse in "PJETSLEC" or "AFTERTJ")

If EDOT < 2^{-17}, TSt = 0

Proceed to "RETURNTJ"

ZONE4 k = AXISCTR + 1

If TJ_k ROTSENSE < 0: (jets on and firing toward desired state)

If FLAT = 0 and $AXISDIST_i$ - FIREFCT > 0:

Perform "Z123COMP"

TS = FIREFCT

Proceed to "ZONE1"

If FLAT > 0 and FIREFCT - 2 $FIREDB_i$ $\leq$ 0:

Perform "Z123COMP"

If TTOAXIS > ZONE3LIM, proceed to "ZONE2"

Proceed to "ZONE3"

(In all other cases, coast)

TSt = 0

Proceed to "RETURNTJ"

ZONE5 If ROTSENSE < 0, j = j - 1 (indices were chosen to select parameters opposing EDOT; in zone 5, acceleration should be in same direction as EDOT)

If ROTSENSE > 0, j = j + 1

HH = 2 TS $ACCFCT_j$

```
            TTOAXIS = 1dANET_j EDOT
            TS = HH - K:100msB2 TTOAXIS - K:50mssq     (more sig half of HH)
            If TS ≤ 0:
                 TSt = ½ HH / (TTOAXIS + K:25msB2)
                 If TSt ≤ K:tjmin, TSt = 0
                 Proceed to "RETURNTJ"
            TS = HH - K:300msB2 TTOAXIS - K:150mssq
            If TS > 0:
                 TSt = K:250msB2
                 Proceed to "RETURNTJ"
            TSt = ½ (HH + K:85mssq) / (TTOAXIS + K:100msB2)
            Proceed to "RETURNTJ"
RUFLAW1     EDOT = EDOT -  K:RUFRATE
            If EDOT ≤ 0:     (rate below 6.50 degrees per second)
                 NUMBERT = 4
                 FIREFCT = - K:posmaxsp       (if P-axis, call for 4 jets)
                 ADRSDIF2 = ADRSDIF2 + 1 signROTSENSE
                 TS = - EDOT
                 Proceed to "RUFLAW12"
            ROTSENSE = - ROTSENSE
            NUMBERT = 4
            FIREFCT = - K:posmaxsp
            TS = EDOT
            Proceed to "RUFLAW12"
RUFLAW2     ROTSENSE = - ROTSENSE
```

NUMBERT = 4

FIREFCT = - K:posmaxsp (if P-axis, call for 4 jets)

TS = EDOT + K:RUFRATE

If overflow ($|TS| \geq 2^{-3}$):

TSt = K:250msB2

Proceed to "RETURNTJ"

<u>RUFLAW12</u> j = ADRSDIF2 + 2

TSt = TS $1dANET_j$

If TSt $\geq$ K:4secB5:

TSt = K:250msB2

Proceed to "RETURNTJ"

TSt = TSt rescaled from B5 to B2

If TSt $\leq$ K:tjmin, TSt = 0

Proceed to "RETURNTJ"

<u>RUFLAW3</u> ROTSENSE = - ROTSENSE

NUMBERT = 4

FIREFCT = - K:posmaxsp (if P-axis, call for 4 jets)

i = ADRSDIF1 + 2 (= ADRSDIF2 + 2)

TS = E + $\frac{1}{2}$ $1dANET_i$ $EDOT^2$ - $FIREDB_{i-2}$

If TS > 0, TSt = K:250msB2

If TS $\leq$ 0, TSt = 0

<u>RETURNTJ</u> i = AXISCTR + 1

TJ_i = TSt ROTSENSE

If TJ_i $ACCSW_{AXISCTR}$ > 0: (Note that for P-axis, this test is not valid, but NUMBERT is set after "TJETLAW")

NUMBERT = 4

Return via HOLDQ

SPSRCS i = AXISCTR + 1

If $TJ_i = 0$:

OLDSENSE = 0

Proceed to "SPSSTART"

OLDSENSE = 1 signTJ_i

TS_1 = - OLDSENSE EDOT

If DAPBOOLS bit 8 (DRIFTBIT) = 0: (powered flight)

$TS_1 = TS_1$ + K:RATEDB1

If $TS_1 > 0$, proceed to 2nd line of "POSTHRST"

SPSSTART If |EDOT K:RATELIM1| ≥ 1: (check to see if outer rate limit has been exceeded)

TS_t = - 0.5 signEDOT

Proceed to "POSTHRST"

TS_2 = DKDB (K:m3tom1 EDOT + E)

TSt = 0

If $|TS_2| < 1$, proceed to "POSTHRST" (E will be less than the deadband within 4 seconds without firing)

If $TS_2 < 0$ and EDOT ≤ K:RATELIM2, TSt = 0.5

If $TS_2 > 0$ and EDOT >-K:RATELIM2, TSt = - 0.5 (if within the inner rate limit fire toward it)

POSTHRST TJ_i = K:B0toB10 TSt

If OLDSENSE = 0, proceed to "CTRCHECK"

If OLDSENSE > 0:

  $TS = TJ_i$

  Skip next step

$TS = -TJ_i$ (OLDSENSE < 0)

If TS > 0, return

$PJETCTR_i = K:UTIME_i$

$TJ_i = 0$

Return

CTRCHECK If $PJETCTR_i = 0$, return

$TJ_i = 0$

Return

RESTORDB If DAPBOOLS bit 5 (DBSLECT2) = 1, proceed to "SETMAXDB"

If DAPBOOLS bit 4 (DBSELECT) = 0, proceed to "SETMINDB"

DB = K:POWERDB

Proceed to 2nd step of "SETMAXDB"

SETMINDB DB = K:NARROWDB

Establish "1/ACCJOB" (pr27)

Return

SETMAXDB DB = K:WIDEDB

Establish "1/ACCJOB" (pr27)

Return

PFLITEDB Perform "ZATTEROR"

DB = K:POWERDB

Establish "1/ACCJOB" (pr27)

Return

(Entered from a verb 48, this is Routine 03)

DAPDATA1 DAPDATR1 = bits 13, 11, 10, 7, 5, 4, 2 and 1 of DAPBOOLS (CSMDOCKD,ACC4OR2X,AORBTRAN, RHCSCALE, DBSLECT2,DBSELECT, AUTRATE2,AUTRATE1)

If FLGWRD10, bit 13(APSFLAG) = 0: DAPDATR1,bit14 = 0, otherwise DAPDATR1 bit 14 = 1

If DAPDATR1, bits 13 and 14 = 0, set bit 13 of DAPDATR1 = 1

Perform "GOXDSPFR" with TS = K:V01N46 (display DAPDATR1)
(If terminate, proceed to "ENDEXT"; If proceed, proceed to "DPDAT1"; other response, skip next two steps)

Perform "BLANKET" with TS = 00006_8

End job

DAPDATR1 = bits 14, 13, 11, 10, 7, 5, 4, 2 and 1 of DAPDATR1

Proceed to third step of "DAPDATA1"

DPDAT1 Inhibit interrupts

FLGWRD10, bit 13 (APSFLAG) = complement of DAPDATR1, bit 14

If DAPDATR1, bits 13 and 14 ≠ 1, bit 13 of DAPDATR1 = 0

Set bits 13,11,10,7,5,4,2 and 1 of DAPBOOLS = bits 13,11,10,7, 5,4,2 and 1 of DAPDATR1

MASS = LEMMASS

If DAPBOOLS bit 13 (CSMDOCKD) = 1, MASS = MASS + CSMMASS

If DAPBOOLS bit 11 (ACC4OR2X) = 1, switch FLAGWRD1 bit 15 (NJETSFLG) to 0

If DAPBOOLS bit 11 (ACC4OR2X) = 0:

Switch FLAGWRD1 bit 15 (NJETSFLG) to 1

RATEINDX = bits 2 and 1 of DAPBOOLS

STIKLOAD STIKSENS = K:FINE

If DAPBOOLS bit 7 (RHCSCALE) = 1:

STIKSENS = STIKSENS + K:NORMAL

RATEDB = K:m0.6DdS

If DAPBOOLS bit 13 (CSMDOCKD) = 1:

STIKSENS = K:1d10 STIKSENS

RATEDB = K:m0.3DdS

Release interrupt inhibit

<u>DAPDATA2</u> Perform "GOXDSPFR" with TS = K:V06N47 (LEMMASS, CSMMASS)
(If terminate, proceed to "ENDR03"; if proceed, skip next two steps; if other response, repeat this step.)

Perform "BLANKET" with TS = 00004_8

End Job

TS = K:MINMINLM (ascent)

If FLGWRD10 bit 13 (APSFLAG) = 0, TS = TS + K:MINLMD (descent)

If LEMMASS ≤ TS, proceed to "DAPDATA2"

MASS = LEMMASS

If DAPBOOLS bit 13 (CSMDOCKD) = 1:

If CSMMASS ≤ K:MINCSM, proceed to "DAPDATA2"

MASS = LEMMASS + CSMMASS

Perform "RESTORDB" with interrupts inhibited

If FLGWRD10 bit 13 (APSFLAG) = 1, proceed to "ENDEXT" (ascent)

Perform "GOXDSPFR" with TS = K:V06N48 (PITTIME, ROLLTIME)
(If terminate, proceed to "ENDEXT"; if proceed, skip next two steps; if other response, repeat this step)

Perform "BLANKET" with TS = 00004_8 (blank R3)

End Job

Call "TRIMGIMB" in 0.01 second

End Job

TRIMGIMB Switch FLAGWRD6 bit 10 (GMBDRVSW) to 0

Switch bits 12 and 10 of channel 12 to 0 (-Q and -R off)

Switch bits 11 and 9 of channel 12 to 1 (+Q and +R on)

Delay 60 seconds (drive gimbals to stops)

Switch bits 11 and 9 of channel 12 to 0

Switch bits 12 and 10 of channel 12 to 1

Call "PITCHOFF" in PITTIME centiseconds

Delay ROLLTIME centiseconds

Switch bit 12 of channel 12 to 0

Skip next step

PITCHOFF Switch bit 10 of channel 12 to 0

If FLAGWRD6 bit 10 (GMBDRVSW) = 1, establish "TRIMDONE" (pr10)

If FLAGWRD6 bit 10 = 0, switch FLAGWRD6 bit 10 to 1

End task

TRIMDONE Perform "GOMARK3R" with TS = K:V50N48 (PITTIME, ROLLTIME) The TS is formed by adding 13000_8 to K:V06N48

(If terminate, proceed to "ENDEXT"; if proceed, proceed to "ENDEXT"; If other response, proceed to "ENDEXT")

Perform "BLANKET" with TS = 00024_8 (PERFREQ and R3BLNK)

End job

ENDR03 Inhibit interrupts

Perform "RESTORDB"

Proceed to "ENDEXT"

COMMEQS If 1JACCR > 1JACCQ, proceed to "BIGIQ"

EPSILON = (1JACCQ - 1JACCR) / 1JACCQ

If EPSILON > K:EPSMAX, EPSILON = K:EPSMAX

COEFFR = K:0.707 + K:0.35356 EPSILON

COEFFQ = (-1 + EPSILON) COEFFR

Return

BIGIQ mEPSILON = (1JACCQ - 1JACCR) / 1JACCR

If |mEPSILON| > K:EPSMAX, mEPSILON = -K:EPSMAX

COEFFQ = -K:0.707 + K:0.35356 mEPSILON

COEFFR = (-1 - mEPSILON) COEFFQ

Return

$1dACOAST_{-16}$, $1dACOAST_0$, $1dACOAST_{16}$, $1dACOSTT_0$: Single precision inverse of magnitude of offset acceleration expected to oppose positive jet torques, scaled B8 in units of seconds squared per revolution. Limited to a maximum value. See description of storage sequence below.

$1dACOAST_{-15}$, $1dACOAST_1$, $1dACOAST_{17}$, $1dACOSTT_1$: Single precision inverse of magnitude of offset acceleration expected to oppose negative jet torques, scaled B8 in units of seconds squared per revolution. Limited to a maximum value. See description of storage sequence below.

$1dANET_0$, $1dANET_{16}$, $1dATEM1_0$, 1dATEMP: Single precision inverse of the 1-jet, net acceleration expected in a negative sense around an axis, scaled B8 in units of seconds squared per revolution. If this 1-jet acceleration cannot counteract an opposing offset acceleration and the required jets are not failed, the appropriate ACCSW is set to -1 and the inverse of the 2-jet, net acceleration is stored in this cell. See description of storage sequence below.

$1dANET_1$, $1dANET_{17}$, $1dATEM1_1$, 1dATEMP: Single precision inverse of the 1-jet, net acceleration expected in a positive sense around an axis, scaled B8 in units of seconds squared per revolution. If this 1-jet acceleration cannot counteract an opposing offset acceleration and the required jets are not failed, the appropriate ACCSW is set to 1 and the inverse of the 2-jet, net acceleration is stored in this cell. See description of storage sequence below.

$1dANET_{-14}$, $1dANET_2$, $1dANET_{18}$, $1dATEM1_2$, 1dATEMP: Single precision inverse of the 2-jet, net acceleration expected in a negative sense around an axis, scaled B8 in units of seconds squared per revolution. If a jet failure is present for the axis in question, the 1-jet, net acceleration is stored in this cell. See description of storage sequence below.

$1dANET_{-13}$, $1dANET_3$, $1dANET_{19}$, $1dATEM1_3$, 1dATEMP: Single precision inverse of the 2-jet, net acceleration expected in a positive sense around an axis, scaled B8 in units of seconds squared per revolution. If a jet failure is present for the axis in question, the 1-jet, net acceleration is stored in this cell. See description of storage sequence below.

1JACCP, 1JACCQ, 1JACCR: (Program notation also 1JACC, 1JACC +1, and 1JACC +2, respectively). Single precision angular accelerations expected from a single RCS jet fired around the P,Q and R axes, respectively; computed in "1/ACCS" or in "DOCKED" from empirical functions of the mass of the vehicle, scaled B-3 in units of revolutions per second squared.

1JACCU: Single precision angular acceleration expected from a single RCS jet fired around the U or V axes; computed from a function of 1JACCQ and 1JACCR and scaled B-2 in units of revolutions per second squared.

2JETLIM: See DAPA section.

ABDELV: Single precision magnitude of sensed change in velocity, scaled B14 in units of centimeters per second.

ABSAOS: Single precision magnitude of sensed offset acceleration, scaled B-2 in units of revolutions per second squared.

ACCDOTQ, ACCDOTR: Magnitude of rate of change of the offset acceleration; a function of inertia and c.g. position for the DPS whose gimbals are driven at a constant rate; zero for the APS which has no gimbals; scaled B-8 in units of revolutions per second cubed. Single precision.

$ACCFCT_0$, $ACCFCT_{16}$, $ACFTEM_0$: Single precision function defining the intersection on the phase plane of two parabolic trajectories (paths of constant acceleration), scaled B8 in units of seconds squared per revolution. One trajectory corresponds to the 1-jet, net acceleration expected in a negative sense around an axis, and the other trajectory corresponds to the offset acceleration expected in a positive sense around an axis. If no offset acceleration is expected in a positive sense around the axis, the second parabola is based on a minimum acceleration and defines the minimum limit cycle that can be achieved. See description of storage sequence below.

$ACCFCT_1$, $ACCFCT_{17}$, $ACFTEM_1$: Single precision function defining the intersection on the phase plane of two parabolic trajectories (paths of constant acceleration), scaled B8 in units of seconds squared per revolution. One trajectory corresponds to the 1-jet, net acceleration expected in a positive sense around an axis, and the other trajectory corresponds to the offset acceleration expected in a negative sense around an axis. If no offset acceleration is expected in a negative sense around the axis, the second parabola is based on a minimum acceleration and defines the minimum limit cycle that can be achieved. See description of storage sequence below.

$ACCFCT_{-14}$, $ACCFCT_2$, $ACCFCT_{18}$, $ACFTEM_2$: Single precision function defining the intersection on the phase plane of two parabolic trajectories (paths of constant acceleration), scaled B8 in units of seconds squared per revolution. One trajectory corresponds to the 2-jet, net acceleration expected in a negative sense around an axis, and the other trajectory corresponds to the offset acceleration expected in a positive sense around an axis. If no offset acceleration is expected in a positive sense around the axis, the second parabola is based on a minimum acceleration and defines the minimum limit cycle that can be achieved. See description of storage sequence below.

$ACCFCT_{-13}$, $ACCFCT_3$, $ACCFCT_{19}$, $ACFTEM_3$: Single precision function defining the intersection on the phase plane of two parabolic trajectories (paths of constant acceleration), scaled B8 in units of seconds squared per revolution. One trajectory corresponds to the 2-jet, net acceleration expected in a positive sense around an axis, and the other trajectory corresponds to the offset acceleration expected in a negative sense around an axis. If no offset acceleration is expected in a negative sense around the axis, the second parabola is based on a minimum acceleration and defines the minimum limit cycle that can be achieved. See description of storage sequence below.

$ACCSW_0$, $ACCSW_1$, ACCSWTEM: Single precision, three-option switches for the U and V axes, respectively, scaled B14 and unitless. Set if the net acceleration in one direction is very small (large opposing offset acceleration) to indicate to "TJETLAW" that maximum jets must be used if a rotation in that direction is required. (See "RETURNTJ".)

ADRSDIF1, ADRSDIF2: Single precision address indices, scaled B14 and unitless. ADRSDIF1 distinguishes among the 3 axes and between positive and negative torque and acceleration parameters; ADRSDIF2 distinguishes between one and two jet parameters. (Only 2-jet parameters computed for the P-axis.)

ALLOWGTS: See DAPA Section.

ALPHAQ, ALPHAR: See DAPA Section.

ANET: Single precision expected net acceleration, scaled B-2 in units of revolutions per second squared.

AOSQ, AOSR: See DAPA Section.

AOSU, AOSV: Single precision disturbing acceleration due to thrust vector/c.g. offset or other external torques around the U and V axes respectively, scaled B-2 in units of revolutions per second squared.

AXISCTR: See DAPA Section.

$AXISDIST_i$: Single precision difference between the E axis intercept defining the zone 1 boundary and the E axis intercept defining the cutoff parabola, scaled B-3 in units of revolutions. The cutoff parabola is one with slope defined by Amin and intercept at FIREDB + FLAT on the same side of the EDOT axis, or the cutoff parabola is one with slope defined by offset acceleration and intercept at 0.75 FIREDB on the other side of the EDOT axis.

$AXISDIST_{-16}$, $AXISDIST_0$, $AXISDIST_{16}$, $AXDSTEM_0$, $UAXDIST_0$: AXISDIST for positive EDOT.

$AXISDIST_{-15}$, $AXISDIST_1$, $AXISDIST_{17}$, $AXDSTEM_1$, $UAXDIST_1$: AXISDIST for negative EDOT.

CH5MASK: See DAPA Section.

$COASTDB_i$ (i = -16, -15, 0, 1, 16, 17): Single precision deadband defining the ZONE4, ZONE5 border, scaled B-3 in units of revolutions. See description of storage sequence below.

COEFFQ, COEFFR: See DAPA Section.

CSMMASS: Single precision astronaut input of the mass of the CSM, scaled B16 in units of kilograms. Part of erasable load.

DAPBOOLS: See DAPA Section.

DAPDATR1: Single precision communication cell used in the astronaut interface routine to allow him to change APSFLAG and selected bits of DAPBOOLS.

DB: Single precision RCS deadband, set by the astronaut or by internal programs at one of 3 fixed values, scaled B-3 in units of revolutions.

DBB1, DBB4: FIREDB and COASTDB for positive EDOT, scaled B-3 in units of revolutions.

DBB2, DBB3: FIREDB and COASTDB for negative EDOT, scaled B-3 in units of revolutions.

DBVAL1, DBVAL2, DBVAL3: Single precision modifications to the basic deadband used to define COASTDB and FIREDB under various different operational conditions of the DAP, scaled B-3 in units of revolutions.

DKDB: Single precision inverse of the attitude deadband in the CSM-docked RCS control law, scaled B15 in units of $revs^{-1}$; initialized to 00200 (corresponding to 1/1.4 deg) by a fresh start. Part of erasable load.

DOCKTEMP: Single precision storage for bit 13 of DAPBOOLS scaled B2 and unitless. Used to eliminate unnecessary masking operations.

DRIFTER: Single precision storage for bit 8 of DAPBOOLS to assure uniformity of drifting flight assumptions throughout "1/ACCS," scaled B14 and unitless.

E, EDOT: Single precision attitude error and attitude rate error used to define positions on the phase plane and calculate the duration of any RCS jet firings that are deemed necessary, scaled B-1 and B-3, respectively in units of revolutions and revolutions per second at entry to "TJETLAW" and "SPSRCS" subroutines. Note that EDOT is also used as rate-to-be-gained in response to RHC commands.

EDOTSQ: Single precision square of EDOT, scaled B-10 in units of revolutions squared per second squared.

EPSILON: Single precision quantity providing a measure of control torque cross-coupling. It is equal to the ratio of the Q-axis inertia to the R-axis inertia subtracted from one. Scaled B0 and unitless.

$FIREDB_i$ (i = -16, -15, 0, 1, 16, 17): Single precision deadband defining the ZONE123, ZONE4 border, scaled B-3 in units of revolutions.

FIREFCT: Single precision function of E, EDOT, net acceleration and FIREDB used to define zones 1-2-3 and 4-5 and used with ACCFCT and TTOAXIS to calculate the duration of the RCS jet firing time, scaled B-3 in units of revolutions. See description of storage sequence below.

FLAT, FLATEMP: Single precision deadband defining the ZONE1, ZONE2-3 boundary, scaled B-3 in units of revolutions.

HH: The double precision square of the time from the E-axis to the intersection of the parabolic phase plane trajectories and the parabolic switch curve, scaled B4 in units of seconds squared.

HIASCENT: Single precision upper bound on the mass of the ascent stage, scaled B16 in units of kilograms. Initialized in fresh start and/or loaded with the erasable load.

HOLDQ: Single precision octal return address storage.

INGTS: See DAPA Section.

K:0.35356: Single precision constant, stored as 13241_8. It is equal to $\sqrt{2}/4$, scaled B0 and unitless.

K:0.707: Single precision constant, program notation .7071, stored as 26501_8, scaled B0 and unitless. Equation value: 0.70709

K:100msB2: Single precision constant, program notation -.05AT2 or .1AT4, stored at $\pm$0.025, scaled B2 in units of seconds. Equation value: +0.1 (equivalent to 100 milliseconds)

K:150mssq: Single precision constant, program notation -.0112A8, stored as 77750_8, scaled B4 in units of seconds squared. Equation value: +0.02246 (equivalent to 149.9^2 milliseconds^2)

K:1d10: Single precision constant stored as 03146_8, program notation 1/10, scaled B0 and unitless. Equation value: 0.1

K:1dp03: Single precision constant stored as 37777_8, scaled B8 in units of seconds squared per revolution. Equation value: 255.98 (equivalent to 1/(1.407 deg/sec^2))

K:1JACCON: Single precision constant stored as 00167_8 and scaled B13 in units of rev-Kg/sec^2. Equation value 59.50. Used to obtain P-axis control authority for the docked configuration.

K:200msB2: Single precision constant, program notation .1AT2, stored as 01463_8, scaled B2 in units of seconds. Equation value: 0.2

K:250msB2: Single precision constant, program notation BIT11, stored as 02000_8, scaled B2 in units of seconds. Equation value: 0.25.

K:25msB2: Single precision constant, program notation -.025AT4, stored as 77631_8, scaled B2 in units of seconds. Equation value: 0.025

K:2jlimdwn: Single precision constant stored as 01000_8, program notation BIT10 scaled B-3 in units of revolutions per second. Equation value: 0.00391 (equivalent to 1.4 degrees per second)

K:300msB2: Single precision constant, program notation -.15AT2, stored as 75462_8, scaled B2 in units of seconds. Equation value: 0.3

K:37.5msB2: Single precision constant, program notation .0375AT4, stored as 00232_8, scaled B2 in units of seconds. Equation value: 0.0376.

K:4secB5: Single precision constant stored as 73777_8, program notation -1/8, scaled B5 in units of seconds. Equation value: +4.0

K:50msB2: Single precision constant, program notation -.025AT2, stored as -0.0125, scaled B2 in units of seconds. Equation value: +0.05 (equivalent to +50 milliseconds)

K:50mssq: Single precision constant, program notation NEG2, stored as 77775_8, scaled B4 in units of seconds squared. Equation value: +0.001953 (equivalent to 44.2^2 milliseconds squared)

K:85mssq: Double precision constant, program notation .00375A8, stored as 0.00375 x 2^{-3}, scaled B4 in units of seconds squared. Equation value: 0.0075 (equivalent to 86.6^2 milliseconds squared)

K:acp: Constant implicit in addition of two quantities of unequal scaling, scaled B-8 in units of revolutions per second squared. Equation value: 0.00390625 (equivalent to 1.4 degrees per second squared)

K:AXISDIFF$_i$: Three single precision constants, scaled B14 and unitless. Equation value: -16, 0 and 16 for i = -1, 0 and 1.

K:B0toB10: Effective scale factor introduced by treating a quantity scaled B0 as if it were scaled B10, scaled B10 in units of centi-seconds. Equation value: 1024 (equivalent to 10.24 seconds)

K:cgcoefA,B,C,D,E,F: Six single precision coefficients for a curve fit of the form $Ax^2 + By^2 + Cxy + Dx + Ey + F$, used to find the rate of change of the DPS moment arm around the c.g. of the docked configuration.

	Program Notation	Stored Value	Scale Factor	Units	Equation Value
A	COEFF+9	-0.37142	B-31	cm sec^{-1} rev rad^{-1} kg^{-2}	-1.7296 E-10
B	COEFF+8	0.75704	B-31	cm sec^{-1} rev rad^{-1} kg^{-2}	3.5252 E-10
C	COEFF+6	0.20096	B-31	cm sec^{-1} rev rad^{-1} kg^{-2}	9.3579 E-11
D	COEFF+11	0.41179	B-15	cm sec^{-1} rev rad^{-1} kg^{-1}	1.2567 E-5
E	COEFF+10	-0.63117	B-15	cm sec^{-1} rev rad^{-1} kg^{-1}	-1.9262 E-5
F	COEFF+7	0.13564	B1	cm sec^{-1} rev rad^{+1}	0.27128

The equation value of each constant embodies the two constant factors $1/2\pi$ and 0.2 degrees per second. The first is a conversion factor to convert radians to revolutions; the second is the DPS gimbal rate. If the equation value is multiplied by 2π and divided by 0.2, the resulting coefficients will give an idea of the programmed value for DPS moment arm per degree of thrust vector/c.g. offset. For example, $(0.27128\ 2\pi/0.2)\ 360 = 2\pi\ 488.3$, the circumference of a circle with radius equal to the c.g. to pivot point distance. Thus, using only the constant term of the polynomial, this distance is about 500 centimeters or about 200 inches.

K:DGBF: Single precision constant, stored as 23146_8, scaled B-1 and unitless. Equation value: 0.3

K:dvtoacc: Constant implicit in the 2-second navigation cycle, scaled B-1 in units of seconds to the minus one power. Equation value: $\frac{1}{2}$

K:EPSMAX: Single precision constant, stored as 62362_8, but used in this writeup as though it were positive. It provides the magnitude limit for EPSILON and mEPSILON. Scaled B0 and unitless. Effectively constrains the U' and V' axes within 15 deg. from the U, V axes. Equation value: 0.422668

K:FINE: Single precision constant stored as 05220_8, scaled B-15 in units of revolutions per second per RHC-count squared. See definition of STIKSENS in the DAPA Section. Equation value: 0.0000050365.

K:FLATVAL: Single precision constant, stored as 00443_8, scaled B-3 in units of revolutions. Equation value: 0.00222 (equivalent to 0.8 degrees)

K:GFACTM: Single precision constant stored as 00337_8, used to convert from units of pounds (force) to units of kg cm/second^2; scaled B15 in units of kg cm sec^{-2}/lbs. Equation value: 446 (corresponds to 4.4482 newtons per pound times 100 cm/meter)

K:HIDESCNT: Single precision constant stored as 07361_8, scaled B16 in units of kilograms. It is the upper bound on descent stage mass. Equation value: 15300

K:INERCONA$_i$: Set of six double precision constants scaled B13 in units of revolutions/sec^2 kg^{-1}.

i	Stored value	Equation value
0	0.005 934 7674	48.617 6145
2	0.001 497 9264	12.271 0131
4	0.001 045 1889	8.562 1875
6	0.006 544 3852	53.611 6036
8	0.003 578 4354	29.314 5428
10	0.005 694 6631	46.650 6801

K:INERCONB$_i$: Set of six single precision constants scaled B-3 in units of revolutions per second squared.

i	Stored value	Equation value
0	0.002 989	3.7363 E-4
2	0.018 791	2.3489 E-3
4	0.021 345	2.6681 E-3
6	0.000 032	4.0000 E-6
8	0.162 862	2.0358 E-2
10	0.009 312	1.1640 E-3

K:INERCONC$_i$: Set of six single precision constants scaled B16 in units of kilograms.

i	Stored value	Equation value
0	0.008 721	571.5
2	-0.068 163	-4467.1
4	-0.066 027	-4327.1
6	-0.006 923	-453.7
8	0.002 588	169.6
10	-0.023 608	-1547.2

K:inrtcofA, B, C, D, E and F: Six single precision coefficients of a curve fit of the form $Ax^2 + By^2 + Cxy + Dx + Ey + F$ used to find the approximate moment of inertia around an axis in the Q-R plane of the combined CSM LM.

	Program Notation	Stored Value	Scale Factor	Units	Equation Value
A	COEFF +3	-0.03709	B6	kg cm^2/rad kg^2	-2.37376
B	COEFF +2	-0.17670	B6	kg cm^2/rad kg^2	-1.13088 E1
C	COEFF +0	0.19518	B6	kg cm^2/rad kg^2	1.24915 E1
D	COEFF +5	0.02569	B22	kg cm^2/rad kg	1.07752 E5
E	COEFF +4	0.06974	B22	kg cm^2/rad kg	2.92511 E5
F	COEFF +1	-0.00529	B38	kg cm^2/rad	-1.45410 E9

K:LconA: Double precision constant, program notation INERCONA -2, stored as 0.0410511917, scaled B19 in units of kilograms feet per radian. Equation value: 21522.647

K:LconB: Single precision constant, program notation INERCONB -2, stored as 0.155044, scaled B3 in units of feet per radian. Equation value: 1.240352

K:LconC: Single precision constant, program notation INERCONC -2, stored as -0.025233, scaled B16 in units of kilograms. Equation value: -1653.7

K:LOASCENT: Single precision constant stored as 2200 x 2^{-16}; the lower bound on ascent stage mass, scaled B16 in units of kilograms. Equation value: 2200

K:LODESCNT: Single precision constant stored as 00666_8, scaled B16 in units of kilograms. It plus HIASCENT is the lower bound on the unstaged LM mass. Equation value 1752

K:m.1875: Single precision constant stored as 71777_8, but used in this writeup as though it were positive. Scaled B-2 with units of revs/sec^2. Equation value: 0.04687 (corresponds to 16.87 deg/sec^2)

K:m0.3DdS: Single precision constant stored as 77622_8, program notation -0.3D/S, scaled B-3 in units of revolutions per second. Equation value: 0.00083 (equivalent to 0.3 degrees per second)

K:m0.6DdS: Single precision constant, program notation -0.6D/S, stored as 77445_8, scaled B-3 in units of revolutions per second. Equation value: 0.00166 (equivalent to 0.6 degrees per second)

K:m3deg: Single precision constant stored as 75673_8. Scaled B-3 in units of revolutions. Equation value: -0.00833 (equivalent to -3 degrees)

K:m3toml: Constant implied in combining EDOT, scaled B-3 in units of revolutions per second, with E, scaled B-1 in units of revolutions; scale factor B2, units seconds. This is the inverse of the slope of the switch curves in the docked RCS phase plane. Equation value: 4

K:MINCSM: Single precision constant stored as 02000_8, scaled B16 in units of kilograms. Equation value: 4096

K:miniacc: Single precision constant, program notation -.03R/S2, stored as 77377_8, scaled B-2 in units of revolutions per second squared. Equation value: +0.0039 (corresponds to 0.02454 radians per second squared or $1.406^\circ/sec^2$)

K:minimpt: Single precision constant, stored as 00040_8, program notation BIT6, scaled B2 in units of seconds. Equation value: 2^{-7} (equivalent to 7.8 ms)

K:MINLMD: Single precision constant stored as 76466_8, but used as positive, scaled B16 in units of kilograms. Equation value: 2852.

K:MINMINLM: Single precision constant stored as 76731_8, but used as positive, scaled B16 in units of kilograms. Equation value: 2200.

K:NARROWDB: Single precision constant stored as 00155_8, scaled B-3 in units of revolutions. Equation value: 0.00083 (equivalent to 0.2994°)

K:nomaxjts: Single precision constant stored as 40000_8, scaled B-3 in units of revolutions per second. Equation value: +0.12499 (equivalent to 44.997 degrees per second)

K:NORMAL: Single precision constant stored as 25101_8, scaled B-15 in units of revolutions per second per RHC-count squared. Equation value: 0.000020148. See definition of STIKSENS in the DAPA Section.

K:POWERDB: Single precision constant stored as 00554_8, scaled B-3 in units of revolutions. Equation value: 0.00277 (equivalent to 1 degree)

K:RATEDB1: Single precision constant, stored as 00045_8, scaled B-3 in units of revolutions per second. Equation value: 0.0002823 (equivalent to $0.102^\circ/sec$)

K:RATELIM1: Single precision constant, stored as 00032_8, scaled B17 in units of seconds per revolution. Equation value: 208 or 1.0/0.0048 (equivalent to $1/1.73^\circ/sec$)

K:RATELIM2: Single precision constant stored as 00632_8, scaled B-3 in units of revolutions per second. Equation value: 0.003128 (equivalent to 1.126°/sec)

K:RUFRATE: Single precision constant stored as 04476_8, scaled B-3 in units of revs/sec. Used as a rate limit in TJETLAW phase plane calculations. Equation value: 0.01805 (equivalent to 6.5 degrees/sec)

K:SENSOR: Single precision constant stored as 14400_8, scaled B8 in units of centiseconds per second. Used to convert scaling used in "TJETLAW" to that used for TIME6 calculations. Equation value: 100

K:tinyacc: Single precision constant, program notation .023R/S2, stored as 00356_8, scaled B-2 in units of revolutions per second squared. Equation value: 0.003631 (equivalent to 0.02281 radians per sec^2 or 1.307 degree/sec^2)

K:tjmax: Single precision constant, program notation -TJMAX, stored as -0.0375, scaled B2 in units of seconds. Equation value: +0.15 (equivalent to +150 milliseconds)

K:tjmin: Single precision constant, program notation -TJMIN or TJMIN, stored as $\pm$0.005, scaled B2 in units of seconds. Equation value: +0.02 (equivalent to 20 milliseconds)

K:TORKJET1: Single precision constant stored as 0.03757, scaled B22 in units of foot-pounds/radians sec^{-1}. Equation value: 157580 (corresponds to about 550/0.00349). The torque expected from one RCS jet is 550 foot-pounds and the rotation rate of the DPS gimbals is nominally 0.2°/sec.

K:TORQCONS: Double precision constant stored as 0.51443 x 2^{-14}, scaled B35 in units of kilogram centimeters squared revolutions/radians seconds squared. Equation value: 1078837.90 (equivalent to 500 foot-pounds x 13557 kg cm^2 per ft-lb x 1/6.2832 rev/rad)

K:$UTIME_i$ (i = 0,1,2): Single precision constants giving time in deciseconds between RCS jet firings for the docked configuration. Stored as: 00004_8 for the P-axis, 00012_8 for the U-axis, 00012_8 for the V-axis. Scaled B14 with units of deciseconds. Equation values: 4, 10, and 10, respectively.

K:WIDEDB: Single precision constant stored as 03434_8, scaled B-3 in units of revolutions. Equation value: 0.013885 (equivalent to 4.999°)

K:ZONE3MAX: Single precision constant stored as 0.004375, scaled B2 in units of seconds. Equation value: 0.0175 (equivalent to 35 msec. of single jet firing)

LEMMASS: Single precision mass of the LM, scaled B16 in units of kilograms; part of the erasable load.

LPVTARM: Single precision proportionality factor between the DPS gimbal angle and the moment arm of the thrust around the c.g., scaled B3 in units of feet per radian, program notation L,PVT-CG. Because the DPS gimbal angles are small, LPVTARM is approximately equal to the distance between the c.g. and the DPS pivot point.

MASS: Double precision mass of the vehicle, scaled B16 in units of kilograms; part of the erasable load.

mEPSILON: Single precision quantity with same function as EPSILON. It is equal to one subtracted from the ratio of the R-axis inertia to the Q-axis inertia. Scaled B0 and unitless.

$MPAC_0$: Single precision working storage used in "1/ACCS" to store the torque expected from the DPS engine, scaled B17 in units of foot-lbs per radian (radians of rotation of the DPS bell); used in "DOCKED" to store the rate of change of the DPS moment arm, scaled B1 in units of centimeters per second-revolutions per radian.

$MPAC_1$: Single precision working storage used in "DOCKED" to store the combined vehicle moment of inertia around an axis in the Q-R plane, scaled B38 in units of kilogram centimeters squared per radian.

NUMBERT: See DAPA Section.

OLDSENSE: Single precision quantity giving the sign of the jet firing time, calculated in the preceding pass through the DAP, for the axis under consideration. Scaled B14 with units of deciseconds.

PITTIME: Single precision time to drive the DPS gimbal in a positive direction around the Q-axis starting at the hard stop to position it prior to a burn, scaled B14 in units of centiseconds. Pad loaded.

$PJETCTR_i$: See DAPA Section.

QACCDOT, RACCDOT: See DAPA Section.

QDAPK, RDAPK: See DAPA Section.

RATEDB: See DAPA Section.

RATEINDX: Single precision index used by large attitude maneuver calculation routines to select the maneuver rate, scaled B13 and unitless. Determined by the setting of bits 1 and 2 of DAPBOOLS: possible values 0, 1, 2, or 3.

ROLLTIME: Single precision time to drive the DPS gimbal in a positive direction around the R-axis starting at the hard stop to position it prior to a burn, scaled B14 in units of centiseconds. Pad loaded.

ROTSENSE: Single precision, two-valued switch specifying the direction of desired rotation, scaled B8 in units of centiseconds per second.

SENSETYP: See DAPA Section.

STIKSENS: See DAPA Section.

T5ADR: See DAPA Section.

TJ_0, TJ_1, TJ_2: See DAPA Section.

TTOAXIS: Time from the present point on the phase plane to the E-axis along the parabolic trajectory defined by the net acceleration, scaled B2 in units of seconds.

$UAXDIST_i$: See AXISDIST pgs. DAPB 25 and 26.

UDB1, UDB2, UDB3, UDB4: Temporary storage for DBB1, DBB2, DBB3, and DBB4 as computed for the U-axis, scaled B-3 in units of revolutions.

UV: Single precision index to distinguish between U and V axes, scaled B14 and unitless.

Z3TEM: Temporary storage with same scaling as K:ZONE3MAX.

ZONE3LIM: Single precision time defining the border between ZONE2 and ZONE3, scaled B2 in units of seconds.

Storage Sequence for "TJETLAW" Parameters

E-memory Register	Contents of cell	Address used in Computations
BLOCKTOP+0	ACCSWU	$ACCSW_{0}$
BLOCKTOP+1	ACCSWV	$ACCSW_{1}$
BLOCKTOP+2	1dANET for negative, 2-jet acceleration around P-axis	$1dANET_{-14}$
BLOCKTOP+3	1dANET for positive, 2-jet acceleration around P-axis	$1dANET_{-13}$
BLOCKTOP+4	1dACOAST, negative around P	$1dACOAST_{-16}$
BLOCKTOP+5	1dACOAST, positive around P	$1dACOAST_{-15}$
BLOCKTOP+6	FLAT	FLAT
BLOCKTOP+7	ZONE3LIM	ZONE3LIM
BLOCKTOP+8	ACCFCT corresponding to negative, 2-jet acceleration around P	$ACCFCT_{-14}$
BLOCKTOP+9	ACCFCT corresponding to positive, 2-jet acceleration around P	$ACCFCT_{-13}$
BLOCKTOP+10	FIREDB for positive EDOT around P	$FIREDB_{-16}$
BLOCKTOP+11	FIREDB for negative EDOT around P	$FIREDB_{-15}$
BLOCKTOP+12	COASTDB for positive EDOT around P	$COASTDB_{-16}$
BLOCKTOP+13	COASTDB for negative EDOT around P	$COASTDB_{-15}$
BLOCKTOP+14	AXISDIST for positive EDOT, P-axis	$AXISDIST_{-16}$
BLOCKTOP+15	AXISDIST for negative EDOT, P-axis	$AXISDIST_{-15}$
BLOCKTOP+16	1dANET for negative, 1-jet acceleration around U-axis	$1dANET_{0}$
BLOCKTOP+17	1dANET for positive, 1-jet acceleration around U-axis	$1dANET_{1}$
BLOCKTOP+18	1dANET for negative, 2-jet acceleration around U-axis	$1dANET_{2}$
BLOCKTOP+19	1dANET for positive, 2-jet acceleration around U-axis	$1dANET_{3}$
BLOCKTOP+20	1dACOAST, negative around U	$1dACOAST_{0}$
BLOCKTOP+21	1dACOAST, positive around U	$1dACOAST_{1}$
BLOCKTOP+22	ACCFCT corresponding to negative, 1-jet acceleration around U	$ACCFCT_{0}$
BLOCKTOP+23	ACCFCT corresponding to positive, 1-jet acceleration around U	$ACCFCT_{1}$
BLOCKTOP+24	ACCFCT corresponding to negative, 2-jet acceleration around U	$ACCFCT_{2}$
BLOCKTOP+25	ACCFCT corresponding to positive, 2-jet acceleration around U	$ACCFCT_{3}$
BLOCKTOP+26	FIREDB for positive EDOT around U	$FIREDB_{0}$
BLOCKTOP+27	FIREDB for negative EDOT around U	$FIREDB_{1}$
BLOCKTOP+28	COASTDB for positive EDOT, U-axis	$COASTDB_{0}$
BLOCKTOP+29	COASTDB for negative EDOT, U-axis	$COASTDB_{1}$
BLOCKTOP+30	AXISDIST for positive EDOT, U-axis	$AXISDIST_{0}$
BLOCKTOP+31	AXISDIST for negative EDOT, U-axis	$AXISDIST_{1}$
BLOCKTOP+32 through BLOCKTOP+47	V-axis parameters identical in description to U-axis parameters but around V-axis instead of U-axis	U-axis subscripts plus 16

DATA

NVSUB (Entered with TSvn and TSmonopt in A and L)

NVTEMP = TSvn (verb-noun code)

FREEDSKY = 0

If DSPLOCK > 0, return (astronaut using DSKY)

If bit 14 of MONSAVE1 = 1, return (externally initiated monitor)

NVQTEM = return address

MONSAVE2 = TSmonopt (monitor options)

MONSAVE1 = 40000_8 (terminate monitor)

ENTRET = "NVSUBEND"

If |NVTEMP| = 0, proceed to "DSPALARM"

If NVTEMP < 0, proceed to "BLANKDSP"

TSnoun = low 7 bits of NVTEMP ($MPAC_3$)

TSverb = bits 14-8 of NVTEMP shifted right 7 places ($MPAC_4$)

If TSnoun = 0:

  VERBREG = TSverb

  Perform "UPDATVB"

  REQRET = +0 (process verb-noun information)

  Proceed to "NVSUBEND"

If TSverb = 0:

  NOUNREG = TSnoun

  Perform "UPDATNN"

  Proceed to "NVSUBEND"

TSadr = $MPAC_2$ ($MPAC_5$)

VERBREG = TSverb

Perform "UPDATVB"

NOUNREG = TSnoun

Perform "UPDATNN"

LOADSTAT = +0

CLPASS = 0

REQRET = +0

$MPAC_2$ = TSadr

Proceed to "ENTPASO"

BLANKDSP CODE = 6 + |NVTEMP| (CODE = 7, 8, 9 or 10 for legal options; illegal options cause error)

Inhibit interrupts

Perform indented steps for i = CODE, CODE - 1, ... 4, 3, 2, 1, 0

If $DSPTAB_i > 0$, NOUT = NOUT + 1

$DSPTAB_i = 04000_8$ complemented to flag for output

(end of "indented steps")

Release interrupt inhibit

If NVTEMP ≤ -3, VERBREG = 0

If NVTEMP ≤ -2, NOUNREG = 0

CLPASS = 0

DSPCOUNT = - 19 (inhibit all numerical inputs)

Switch bit 6 of channel 11 to 0 (flash off)

REQRET = +0 (process verb-noun information)

NVSUBEND FREEDSKY = 1

Return via NVQTEM

ENTPASO Switch DECBRNCH to 00000_8

DSPCOUNT = - 19 (to block entry of numerical characters)

VERBSAVE = - VERBREG

If VERBREG $\geq$ 28, proceed to "VERBFAN"

TESTNN Perform "LODNNTAB"

If MIXBR = 2, proceed to "MIXNOUN"

If NNADTEM > 0: (normal noun address)

  NOUNCADR = NNADTEM

  EBANK = bits 11-9 of NNADTEM

  NOUNADD = 01400_8 + bits 8-1 of NOUNCADR

  Proceed to "VERBFAN"

If NNADTEM = +0, proceed to "DSPALARM" (noun is not valid)

If NNADTEM = -0: (increment present noun address)

  NOUNCADR = NOUNCADR + 1

  EBANK = bits 11-9 of NOUNCADR

  NOUNADD = 01400_8 + bits 8-1 of NOUNCADR

  If VERBREG $\neq$ 5: (verb 5 uses R3; cannot display NOUNCADR)

    DSPCOUNT = K:R3D1

    Perform "DSPOCTWD" with TSwd = NOUNCADR

  Proceed to "VERBFAN"

(Otherwise, address is to be specified)

CLPASS = - K:posmaxsp (to prevent multiple clears)

If ENTRET $\neq$ "ENDOFJOB": (internal)

  NOUNCADR = $MPAC_2$

  EBANK = bits 11-9 of NOUNCADR

(If internal "address to be specified")

NOUNADD = 01400_8 + bits 8-1 of NOUNCADR

If VERBREG $\neq$ 5:

DSPCOUNT = K:R3D1

Perform "DSPOCTWD" with TSwd = NOUNCADR

Proceed to "VERBFAN"

Perform "REQDATZ" (request noun address; return via REQRET)

If DECBRNCH $\neq 00000_8$, proceed to "ALMCYCLE"

DSPCOUNT = - 19 (to block further numerical characters)

If CADRSTOR $\neq$ +0: (internal display request interrupted)

Switch bit 6 of channel 11 to 1 (leave flash on)

NOUNCADR = ZREG (loaded in "REQDATZ")

EBANK = bits 11-9 of NOUNCADR

NOUNADD = 01400_8 + bits 8-1 of NOUNCADR

Perform "LODNNTAB" (reload NNTYPTEM)

Proceed to "VERBFAN"

LODNNTAB NNADTEM = K:$NNADTAB_{NOUNREG}$

NNTYPTEM = K:$NNTYPTAB_{NOUNREG}$

If NOUNREG < 40: ("normal" noun)

MIXBR = 1

Return

MIXBR = 2 (mixed noun)

RUTMXTEM = K:$RUTMXTAB_{NOUNREG}$

i = low 10 bits of NNADTEM

$IDADTEM_1$ = K:$IDADTAB_i$

$IDADTEM_2$ = K:$IDADTAB_{i+1}$

$IDADTEM_3$ = K:$IDADTAB_{i+2}$

Return

MIXNOUN If NNADTEM = +0, proceed to "DSPALARM" (noun not valid)

If VERBREG > 6, proceed to "VERBFAN" (not a display verb)

Perform the indented steps for i = 2, 1, and 0

NOUNTEM = $IDADTEM_{i+1}$

TS = high 5 (i = 2), mid 5 (i = 1) or low 5 (i = 0) bits of RUTMXTEM shifted right 10, 5 or 0 places to bit positions 5 through 1

If TS = 4, 5, 7 or 10: (double precision)

NOUNTEM = NOUNTEM + 1 (specify minor part for octal display)

EBANK = bits 11-9 of NOUNTEM

TSadr = 01400_8 + low 8 bits of NOUNTEM

$MIXTEMP_i$ = contents of address specified in TSadr

(End of "indented steps")

NOUNADD = "$MIXTEMP_0$" (In a routine such as "DSPABC" then, the "contents of cell specified by (2 + NOUNADD)" will be the contents of $MIXTEMP_2$, loaded above.)

Proceed to "VERBFAN"

DSPALARM If ENTRET = "NVSUBEND", proceed to "POODOO" with TS = 21501_8

If ENTRET = "PASTEVB":

MONSAVE1 = 40000_8

Switch bit 7 of channel 11 to 1 (operator error)

Proceed to "PASTEVB"

Switch bit 7 of channel 11 to 1

End job

VERBFAN If VERBREG ≥ 40: (extended verb)

TSextfan = VERBREG - 40

Perform "RELDSP"

Proceed to "GOEXTVB"

Proceed to routine specified by the contents of VERBREG

VERBREG	Starting address of routine	Function
0	DSPALARM	not defined
1	DSPA	octal display in R1 from NOUNADD
2	DSPB	octal display in R1 from 1 + NOUNADD
3	DSPC	octal display in R1 from 2 + NOUNADD
4	DSPAB	octal display in R1-R2 from NOUNADD+0,1
5	DSPABC	octal display in R1,R2,R3 from NOUNADD+0,1,2
6	DECDSP	decimal display in R1, R1-R2 or R1-R2-R3
7	DSPDPDEC	decimal display in R1-R2 from NOUNADD
8	DSPALARM	not defined
9	DSPALARM	not defined
10	DSPALARM	not defined
11	MONITOR	monitor with verb 1
12	MONITOR	monitor with verb 2
13	MONITOR	monitor with verb 3
14	MONITOR	monitor with verb 4
15	MONITOR	monitor with verb 5
16	MONITOR	monitor with verb 6
17	MONITOR	monitor with verb 7
18	DSPALARM	not defined
19	DSPALARM	not defined
20	DSPALARM	not defined
21	ALOAD	load 1 component via R1 by NOUNADD
22	BLOAD	load 1 component via R2 by NOUNADD+1
23	CLOAD	load 1 component via R3 by NOUNADD+2
24	ABLOAD	load 2 components via R1-R2 by NOUNADD+0,1
25	ABCLOAD	load 3 components via R1-R2-R3 by NOUNADD+0,1,2
26	DSPALARM	not defined
27	DSPFMEM	octal display from fixed memory
28	DSPALARM	not defined
29	DSPALARM	not defined
30	VBRQEXEC	Request executive
31	VBRQWAIT	Request waitlist
32	VBRESEQ	Resequence
33	VBPROC	Proceed (without data)
34	VBTERM	Terminate current test or request
35	VBTSTLTS	Lamp test
36	SLAP1	Fresh start
37	MMCHANG	Change major mode (program)
38	DSPALARM	not defined
39	DSPALARM	not defined

DSPA If MIXBR = 1, TS = high 5 bits of NNTYPTEM (xxx xx0 000 000 000)

If MIXBR = 2, TS = high 5 bits of NNADTEM

If bit 14 of TS = 1, proceed to "DSPALARM" (noun is decimal only)

If NNADTEM = -1: (noun specified a channel)

BUF_0 = -(contents of channel specified by low 9 bits of NOUNCADR)
Proceed to "DSPCOM2"

If MIXBR = 1: (minor part already specified for mixed nouns in "MIXNOUN" logic)

TS = mid 5 bits of NNTYPTEM shifted right 5

If TS = 4, 5, 7 or 10; NOUNADD = NOUNADD + 1 (specify minor part if double precision)

BUF_0 = - contents of cell specified by NOUNADD (mixed noun specifies $MIXTEMP_0$

Proceed to "DSPCOM2"

DSPB If MIXBR = 1, TS = high 5 bits of NNTYPTEM

If MIXBR = 2, TS = high 5 bits of NNADTEM

If bit 14 of TS = 1, proceed to "DSPALARM"

TS = low 2 bits of (TS shifted right 10)

If TS < 1, proceed to "DSPALARM" (noun has no second component)

BUF_0 = - contents of cell specified by (1 + NOUNADD) (mixed noun specifies $MIXTEMP_1$)

Proceed to "DSPCOM2"

DSPC If MIXBR = 1, TS = high 5 bits of NNTYPTEM

If MIXBR = 2, TS = high 5 bits of NNADTEM

If bit 14 of TS = 1, proceed to "DSPALARM"

TS = low 2 bits of (TS shifted right 10)

If TS < 2, proceed to "DSPALARM"

BUF_0 = - contents of cell specified by (2 + NOUNADD)

Proceed to "DSPCOM2"

<u>DSPAB</u> If MIXBR = 1, TS = bits 12-11 of NNTYPTEM shifted right 10

If MIXBR = 2, TS = bits 12-11 of NNADTEM shifted right 10

If TS < 1, proceed to "DSPALARM" (noun has no second component)

BUF_1 = - contents of cell specified by (1 + NOUNADD)

Proceed to "DSPA"

<u>DSPABC</u> If MIXBR = 1, TS = bits 12-11 of NNTYPTEM shifted right 10

If MIXBR = 2, TS = bits 12-11 of NNADTEM shifted right 10

If TS < 2, proceed to "DSPALARM" (noun has no third component)

BUF_2 = - contents of cell specified by (2 + NOUNADD)

Proceed to "DSPAB"

<u>DSPCOM2</u> i = 0

If VERBREG = 4, i = 1

If VERBREG = 5, i = 2

If i = 2:

DSPCOUNT = K:R3D1 (4)

Perform "DSPOCTWD" with TSwd = - BUF_2

i = i - 1

If i = 1:

DSPCOUNT = K:R2D1 (9)

Perform "DSPOCTWD" with TSwd = - BUF_1

DSPCOUNT = K:R1D1 (14)

Perform "DSPOCTWD" with TSwd = - BUF_0

If ENTRET = "ENDOFJOB", end job

Proceed to address specified by ENTRET

DECDSP If MIXBR = 1, DECOUNT = bits 12-11 of NNTYPTEM (rescaled from B4 to B14)

If MIXBR = 2, DECOUNT = bits 12-11 of NNADTEM

If DECOUNT = 2: (three components)

ZREG = - contents of cell specified by (2 + NOUNADD)

If DECOUNT $\geq$ 1: (two or three components)

YREG = - contents of cell specified by (1 + NOUNADD)

XREG = - contents of cell specified by NOUNADD

DSPDCPUT If DECOUNT = 2:

DSPCOUNT = K:R3D1 (4)

$MPAC_{sp}$ = - ZREG

If MIXBR = 1, i = low 5 bits of NNTYPTEM

If MIXBR = 2, i = high 5 bits of NNTYPTEM shifted right 10

If DECOUNT = 1:

DSPCOUNT = K:R2D1 (9)

$MPAC_{sp}$ = - YREG

If MIXBR = 1, i = low 5 bits of NNTYPTEM

If MIXBR = 2, i = mid 5 bits of NNTYPTEM shifted right 5

If DECOUNT = 0:

DSPCOUNT = K:R1D1 (14)

$MPAC_{sp}$ = - XREG

If MIXBR = 1, i = low 5 bits of NNTYPTEM

If MIXBR = 2, i = low 5 bits of NNTYPTEM

SFTEMP1 = $K:SFOUTAB_i$

If MIXBR = 2, i = high 5, mid 5 or low 5 bits of RUTMXTEM shifted right 10, 5 or 0 according to whether DECOUNT = 2, 1 or 0

If MIXBR = 1, i = mid 5 bits of NNTYPTEM shifted right 5

(i now contains the index determining the type of display)

DECDSP3 If i = 0: (octal only noun)

DSPCOUNT = -19

Proceed to "DSPALARM"

If i = 1, proceed to "DSPDCEND"

If i = 2, proceed to "DEGOUTSF"

If i = 3, proceed to "ARTOUTSF"

If i = 4:

Perform "DPOUT"

$MPAC_{dp}$ = K:b14tob0 $MPAC_{tp}$

If i = 5 or 10:

Perform "DPOUT"

If i = 6, proceed to "LRPOSOUT"

If i = 7:

Perform "DPOUT"

$MPAC_{dp}$ = K:b7tob0 $MPAC_{tp}$

If i = 8, proceed to "HMSOUT"

If i = 9, proceed to "M/SOUT"

If i = 11, proceed to "AROUT1SF"

If i = 12, proceed to "2INTOUT"

If i = 13, proceed to "360-CDUO"

If i = 14, proceed to "RRANGOUT"

If i = 15, proceed to "RRDOTOUT"

Proceed to "DSPDCEND"

ARTOUTSF If $MPAC_0$ = -0:

$MPAC_{dp}$ = -0, skip next line

$MPAC_{dp}$ = SFTEMP1 $MPAC_{sp}$

Proceed to "DSPDCEND"

DEGOUTSF (Entered with SFTEMP1 = 0 and $MPAC_{sp}$ between $-\frac{1}{2}$ and $+\frac{1}{2}$ revs B-1)

MPAC = $MPAC_{sp}$ converted to one's complement form

If MPAC $<$ 0:

MPAC = MPAC + $\frac{1}{2}$

SFTEMP1 = $\frac{1}{2}$

$MPAC_{dp}$ = K:0.180(MPAC + SFTEMP1)

(exit with $MPAC_{dp}$ between 0 and 360 degrees scaled E3)

Proceed to "DSPDCEND"

DPOUT If MIXBR = 1, TS = NOUNADD

If MIXBR = 2:

i = DECOUNT + 1

EBANK = bits 11-9 of $IDADTEM_i$

TS = 01400_8 + bits 8-1 of $IDADTEM_i$

$MPAC_{dp}$ = contents of double precision register whose address is specified by TS

Force sign agreement between two halves of $MPAC_{dp}$

$MPAC_{tp}$ = $MPAC_{dp}$ SFTEMP1

Return

LRPOSOUT $MPAC_{sp}$ = - bits 7-6 of channel 33 shifted right 5

Proceed to "ARTOUTSF"

HMSOUT If MIXBR = 2:

i = DECOUNT + 1

EBANK = bits 11-9 of $IDADTEM_i$

TS = 01400_8 + bits 8-1 of $IDADTEM_i$

If MIXBR = 1, TS = NOUNADD

$MPAC_{dp}$ = contents of double precision register whose address is specified by TS

Force sign agreement between two halves of $MPAC_{dp}$

TS_{tp} = K:SECON1 $MPAC_{dp}$

HITEMOUT = most significant third of TS_{tp}

LOTEMOUT = second most significant third of TS_{tp}

$MPAC_{dp}$ = fractional part of TS_{tp}

$MPAC_{dp}$ = K:SECON2 $MPAC_{dp}$

DSPCOUNT = K:R3D1

Perform "DSPDECWD"

Discard fractional part of LOTEMOUT (shift right 12 then left 12)

$MPAC_{tp}$ = K:MINCON1 (HITEMOUT + LOTEMOUT added with regard for scaling) (yields hours in $MPAC_0$ and minutes/60 in $MPAC_1$, scaled B14 and B0 respectively)

HITEMOUT = most significant third of $MPAC_{tp}$ ($MPAC_0$)

If $MPAC_0$ = -0:

    $MPAC_{dp}$ = -0, skip next line

$MPAC_{dp}$ = K:MINCON2 $MPAC_1$

DSPCOUNT = K:R2D1

Perform "DSPDECWD:

If HITEMOUT = -0:

    $MPAC_{dp}$ = -0, skip next line

$MPAC_{dp}$ = K:HRCON1 HITEMOUT

DSPCOUNT = K:R1D1

Perform "DSPDECWD"

If ENTRET = "ENDOFJOB", end job

Proceed to address specified by ENTRET

M/SOUT If MIXBR = 1, TS = NOUNADD

If MIXBR = 2:

i = DECOUNT + 1

EBANK = bits 11-9 of $IDADTEM_i$

TS = 01400_8 + bits 8-1 of $IDADTEM_i$

$MPAC_{dp}$ = contents of double precision register whose address is specified by TS

Force sign agreement between two halves of $MPAC_{dp}$

If $|MPAC_{dp}| \geq |K:MdSCON1|$, $MPAC_{dp}$ = K:MdSCON3 sign$MPAC_{dp}$

If $|MPAC_{dp}| < |K:MdSCON1|$, $MPAC_{dp}$ = $MPAC_{dp}$ + K:RNDCON sign$MPAC_{dp}$

TS_{tp} = K:SECON1 $MPAC_{dp}$

HITEMOUT = most significant third of TS_{tp}

LOTEMOUT = second most significant third of TS_{tp}

$MPAC_{dp}$ = fractional part of TS_{tp}

$MPAC_{dp}$ = K:HISECON $MPAC_{dp}$

DSPCOUNT = DSPCOUNT - 3 (previously set in "DSPDCPUT")

Perform "DSPDC2NR" (display seconds in RxD4 and RxD5)

CODE = 0

TS = K:R1D1, K:R2D1 or K:R3D1 according to whether DECOUNT = 0,1 or 2

COUNT = TS - 2

Perform "DSPIN" (blank middle digit)

Discard fractional part of LOTEMOUT (shift right 12 then left 12)

$MPAC_{tp}$ = K:MINCON1 (HITEMOUT + LOTEMOUT added with regard for scaling) (yields fraction of hours in $MPAC_1$ scaled B0)

$MPAC_{dp}$ = K:HIMINCON $MPAC_1$

DSPCOUNT = K:R3D1, K:R2D1 or K:R1D1 according to whether DECOUNT equals 2, 1 or 0

Perform "DSPDC2NR"   (display minutes in RxD1 and RxD2)

Proceed to second step of "DSPDCEND"   (display already performed)

<u>2INTOUT</u> Perform "5BLANK"

Perform "+ON"

Perform "DSPDECVN" with TS = $MPAC_{sp}$ (first integer to RxD1 and D2)

TS = K:R1D1, K:R2D1 or K:R3D1 according to whether DECOUNT = 0, 1 or 2

DSPCOUNT = TS - 3

If MIXBR = 1, TS = 1 + NOUNADD

If MIXBR = 2:

i = DECOUNT + 1

EBANK = bits 11-9 of $IDADTEM_i$

TS = 01400_8 + 1 + bits 8-1 of $IDADTEM_i$

$MPAC_1$ = contents of single precision register whose address is specified by TS

Perform "DSPDECVN" with TS = $MPAC_1$ (second to RxD4 and RxD5)

Proceed to second step of "DSPDCEND"   (display already performed)

<u>360-CDUO</u> If $MPAC_{sp} \neq 0$ or $-\frac{1}{2}$:

$MPAC_{sp} = - MPAC_{sp}$ (two's complement)

Proceed to "DEGOUTSF"

<u>DSPDCEND</u> Perform "DSPDECWD"

If DECOUNT > 0:

DECOUNT = DECOUNT - 1

Proceed to "DSPDCPUT"

If ENTRET = "ENDOFJOB", end job

Proceed to address specified by ENTRET

<u>DSPDPDEC</u> If MIXBR = 2, proceed to "DSPALARM"

$MPAC_{dp}$ = contents of double precision register whose address is specified by NOUNADD

DSPCOUNT = K:R1D1

Force sign agreement in $MPAC_{dp}$

Perform "DSP2DEC"

If ENTRET = "ENDOFJOB", end job

Proceed to address specified by ENTRET

<u>MONITOR</u> TSadr = NOUNCADR with bits 15 and 14 switched to 0

If ENTRET = "ENDOFJOB" (externally initiated)

Switch bit 14 of TSadr to 1

MONSAVE2 = 00000_8 (set in "NVSUB" if internal)

TSvn = (bits 7-1 of VERBREG shifted left 7) + NOUNREG

DSPLOCK = 0

If CADRSTOR = 0, perform "RELDSP1"

Inhibit interrupts

If MONSAVE = 0, call "MONREQ" in 0.01 second

MONSAVE = TSvn

MONSAVE1 = TSadr

Release interrupt inhibit

If ENTRET = "ENDOFJOB", end job

Proceed to address specified by ENTRET

<u>MONREQ</u> SAMPTIME = TIMENOW

If bit 15 of MONSAVE1 = 1:

MONSAVE = 0

(If bit 15 of MONSAVE1 = 1:)

MONSAVE1 = 00000_8

End task

Call "MONREQ" in 1 second

Establish "MONDO" (pr30)

End task

MONDO If bit 15 of MONSAVE1 = 1, end job

If DSPLOCK > 0:

Switch bit 5 of channel 11 to 1 (key release lamp)

End job

NOUNREG = bits 7-1 of MONSAVE

Perform "UPDATNN"

VERBREG = (bits 14-8 of MONSAVE shifted right 7) - 10

ENTRET = "PASTEVB"

$MPAC_2$ = bits 13-1 of MONSAVE1 (address)

Proceed to "TESTNN" (continues at "PASTEVB" after display verb is executed)

PASTEVB TS = bits 14-8 of MONSAVE2

If TS = 0, TS = bits 14-8 of MONSAVE

VERBREG = TS shifted right 7 places

Perform "UPDATVB"

REQRET = +0

TSblank = MONSAVE2

Perform "BLANKSUB"

End job

UPDATNN Perform "LODNNTAB"

If NNADTEM $\geq$ +0:

NOUNCADR = NNADTEM

EBANK = bits 11-9 of NNADTEM

NOUNADD = 01400_8 + bits 8-1 of NNADTEM

DSPCOUNT = K:ND1

Perform "DSPDECVN" with TS = NOUNREG

Return

UPDATVB DSPCOUNT = K:VD1

Perform "DSPDECVN" with TS = VERBREG

Return

DSPMMJOB TSmmtemp = DSPCOUNT

DSPCOUNT = K:MD1

If MODREG = -0, perform "2BLANK"

If MODREG $\geq$ +0, perform "DSPDECVN" with TS = MODREG

DSPCOUNT = TSmmtemp

End job

ALMCYCLE Switch bit 7 of channel 11 to 1 (operator error lamp)

REQRET = - VERBSAVE (to make it positive)

VERBREG = - VERBSAVE

Perform "UPDATVB"

CLPASS = 0

ENTRET = "ENDOFJOB"

Proceed to "ENTPASO"

<u>ALOAD</u> Perform "REQDATX" (return is via REQRET after data entry)

Perform "LODNNTAB"

Perform "PUTCOM" with DECOUNT = 0

Store TS from "PUTCOM" in address specified by NOUNADD

Proceed to "LOADLV"

<u>BLOAD</u> Perform "GETCOMP"

TS = low 2 bits of (TS shifted right 10)

If TS < 1, proceed to "DSPALARM" (noun has no 2nd component)

CLPASS = − K:posmaxsp

Perform "REQDATY" (return is via REQRET after data entry)

Perform "LODNNTAB"

Perform "PUTCOM" with DECOUNT = 1

Store TS from "PUTCOM" in address specified by (1 + NOUNADD)

Proceed to "LOADLV"

<u>CLOAD</u> Perform "GETCOMP"

TS = low 2 bits of (TS shifted right 10)

If TS < 2, proceed to "DSPALARM" (noun has no 3rd component)

CLPASS = − K:posmaxsp

Perform "REQDATZ" (return is via REQRET after data entry)

Perform "LODNNTAB"

Perform "PUTCOM" with DECOUNT = 2

Store TS from "PUTCOM" in address specified by (2 + NOUNADD)

Proceed to "LOADLV"

ABLOAD Perform "GETCOMP"

TS = low 2 bits of (TS shifted right 10)

If TS < 1, proceed to "DSPALARM" (noun has no 2nd component)

Perform "GETCOMP"

If bit 15 of TS = 1, proceed to "DSPALARM" ("no-load" noun)

VERBREG = K:VB21

Perform "UPDATVB"

Perform "REQDATX" (return is via REQRET after data entry)

VERBREG = K:VB22

Perform "UPDATVB"

Perform "REQDATY"

Bits 5 and 4 of DECBRNCH now indicate whether the numbers loaded were decimal (1) or octal (0). (See routine "BOTHSGN") If both are not the same (one component octal, the other decimal):

Proceed to "ALMCYCLE"

Perform "LODNNTAB"

Perform "PUTCOM" with DECOUNT = 0

Store TS from "PUTCOM" in address specified by NOUNADD

Perform "PUTCOM" with DECOUNT = 1

Store TS from "PUTCOM" in address specified by (1 + NOUNADD)

Proceed to "LOADLV"

ABCLOAD Perform "GETCOMP"

TS = low 2 bits of (TS shifted right 10)

If TS < 2, proceed to "DSPALARM" (noun has no 3rd component)

Perform "GETCOMP"

If bit 15 of TS = 1, proceed to "DSPALARM" ("no-load" noun)

VERBREG = K:VB21

Perform "UPDATVB"

Perform "REQDATX"   (return is via REQRET after data entry)

VERBREG = K:VB22

Perform "UPDATVB"

Perform "REQDATY"

VERBREG = K:VB23

Perform "UPDATVB"

Perform "REQDATZ"

Bits 3, 4 and 5 of DECBRNCH now indicate whether the numbers loaded were decimal (1) or octal (0). If the three bits are not all 1 or all 0 (some components octal and some decimal):

Proceed to "ALMCYCLE"

Perform "LODNNTAB"

Perform "PUTCOM" with DECOUNT = 0

Store TS from "PUTCOM" in address specified by NOUNADD

Perform "PUTCOM" with DECOUNT = 1

Store TS from "PUTCOM" in address specified by (1 + NOUNADD)

Perform "PUTCOM" with DECOUNT = 2

Store TS from "PUTCOM" in address specified by (2 + NOUNADD)

If NOUNREG $\neq$ 7, proceed to "LOADLV"

Inhibit interrupts

TS = XREG - 30_8

If TS $\leq$ 0, proceed to "CHANBITS"

EBANK = bits 11-9 of XREG

NOUNADD = 01400_8 + (bits 8-1 of XREG)

Channel 1 = contents of location specified by NOUNADD

XREG = 1   (channel 1 is the computer L register)

TS = 1

CHANBITS   If TS + 21_8 = 0, proceed to "BITSOFF2"   (channel 7)

If ZREG $\leq$ 0:

Set those bits of $\text{channel}_{\text{XREG}}$ = 0 that are 1 in YREG

Proceed to "BITSOFF1"

Set those bits of $channel_{XREG}$ = 1 that are 1 in YREG

BITSOFF1 If XREG = 1 or XREG < 0, $E_{NOUNADD}$ = Channel 1 (computer L register

BITSOFF2 Release interrupt inhibit

Proceed to "LOADLV"

GETCOMP If MIXBR = 1, TS = high 5 bits of NNTYPTEM

If MIXBR = 2, TS = high 5 bits of NNADTEM

Return

PUTCOM DECRET = return address

Set overflow indicator to 0

$MPAC_{dp}$ = (XREG + XREGLP), (YREG + YREGLP) or (ZREG + ZREGLP) according to whether DECOUNT = 0, 1 or 2

If MIXBR = 1, proceed to "PUTNORM"

i = DECOUNT + 1

NOUNCADR = low 11 bits of $IDADTEM_i$

EBANK = bits 11-9 of NOUNCADR

NOUNADD = (01400_8 + bits 8-1 of NOUNCADR) - DECOUNT

If DECBRNCH > 0: (decimal)

Perform "GETI" with TS = NNTYPTEM

SFTEMP1 = $K:SFINTAB_i$

Perform "GETI" with TS = RUTMXTEM

Proceed to "PUTDCSF2"

Perform "GETCOMP"

If bit 14 of TS = 1, proceed to "ALMCYCLE" (decimal only)

Perform "GETI" with TS = RUTMXTEM

If i = 4, 5, 7 or 10: (double precision noun)

Set (the more significant half of the double precision register specified by NOUNADD + DECOUNT) = 0

NOUNADD = NOUNADD + 1 (specify minor part)

Proceed to "PUTCOM2"

GETI i = high 5 bits (DECOUNT = 2), mid 5 bits (DECOUNT = 1) or low 5 bits (DECOUNT = 0) of TS shifted right 10, 5 or 0 places according to whether DECOUNT = 2, 1 or 0.

(i is of the form $000\ 000\ 000\ 0xx\ xxx_2$)

Return

PUTNORM EBANK = bits 11-9 of NOUNCADR (NOUNCADR set in "TESTNN")

NOUNADD = 01400_8 + bits 8-1 of NOUNCADR

If DECBRNCH > 0: (decimal)

i = low 5 bits of NNTYPTEM

SFTEMP1 = K:$SFINTAB_i$

i = mid 5 bits of NNTYPTEM shifted right 5

Proceed to "PUTDCSF2"

Perform "GETCOMP"

If bit 14 of TS = 1, proceed to "ALMCYCLE" (decimal only)

i = mid 5 bits of NNTYPTEM shifted right 5

If i = 4, 5, 7 or 10: (double precision noun)

Set (the more significant half of the double precision register specified by NOUNADD) = 0

NOUNADD = NOUNADD + 1

Proceed to "PUTCOM2"

If NNADTEM = -1: (channel load)

If NOUNCADR = 7, proceed to "LOADLV"

Set channel specified by NOUNCADR equal to the more significant half of $MPAC_{dp}$ (in low 9 bits of NOUNCADR)

Proceed to "LOADLV"

Proceed to "PUTCOM2"

PUTDCSF2 If i = 0, proceed to "ALMCYCLE" (octal only)

If i = 1, proceed to "BINROUND"

If i = 2, proceed to "DEGINSF"

If i = 3:

$MPAC_{dp}$ = SFTEMP1 K:b0tobm14 $MPAC_{dp}$

If overflow, proceed to "ALMCYCLE"

Proceed to "BINROUND"

If i = 4 or 7:

$MPAC_{tp}$ = SFTEMP1 $MPAC_{dp}$

Proceed to "DPINSF+2"

If i = 5:

$MPAC_{tp}$ = SFTEMP1 K:b0tobm7 $MPAC_{dp}$

Proceed to "DPINSF+2"

If i = 6, proceed to "DSPALARM" (LR position is display only)

If i = 8, proceed to "HMSIN"

If i = 9, proceed to "DSPALARM" (min/sec cannot be loaded)

If i = 10:

$MPAC_{tp}$ = SFTEMP1 K:b0tobm3 $MPAC_{dp}$

Proceed to "DPINSF+2"

If i = 11:

  $MPAC_{dp}$ = $MPAC_{dp}$ SFTEMP1

  Proceed to "BINROUND"

If i = 12, proceed to "DSPALARM" (2INT cannot be loaded)

If i = 13, proceed to "DEGINSF" (test for 360-CDU in "DEGINSF")

If i = 14 or 15, proceed to "DSPALARM" (RR data cannot be loaded)

<u>BINROUND</u> $MPAC_{sp}$ = $MPAC_{dp}$ rounded off

If overflow, proceed to "ALMCYCLE"

Proceed to "PUTCOM2"

<u>DPINSF+2</u> $MPAC_{dp}$ = $MPAC_{tp}$ rounded off

If overflow, proceed to "ALMCYCLE"

If MIXBR = 1, TS = NOUNADD

If MIXBR = 2, TS = NOUNADD + DECOUNT

Store less significant half of $MPAC_{dp}$ in less significant half of double precision register specified by TS

$MPAC_{sp}$ = more significant half of $MPAC_{dp}$

Proceed to "PUTCOM2"

<u>DEGINSF</u> $MPAC_{dp}$ = K:DEGCON1 $MPAC_{dp}$

$MPAC_{sp}$ = $MPAC_{dp}$ rounded off and rescaled to B-1 (s13)

If $|MPAC_{sp}| \geq 1$ (360°), proceed to "ALMCYCLE"

MPAC = $MPAC_{sp}$ converted to two's complement form

If MPAC $\geq \frac{1}{2}$ (180°):

  MPAC = $-$(1 - MPAC)

If MPAC $< -\frac{1}{2}$:

  MPAC = MPAC + 1

If MIXBR = 1, i = mid 5 bits of NNTYPTEM shifted right 5

If MIXBR = 2, perform "GETI" with TS = RUTMXTEM

If $i \neq 2$: (360-CDU)

If $MPAC_{sp} \neq 0$ or $-\frac{1}{2}$, $MPAC_{sp} = -MPAC_{sp}$ (two's complement)

$MPAC_{sp}$ = MPAC

Proceed to "PUTCOM2"

<u>HMSIN</u> If bits 3, 4 and 5 of DECBRNCH are not all 1:
(three decimal components have not been loaded)

VERBSAVE = - K:VB25 (initiate ABCLOAD)

Proceed to "ALMCYCLE"

TS = K:HRCON (10^5 $MPAC_{dp}$ rounded to whole hours)

If $|TS| \geq 2^{28}$, proceed to "ALMCYCLE" (745 hour max)

HITEMIN = TS

TS = 10^5 (YREG + YREGLP) rounded to whole minutes

If $|TS| >$ K:59MIN, proceed to "ALMCYCLE"

HITEMIN = HITEMIN + K:MINCON TS

If $|HITEMIN| \geq 2^{28}$, proceed to "ALMCYCLE"

TS = 10^5 (ZREG + ZREGLP) rounded to whole centiseconds

If $|TS| >$ K:59.99SEC, proceed to "ALMCYCLE"

TS = HITEMIN + TS

If $|TS| \geq 2^{28}$, proceed to "ALMCYCLE"

$MPAC_{dp}$ = TS with forced sign agreement between two halves

Store $MPAC_{dp}$ in double precision register specified by NOUNADD

Proceed to "LOADLV"

<u>PUTCOM2</u> TS = $MPAC_{sp}$

Return via DECRET

__DSPFMEM__ DSPCOUNT = K:R1D1

If NOUNCADR $\neq$ 11x xxx xxx xxx xxx_2:

TS = contents of fixed memory cell whose address is specified by information in NOUNCADR (Standard fixed memory CADR format contains FBANK information in bits 15-11 and S-register information in bits 10-1.)

If NOUNCADR = 11x xxx xxx xxx xxx_2:

TS = contents of fixed memory cell whose address is specified by information in NOUNCADR and in $DSPTEM1_2$. ($DSPTEM1_2$ contains FBANK extension or "Superbank" information in bits 7-5 and must be loaded prior to verb 27 entry with a verb 23, noun 26 for access to fixed memory banks 30_8 through 43_8.)

Perform "DSPOCTWD"

End job

__MMCHANG__ Perform "REQMM" (return is via REQRET on data entry)

If DSPCOUNT $\neq$ -16, proceed to "ALMCYCLE"

TS = NOUNREG (which contains desired major mode)

NOUNREG = 0

Perform "2BLANK" with DSPCOUNT = K:ND1

DSPCOUNT = -19 (to block further numerical entries)

MMNUMBER = TS

Proceed to "V37"

__RRANGOUT__ $MPAC_{dp}$ = $MPAC_{sp}$ (15 magnitude bits) converted to double precision

If RADMODES bit 3 (RRRSFLAG) = 0: (low scale)

Skip next step

$MPAC_{dp}$ = 8 $MPAC_{dp}$ (high scale)

$MPAC_{tp}$ = $MPAC_{dp}$ SFTEMP1

$MPAC_{dp}$ = K:b14tob0 $MPAC_{tp}$

Proceed to "DSPDCEND"

RRDOTOUT $MPAC_{dp} = MPAC_{sp}$ (15 magnitude bits) converted to double precision

$MPAC_{dp} = MPAC_{dp} - 17000$ (subtract bias of 17000 counts)

$MPAC_{tp} = MPAC_{dp}$ SFTEMP1

$MPAC_{dp}$ = K:b14tob0 $MPAC_{tp}$

Proceed to "DSPDCEND"

"AROUT1SF" If $MPAC_0 = -0$:

$MPAC_{dp} = -0$

Skip next line

$MPAC_{dp}$ = K:b14tob0 SFTEMP1 $MPAC_{sp}$

Proceed to "DSPDCEND"

BUF_i (i = 0,1,2): Three single precision octal working storage cells.

CADRSTOR: See DINT section.

CLPASS, CODE, COUNT: See DSKY section.

DECBRNCH: See DSKY section.

DECOUNT: Single precision number of components in a noun display, extracted from the relevant bits of NNTYPTEM or NNADTEM, scaled B14 and unitless. (Number of components = DECOUNT + 1.)

DECRET: Single precision octal return address storage.

DSPCOUNT, DSPLOCK: See DSKY section.

$DSPTAB_i$ (i = 0-10): See DSKY section.

DSPTEM1, DSPTEM2, DSPTEMX: Single, double or triple precision display interface registers with variable scaling and units. See tables below.

EBANK: See MATX section.

ENTRET: Single precision octal address indicating whether the data input/output is under control of the astronaut or of internal programs.

FBANK: See MATX section.

FREEDSKY: Variable introduced as a substitute for variable return address; set to 1 or 0 to indicate whether the DSKY is free or unavailable for use by internal programs.

HITEMIN: Double precision working storage in "HMSIN", scaled B28 in units of centiseconds.

HITEMOUT: Single precision working storage for minutes scaled B16 or hours scaled B14.

$IDADTEM_i$ (i = 1,2,3): Three single precision octal words containing the address and EBANK information for each of the three separate registers that can be displayed or loaded by a mixed noun. (Equal to +0 if the component of the mixed noun is not utilized.)

K:0.180: Double precision constant stored as 05605_8 03656_8, scaled B1 in units of degrees per revolution; program notation is DEGTAB. Equation value: 360/1000.

K:59MIN: Single precision constant stored as 00073_8, scaled B14 in units of minutes. Equation value: 59.

K:59.99SEC: Single precision constant stored as 13557_8, scaled B14 in units of centiseconds. Equation value: 5999.

K:b0tobm14: Constant representing the effect of a left shift of 14, scaled B-14 and unitless. Equation value: 1.

K:b0tobm3: Constant representing the effect of a left shift of 3, scaled B-3 and unitless. Equation value: 1.

K:b0tobm7: Constant representing the effect of a left shift of 7, scaled B-7 and unitless. Equation value: 1.

K:b14tob0: Constant representing the effect of a left shift of 14, scaled B-14 and unitless. Equation value: 1.

K:b7tob0: Constant representing the effect of a left shift of 7, scaled B-7 and unitless. Equation value: 1.

K:DEGCON1: Double precision constant stored as $5.555555555 \times 2^{-3}$, scaled B2 in units of revolutions per degree. Equation value: 1000 / 360.

K:HIMINCON: Single precision constant stored as 23346_8, scaled B0 in units of minutes per hour. Equation value: 60.8 / 100 (for round off).

K:HISECON: Single precision constant stored as 23147_8, scaled B0 in units of seconds per minute. Equation value: 60 / 100.

K:HRCON: Double precision constant stored as 00025_8 37100_8, scaled B28 in units of centiseconds per hour. Equation value: 360,000.

K:HRCON1: Double precision constant stored as 0.16384, scaled B-14 and unitless. Equation value: 0.00001.

$K:IDADTAB_i$ (i = 0-179): Table of single precision addresses for mixed nouns, loaded into $IDADTEM_{1,2,3}$ according to the value of NNADTEM Program name; IDADDTAB.

K:MD1: Single precision constant to cause a numerical display to be started in the first digit of the major mode (program) register, scaled B14 and unitless. Equation value: 21.

K:MdSCON1: Double precision constant stored as 77753_8 41126_8, scaled B28 in units of centiseconds. Equation value: -359,850 (Equivalent to -59:58.5 = -25_8 36652_8 because of CCS.)

K:MdSCON3: Double precision constant stored as 00025_8 37016_8, scaled B28 in units of centiseconds. Equation value: 359,950. (Equivalent to 59:59.5.)

K:MINCON: Single precision constant stored as 13560_8, scaled B14 in units of centiseconds per minute. Equation value: 6000.

K:MINCON1: Double precision constant stored as 02104_8 10422_8, scaled B-2 in units of hours per minute. Equation value: 1/60.

K:MINCON2: Double precision constant stored as 00011_8 32445_8, scaled B0 in units of minutes per hour. Equation value: 60/100000.

K:ND1: See DSKY section.

$K:NNADTAB_i$ (i = 0-99): Table of single precision constants to be loaded into NNADTEM according to the value of NOUNREG. See tables 1 and 2 below.

$K:NNTYPTAB_i$ (i = 0-99): Table of single precision constants to be loaded into NNTYPTEM according to the value of NOUNREG. See tables 1 and 2 below.

K:posmaxsp: See "Major Variables" section.

K:R1D1, K:R2D1, K:R3D1: See DSKY section.

K:RNDCON: Double precision constant stored as 00000_8 00062_8, scaled B28 in units of centiseconds. Equation value: 50.

$K:RUTMXTAB_i$ (i = 40-99): Table of single precision constants to be loaded into RUTMXTEM according to the value of NOUNREG. See table 2 below.

K:SECON1: Double precision constant stored as 1.666666666 E-4 x 2^{12}, scaled B-12 in units of minutes per centisecond. Equation value: 1/6000.

K:SECON2: Double precision constant stored as 01727_8, 01217_8, scaled B-14 in units of seconds per minute. Equation value: 60/1000.

K:SFINTAB$_i$ (i = 0-28): Table of double precision constants with variable scaling and units, used to convert from units used on a DSKY display into units used in the LGC. See table 3 below.

K:SFOUTAB$_i$ (i = 0-28): Table of double precision constants with variable scaling and units, used to convert from units used in the LGC to units used on a DSKY display. See table 3 below.

K:VB21, K:VB22, K:VB23, K:VB25: Single precision constants stored as 21, 22, 23, and 25 times two to the minus fourteenth power, scaled B14 and unitless. Equation values respectively: 21, 22, 23, 25.

K:VD1: See DSKY section.

LOADSTAT: See DINT section.

LOTEMOUT: Single precision working storage for minutes, scaled B2.

MIXBR: Single precision index indicating whether the noun being processed is a "mixed" noun (addresses non-consecutive E-memory cells) or a "normal" noun (addresses one or more consecutive E-memory cells); scaled B14 and unitless.

MIXTEMP$_i$ (i = 0,1,2): Three consecutive single precision E-memory cells loaded with the values of the three non-consecutive registers addressed by a mixed noun so that the same display logic can be used for both normal and mixed after the MIXTEMP$_i$ are loaded and NOUNADD is set equal to the address of MIXTEMP$_0$.

MMNUMBER: See PGSR section.

MODREG: Single precision register reflecting the status of the "major mode" or "program" number on the DSKY, scaled B14 and unitless.

MONSAVE: Single precision storage for monitor verb and noun (verb number in bits 14-8, noun in bits 7-1).

MONSAVE1: Single precision octal storage for the address of the noun to be displayed by the monitor routines. Bits 15 and 14 are used as flag bits; Bit 15 is set to terminate the monitor, and bit 14 is set to indicate that the monitor was initiated by the astronaut and thus takes priority over displays requested by the program.

MONSAVE2: Single precision storage for an octal blanking code and/or a verb to be "pasted" over the display verb during a monitor.

MPAC, $MPAC_{sp}$, $MPAC_{tp}$: Single, double and triple precision working storage cells: $MPAC_{sp} = MPAC_0$, $MPAC_{dp} = MPAC_{0,1}$, $MPAC_{tp} = MPAC_{0,1,2}$.

$MPAC_2$: Single precision storage for an octal address when an "address-to-be-specified noun" is used by LGC programs. Instead of requesting the address from the astronaut, the program finds it in $MPAC_2$. (see "TESTNN").

NNADTEM: Single precision octal word containing the following information. If the noun is a normal noun, NNADTEM contains the normal noun address and EBANK. If the noun is a mixed noun, NNADTEM contains the "no-load" and "decimal only" indicators (bits 15 and 14), the indication of the number of components in the noun (bits 12-11), and the index used to load $IDADTEM_i$ (see "LODNNTAB").

NNTYPTEM: Single precision octal word containing the following information. If the noun is a normal noun, NNTYPTEM contains the "no-load" and "decimal only" indicators (bits 15 and 14), the indication of the number of components in the noun (bits 12-11), the specification of the routine to be used for input/output (bits 10-6), and the index used in selecting the scale factor to be used in input/output scaling (bits 5-1). If the noun is a mixed noun, NNTYPTEM contains the index used in selecting the scale factor for each of the three components. Bits 15-11 contain the index for the third component, bits 10-6 contain the index for the second component, and bits 5-1 contain the index for the first component.

NOUNADD: Single precision octal address of register or registers to be displayed or loaded by the noun being processed.

NOUNCADR: Single precision octal address of the most recent register displayed by a normal noun or loaded.

NOUNREG: Single precision storage for the value of the number currently displayed in the noun register on the DSKY, scaled B14 and unitless.

NOUNTEM: Single precision temporary storage for the address of one of the components of a mixed noun.

NOUT: See INTR section.

NVQTEM: Double precision octal return address storage. Program notation for least significant half is NVBNKTEM.

NVTEMP: Single precision storage for verb-noun combination, the noun number stored in bits 7-1 and the verb number stored in bits 14-8; used instead to indicate a desired blanking option if bit 15 is set (if NVTEMP is negative).

OPTION1, OPTION2, OPTION3: Three single precision option codes for display to the astronaut via noun 6. The first indicates the subject of the decision to be made, the second indicates the choice made, which he may accept or change, the third indicates flagbit settings.

RADMODES: See RADR section.

REQRET: See DSKY section.

RUTMXTEM: Single precision octal word loaded only for mixed nouns. Bits 15-11 specify the routine to be used in input/output of the third component of the mixed noun; bits 10-6 specify the routine for the second component; bits 5-1 specify the routine for the first component.

SAMPTIME: See DSKY section.

SFTEMP1: Double precision storage for the conversion/scale factor in decimal input/output routines.

TIMENOW: See EXVB section.

VERBREG: Single precision storage for the value of the number currently displayed in the verb register on the DSKY, scaled B14 and unitless.

VERBSAVE: Single precision storage for the value of VERBREG (complemented at the beginning of verb processing, see "ALMCYCLE").

VGDISP: See DELVSAB in BURN section.

XREG, XREGLP: Two halves (most and least significant) of the five digit number currently input into the first data register on the DSKY (R1), scaled B0 assuming that the decimal point is on the left of the display register.

YREG, YREGLP: The equivalent of XREG and XREGLP for R2 instead of R1.

ZREG, ZREGLP: The equivalent of XREG and XREGLP for R3 instead of R1.

Table 1

Normal Nouns

Noun	K:NNADTAB (addresses are normally positive)	K:NNTYPTAB bits 15-13	bits 12-11	bits 10-6	bits 5-1
0	+0	0 0 0	0	0	0
1	-K:posmaxsp	0 0 0	2	1	0
2	-K:posmaxsp	0 0 0	2	3	0
3	-K:posmaxsp	0 0 0	2	2	2
4	DSPTEM1	0 0 0	0	10	4
5	DSPTEM1	0 0 0	0	10	4
6	OPTION1,OPTION2,OPTION3	0 0 0	2	0	0
7	XREG, YREG, ZREG	0 0 0	2	0	0
8	$ALMCADR_{dp}$, ERCOUNT	0 0 0	2	0	0
9	$FAILREG_{0,1,2}$	0 0 0	2	0	0
10	-1	0 0 0	0	0	0
11	TCSI	0 1 0	2	8	0
12	OPTIONX	0 0 0	1	0	0
13	TCDH	0 1 0	2	8	0
14	$DSPTEMX_{0,1,2}$	0 0 0	2	3	0
15	-0	0 0 0	0	0	0
16	$DSPTEMX_{dp}$	0 1 0	2	8	0
17	+0	0 0 0	0	0	0
18	FDAI	0 0 0	2	2	2
19	+0	0 0 0	0	0	0
20	CDU	0 0 0	2	2	2
21	PIPA	0 0 0	2	3	0
22	THETAD	0 0 0	2	2	2
23	+0	0 0 0	0	0	0
24	$DSPTEMX_{dp}$	0 1 0	2	8	0
25	$DSPTEM1_{0,1,2}$	0 0 0	2	3	0
26	$DSPTEM1_{0,1,2}$	0 0 0	2	0	0
27	SMODE	0 0 0	0	3	0
28	+0	0 0 0	0	0	0
29	+0	0 0 0	0	0	0
30	+0	0 0 0	0	0	0
31	+0	0 0 0	0	0	0
32	mTPER	0 1 0	2	8	0
33	TIG	0 1 0	2	8	0
34	$DSPTEM1_{dp}$	0 1 0	2	8	0
35	TTOGO	0 1 0	2	8	0
36	TIMENOW	0 1 0	2	8	0
37	TTPI	0 1 0	2	8	0
38	TET	0 1 0	2	8	0
39	+0	0 0 0	0	0	0

Table 2

Mixed Nouns

Noun	K:NNADTAB (bits 15,14,13 12-11, 10-1)	i	K:IDADTAB K:IDADTAB^{i} K:IDADTAB$^{i+1}_{i+2}$	K:NNTYPTAB (binary)	K:RUTMXTAB (binary)
40	1 1 0 2	0	TTOGO VGDISP DVTOTAL	01010 01010 00000	00111 00111 01001
41	0 0 0 1	3	$DSPTEM1_0$ $DSPTEM1_1$ +0	00000 01011 00010	00000 00011 00010
42	0 1 0 2	6	HAPO HPER VGDISP	01010 01000 01000	00111 00111 00111
43	0 1 0 2	9	LAT LONG ALT	01000 00100 00100	00111 01010 01010
44	1 1 0 2	12	HAPOX HPERX TFF	00000 01000 01000	01001 00111 00111
45	1 1 0 2	15	TRKMKCNT TTOGO pMGA	00100 00000 00000	01010 01001 00011
46	0 0 0 0	18	DAPDATR1 +0 +0	00000 00000 00000	00000 00000 00000
47	0 1 0 1	21	LEMMASS CSMMASS +0	00000 00110 00110	00000 01011 01011
48	0 1 0 1	24	PITTIME ROLLTIME +0	00000 10111 10111	00000 00011 00011
49	0 1 0 2	27	R22DISP R22DISP+2 WHCHREAD	00000 01010 01000	00011 00111 00111
50	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000
51	0 1 0 1	33	ALPHASB BETASB +0	00000 00100 00100	00000 01010 01010
52	0 0 0 0	36	ACTCENT +0 +0	00000 00000 00100	00000 00000 01010
53	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000

Table 2 continued

Noun	K:NNADTAB		K:IDADTAB	K:NNTYPTAB	K:RUTMXTAB
54	0 1 0 2	42	RANGE RRATE RTHETA	00100 01010 00111	01010 00111 00100
55	0 1 0 2	45	NN ELEV CENTANG	00100 00100 00000	01010 01010 00011
56	0 0 0 1	48	RR-AZ RR-ELEV +0	00000 00100 00100	00000 01010 01010
57	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000
58	0 1 0 2	54	POSTTPI DELVTPI DELVTPF	01010 01010 01000	00111 00111 00111
59	0 1 0 2	57	$DVLOS_x$ $DVLOS_y$ $DVLOS_z$	01010 01010 01010	00111 00111 00111
60	0 1 0 2	60	FORVEL HDOTDISP HCALC1	11000 01010 10001	00100 00111 00011
61	1 1 0 2	63	TTFDISP TTOGO OUTOFPLN	10110 00000 00000	01010 01001 01001
62	1 1 0 2	66	ABVEL TTOGO DVTOTAL	01010 00000 01010	00111 01001 00111
63	0 1 0 2	69	ABVEL HDOTDISP HCALC1	11000 01010 01010	00100 00111 00111
64	1 1 0 2	72	FUNNYDSP HDOTDISP HCALC	11000 01010 00000	00100 00111 01100
65	0 1 0 2	75	SAMPTIME SAMPTIME SAMPTIME	00000 00000 00000	01000 01000 01000
66	1 1 0 1	78	$RSTACK_6$ +0 +0	00000 00000 01110	00000 00110 00100
67	0 0 0 2	81	$RSTACK_0$ $RSTACK_2$ $RSTACK_4$	10101 10100 10011	00100 00100 00100
68	1 1 0 2	84	RANGEDSP TTFDISP DELTAH	11000 00000 10110	00100 01001 01010
69	0 1 0 2	87	DLANDZ DLANDY DLANDX	11000 11000 11000	00100 00100 00100

Table 2 continued

Noun	K:NNADTAB					K:IDADTAB	K:NNTYPTAB	K:RUTMXTAB
70	0	0	0	2	90	AOTCODE $AOTCODE_1$ $AOTCODE_2$	00000 00000 00000	00000 00000 00000
71	0	0	0	2	93	AOTCODE $AOTCODE_1$ $AOTCODE_2$	00000 00000 00000	00000 00000 00000
72	0	0	0	1	96	CDU_t CDU_s +0	00000 00010 00010	00000 00010 01101
73	0	0	0	1	99	$TANG_0$ $TANG_1$ +0	00000 00010 00010	00000 00010 01101
74	1	1	0	2	102	TTOGO YAW PITCH	00100 00100 00000	01010 01010 01001
75	1	1	0	2	105	DIFFALT T1TOT2 T2TOT3	00000 00000 01000	01001 01001 00111
76	0	1	0	2	108	ZDOTD RDOTD XRANGE	01000 01010 01010	00111 00111 00111
77	1	1	0	1	111	TTOGO YDOT +0	00000 01010 00000	00000 00111 01001
78	1	1	0	2	114	DNRRANGE DNRRDOT TTOTIG	00000 01101 01100	01001 01111 01110
79	0	1	0	2	117	CURSOR SPIRAL POSCODE	00000 00010 00010	00011 00010 00010
80	0	0	0	1	120	DATAGOOD OMEGDISP +0	00000 00100 00000	00000 01010 00011
81	0	1	0	2	123	$DELVLVC_x$ $DELVLVC_y$ $DELVLVC_z$	01010 01010 01010	00111 00111 00111
82	0	1	0	2	126	$DELVLVC_x$ $DELVLVC_y$ $DELVLVC_z$	01010 01010 01010	00111 00111 00111
83	0	1	0	2	129	$DELVIMU_x$ $DELVIMU_y$ $DELVIMU_z$	01010 01010 01010	00111 00111 00111
84	0	1	0	2	132	$DELVOV_x$ $DELVOV_y$ $DELVOV_z$	01010 01010 01010	00111 00111 00111

Table 2 continued

Noun	K:NNADTAB		K:IDADTAB	K:NNTYPTAB	K:RUTMXTAB
85	0 1 0 2	135	$VGBODY_x$ $VGBODY_y$ $VGBODY_z$	01010 01010 01010	00111 00111 00111
86	0 1 0 2	138	DELVLVCx DELVLVCy DELVLVCz	01010 01010 01010	00111 00111 00111
87	0 0 0 1	141	AZ EL +0	00000 00010 00010	00000 00010 00010
88	0 1 0 2	144	$STARAD0_x$ $STARAD0_y$ $STARAD0_z$	00000 00000 00000	00001 00001 00001
89	0 1 0 2	147	LANDLAT LANDLONG LANDALT	00111 00011 00011	00100 00111 00111
90	0 1 0 2	150	RANGE RRATE RTHETA	00100 01010 00111	01010 00111 00100
91	0 0 0 2	153	P21ALT P21VEL P21GAM	00100 01001 01000	01010 01010 00111
92	0 0 0 2	156	THRDISP HDOTDISP HCALC1	11000 01010 00000	00100 00111 00011
93	0 0 0 2	159	OGC IGC MGC	00011 00011 00011	00111 00111 00111
94	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000
95	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000
96	0 0 0 0	0	+0 +0 +0	00000 00000 00000	00000 00000 00000
97	0 0 0 2	171	DSPTEM1 $DSPTEM1_1$ $DSPTEM1_2$	00000 00000 00000	00011 00011 00011
98	0 0 0 2	174	DSPTEM2 $DSPTEM2_1$ $DSPTEM2_2$	00000 00000 00000	00011 00001 00011
99	0 1 0 2	177	WWPOS WWVEL WWBIAS	11100 11011 11010	01010 00101 00111

Table 3

Input/Output Scaling

Each description in the table is arranged in the following order:

Equation value
Scale factor and units
Stored value
(comment)

Index	K:SFINTAB$_i$	K:SFOUTAB$_i$	
0	10^5 B28, unitless 00006_8 03240_8	10^{-5} B-14, unitless 05174_8 13261_8	
	(used with nouns 2,14,21,25,27,45,49,55,79,80,92,97,98)		
1	+0	+0	(not used)
2	0 B-1, revolutions 0	+0	
	(used with nouns 3,18,20,22,41,72,73,79,87)		
3	when used with noun 89		
	$(100 / 360) + 2^{-28}$ B0, revolutions per degree 10707_8 03435_8	$360 / 100$ B7, degrees per revolution 00714_8 31463_8	
3	when used with noun 93		
	$(100 / 360)\ 2^{-21} + 2^{-7}$ B-21, gyro torque pulses/deg 10707_8 03435_8	$2^{21}\ 360 / 100$ B28 degrees per gyro pulse 00714_8 31463_8	
4	$(1000 / 360) + 2^{-25}$ B3, revolutions per degree 13070_8 34345_8	$360 / 1000$ B0, degrees per revolution 13412_8 07534_8	
	(used with nouns 4,5,43,45,51,52,54,55,56,74,80,90,91)		
5	$1000 / 360$ B13, revolutions per degree 00005_8 21616_8	$360 / 1000$ B1, degrees per revolution 05605_8 03656_8	
	(not used)		

Index	K:SFINTAB$_i$	K:SFOUTAB$_i$
6	$10^5 \times 0.45359237$ B16, kilograms per pound 26113_8 31713_8	$2.2046268 / 10^5$ B-2, pounds per kilogram 00001_8 16170_8
	(used with noun 47)	
7	1852×10^3 B29, meters per nmi 00070_8 20460_8	$5.3996 \times 10^{-4} / 10^3$ B-15, nmi per meter 00441_8 34306_8
	(used with nouns 54,89,90)	
8	1852×10^4 B29, meters per nmi 01065_8 05740_8	$5.3996 \times 10^{-4} / 10^4$ B-22, nmi per meter 07176_8 21603_8
	(used with nouns 42, 43, 44, 49, 58, 75, 76, 91)	
9	$(0.3048 / 100) \times 10^5$ B10, meters/cs per fps 11414_8 31463_8	$(100 / 0.3048) \times 10^5$ B0 , fps per meter/cs 15340_8 15340_8 (equal halves)
	(used with noun 91)	
10	$(0.3048 / 100) \times 10^4$ B7, meters/cs per fps 07475_8 16051_8	$(100 / 0.3048)\ 10^{-4} + 2^{-21}$ B0, fps per meter/cs 01031_8 21032_8
	(used with nouns 40,42,49,54,58,59,60,62,63,64,76,77,81-86,90,92)	
11	$10^2 / 360$ B12, revolutions per degree 00001_8 03434_8	$360 / 10^2$ B2, degrees per revolution 34631_8 23146_8
	(used with noun 41)	
12	$1852\ 10^3 / 2.859026$ B28, RR range counts per nmi 00047_8 21135_8	$2.859024 / 10^3\ 1852$ B-14 nmi per RR range count 00636_8 14552_8
	(used with noun 78)	
13	-1.59286×10^5 B28, RR rate counts per fps 77766_8 50711_8	$-0.6278 / 10^5$ B-14, fps per RR rate count 74552_8 70307_8
	(used with noun 78)	

Index	$K{:}SFINTAB_i$	$K{:}SFOUTAB_i$
14	10^5 / 1.0790 B28, LR alt counts per foot 0.9267840599 E5 x 2^{-28} (used with noun 66)	1.0790 / 10^5 B-14, feet per LR alt count 1.079 E-5 x 2^{14}
15	10^5 / 2.345 B28, bits per foot 00002_8 23224_8 (not used)	2.345 / 10^5 B-14, feet per bit 14226_8 31757_8
16	10^5 / 0.5 B28, bits per fps 00014_8 06500_8 (not used)	0.5 / 10^5 B-14, fps per bit 02476_8 05531_8
17	0.18125 x E5 B28, bits per fps 0.18125 E5 x 2^{-28} (used with noun 60)	5.517 / 10^5 B-14, fps per bit 5.517 E-5 2^{14}
18	10^5 / 360 B11, rev per deg/sec 04256_8 07071_8 (not used)	360 / 10^5 B-3, deg/sec per rev 00007_8 13734_8
19	-1.55279503 x 10^5 B28, LRVX counts per fps -1.55279503 E5 x 2^{-28} (used with noun 67)	-0.6440 / 10^5 B-14, fps per LRVX count -0.6440 E-5 x 2^{14}
20	0.8250825087 x 10^5 B28, LRVY counts per fps 0.8250825087 E5 x 2^{-28} (used with noun 67)	1.212 / 10^5 B-14, fps per LRVY count 1.212 E-5 x 2^{14}
21	1.153668673 x 10^5 B28, LRVZ counts per fps 1.153668673 E5 x 2^{-28} (used with noun 67)	0.8668 / 10^5 B-14, fps per LRVZ count 0.8668 E-5 x 2^{14}

Index	K:SFINTAB$_i$	K:SFOUTAB$_i$
22	1852×10^4 B27, meters per nmi 04324_8 27600_8	$5.399568 \times 10^{-4} / 10^4$ B-24, nmi per meter 34772_8 07016_8
	(used with nouns 61,68)	
23	$10^3 / 0.002$ B28, centiseconds per deg 00036_8 20440_8	$0.002 / 10^3$ B-14, deg per centisecond 01030_8 33675_8
	(used with noun 48)	
24	0.3048×10^5 B24, meters per foot 00035_8 30400_8	$3.2808399 / 10^5$ B-10, feet per meter 01046_8 15700_8
	(used with nouns 60,63,64,68 ,69,92)	
25	10^4 B14, unitless 23420_8 00000_8	10^{-4} B-7, unitless 00321_8 26706_8
	(not used)	
26	30480 B19, meters per foot 30480×2^{-19}	17.2010499 B7, feet per meter 17.2010499×2^{-7}
	(used with noun 99)	
27	0.003048×10^4 B7, meters/cs per fps 30.48×2^{-7}	$328.08399 / 10^4$ B0, fps per meter 0.032808399
	(used with noun 99)	
28	100 B8, unitless 100×2^{-8}	$2^5 \times 10^{-2}$ B0, unitless 0.32
	(used with noun 99)	

Display Quantities

ABVEL: See SERV section.

ACTCENT: See TRGL section.

ALMCADR: See PGSR section.

ALPHASB: Same as PITCHANG, see EXVB section.

ALT: See COOR section.

AOTCODE: See ALIN section.

AZ: See ALIN section.

BETASB: Same as YAWANG, see EXVB section.

$\underline{C}$DU: See IMUC section.

CDU_s: See RADR section.

CDU_t: See RADR section.

CENTANG: See TRGL section.

CSMMASS: See DAPB section.

CURSOR: See ALIN section.

DAPDATR1: See DAPB section.

DATAGOOD: See RNAV section.

DELTAH: See SERV section.

$DELVIMU_{x,y,z}$: See BURN section.

$DELVLVC_{x,y,z}$: See TRGX section.

$DELVOV_{x,y,z}$: See ORBI section.

DELVTPF: See TRGL section.

DELVTPI: See TRGL section.

DIFFALT: See TRGX section.

DLANDX,DLANDY,DLANDZ: See DESC section.

DNRRANGE: See RADR section.

DNRRDOT: See RADR section.

DSPTEMX: See DATA section.

DSPTEM1: See DATA section.

DSPTEM2: See DATA section.

$DVLOS_{x,y,z}$: See TRGL section.

DVTOTAL: See SERV section.

EL: See ALIN section.

ELEV: See TRGL section.

ERCOUNT: See TEST section.

FAILREG: See PGSR section.

FDAI: See ATTM section.

FORVEL: See SERV section.

FUNNYDSP: See DESC section.

HAPO: See TRGX section.

HAPOX: See EXVB section.

HCALC: See SERV section.

HCALC1: See SERV section.

HDOTDISP: See SERV section.

HPER: See TRGX section.

HPERX: See EXVB section.

IGC: See COOR section.

LANDALT: See ALIN section.

LANDLAT: See ALIN section.

LANDLONG: See ALIN section.

LAT: See COOR section.

LEMMASS: See DAPB section.

LONG: See COOR section.

MGC: See COOR section.

mTPER: See EXVB section.

NN: See TRGX section.

OGC: See COOR section.

OMEGDISP: See RNAV section.

OPTIONX: See EXVB section.

OPTION1: See DATA section.
OPTION2: See DATA section.
OPTION3: See DATA section.
OUTOFPLN: See DESC section.

PIPA: See IMUC section.

PITCH: See ASCT section.

PITTIME: See DAPB section.

pMGA: See TRGX section.

POSCODE: See ALIN section.

POSTTPI: See TRGL section.

P21ALT: See RNAV section.

P21GAM: See RNAV section.

P21VEL: See RNAV section.

RANGE: See EXVB section.

RANGEDSP: See DESC section.

RDOTD: See ASCT section.

ROLLTIME: See DAPB section.

RRATE: See EXVB section.

RR-AZ: See RNAV section.

RR-ELEV: See RNAV section.

RSTACK: See RADR section.

RTHETA: See EXVB section.

R22DISP: See R22DISPR in RNAV section.

R22DISP+2: See R22DISPV in RNAV section.

SAMPTIME: See DSKY section.

SMODE: See TEST section.

SPIRAL: See ALIN section.

$STARAD_{x,y,z}$: See ALIN section.

TANG: See RADR section.

TCDH: See TRGX section.

TCSI: See TRGX section.

TET: See ORBI section.

TFF: See EXVB section.

THETAD: See IMUC section.

THRDISP: See DESC section.

TIG: See BURN section.

TIMENOW: See EXVB section.

TRKMKCNT: See RNAV section.

TTFDISP: See DESC section.

TTOGO: See BURN section.

TTOTIG: See RADR section.

TTPI: See TRGL section.

T1TOT2: See TRGX section.

T2TOT3: See TRGX section.

$VGBODY_{x,y,z}$: See BURN section.

VGDISP: Same as DELVSAB, see TRGX section.

WHCHREAD: See RNAV section.

WWBIAS: See RNAV section.

WWPOS: See RNAV section.

WWVEL: See RNAV section.

XRANGE: See ASCT section.

XREG: See DATA section.

YAW: See ASCT section.

YDOT: See ASCT section.

YREG: See DATA section.

ZDOTD: See ASCT section.

ZREG: See DATA section.

-K:posmaxsp: See "Major Variables" section.

DESC

Descent Guidance

P63LM Perform "R02BOTH" (assure that IMU is operating)

WHICH = "P63TABLE"

DVTHRUSH = K:DPSTHRSH

DVCNTR = 4

WCHPHASE = - 1

FLPASS0 = 0

Switch bit 14 of channel 12 to 0 (disable RR tracker)

Switch FLAGWRD5 bit 12 (NOTHROTL) to 0

Switch FLAGWRD6 bit 6 (REDFLAG) to 0

Switch FLGWRD11 bit 15 (LRBYPASS) to 0

Switch FLAGWRD6 bit 8 (MUNFLAG) to 1

Switch FLAGWRD0 bit 9 (P25FLAG) to 0

Switch FLAGWRD0 bit 7 (RNDVZFLG) to 0

TPIP = TLAND

TSt = TLAND

Perform "MOONMX"

$\underline{LAND} = \left[REFSMMAT\right] \left[MOONMAT\right]^{T} (\underline{RLS} + \underline{LM504} * \underline{RLS})$

TSt = TIMENOW

Perform "MOONMX"

$\underline{WM} = K{:}MOONRATE \left[REFSMMAT\right] \left[MOONMAT\right]^{T} (\underline{K}{:}UNITZ + \underline{LM504} * \underline{K}{:}UNITZ)$

$LANDMAG = \left|\underline{RLS}\right|$

TDEC1 = TLAND - K:GUIDDURN

Perform "LEMPREC"

NIGNLOOP = 40

$$[\text{GCMAT}] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

DELTAH = K:99999CON

UNFC = 0

TTF = 0

IGNALOOP PIPTIME1 = TAT

R = [REFSMMAT] RATT

Perform "MUNGRAV" with TSr = R

GDT = GDT1

NGUIDSUB = 2

Proceed to the second step of "GUILDRET" (continues at "EXGSUB" after one iteration of guidance computations)

EXGSUB UNFC = K:TRIMACCL ZOOMTIME unitUNFC (argument of unit operation adjusted to reduce the possibility of overflow)

If NGUIDSUB > 0:

NGUIDSUB = NGUIDSUB - 1

Proceed to "CALCRGVG"

If NIGNLOOP = 0:

Perform "ALARM" with TS = 01412_8

TS = 3313_8 (S-register portion of address of cell containing alarm pattern)

If NIGNLOOP > 0:

TS = NIGNLOOP - 1

NIGNLOOP = TS

TSden = VGU_z - DESKIGNX VGU_x

TSnum = (DESIGNRZ - RGU_z) + DESKIGNY $\text{RGU}_y{}^2$ + DESKIGNX (RGU_x - DESIGNRX)

TSv = DESKIGNV (|VGU| - DESIGNV)

TSt = (TSv + TSnum) / TSden

TDEC1 = PIPTIME1 + TSt

If $|TSt| \geq$ K:DDUMCRIT: (reiterate)

Perform "INTSTALL"

Switch FLAGWRD3 bit 4 (INTYPFLG) to 1

Switch FLAGWRDO bit 12 (MOONFLAG) to 1

TET = PIPTIME1

$\underline{R}$CV = $\underline{R}$ATT

$\underline{V}$CV = $\underline{V}$ATT

Perform "INTEGRVS"

Proceed to "IGNALOOP"

TIG = TDEC1 - ZOOMTIME

OUTOFPLN = unit($\underline{V}$ * $\underline{R}$) · $\underline{L}$AND

$\underline{R}$60SAVE = $\underline{U}$NFC

DISPDEX = - 21 (enable astronaut branch to "ASTNRET")

Perform "STCLOK3"

End job

ASTNRET Proceed to "GOPERF1" with TS = 00014_8
(If terminate, proceed to "GOTO POOH"; if proceed, continue at next step; if other response, skip next step.)

Perform "R51"

$\underline{P}$OINTVSM = unit$\underline{R}$60SAVE

$\underline{S}$CAXIS = $\underline{K}$:UNITX

Perform "PFLITEDB" with interrupts inhibited

Perform "R60LEM"

If bit 6 of channel 33 = 1: (LR not in position #1)

Proceed to "GOPERF1" with TS = 00500_8
(If terminate, proceed to "GOTO POOH"; if proceed, continue at previous step; if other response, continue at next step.)

Perform "SETPOS1" (Initialize landing radar control)

Proceed to "BURNBABY"
(Standard pre-ignition sequence; initializes average-g navigation at TIG-30 seconds; calls "P63IGN" at time of ignition which sets AVEGEXIT to "LUNLAND" establishing the two-second guidance loop; calls "P63ZOOM" at throttle-up time.)

LUNLAND If FLAGWRD5 bit 8 (ZOOMFLAG) = 0: (R13)

Proceed to "DISPEXIT" (do display only; no throttle-up yet)

If MODREG = 66: (in P66)

If FLAGWRD1 bit 12 (RODFLAG) = 0:

Proceed to "STRTP66A"

Proceed to "P66"

If bit 13 of channel 31 = 0 and RODCOUNT ≠ 0:

Proceed to "STARTP66"

GUILDRET RODCOUNT = 0

TPIPOLD = TPIP

TPIP = PIPTIME1

TTFTMP = TTF

If FLPASS0 > 0, proceed to "TTFINCR"

Proceed to $K:NEWPHASE_{WCHPHASE}$
(TTFINCR, TTFINCR, STARTP64, STARTP66)

STARTP64 MODREG = 64

Establish "DSPMMJOB" (pr30)

TTFTMP = TTFTMP + DELTTFAP

Inhibit interrupts

Perform "C13STALL"

Switch bit 12 of channel 13 to 1 (enable RHC interrupt #10)

DB = K:P64DB

Switch FLAGWRD6 bit 6 (REDFLAG) to 0

Release interrupt inhibit

Proceed to "TTFINCR"

P65START MODREG = 65 (this coding not exercised)

Establish "DSPMMJOB" (pr30)

WCHVERT = 0

Switch DAPBOOLS bit 9 (XOVINHIB) to 0 (permit X-axis override)

TTFINCR TSt = TPIP - TPIPOLD (rescaled to B17 centiseconds)

$\underline{LANDTEMP}$ = LANDMAG unit($\underline{LAND}$ - TSt $\underline{LAND}$ * $\underline{WM}$) (argument of unit operation adjusted to reduce the possibility of overflow)

TTFTMP = TTFTMP + TSt

TTF = TTFTMP

Perform "TDISPSET"

Change job priority to 31 (pr31)

$\underline{LAND}$ = $\underline{LANDTEMP}$ + $\underline{DLAND}$

LANDMAG = $|\underline{LAND}|$

$\underline{DLAND}$ = 0

Change job priority to 20 (pr20)

Proceed to $K:PREGUIDE_{WCHPHASE}$
(CALCRGVG, RGVGCALC, REDESIG, RGVGCALC)

REDESIG If FLAGWRD6 bit 6 (REDFLAG) = 0 or if TREDES = 0:

Proceed to "RGVGCALC"

Inhibit interrupts

$ELINCR_{dp}$ = (ELINCR1, 0)

$AZINCR_{dp}$ = (AZINCR1, 0)

ELINCR1 = 0

AZINCR1 = 0

(AZINCR1 and ELINCR1 are updated in routines "PITFALL" and "REDESMON" which are called by program interrupt #10)

Release interrupt inhibit

$\underline{TS}$ = unit($\underline{LAND}$ - $\underline{R}$) (argument of unit operation adjusted to reduce the possibility of overflow)

$\underline{TS}$ = $\underline{TS}$ + AZINCR $\underline{YNBPIP}$ - ELINCR $\underline{TS}$ * $\underline{YNBPIP}$

If $TS_x \geq$ K:DEPRCRIT, TS_x = K:DEPRCRIT

$\underline{LANDTEMP}$ = LANDMAG unit($\underline{R}$ + $\underline{TS}$ ($LAND_x$ - R_x) / TS_x)

$\underline{LAND}$ = $\underline{LANDTEMP}$

Proceed to "RGVGCALC"

CALCRGVG $\underline{V}$ = [REFSMMAT] $\underline{VATT}$ + $\underline{UNFC}$ (VATT used here is VATT1 scaled B5; see VATT of the ORBI section.)

RGVGCALC $\underline{ANGTERM}$ = $\underline{R}$ * $\underline{WM}$ + $\underline{V}$

$\underline{VGU}$ = [GCMAT] $\underline{ANGTERM}$

$\underline{TS}$ = $\underline{R}$ - $\underline{LAND}$

$\underline{RGU}$ = [GCMAT] $\underline{TS}$

RANGEDSP = |$\underline{RGU}$|

LOOKANGL = K:180degs ($arcsin_{sp}$(unit*$\underline{TS}$ · $\underline{XNBPIP}$) + K:1d2DEG + ELBIAS)

Proceed to $K{:}WHATGUID_{WCHPHASE}$ (TTF/8CL, TTF/8CL, TTF/8CL, CGCALC)

*(argument of unit operation adjusted to reduce the possibility of overflow)

TTF/8CL LUNDEX = $K{:}TARGTDEX_{WCHPHASE}$ (0, 0, 28)

A_3 = $TTFJDGZ_{LUNDEX}$ (j_{DZG})

A_2 = $TTFADGZ_{LUNDEX}$ (6 a_{DZG})

A_1 = K:ttf6b3 VGU_z + $TTFVDGZ_{LUNDEX}$ (6 VGU_z + 18 v_{DZG})

$\underline{TS}$ = $\underline{TARGRDG}_{LUNDEX}$

A_0 = K:ttf24b6 (TS_z - RGU_z)

PREC = 2^{-7}

ROOTPS = TTF

n = 3

Perform "ROOTPSRS"

If ROOTGOOD = 0, proceed to $K{:}WHATALM_{WCHPHASE}$
(1406P00, 1406ALM, 1406ALM)

TTF = ROOTPS

Perform "TDISPSET"

QUADGUID $TSt = - TTF_{ms} + LEADTIME$

If $TSt < 0$, $TSt = 0$

$RA = TSt / TTF_{ms}$ (- r)

$TS1 = 2\ RA^2 + RA$ $(2\ r^2 - r)$

$TS2 = 3\ RA^2 + 2\ RA$ $(3\ r^2 - 2\ r)$

$TS3 = 4\ RA^2 + 3\ RA$ $(4\ r^2 - 3\ r)$

$TS4 = 6\ (RA^2 + RA) + K{:}posmaxsp$ $(6\ r^2 - 6\ r + 1)$

$LUNDEX = K{:}TARGTDEX_{WCHPHASE}$ (0, 0, 28)

$\underline{T}Sb = TS1\ \underline{V}GU$

$\underline{T}Sc = TS3\ \underline{T}ARGVDG_{LUNDEX}$

$\underline{T}Sd = 2\ TS2\ (\underline{T}ARGRDG_{LUNDEX} - \underline{R}GU) / TTF$

$\underline{T}Se = K{:}ttf6b3\ (\underline{T}Sb + \underline{T}Sc + \underline{T}Sd) / TTF$

$\underline{T}Sa = TS4\ \underline{T}ARGADG_{LUNDEX} + \underline{T}Se$ (desired acceleration)

AFCCALC1 $\underline{U}NFC = [GCMAT]^T\ \underline{T}Sa - \underline{G}DT / K{:}GSCALE$

$AFCMAG = |\underline{U}NFC|$

$TS = (K{:}HIGHESTF / MASS)^2 - UNFC_y^{\ 2} - UNFC_x^{\ 2}$

If $TS < 0$, $TS = 0$

If $UNFC_z < -\sqrt{TS}$, $UNFC_z = -\sqrt{TS}$

WCHPHOLD = WCHPHASE

FLPASSO = FLPASSO + 1

Proceed to $K{:}AFTRGUID_{WCHPHASE}$
(CGCALC, EXTLOGIC, EXTLOGIC, STEER?)

EXTLOGIC $TSt = TEND_{WCHPHASE} + TTF_{ms}$

If $TSt > 0$:

WCHPHASE = WCHPHOLD + 1

FLPASSO = 0

CGCALC $i = K{:}TARGTDEX_{WCHPHASE}$ (0,0,28)

If $TTF_{ms} < - TCGI_i$ or $TTF_{ms} > - TCGF_i$:

Proceed to $K{:}WHATEXIT_{WCHPHOLD}$
(EXGSUB, EXBRAK, EXNORM, -----)

$LUNDEX = K{:}TARGTDEX_{WCHPHASE}$ (0, 0, 28)

$\underline{T}Sa = \text{unit}\underline{L}AND$

$\underline{T}Sb = \text{unit}^*(\text{unit}^*(GAIN_{LUNDEX}\ TTF\ \underline{A}NGTERM + 4\ \underline{L}AND - 4\ \underline{R}) * \underline{L}AND)$

$\underline{T}Sc = \underline{T}Sa * \underline{T}Sb$

$$[GCMAT] = \begin{bmatrix} TSa_x & TSa_y & TSa_z \\ TSb_x & TSb_y & TSb_z \\ TSc_x & TSc_y & TSc_z \end{bmatrix}$$

Proceed to $K{:}WHATEXIT_{WCHPHOLD}$
(EXGSUB, EXBRAK, EXNORM, ----)

*(argument of unit operation adjusted to reduce the possibility of overflow)

EXBRAK $\underline{U}NWC = \underline{U}NITR$

Proceed to "STEER?"

EXNORM $\underline{U}NWC = \text{unit}^*(\underline{L}AND - \underline{R})$

$\underline{T}S$ = second row of [GCMAT] $(\underline{Y}DGC_{sm})$

$PROJ = \underline{U}NWC * \underline{X}NBPIP \cdot \underline{T}S$

PROJ1 = K:PROJMAX - PROJ

If $PROJ1 \leq 0$, PROJ1 = 0

PROJ2 = PROJ - K:PROJMIN

If PROJ2 $\leq$ 0, PROJ2 = 0

$UNWC_z$ = PROJ1 $GCMAT_{33}$ + PROJ2 $UNWC_z$

$UNWC_y$ = PROJ1 $GCMAT_{32}$ + PROJ2 $UNWC_y$

$UNWC_x$ = PROJ1 $GCMAT_{31}$ + PROJ2 $UNWC_x$

OGABIAS = AZBIAS

STEER? If overflow occurred anywhere above: (interpretive language overflow)

Perform "ALARM" with TS = 01410_8

Perform "STOPRATE"

Proceed to "DISPEXIT"

If FLAGWRD2 bit 11 (STEERSW) = 0:

Perform "STOPRATE"

Proceed to "DISPEXIT"

Perform "THROTTLE"

Perform "FINDCDUW"

DISPEXIT If FLAGWRD8 bit 10 (FLUNDISP) = 1, end job

Proceed to $K{:}WHATDISP_{WCHPHOLD}$
(---, P63DISPS, P64DISPS, VERTDISP)

P63DISPS Proceed to "REGODSP" with TS = K:V06N63 (ABVEL, HDOTDISP, HCALC1)

P64DISPS If TREDES = 0:

Switch FLAGWRD6 bit 6 (REDFLAG) to 0

Proceed to "REGODSP" with TS = K:V06N64

If FLAGWRD6 bit 6 (REDFLAG) = 1:

Proceed to "REGODSP" with TS = K:V06N64

Proceed to "REFLASH" with TS = K:V06N64 (FUNNYDSP, HDOTDISP, HCALC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, proceed to "P64DISPS".)

ELINCR1 = 0

AZINCR1 = 0

Switch FLAGWRD6 bit 6 (REDFLAG) to 1

End job

VERTDISP Proceed to "REFLASH" with TS = K:V06N60 (FORVEL, HDOTDISP, HCALC1)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, continue at next step.)

WCHVERT = K:posmaxsp

End job

TDISPSET TTFDISP = K:TSCALINV TTF_{ms}

TSt = K:TREDESCL ($TCGF_{28}$ + TTF_{ms}) - 103

If TSt $\geq$ 0:

TREDES = 99

Return

TSt = TSt + 99

If TSt $\leq$ 0:

TREDES = 0

Return

TREDES = TSt

Return

STARTP66 MODREG = 66

Establish "DSPMMJOB" (pr30)

VDGVERT = HDOTDISP

WCHVERT = - 0

CNTTHROT = - TOOFEW

VHZC = WM * R

STRTP66A TS = (PIPABIAS$_x$, PIPABIAS$_y$, PIPABIAS$_z$)

VBIAS = K:BIASFACT TS

Switch FLAGWRD1 bit 12 (RODFLAG) to 1

OLDPIPA = - TEM

DELVROD = 0

RODSCAL1 = RODSCALE

LASTTPIP = PIPTIME

FCOLD$_{sp}$ = 0

FWEIGHT$_{dp}$ = 0

WCHPHOLD = 2

WCHPHASE = 2

Perform "STOPRATE"

Switch DAPBOOLS bit 9 (XOVINHIB) to 0

Switch FLAGWRD6 bit 6 (REDFLAG) to 0

P66 TSt = TIMENOW - PIPTIME

If 2LATE466 $\geq$ TSt:

Proceed to "P66HZ"

Switch FLAGWRD1 bit 12 (RODFLAG) to 0

TS = CNTTHROT

CNTTHROT = - TOOFEW

If $TS \leq 0$:

Perform "ALARM" with TS = 01466_8

Proceed to "DISPEXIT"

P66HZ $\underline{TS} = (\underline{V}HZC - \underline{V})/TAUHZ - QHZ\ \underline{UNFC}$

Change job priority to 21_8 (pr21)

$TS_{x_{sp}} = K{:}GHZ$

If $|TS_{y_{sp}}| > AHZLIM$, $TS_{y_{sp}} = AHZLIM\ signTS_{y_{sp}}$

If $|TS_{z_{sp}}| > AHZLIM$, $TS_{z_{sp}} = AHZLIM\ signTS_{z_{sp}}$

If overflow anywhere above: (interpretive language overflow)

Perform "ALARM" with TS = 01410_8

Perform "STOPRATE"

Proceed to "P66VERT"

If bit 3 of channel 30 = 1: (Engine not armed)

If WCHVERT $>$ 0:

T5ADR = "DAPIDLER"

Proceed to "P66VERT"

If WCHVERT $<$ - 0:

Proceed to "P66VERT"

If WCHVERT = - 0:

Continue at next step

$\underline{UNFC} = \underline{TS}$

Perform "FINDCDUW"

Proceed to "P66VERT"

P66VERT Call "RODTASK" in 1.0 second

Proceed to "RODCOMP" (note that "RODCOMP" will be performed now and again in 1.0 second)

RODTASK Establish "RODCOMP" (pr22)

End task

RODCOMP Change job priority to 23_8 (pr23)

Inhibit interrupts

VDGVERT = VDGVERT + RODCOUNT RODSCAL1 (activation of the R.O.D. switch causes routine "DESCBITS" to be entered which updates RODCOUNT)

RODCOUNT = 0

POLDPIPA = OLDPIPA

OLDPIPA = PIPA

THISTPIP = TIMENOW

TS_{sp} = OLDPIPA + PIPATMP

DELVROD = TEM - OLDPIPA + POLDPIPA

TS_{dp} = TS_{sp} (least significant components set to 0)

TEM = 0

Release interrupt inhibit

TS = K:KPIP1 TS_{dp}

TSdelt = THISTPIP - PIPTIME

TSv = (TSdelt / K:4SECb28) ($\frac{1}{2}$ GDT - $\frac{1}{2}$ VBIAS) + V + TS (updated velocity stored in PDL24-29 at B7 m/cs)

HDOTDISP = TSv · unitR

HCALC1 = TSdelt HDOTDISP + |R| - LANDMAG

TS1 = (VDGVERT - HDOTDISP) / TAUROD (PDL0-1; B-2 m/cs^2)

TS2 = |GDT| / K:GSCALE (PDL20-21; B-2)

TS3 = TS2 + TS1 (PDL0-1; B-2)

Perform "CDUTRIG"

Perform "NBTOSM"

TS = [NBSMMAT] K:UNITX

TS4 = TS · unitR (PDL22-23; B2)

AFCMAG = TS3 / TS4

TS1 = |K:KPIP1 DELVROD + $\frac{1}{2}$ VBIAS| (PDL0-1; B7)

TSt3 = THISTPIP - LASTTPIP (PDL2-3; B28)

LASTTPIP = THISTPIP

TSacc = TS1 / (TSt3 / K:SHFTFACT) (measured acceleration in PDL0-1 at B-4 m/cs^2)

TS5 = (FWEIGHT K:BIT1H) / (MASS K:SCALEFAC) + TSacc (PDL2-3; B-4)

AFCMAG = AFCMAG + LAGdTAU ((TS2 / TS4) - TS5) (PDL2-3; B-4)

TSafcmax = MAXFORCE / MASS (PDL4-5; B-4)

TSafcmin = MINFORCE / MASS (PDL6-7; B-4)

If AFCMAG < TSafcmin , AFCMAG = TSafcmin

If AFCMAG ≥ TSafcmax , AFCMAG = TSafcmax

TSthrot = TSacc

Perform "THROTTLE" (starting at second step; return will be to the next step.)

CNTTHROT = CNTTHROT + 1

Proceed to "DISPEXIT"

THRTOTTLE

<u>THROTTLE</u> TSthrot = K:ABAFCNST ABDELV

RTNHOLD = return address

FP_{dp} = K:SCALEFAC MASS TSthrot

If $FP_{dp} \geq$ K:fmax, FP_{sp} = K:posmaxsp

$FCODD_{dp}$ = K:SCALEFAC MASS AFCMAG

If $FCODD_{dp} \geq$ K:fmax, $FCODD_{sp}$ = K:posmaxsp

FC = FCODD

TS = FC

If TS $\geq 2^{13}$ throttle pulses, truncate bits $\geq 2^{13}$

THRDISP = (TS / K:4FMAXNOM) 400

TSt = (less significant half of TIMENOW) - TTHROT

If TSt $\leq$ 0, TSt = 16384 + TSt

If TSt $<$ K:3SECS:

    FP_{dp} = FP + FWEIGHT

PIFPSET = 0   (-0)

If FCOLD $>$ HIGHCRIT:

    If $FCODD_{sp} \leq$ LOWCRIT:

        PIFPSET = FP_{sp} - K:FMAXODD

    If $FCODD_{sp} >$ LOWCRIT:

        FCODD = FP

        PIFPSET = K:FEXTRA

If FCOLD $\leq$ HIGHCRIT:

    If $FCODD_{sp} >$ HIGHCRIT:

        FCODD = K:FMAXPOS

        PIFPSET = K:FEXTRA

FCOLD = FCODD

PIF = FCODD - FP

Proceed to "DOIT"

FLATOUT PIFPSET = K:FEXTRA

FCOLD = 0 (-0)

PIF = 0 (-0)

RTNHOLD = return address (to caller of "FLATOUT")

DOIT PSEUDO55 = PIF_{sp} + PIFPSET

THRUST = PSEUDO55

Switch bit 4 of channel 14 to 1 (send throttle command from THRUST)

TTHROT = less significant half of TIMENOW

TS1 = $THISTPIP_{ls}$

TS2 = K:2SECS

If MODREG $\neq$ 66:

  TS1 = $PIPTIME_{ls}$

  TS2 = K:4SECS

TS3 = TS2 K:BIT6

TS4 = $TS3_{ls}$

TSt = K:THROTLAG + $TIMENOW_{ls}$ - TS1

If TSt $<$ 0, TSt = $|TSt|$

If TSt $\geq 2^8$ centiseconds, truncate bits $\geq 2^8$ (i.e. subtract 2^8 until TSt is less than 2^8 centiseconds)

FWEIGHT = 2 PIF TSt / TS2

FWEIGHT = FWEIGHT + $|PIF|$ PIF / TS4

Return via RTNHOLD

PITFALL (Entered on program interrupt #10)

If MODREG ≠ 64, resume

ELVIRA = bits 6, 5, 2 and 1 of -channel 31 (bits on channel 31 logically inverted)

ZERLINA = 2

Call "REDESMON" in 0.05 second

Resume

REDESMON TS = ELVIRA

ELVIRA = bits 6, 5, 2 and 1 of -channel 31

If ELVIRA ≠ 00000_8: (LPD still out of detent)

    Delay 0.07 second

    Proceed to "REDESMON"

If TS = 00000_8:

    If ZERLINA > 0:

        ZERLINA = ZERLINA - 1

        Delay 0.07 second

        Proceed to "REDESMON"

    Perform "C13STALL"

    Switch bit 12 of channel 13 to 1 (re-enable RHC interrupt #10)

    End task

If bit 13 of channel 31 = 0:

    Perform "C13STALL"

    Switch bit 12 of channel 13 to 1 and end task

If bit 6 of TS = 1, AZINCR1 = AZINCR1 - K:AZEACH (-AZ LPD)

If bit 5 of TS = 1, AZINCR1 = AZINCR1 + K:AZEACH (+AZ LPD)

If bit 1 of TS = 1, ELINCR1 = ELINCR1 - K:ELEACH (+EL LPD)

If bit 2 of TS = 1, ELINCR1 = ELINCR1 + K:ELEACH (-EL LPD)

Perform "C13STALL"

Switch bit 12 of channel 13 to 1 (re-enable RHC interrupt #10)

End task

<u>DESCBITS</u> (Entered from "SOMEKEY" with contents of channel 16 in TS)

If bit 7 of TS = 1, RODCOUNT = RODCOUNT - 1

If bit 7 of TS = 0, RODCOUNT = RODCOUNT + 1 (assume bit 6 = 1)

Resume

<u>ROOTPSRS</u> DXCRIT = $|$PREC ROOTPS$|$

p = n - 1

<u>DERCLOOP</u> $DA_p = (p + 1)\ A_{p+1}$

If p > 0:

p = p - 1

Proceed to "DERCLOOP"

<u>ROOTLOOP</u> $TSderiv = DA_0 + DA_1\ ROOTPS + \ldots + DA_{n-1}\ ROOTPS^{n-1}$

$TSfunct = A_0 + A_1\ ROOTPS + \ldots + A_{n-1}\ ROOTPS^{n-1} + A_n\ ROOTPS^n$

TSdelt = - TSfunct / TSderiv

ROOTPS = ROOTPS + TSdelt

If p = 8:

ROOTGOOD = 0

Return

p = p + 1

If $|$TSdelt$|$ > DXCRIT, proceed to "ROOTLOOP"

ROOTGOOD = 2

Return

<u>1406POO</u> Proceed to "POODOO" with TS = 21406_8

<u>1406ALM</u> Perform "ALARM" with TS = 01406_8

Perform "STOPRATE"

Proceed to "DISPEXIT"

LANDJUNK Inhibit interrupts (program 68)

Perform "ZATTEROR"

Release interrupt inhibit

Switch DAPBOOLS bit 15 (PULSES) to 1

Switch FLAGWRD8 bit 8 (SURFFLAG) to 1

Switch FLAGWRD9 bit 9 (LETABORT) to 0

Switch FLGWRD10 bit 13 (APSFLAG) to 1

$\underline{A}$LPHAV = $\underline{R}$N

TSt = PIPTIME

Switch FLAGWRD3 bit 12 (LUNAFLAG) to 1

Perform "LAT-LONG"

TSt = PIPTIME

Perform "MOONMX"

$\underline{R}$LS = [MOONMAT] ($\underline{R}$N - ([MOONMAT]^T $\underline{L}$M504) * $\underline{R}$N)

Proceed to "GOFLASH" with TS = K:V06N43 (LAT, LONG, ALT)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; if other response, repeat this step.)

$\underline{G}$SAV = $\underline{K}$:UNITX

Perform "REFMF"

Proceed to "GOTOPOOH"

2LATE466: Double precision erasable memory constant representing the time delay after the accelerometer readings beyond which further P66 computations are bypassed, scaled B28 in units of centiseconds; part of the erasable load.

A_i (i = 0,1,2,...): Double precision coefficients of the polynomial input to "ROOTPSRS", unitless and scaled B30, B13, B-4 and B-21 when generated in "TTF/8CL".

ABDELV, ABVEL: see SERV section.

AFCMAG: Double precision magnitude of desired thrust acceleration, program notation /AFC/, scaled B-4 in units of meters per centisecond squared.

AHZLIM: Single precision erasable memory constant representing the maximum commanded pitch and roll attitude for horizontal nulling in P66 Auto, scaled B-4 in units of meters per centisecond squared; part of the erasable load.

$\underline{A}$LPHAV: see COOR section.

ALT: see SERV section.

$\underline{A}$NGTERM: Double precision velocity of the LM vehicle relative to the rotating moon, scaled B9 in units of meters per centisecond and expressed in the Platform coordinate system.

AZBIAS: Single precision quantity representing the desired outer gimbal angle bias for window pointing commands in P64 to account for window bending due to cabin pressurization, scaled B-1 in units of revolutions; part of the erasable load

AZINCR, AZINCR1: Double precision and single precision storage for the desired addition to the landing site azimuth, scaled BO in units of radians.

CNTTHROT: Single precision counter of the number of times that the "THROTTLE" routine is entered in P66; initialized to -TOOFEW, scaled B14 and unitless.

DA_i (i = 0,1...): Double precision coefficients of the polynomial derivative of the polynomial input to "ROOTPSRS"; unitless and scaled B13, B-4, and B-21 when generated in response to the polynomial input from "TTF/8CL".

DAPBOOLS: see DAPA section.

DB: see DAPB section.

DELTAH: see SERV section.

DELTTFAP: Single precision time constant added to TTF at the start of P64, scaled B17 in units of centiseconds. DELTTFAP is a negative number and is part of the erasable load.

DELVROD: Double precision sensed-change-in-velocity vector for P66 computations, scaled B14 in units of centimeters per second.

DESIGNRX, DESIGNRZ: Double precision components of desired position relative to the landing site (desired crossrange position component is zero), scaled B24 in units of meters and expressed in the Descent Guidance coordinate system; program notations RIGNX and RIGNZ respectively; part of the erasable load.

DESIGNV: Double precision speed desired at ignition, relative to the rotating moon, scaled B10 in units of meters per centisecond; program notation VIGN; part of the erasable load.

DESKIGNV: Double precision speed error scale factor use in the ignition-time test quantity, scaled B18 in units of centiseconds; program notation KIGNV/B4; part of the erasable load.

DESKIGNX: Double precision landing site vertical error scale factor used in the ignition-time test quantity, scaled B4 and unitless; program notation KIGNX/B4; part of the erasable load.

DESKIGNY: Double precision crossrange error scale factor used in the ignition-time test quantity, scaled B-16 in units of meters to the minus one power; program notation KIGNY/B8; part of the erasable load.

DISPDEX: see BURN section.

DLAND: Double precision vector expressed in the Platform coordinate system representing the correction to the Landing site vector LAND, scaled B24 in units of meters. DLAND is padloaded to zero and may be loaded by the crew in Noun 69 in the order $DLAND_z$, $DLAND_y$, $DLAND_x$.

DVCNTR, DVTHRUSH: see SERV section.

DXCRIT: Double precision criterion for the convergence of the iterative calculation in "ROOTPSRS", with scaling and units identical to those of ROOTPS.

ELBIAS: Single precision quantity representing the LPD elevation angle bias used in calculating LOOKANGL to account for window bending due to cabin pressurization, scaled B-1 in units of revolutions; part of the erasable load.

ELINCR, ELINCR1: Double precision and single precision storage for the complement of the desired addition to landing site elevation, scaled B0 in units of radians. (Sign changed to compensate for the inversion of the cross product in "REDESIG".)

ELVIRA: Single precision storage for the status of the landing site redesignation discretes from channel 31.

FC: Single precision storage for the magnitude of desired thrust, scaled B14 in units of DPS throttle pulses.

FCODD: Double precision magnitude of desired thrust, scaled B14 in units of DPS throttle pulses. (The less significant half is not always maintained.)

FCOLD: Single precision magnitude of previous value of desired thrust, scaled B14 in units of DPS throttle pulses.

FLPASS0: Single precision flag set to zero at the beginning of a new guidance phase (except at the beginning of P66) to initialize guidance quantities for the new guidance phase.

FORVEL: see SERV section.

FP: Double precision estimate of the magnitude of the present thrust, scaled B14 in units of DPS throttle pulses. (The less significant half is not always maintained.)

FUNNYDSP: Special display of LOOKANGL and TREDES in the same display register, both displayed in two digits only.

FWEIGHT: Double precision change in sensed thrust expected to have occurred since the sampling of the accelerometers, scaled B14 in units of DPS throttle pulses.

$GAIN_0$: Double precision gain constant used in the computation of the orientation of the Descent Guidance Coordinate System for the braking phase, scaled B0 and unitless. Program notation: GAINBRAK; part of the erasable load.

$GAIN_{28}$: Double precision gain constant used in the computation of the orientation of the Descent Guidance Coordinate System for the approach phase, scaled B0 and unitless. Program notation: GAINAPPR; part of the erasable load.

[GCMAT] : Double precision, 3X3 transformation matrix defined such that Adgc = [GCMAT] Asmc, where A is a vector expressed in the Descent Guidance and Platform (sm) coordinate systems respectively; scaled B1 and unitless; program notation CG+0 through CG+17.

The Descent Guidance coordinate system is an orthogonal, cartesian system where the X axis is along the radius from the center of the moon through the present landing site, the Y axis is defined such that the velocity, acceleration and jerk vectors at the landing site lie entirely in the X-Z plane, and the Z axis is defined such as to complete the right handed system.

GDT, GDT1: See SERV section.

GSAV: See ALIN section

HCALC, HDOTDISP: See SERV section.

HCALC1: Double precision calculated altitude above the landing site radius for display in Nouns 60,63 ~~and~~ 92, scaled B24 in units of meters. HCALC1 is set to HCALC in the SERV section and is calculated once per second in "RODCOMP"..

HIGHCRIT: Single precision upper limit on the variable throttle region in a situation of increasing thrust commands, scaled B14 in units of DPS throttle pulses. If the throttle setting is in the variable region, the throttle setting commanded by the program will correspond directly with the desired thrust until the desired thrust exceeds HIGHCRIT. Then the program will command full throttle. HIGHCRIT is part of the erasable load.

K:180DEGS: Single precision constant stored as 180×2^{-14}, scaled B15 in units of degrees per revolution. Equation value: 360.

K:1d2DEG: Single precision constant stored as 0.00278, scaled B-1 in units of revolutions. Equation value: 0.00139. (Equivalent to one-half of one degree.)

K:2SECS: Single precision constant stored as 200×2^{-14}, scaled B14 in units of centiseconds. Equation value: 200.

K:3SECS: Single precision constant stored as 300 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 300.

K:4FMAXNOM: Single precision constant stored as 14908 x 2^{-14}, scaled B14 in units of DPS throttle pulses. Corresponds to 4 x 3727 x 2^{-14}. The 3727 corresponds to 10,500 lbf. converted to throttle pulses. Equation value: 14908.

K:4SECb28: **Double precision constant stored as** 400 x 2^{-28}, **scaled** B26 in **units of centiseconds. Equation value:** 100.

K:4SECS: Single precision constant stored as 400 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 400.

K:99999CON: Double precision constant stored as 30479.7 x 2^{-24}, scaled B24 in units of meters. Equation value: 30479.7. (Equivalent to 99999 feet.)

K:ABAFCNST: Single precision constant stored as 0.13107, program notation /AF/CNST, scaled B-18 in units of meters per centisecond squared / centimeters per second per guidance cycle. Equation value: 5 E-7. (Equivalent to ½ x 0.01 cubed.)

K:AFTRGUID$_i$ (i = -1 thru 2): Table of single precision addresses for branching. Indexed in the order -1 thru 2, they are the addresses of: CGCALC, **EXTLOGIC, EXTLOGIC, STEER?.**

K:AZEACH: Single precision constant stored as 0.03491, scaled B0 in units of radians. Equation value: 0.03491. (Equivalent to 2 degrees.)

K:BIASFACT: Double precision constant stored as 655.36 x 2^{-26}, scaled B11 in units of seconds meters per centimeter. Equation value: 0.02. (Stored value corresponds to 2 sec x 0.01 m/cm x 2^{-11}.)

K:BIT1H: Double precision constant stored as 1 x 2^{-14}, scaled B14 and unitless. Equation value: 1.0

K:BIT6: Single precision constant stored as 00040_8, scaled B14 in units of DPS throttle pulses per centisecond. Equation value: 32.

K:DDUMCRIT: Double precision constant stored as 8 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 8.

K:DEPRCRIT: Double precision constant stored as -0.02 x 2^{-1}, scaled B1 in units of radians. Equation value: -0.02.

K:DPSTHRSH: Single precision constant stored as 36 x 2^{-14}, scaled B14 in units of centimeters per second. Equation value: 36. (Equivalent to K:THRESH1 + K:THRESH3 of the BURN section.)

K:ELEACH: Single precision constant stored as 0.00873, scaled B0 in units of radians. Equation value: 0.00873. (Equivalent to one-half of one degree.)

K:FEXTRA: Single precision constant stored as 10000_8, program notation also BIT13, scaled B14 in units of DPS throttle pulses. Equation value: 4096. (Equivalent to 51,331 newtons or 11,540 pounds force based on the value of K:SCALEFAC.)

K:fmax: Value of overflow bit on a quantity scaled B14 in units of DPS throttle pulses. Equation value: 16384.

K:FMAXODD: Single precision constant stored as 3841×2^{-14}, scaled B14 in units of DPS throttle pulses. Equation value: 3841. (Equivalent to 48,135 newtons or 10,821 pounds force based on the value of K:SCALEFAC.)

K:FMAXPOS: Single precision constant stored as 3467×2^{-14}, scaled B14 in units of DPS throttle pulses. Equation value: 3467. (Equivalent to 43,448 newtons or 9,767 pounds force based on the value of K:SCALEFAC.)

K:GHZ: Single precision constant stored as $1.62292\ E{-4} \times 2^{4}$, scaled B-4 in units of meters per centisecond squared. Equation value: 1.62292 E-4

K:GSCALE: Double precision constant stored as 100×2^{-11}, scaled B12 in units of centiseconds per navigation cycle. Equation value: 200.

K:GUIDDURN: Double precision constant stored as 66440×2^{-28}, scaled B28 in units of centiseconds. Equation value: 66440.

K:HIGHESTF: Double precision constant stored as $4.34546769 \times 2^{-12}$; scaled B12 in units of kilogram meters per centisecond squared. Equation value: 4.34546769. (Equivalent to 9,769 pounds force.)

K:KPIP1: See SERV section.

K:MOONRATE: Double precision constant stored as $0.2661699489\ E{-7} \times 2^{19}$, scaled B-19 in units of radians per centisecond. Equation value: 0.2661699489 E-7.

K:$NEWPHASE_i$ (i = -1 thru 2): Table of single precision addresses for branching. Indexed in the order -1 through 2, they are the addresses of: TTFINCR, TTFINCR, STARTP64, STARTP66.

K:P64DB: Single precision constant stored as 00155_8, scaled B-3 in units of revolutions. Equation value: 0.00083. (Equivalent to 0.2994 degrees.)

K:posmaxsp: See Major Variables.

K:$PREGUIDE_i$ (i = -1 thru 2): Table of single precision addresses for branching. Indexed in the order - 1 through 2, they are the addresses of: CALCRGVG, RGVGCALC, REDESIG, RGVGCALC.

K:PROJMAX: Single precision constant stored as 0.42262×2^{-3}, scaled B3 and unitless. Equation value: 0.42262. (Equivalent to the sine of 25 degrees.)

K:PROJMIN: Single precision constant stored as 0.25882×2^{-3}, scaled B3 and unitless. Equation value: 0.25882. (Equivalent to the sine of 15 degrees.)

K:SCALEFAC: Double precision constant stored as 797.959872 x 2^{-16}, scaled B16 in units of DPS throttle pulses / kilogram meter per centisecond squared. Equation value: 797.959872. (Equivalent to 12.532 newtons or 2.8173 pounds force per pulse.)

K:SHFTFACT: Double precision constant stored as 1 x 2^{-17}, scaled B17 and unitless. Equation value: 1.0.

K:TARGTDEX$_i$ (i = -1 thru 1): Table of single precision indexes, scaled B14 and unitless. Equation value indexed in the order -1 through 1: 0, 0, 28.

K:THROTLAG: Single precision constant stored as 20. x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 20.

K:TREDESCL: Single precision constant stored as -0.08, scaled B-3 in units of seconds per centisecond . Equation value: -0.01.

K:TRIMACCL: Double precision constant stored as 3.50132708 E-5 x 2^8, scaled B-8 in units of meters per centisecond squared. Equation value: 3.50132708 E-5.

K:TSCALINV: Single precision constant stored as 00010_8, scaled B11 and unitless. Equation value: 1.

K:ttf6b3: Double precision constant stored as 0.75, program notation 3/4DP, scaled B3 and unitless. Equation value: 6.

K:ttf24b6: Double precision constant stored as 0.375, program notation 3/8DP, scaled B6 and unitless. Equation value: 24.

K:UNITX, K:UNITZ: Double precision vector constants, stored as (0.5, 0, 0) and (0, 0, 0.5) respectively, scaled B1 and unitless. Equation values: (1, 0, 0) and (0, 0, 1).

K:WHATALM$_i$ (i = -1 thru 1): Table of single precision addresses for branching. Indexed in the order -1 through 1 they are the addresses of: 1406P00, 1406ALM, 1406ALM.

K:WHATDISP$_i$ (i = 0 thru 2): Table of single precision addresses for branching. Indexed in the order 0 through 2 they are the addresses of: P63DISPS, P64DISPS, VERTDISP.

K:WHATEXIT$_i$ (i = -1 thru 1): Table of single precision addresses for branching. Indexed in the order -1 through 1, they are the addresses of: EXGSUB, EXBRAK, EXNORM.

K:WHATGUID$_i$ (i = -1 thru 2): Table of single precision addresses for branching. Indexed in the order -1 through 2, they are the addresses of: TTF/8CL, TTF/8CL, TTF/8CL, CGCALC.

LAGdTAU: Double precision lag time divided by TAUROD, scaled B0 and unitless. Program notation LAG/TAU; part of the erasable load.

LAND, LANDTEMP: Double precision position vector of the landing site, scaled B24 in units of meters, measured from the center of the moon and expressed in the Platform coordinate system.

LANDMAG: Double precision radius magnitude of the landing site, scaled B24 in units of meters; program notation /LAND/.

LASTTPIP: Double precision storage for the time of the previous PIPA reading during P66 (R.O.D.) computations, scaled B28 in units of centiseconds.

LAT: see COOR section.

LEADTIME: Single precision negative of the time increment specifying how far the guidance computations are to be projected forward in P63 and P64, scaled B17 in units of centiseconds; part of the erasable load.

LM504: see COOR section.

LONG: see COOR section.

LOOKANGL: Single precision landing site elevation angle, scaled B14 in units of degrees. LOOKANGL is calculated as the complement of the angle between the LM +X axis and the negative LOS, which is equivalent to the angle between the LM YZ plane and the positive LOS.

LOWCRIT: Single precision upper limit on the variable throttle region in a situation of decreasing thrust commands, scaled B14 in units of DPS throttle pulses. If the throttle is set at maximum thrust, the desired thrust must fall below this limit before the program will command a throttle setting below maximum. LOWCRIT is part of the erasable load.

LUNDEX: Single precision index scaled B14 and unitless.

MASS: See SERV section.

MAXFORCE: Double precision maximum thrust that P66 will command, scaled B12 in units of kilogram meter per centisecond squared; part of the erasable load.

MINFORCE: Double precision minimum thrust that P66 will command, scaled B12 in units of kilogram meter per centisecond squared; part of the erasable load.

MODREG: See DATA section.

[MOONMAT]: See COOR section.

[NBSMMAT]: See COOR section.

NGUIDSUB: Single precision counter scaled B14 and unitless.

NIGNLOOP: Single precision counter scaled B14 and unitless.

OGABIAS: See BURN section.

OLDPIPA: Single precision storage for the accelerometer readings (PIPA) performed at time THISTPIP for P66 computations, scaled B14 in units of centimeters per second. Note that this is different from the normal two second cycle PIPA reading which is made at PIPTIME.

OUTOFPLN: Double precision distance of the landing site from the LM orbital plane at the projected time of ignition, scaled B24 in units of meters. (Positive if the orbital plane is to the right of the landing site, looking in the direction of travel.)

PIF: Double precision change in the desired thrust level, scaled B14 in units of DPS throttle pulses.

PIFPSET: Single precision bias on the throttle command, scaled B14 in units of DPS throttle pulses.

PIPA, PIPATMP: See SERV section.

$PIPABIAS_x$, $PIPABIAS_y$, $PIPABIAS_z$: See IMUC section.

PIPTIME, PIPTIME1: See SERV section.

POINTVSM: See ATTM section.

POLDPIPA: Single precision storage for the previous cycle value of OLDPIPA, scaled B14 in units of centimeters per second; program notation RUPTREG.

PREC: Single precision specification of the precision to which "ROOTPSRS" is to converge, scaled B0 and unitless.

PROJ, PROJ1, PROJ2: Single precision projection of the Y Descent Guidance coordinate system axis onto the unit normal to the plane defined by the X body axis and the line-of-sight vector, and the difference between that projection and its upper and lower bounds; scaled B3 and unitless.

PSEUDO55: Single precision storage for telemetry of the throttle command sent to the descent engine, scaled B14 in units of throttle pulses. (See definition of THRUST.)

QHZ: Single precision erasable memory constant representing the gain on UNFC, scaled B0 and unitless; part of the erasable load.

R60SAVE: Double precision temporary storage for the UNFC vector, scaled B7 in units of meters per centisecond.

R: Double precision navigated present position vector of the LM, scaled B24 in units of meters, measured from the center of the moon and expressed in the Platform coordinate system.

RA: Single precision ratio of the lag-diminished TTF to TTF, scaled B0 and unitless.

RANGEDSP: Double precision distance from the LM to the estimated landing site, scaled B24 in units of meters (displayed by noun 68).

RATT, VATT : see ORBI section.

RCV, VCV: see ORBI section.

[REFSMMAT]: see COOR section.

RGU: Double precision position vector of the LM, scaled B24 in units of meters, measured from the landing site on the moon's surface and expressed in the Descent Guidance coordinate system.

RLS: Double precision vector position of the landing site relative to the center of the moon, scaled B27 in units of meters and expressed in the Selenographic (moon-fixed) coordinate system; part of the erasable load.

RN, VN: see SERV section.

RODCOUNT: Single precision count of the number and direction of astronaut deflections of the rate-of-descent switch, scaled B14 and unitless.

RODSCAL1: Single precision working storage for RODSCALE, scaled B-7 in units of meters per centisecond.

RODSCALE: Single precision erasable memory quantity representing the velocity increment to be added or subtracted per each deflection of the R.O.D. switch during P66, scaled B-7 in units of meters per centisecond; part of the erasable load.

ROOTGOOD: Variable introduced as a substitute for a variable return address: Set to 2 or 0 to indicate a successful or non-successful convergence on the root of the "ROOTPSRS" polynomial.

ROOTPS: Double precision root extracted from an arbitrary polynomial by the Newton iteration method, scaling and units variable.

RTNHOLD: Single precision octal return address storage.

SCAXIS: see ATTM section.

T5ADR: see DAPA section.

TARGADG$_0$: Double precision Hi-gate acceleration aimpoint vector, scaled B-4 in units of meters per centisecond squared and expressed in the Descent Guidance coordinate system; program notation ADG or ABRFG; part of the erasable load.

$TARGADG_{28}$: Double precision Lo-gate acceleration aimpoint vector, scaled B-4 in units of meters per centisecond squared and expressed in the Descent Guidance coordinate system; program notation AAPFG; part of the erasable load.

$TARGRDG_{0}$: Double precision Hi-gate position aimpoint vector, scaled B24 in units of meters and expressed in the Descent Guidance coordinate system; program notation RDG or RBRFG; part of the erasable load.

$TARGRDG_{28}$: Double precision Lo-gate position aimpoint vector, scaled B24 in units of meters and expressed in the Descent Guidance coordinate system; program notation RAPFG; part of the erasable load.

$TARGVDG_{0}$: Double precision Hi-gate velocity aimpoint vector, scaled B10 in units of meters per centisecond and expressed in the Descent Guidance coordinate system; program notation VDG or VBRFG; part of the erasable load.

$TARGVDG_{28}$: Double precision Lo-gate velocity aimpoint vector, scaled B10 in units of meters per centisecond and expressed in the Descent Guidance coordinate system; program notation VAPFG; part of the erasable load.

TAT: See ORBI section.

TAUHZ: Double precision time constant for the horizontal velocity nulling in P66 Auto, scaled B11 in units of centiseconds; part of the erasable load.

TAUROD: Double precision time constant for the rate-of-descent equations in P66, scaled B9 in units of centiseconds; part of the erasable load.

$TCGF_{0}$: Single precision quantity representing the latest time at which the Descent Guidance coordinate system is erected in the braking phase (P63, WCHPHASE = 0), scaled B17 in units of centiseconds; program notation TCGFBRAK; part of the erasable load.

$TCGF_{28}$: Single precision quantity representing the latest time at which the Descent Guidance coordinate system is erected in the approach phase (P64, WCHPHASE = 1), scaled B17 in units of centiseconds; program notation TCGFAPPR; part of the erasable load.

$TCGI_{0}$: Single precision quantity representing the earliest time at which the Descent Guidance coordinate system is erected in the Braking phase (P63, WCHPHASE = 0), scaled B17 in units of centiseconds; program notation TCGIBRAK; part of the erasable load.

$TCGI_{28}$: Single precision quantity representing the earliest time at which the Descent Guidance coordinate system is erected in the approach phase (P64, WCHPHASE = 1), scaled B17 in units of centiseconds; program notation TCGIAPPR; part of the erasable load.

TDEC1: See ORBI section.

TEM: See SERV section.

$TEND_0$: Single precision quantity representing the time at which the approach phase (P64) is selected (i.e. WCHPHASE goes from 0 to 1 thus selecting P64), scaled B17 in units of centiseconds; program notation TENDBRAK; part of the erasable load.

$TEND_1$: Single precision quantity representing the time at which the vertical phase (P66) is automatically selected (i.e. WCHPHASE goes from 1 to 2 thus selecting P66 Auto provided the Mode Control switch is in the Auto position ; otherwise P66 "Manual" is selected), scaled B17 in units of centiseconds; program notation TENDAPPR; part of the erasable load.

TET: See ORBI section.

THISTPIP: Double precision time of PIPA readings for P66 (R.O.D.) computations, scaled B28 in units of centiseconds. Note that this is a different reading than that which is taken at PIPTIME.

THRDISP: Single precision quantity representing the percent that desired thrust is of 10,500 lbf. for display in Noun 92, scaled B14 and unitless.

THRUST: Cell used to provide DPS throttle commands by setting bit 4 of channel 14; scaled B14 in units of DPS throttle pulses. One pulse corresponds to about 12.532 newtons or 2.8173 pounds force (depending on erosion of the DPS nozzle), and the maximum command recognized by the throttle is 3428 pulses or about 42,960 newtons or 9658 pounds force.

TIG: see BURN section.

TIMENOW: see EXVB section.

TLAND: Double precision nominal time of lunar landing, scaled B28 in units of centiseconds; part of the erasable load.

TOOFEW: Single precision erasable memory constant used to initialize CNTTHROT, scaled B14 and unitless; part of the erasable load.

TPIP, TPIPOLD: Double precision storage for consecutive times of entry in the TTF incrementing routine, scaled B28 in units of centiseconds and used to increment TTF.

TREDES: Single precision time remaining to redesignate the landing site, scaled B14 in units of seconds (limited to 99).

TTF,TTFTMP: Double precision time from now until the achievement of the target conditions of the present guidance phase, scaled B17 in units of centiseconds.

$TTFADGZ_0$: Double precision Z component of $\underline{T}ARGADG_0$ multiplied by 6; scaled B-4 in units of meters per centisecond squared; program notation ABRFG* or $ADG2TTF_0$; part of the erasable load.

$TTFADGZ_{28}$: Double precision Z component of $\underline{T}ARGADG_{28}$ multiplied by 6; scaled B-4 in units of meters per centisecond squared; program notation AAPFG* or $ADG2TTF_{28}$; part of the erasable load.

TTFDISP: Double precision storage for TTF for display purposes, scaled B28 in units of centiseconds.

$TTFJDGZ_0$: Double precision Hi-gate jerk aimpoint, Z component only, scaled B-21 in units of meters per centisecond cubed; program notation JBRFG* or $JDG2TTF_0$; part of the erasable load.

$TTFJDGZ_{28}$: Double precision Lo-gate jerk aimpoint, Z component only, scaled B-21 in units of meters per centisecond cubed; program notation JAPFG* or $JDG2TTF_{28}$; part of the erasable load.

$TTFVDGZ_0$: Double precision Z component of $\underline{T}ARGVDG_0$ multiplied by 18; scaled B13 in units of meters per centisecond; program notation VBRFG* or $VDG2TTF_0$; part of the erasable load.

$TTFVDGZ_{28}$: Double precision Z component of $\underline{T}ARGVDG_{28}$ multiplied by 18; scaled B13 in units of meters per centisecond; program notation VAPFG* or $VDG2TTF_{28}$; part of the erasable load.

TTHROT: Single precision time of the last throttle command, scaled B14 in units of centiseconds.

$\underline{U}$NITR: See SERV section.

$\underline{U}$NFC: See BURN section. During the pre-ignition phase computations for the powered descent maneuver (P63), $\underline{U}$NFC represents the Delta-V vector for the pre-full throttle thrust, scaled B7 in units of meters per centisecond.

$\underline{U}$NWC: See BURN section.

$\underline{V}$: Double precision present navigated velocity vector of the LM, scaled B7 in units of meters per centisecond and expressed in the Platform coordinate system.

VBIAS: Double precision velocity bias factor based on PIPA bias values for P66 (R.O.D.) computations, scaled B8 in units of meters per centisecond.

VDGVERT: Double precision vertical component of velocity desired in the final (vertical) phase of descent, scaled B7 in units of meters per centisecond; altered in response to astronaut commands during manual descent control.

VGU: Double precision velocity vector of the LM relative to the rotating moon, scaled B10 in units of meters per centisecond and expressed in the Descent Guidance coordinate system.

VHZC: Double precision velocity vector representing the inertial velocity of a point at the LM as if it were rotating with the moon, scaled B7 in units of meters per centisecond.

WCHPHASE: Single precision index scaled B14 and unitless. Set to -1 in the pre-ignition phase ("P63LM"), 0 at ignition ("P63IGN"), 1 when TTF (negative) is greater than minus $TEND_0$ ("EXTLOGIC") thus selecting P64, and 2 when TTF is greater than minus $TEND_1$ ("EXTLOGIC") thus selecting P66 Auto or when the astronaut switches out of automatic control selecting P66 "Manual" ("STRTP66A")

WCHPHOLD: Single precision storage for WCHPHASE to preserve the present guidance mode through the present guidance cycle when WCHPHASE changes.

WCHVERT: Single precision flag to indicate whether the astronaut has responded to the flashing V06N60 display in P66 with either a PROCEED or ENTER.

WHICH: See BURN section.

WM: Double precision mean angular velocity vector of the moon, scaled B-17 in units of radians per centisecond and expressed in the Platform coordinate system.

XNBPIP, YNBPIP, ZNBPIP: See SERV section.

ZERLINA: Single precision counter scaled B14 and unitless.

ZOOMTIME: See BURN section.

DINT

GODSP

TSaddr = address of step that proceeded to "GODSP"

$MPAC_1$ = TS

$MPAC_4$ = 00000_8

Proceed to "GOFLASH2"

GODSPR

TSaddr = address of step that performed "GODSPR"

$MPAC_1$ = TS

$MPAC_4$ = 00000_8

Proceed to "GODSPRS1"

GODSPRET

TSaddr = address of step that performed "GODSPRET"

$MPAC_1$ = TS

$MPAC_4$ = 00040_8 (DOTHNRET)

Proceed to "GOFLASH2"

GOFLASH

TSaddr = address of step that proceeded to "GOFLASH"

$MPAC_1$ = TS

$MPAC_4$ = 00010_8 (FLREQ)

Proceed to "GOFLASH2"

GOFLASHR

TSaddr = address of step that performed "GOFLASHR"

$MPAC_1$ = TS

$MPAC_4$ = 00010_8 (FLREQ)

Proceed to "GODSPRS1"

GOPERF1

TSaddr = address of step that proceeded to "GOPERF1"

DSPTEM1 = TS

$MPAC_1$ = K:V01N25 (octal display of DSPTEM1 in R1)

$MPAC_4$ = 00036_8 (PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GOFLASH2"

GOPERF1R TSaddr = address of step that performed "GOPERF1R"

DSPTEM1 = TS

$MPAC_1$ = K:V01N25 (octal display of DSPTEM1 in R1)

$MPAC_4 = 00036_8$ (PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GODSPRS1"

GOPERF2 TSaddr = address of step that proceeded to "GOPERF2"

$MPAC_1$ = TS

$MPAC_4 = 00030_8$ (PERFREQ, FLREQ)

Proceed to "GOFLASH2"

GOPERF2R TSaddr = address of step that performed "GOPERF2R"

$MPAC_1$ = TS

$MPAC_4 = 00030_8$ (PERFREQ, FLREQ)

Proceed to "GODSPRS1"

GOPERF4 TSaddr = address of step that proceeded to "GOPERF4"

$MPAC_1$ = K:V04N06 (octal display of OPTION1 and OPTION2)

$MPAC_4 = 00014_8$ (FLREQ, R3BLNK)

Proceed to "GOFLASH2"

GOPERF4R TSaddr = address of step that performed "GOPERF4R"

$MPAC_1$ = K:V04N06 (octal display of OPTION1 and OPTION2)

$MPAC_4 = 00014_8$ (FLREQ, R3BLNK)

Proceed to "GODSPRS1"

REGODSP TSaddr = address of step that proceeded to "REGODSP"

$MPAC_1$ = TS

$MPAC_4 = 20000_8$ (RESETREQ)

Proceed to "GOFLASH2"

REGODSPR TSaddr = address of step that performed "REGODSPR"

$MPAC_1$ = TS

$MPAC_4$ = 20000_8 (RESETREQ)

Proceed to "GODSPRS1"

REFLASH TSaddr = address of step that proceeded to "REFLASH"

$MPAC_1$ = TS

$MPAC_4$ = 20010_8 (RESETREQ, FLREQ)

Proceed to "GOFLASH2"

REFLASHR TSaddr = address of step that performed "REFLASHR"

$MPAC_1$ = TS

$MPAC_4$ = 20010_8 (RESETREQ, FLREQ)

Proceed to "GODSPRS1"

CLEANDSP TSaddr = address of step that proceeded to "CLEANDSP"

$MPAC_1$ = 00000_8

$MPAC_4$ = 20010_8 (RESETREQ, FLREQ)

Proceed to "GOFLASH2"

GOXDSP TSaddr = address of step that proceeded to "GOXDSP" (GOMARK)

$MPAC_1$ = TS

$MPAC_4$ = 40000_8 (MKEXREQ)

Proceed to "GOFLASH2"

GOXDSPR TSaddr = address of step that performed "GOXDSPR" (GOMARKR)

$MPAC_1$ = TS

$MPAC_4$ = 40000_8 (MKEXREQ)

Proceed to "GODSPRS1"

EXDSPRET TSaddr = address of step that performed "EXDSPRET"

$MPAC_1$ = TS

$MPAC_4 = 40040_8$ (MKEXREQ, DOTHNRET)

Proceed to "GOFLASH2"

GOXDSPF TSaddr = address of step that proceeded to "GOXDSPF" (GOMARKF)

$MPAC_1$ = TS

$MPAC_4 = 40010_8$ (MKEXREQ, FLREQ)

Proceed to "GOFLASH2"

GOXDSPFR TSaddr = address of step that performed "GOXDSPFR" (GOMARKFR)

$MPAC_1$ = TS

$MPAC_4 = 40010_8$ (MKEXREQ, FLREQ)

Proceed to "GODSPRS1"

GOMARK2 TSaddr = address of step that proceeded to "GOMARK2"

$MPAC_1$ = TS

$MPAC_4 = 40030_8$ (MKEXREQ, PERFREQ, FLREQ)

Proceed to "GOFLASH2"

GOMARK2R TSaddr = address of step that performed "GOMARK2R"

$MPAC_1$ = TS

$MPAC_4 = 40030_8$ (MKEXREQ, PERFREQ, FLREQ)

Proceed to "GODSPRS1"

GOMARK3 TSaddr = address of step that proceeded to "GOMARK3"

$MPAC_1$ = TS

$MPAC_4 = 40230_8$ (MKEXREQ, DECVERB, PERFREQ, FLREQ)

Proceed to "GOFLASH2"

GOMARK3R Analogous to "GOMARK2R" except $MPAC_4 = 40230_8$

<u>GOMARK4</u> TSaddr = address of step that proceeded to "GOMARK4"

$MPAC_1$ = TS

$MPAC_4$ = 40036_8 (MKEXREQ, PERFREQ, FLREQ, R3BLNK, R2BLNK)

Proceed to "GOFLASH2"

<u>KLEENEX</u> TSaddr = address of step that proceeded to "KLEENEX"

$MPAC_1$ = 00000_8

$MPAC_4$ = 40010_8 (MKEXREQ, FLREQ)

Proceed to "GOFLASH2"

<u>PRIODSP</u> TSaddr = address of step that proceeded to "PRIODSP"

$MPAC_1$ = TS

$MPAC_4$ = 00110_8 (PRIOREQ, FLREQ)

Proceed to "GOFLASH2"

<u>PRIODSPR</u> TSaddr = address of step that performed "PRIODSPR" or "PRIOLARM"

$MPAC_1$ = TS

$MPAC_4$ = 00110_8 (PRIOREQ, FLREQ)

Proceed to "GODSPRS1"

<u>GOFLASH2</u> Inhibit interrupts

Set bits 11-9 of $MPAC_4$ = EBANK

$MPAC_3$ = TSaddr

Release interrupt inhibit

Proceed to "MAKEPLAY"

<u>GODSPRS1</u> Inhibit interrupts

Establish "MAKEPLAY" (if a flashing register display use VAC, if not no VAC; priority same as current job)

Set bits 11-9 of $MPAC_4$ = EBANK

$MPAC_3$ = TSaddr

Set $MPAC_i$ of "MAKEPLAY" job = $MPAC_i$ of present job for i = 1-4

Release interrupt inhibit

Return (in listing return is to calling address + 4, except if called by "GODSPR", "REGODSPR", or "GOXDSPR" then to caller + 1)

BLANKET Switch $MPAC_4$ bits indicated by a binary 1 in TS to 1

Set $MPAC_4$ of "MAKEPLAY" job = $MPAC_4$ of present job

Return

MAKEPLAY USERPRIO = priority of present job

Change priority to 33 (higher than "CHARIN")

If bit 15 of $MPAC_4$ (MKEXREQ) = 1, proceed to "MAKEMARK"

If bit 7 of $MPAC_4$ (PRIOREQ) = 1, proceed to "MAKEPRIO"

COPINDEX = 2

If bit 14 of $MPAC_4$ (RESETREQ) = 1: (REGODSP, REFLASH, CLEANDSP REGODSPR, REFLASHR)

If $CADRFLSH_2 \neq MPAC_3$, proceed to "OKTOPLAY"

If DSPLOCK = 0, proceed to "OKTOPLAY"

End job (display is already set and DSKY is busy; see "CLOKJOB" for example)

If $DSPFLG_2$ bit 4 (FLREQ) = 0, proceed to "OKTOPLAY" (Normal displays not requiring astronaut action can be replaced by other normal displays)

If FLAGWRD4 bits 13 (NRMIDFLG), 10 (NWAITFLG), 8 (NRMNVFLG) and 4 (NRUPTFLG) are all zero, proceed to "OKTOPLAY" (not replacing a display that has not yet been displayed)

Proceed to "BAILOUT" with TS = 31502_8 (two simultaneous requests for normal display)

OKTOPLAY Inhibit interrupts

$DSPFLG_2 = MPAC_4$

If $DSPFLG_2$ bit 6 (DOTHNRET) or 4 (FLREQ) = 1:

$CADRFLSH_2 = MPAC_3$

$NVWORD_2 = MPAC_1$

Release interrupt inhibit

If FLAGWRD4 bit 15 (MRKIDFLG), 14 (PRIODFLG), 12 (PDSPFLAG), 11 (MWAITFLG), 9 (MRKNVFLG), 7 (PRONVFLG), 5 (MRUPTFLG) or 1 (XDSPFLAG) = 1:

Switch FLAGWRD4 bit 10 (NWAITFLG) to 1

$MPAC_0$ = COPINDEX - 1

If a job with LOC = "PLAYJUM1" is asleep:

Awaken it and set its LOC = "ENDOFJOB" (kill it)

Put this job to sleep with a LOC = "PLAYJUM1" ($MPAC_{0-7}$ maintained in sleep)

When awakened , proceed to address specified in its LOC

(Otherwise, this job is free to proceed to use the DSKY)

Perform "WITCHONE"

If a job with LOC = TS is asleep:

Awaken it and set its LOC = "ENDOFJOB" (kill it)

PLAYJUM1 COPINDEX = 2 (COPINDEX not maintained by sleeping job)

Proceed to "NVDSP"

MAKEMARK COPINDEX = 1

Inhibit interrupts

$DSPFLG_1 = MPAC_4$

If $DSPFLG_1$ bit 6 (DOTHNRET) or 4 (FLREQ) = 1:

$CADRFLSH_1 = MPAC_3$

$NVWORD_1 = MPAC_1$

Release interrupt inhibit

If FLAGWRD4 bit 14 (PRIODFLG), 13 (NRMIDFLG), 12 (PDSPFLAG), 8 (NRMNVFLG) and 7 (PRONVFLG) are all 0: (no normal or priority display waiting for a response or a key release)

(If FLAGWRD4 bit 14 ... are all 0:)

If FLAGWRD4 bit 9 (MRKNVFLG) = 1, end job

Proceed to "MARKPLAY"

If FLAGWRD4 bit 14 (PRIODFLG), 12 (PDSPFLAG) and 7 (PRONVFLG) are all 0: (interrupt if a normal display)

Switch FLAGWRD4 bit 3 (MKOVFLAG) to 1

$MPAC_0$ = 1

Proceed to "JOBXCHS" (substitute mark for normal display)

If FLAGWRD4 bit 11 (MWAITFLG) or 5 (MRUPTFLG) = 1, end job

(otherwise, put mark to sleep until prio display is over)

Switch FLAGWRD4 bit 11 (MWAITFLG) to 1

$MPAC_0$ = COPINDEX - 1

If a job with LOC = "MARKPLAY" is asleep:

Awaken it and set its LOC = "ENDOFJOB" (kill it)

Put this job to sleep with a LOC = "MARKPLAY"

When awakened, proceed to address specified in its LOC

MARKPLAY Inhibit interrupts

Switch FLAGWRD4 bit 3 (MKOVFLAG) to 0 and bit 1 (XDSPFLAG) to 1

Release interrupt inhibit

If $DSPFLG_1$ bit 5 (PERFREQ) = 1, $NVWORD_1$ = $-NVWORD_1$

COPINDEX = 1

Proceed to "NVDSP"

MAKEPRIO COPINDEX = 0

If bit 14 (RESETREQ) of $MPAC_4$ = 0:

If FLAGWRD4 bit 14 (PRIODFLG) or 7 (PRONVFLG) = 1, proceed to "BAILOUT" with TS = 31502_8 (too many priority displays)

If bit 14 of $MPAC_4$ (RESETREQ) = 1: (not expected)

If $CADRFLSH_0 = MPAC_3$ and DSPLOCK > 0, end job

If FLAGWRD4 bit 15 (MRKIDFLG) or 9 (MRKNVFLG) = 1:

$MPAC_0 = 0$

Proceed to "JOBXCHS" (replace mark with prio display)

If FLAGWRD4 bit 13 (NRMIDFLG) or 8 (NRMNVFLG) = 1:

$MPAC_0 = 1$

Proceed to "JOBXCHS" (replace norm with prio display)

OKTOCOPY COPINDEX = 0

Inhibit interrupts

$DSPFLG_0 = MPAC_4$

If $DSPFLG_0$ bit 6 (DOTHNRET) or 4 (FLREQ) = 1:

$CADRFLSH_0 = MPAC_3$

$NVWORD_0 = MPAC_1$

Release interrupt inhibit

Perform "WITCHONE"

If a job with LOC = TS is asleep:

Awaken it and set its LOC = "ENDOFJOB"

REDOPRIO PRIOTIME = less significant half of TIMENOW

COPINDEX = 0

Proceed to "NVDSP"

JOBXCHS Perform "WITCHONE"

If a job with LOC = TS is asleep:

Awaken it and set its LOC = "XCHSLEEP" (to cause it to continue at "XCHSLEEP" when this job is finished)

(If a job with LOC = TS is asleep:)

Set $MPAC_0$ of awakened job = $MPAC_0$ of this job

If $MPAC_0$ = 0: (mark display replaced by a priority display)

Switch FLAGWRD4 bits 15 (MRKIDFLG) and 9 (MRKNVFLG) to 0

Switch FLAGWRD4 bit 5 (MRUPTFLG) to 1

If $MPAC_0$ = 1: (normal display replaced by mark or prio)

Switch FLAGWRD4 bits 13 (NRMIDFLG) and 8 (NRMNVFLG) to 0

Switch FLAGWRD4 bit 4 (NRUPTFLG) to 1

If FLAGWRD4 bit 3 (MKOVFLAG) = 1, proceed to "MARKPLAY"

Proceed to "OKTOCOPY" (priority)

XCHSLEEP If $MPAC_0$ = 0, TS = "MARKPLAY"

If $MPAC_0$ = 1, TS = "PLAYJUM1"

If a job with LOC = TS is asleep:

Awaken it and set its LOC = "ENDOFJOB" (kill it)

Put this job to sleep and set its LOC = TS

When awakened, proceed to the address specified in its LOC

WITCHONE Switch bit 5 of channel 11 to 0 (key release lamp off)

If FLAGWRD4 bit 9 (MRKNVFLG), 8 (NRMNVFLG) or 7 (PRONVFLG) = 1:

TS = DSPLIST

DSPLIST = +0

Return

(job active in the display interface routines is one that is awaiting astronaut release of the DSKY)

TS = CADRSTOR

CADRSTOR = +0

Return

(job active in the display interface routines is one that is awaiting an astronaut response)

NVDSP i = COPINDEX

GENMASK = $K{:}dspoctab_i$

$MPAC_4$ = $DSPFLG_i$

EBANK = bits 11-9 of $DSPFLG_i$

TSmonopt = bits 3-1 of $DSPFLG_i$

Switch $DSPFLG_i$ bit 13 (2NDPERF) to 0

TSdec = bit 8 of $DSPFLG_i$ (DECVERB)

MPAC2SAV = $MPAC_2$

MARK2PAC = $MPAC_2$

If $NVWORD_i = 0$:

  Establish "JAMTERM" (force return to calling address + 1) (pr32)

  Proceed to second step of "FLASHSUB"

If $NVWORD_i > 0$:

  TSvn = $NVWORD_i$

If $NVWORD_i < 0$: (expected only with marks)

  $NVWORD_1 = - NVWORD_1$

  TS = K:V05N00

  If TSdec = 00200_8 (bit 8 = 1), TS = K:V06N00

  TSvn = TS + low 7 bits of $NVWORD_1$ (verb 5 or 6 with noun)

NV50DSP Perform "NVSUB"

If FREEDSKY = 0: (display system in use externally)

  If CADRSTOR $\neq$ 0, end job

  If COPINDEX = 0 (GENMASK = 20144_8):

    Switch FLAGWRD4 bit 7 (PRONVFLG) to 1

    TSadr = "REDOPRIO"

(If FREEDSKY = 0:)

If COPINDEX = 1 (GENMASK = 42424_8):

Switch FLAGWRD4 bit 9 (MRKNVFLG) to 1

TSadr = "MARKPLAY"

If COPINDEX = 2 (GENMASK = 11254_8):

Switch FLAGWRD4 bit 8 (NRMNVFLG) to 1

TSadr = "PLAYJUM1"

If DSPLIST ≠ 0, proceed to "BAILOUT" with TS = 31206_8 (CADRSTOR = 0 from above)

Switch bit 5 of channel 11 to 1 (light key release lamp)

DSPLIST = TSadr

Put this job to sleep with a LOC = TSadr

When awakened, proceed to address specified in its LOC

Switch bit 6 of channel 11 to 0 (verb-noun flash off)

$MPAC_2$ = MPAC2SAV

i = COPINDEX

GENMASK = K:$dspoctab_i$

$MPAC_4$ = $DSPFLG_i$

EBANK = bits 11-9 of $MPAC_4$

Switch FLAGWRD4 bits 9 (MRKNVFLG), 8 (NRMNVFLG) and 7 (PRONVFLG) to 0

Perform "BLANKSUB" with TSblank = $MPAC_4$ (blank extraneous Registers)

If FREEDSKY = 0, proceed to "NVDSP"

If $MPAC_4$ bit 5 (PERFREQ) = 1 and bit 13 (2NDPERF) = 0:

Switch $DSPFLG_i$ bit 13 (2NDPERF) to 1

TSvn = bits 14-8 of $NVWORD_1$ (mark verb code)

TSmonopt = 00000_8

If $DSPFLG_i$ bit 15 (MKEXREQ) = 1, proceed to "NV50DSP"

(If PERFREQ = 1 and 2NDPERF = 0)

If $DSPFLG_i$ bit 12 (BURNREQ) = 1, TSvn = K:V97N00-NVWORD1

If $DSPFLG_i$ bit 12 (BURNREQ) = 0, TSvn = K:V50N00

Proceed to "NV50DSP"

If $MPAC_4$ bit 4 (FLREQ) = 1, proceed to "FLASHSUB"

If $MPAC_4$ bit 6 (DOTHNRET) = 1:

Change priority of this job to that stored in USERPRIO

Proceed to step following that whose address is stored in $CADRFLSH_{COPINDEX}$

If low 7 bits of $NVWORD_i$ = 0, proceed to "FLASHSUB"

End job

RELDSP Switch bit 14 of MONSAVE1 to 0 (turn off external monitor priority indicator)

If DSPLIST ≠ 0: (display job waiting for key release)

Awaken job with LOC = DSPLIST

DSPLIST = 0

Switch bit 5 of channel 11 to 0 (key release lamp off)

DSPLOCK = 0

Return

RELDSP1 If DSPLIST = 0, switch bit 5 of channel 11 to 0 (key rel off)

DSPLOCK = 0

Return

FLASHSUB Switch bit 6 of channel 11 to 1 (start verb-noun flash)

$MPAC_3$ = COPINDEX

If GENMASK = 20144_8, switch FLAGWRD4 bit 14 (PRIODFLG) to 1

If GENMASK = 42424_8, switch FLAGWRD4 bit 15 (MRKIDFLG) to 1

If GENMASK = 11254_8, switch FLAGWRD4 bit 13 (NRMIDFLG) to 1

If COPINDEX = 1 and R1SAVE = 2:

R1SAVE = 0

Return to address specified by $CADRFLSH_1$ + 3

If CADRSTOR ≠ 0:

If FLAGWRD4 bits 15 (MRKIDFLG) and 6 (PINBRFLG) both = 0:

Proceed to "BAILOUT" with TS = 31502_8

End job

If DSPLIST ≠ 0, proceed to "BAILOUT" with TS = 31206_8

CADRSTOR = K:IDLERET1

Put this job to sleep with a LOC = CADRSTOR

When awakened, proceed to address specified in its LOC

<u>VBTERM</u> LOADSTAT = -1 (verb 34)

Skip next 3 steps

<u>VBPROC</u> LOADSTAT = 1 (verb 33)

Skip next step

<u>VBRESEQ</u> LOADSTAT = -0 (verb 32)

MONSAVE1 = 40000_8 (kill monitor)

Perform "RELDSP"

Switch bit 6 of channel 11 to 0 (verb-noun flash off)

Proceed to "RECALTST"

<u>LOADLV</u> DECBRNCH = 00000_8 (data entry, mark verb etc.)

LOADSTAT = -0

Perform "RELDSP"

DSPCOUNT = -19

Proceed to "RECALTST"

RECALTST If CADRSTOR = 0, end job

TSadr = CADRSTOR

CADRSTOR = 0

Inhibit interrupts

Awaken job with LOC = TSadr

If LOADSTAT = -1, set LOC of awakened job to "TERMATE"

If LOADSTAT = 1, set LOC of awakened job to "PROCEED"

If LOADSTAT = -0, set LOC of awakened job to "IDLERET3"

TSnoun = NOUNREG (MPAC$_1$ of awakened job)

TSverb = VERBREG (MPAC$_0$ of awakened job)

Release interrupt inhibit

Perform "RELDSP"

End job

IDLERET3 If TSverb = 21, 22 or 23: (data load)

i = MPAC$_3$ (COPINDEX)

If TSnoun $\neq$ low 7 bits of NVWORD$_i$, proceed to "PINBRNCH" (load is not in response to request)

OUTHERE = 2

Skip next 3 steps

PROCEED OUTHERE = 1

Skip next step

TERMATE OUTHERE = 0

If FLAGWRD4 bit 15 (MRKIDFLG) = 1, proceed to "MARKRET"

If FLAGWRD4 bit 14 (PRIODFLG) = 0, proceed to "NORMRET"

TS = (less significant half of TIMENOW) - PRIOTIME

If TS < 0, TS = 16384 + TS

If TS ≤ K:2secB14:   (Priority display not up for 2 seconds)

COPINDEX = 0

Proceed to "NVDSP"

NORMRET If FLAGWRD4 bit 11 (MWAITFLG) or 5 (MRUPTFLG) = 1:

Proceed to "MARKWAKE"

If FLAGWRD4 bit 10 (NWAITFLG) or 4 (NRUPTFLG) = 1:

Proceed to "NORMWAKE"

If $DSPFLG_2$ bits 4 (FLREQ) and 6 (DOTHNRET) both = 0 and $NVWORD_2 \neq 0$:

Establish "PLAYJUM1"   (pr15)

Skip next step

MARKRET Switch FLAGWRD4 bit 3 (MKOVFLAG) to 0   (also bit 2 - meaningless)

ENDRET If OUTHERE < 0, end job

If GENMASK = 20144_8, switch FLAGWRD4 bits 14 (PRIODFLG) and 6 (PINBRFLG) to 0

If GENMASK = 42424_8, switch FLAGWRD4 bit 15 (MRKIDFLG) to 0

If GENMASK = 11254_8, switch FLAGWRD4 bits 13 (NRMIDFLG) and 6 (PINBRFLG) to 0

TSvn = -3

TSmonopt = 0

Perform "NVSUB"   (Blank display excepting MM number)

Change priority of this job to that specified in USERPRIO

$i = MPAC_3$ (COPINDEX)

$MPAC_3 = CADRFLSH_i$

If OUTHERE = 0, proceed as specified in the "if terminate" option at the step whose address is stored in $MPAC_3$

If OUTHERE = 1, proceed as specified in the "if proceed" option at the step whose address is stored in $MPAC_3$.

If OUTHERE = 2, proceed as specified in the "other response" option at the step whose address is stored in $MPAC_3$.

NORMWAKE Switch FLAGWRD4 bit 4 (NRUPTFLG) and 10 (NWAITFLG) to 0

Awaken job with LOC = "PLAYJUM1"

Proceed to "ENDRET"

MARKWAKE Switch FLAGWRD4 bit 5 (MRUPTFLG) and 11 (MWAITFLG) to 0

Awaken job with LOC = "MARKPLAY"

Proceed to "ENDRET"

PINBRNCH Release interrupt inhibit (if any)

$MPAC_2$ = MARK2PAC

If FLAGWRD4 bits 15 (MRKIDFLG), 14 (PRIODFLG), and 13 (NRMIDFLG) all = 0:

> TSvn = -3
>
> TSmonopt = 00000_8
>
> Perform "NVSUB"
>
> End job

If FLAGWRD4 bit 15 (MRKIDFLG) = 1, proceed to "MARKPLAY"

Proceed to "NORMBNCH"

ENDEXT EXTVBACT = 00000_8

Inhibit interrupts

Switch FLAGWRD4 bit 1 (XDSPFLAG) to 0

Release interrupt inhibit

OUTHERE = -1

If FLAGWRD4 bit 14 (PRIODFLG) and 13 (NRMIDFLG) are both zero:

> Proceed to "NORMRET"

NORMBNCH Switch FLAGWRD4 bit 6 (PINBRFLG) to 1

If FLAGWRD4 bit 14 (PRIODFLG) = 1:

COPINDEX = 0

Proceed to "NVDSP"

Proceed to "PLAYJUM1"

JAMTERM REQRET = 34

DSPCOUNT = -19

Proceed to "VBTERM"

PRIOLARM Perform "ALARM"

Proceed to "PRIODSPR" with TS = K:V05N09

CLOCPLAY TSaddr = address of step that proceeded to "CLOCPLAY"

$MPAC_1$ = TS

$MPAC_4 = 24030_8$ (RESETREQ, BURNREQ, PERFREQ, FLREQ)

Proceed to "GOFLASH2"

$CADRFLSH_i$ (i = 0,1,2): Three single precision cells for storage of return address information required by priority, mark and normal display requests. In the program itself, CADRFLSH is used for storage of the address of the step after that at which the display interface routine is called; in this writeup, CADRFLSH refers to the address of the step that called a display interface routine.

CADRSTOR: Single precision storage for the address of the job that is asleep while awaiting an astronaut response.

COPINDEX: Single precision index used to determine which display control parameters are relevant (priority, mark/extended verb, or normal = 0, 1, or 2), scaled B14 and unitless.

DECBRNCH: See DSKY section.

DSPCOUNT: See DSKY section.

$DSPFLG_i$ (i = 0,1,2): Three single precision flagwords containing control discretes for one priority (i=0), one mark (i=1) and one normal display (i=2) simultaneously. The individual bits have the same significance in each of the three flagwords (one would not expect bit 15 to be set in $DSPFLG_0$ or $DSPFLG_2$, etc.). The following is a list of the significance of each when it is set (1).

Bit	Mnemonic	Significance
15	MKEXREQ	Mark or extended verb display. (Higher priority than a normal display.)
14	RESETREQ	Reset request; replaces active display of same priority.
13	2NDPERF	Second loop through a "please perform" type display; the first loop displays the noun using a display verb. The second loop replaces the display verb with the appropriate "please perform" verb.
12	BURNREQ	The "please perform" verb is to be 97/99 rather than 50.

Bits 11-9 are used to store the setting of EBANK from the program using a display interface routine.

Bit	Mnemonic	Significance
8	DECVERB	The noun to be displayed with a "please perform" verb requires a decimal display verb in the first loop.
7	PRIOREQ	Priority display. (Higher priority than a mark or a normal display.)
6	DOTHNRET	Return only after display is executed. (Used only with displays that do not require astronaut action.)
5	PERFREQ	A "please perform" type display request.
4	FLREQ	This display requires astronaut response; set the verb and noun registers flashing.

Bit	Mnemonic	Significance
3	R3BLNK	Blank register 3.
2	R2BLNK	Blank register 2.
1	R1BLNK	Blank register 1.

Program notations EBANKSAV, MARKEBAN and EBANKTEM or DSPFLG, MARKFLAG and SAVEFLAG, respectively. $DSPFLG_2$ is set to 00000_8 in "DOFSTRT1".

DSPLIST: Single precision storage for the address of the job that is asleep while waiting for the astronaut to release control of the DSKY.

DSPLOCK: Single precision flagword set positive when the astronaut has control of the DSKY, and reset to zero when the DSKY is free for use by internal programs.

DSPTEM1: See DATA section.

EBANK: See MATX section.

EXTVBACT: See EXVB section.

FREEDSKY: See DATA section.

GENMASK: Single precision flagword indicating which bits of the $DSPFLG_i$ are to be set. Same as $MPAC_6$.

K:2secB14: Single precision constant stored as -200×2^{-14}, program notation -2SEC, scaled B14 in units of centiseconds. Equation value: 200.

K:$dspoctab_i$ (i = 0,1,2): Three single precision octal constants, program notation PRIOOCT+0, +1, +2, stored as 20144_8, 42424_8, and 11254_8.

K:IDLERET1: Single precision octal address changed in "RECALTST" to cause a flashing display interface routine to branch to "TERMATE", "PROCEED" or "IDLERET3".

K:VxxNxx: Single precision constant containing a seven-bit verb code (0-99) in bit positions 14-8 and a seven-bit noun code (0-99) in bit positions 7-1.

LOADSTAT: Single precision flag indicating whether an astronaut response is verb 34, verb 33, or verb 32 or a data entry or entry of a "please perform" verb.

LOC: See MATX section.

MARK2PAC: Single precision storage for $MPAC_2$.

MONSAVE1: See DATA section.

$MPAC_i$ (i = 0-7): A set of eight single precision cells associated with a particular job and used exclusively by that job. When a job is put to sleep or is interrupted by a job of higher priority, $MPAC_{0-7}$ are saved to be reset exactly as they were when the interrupted job is reestablished. See MATX section.

$MPAC_0$: Used as a flag to determine if a job being put to sleep is one handling a normal display or one handling a mark display.

$MPAC_1$: Used to store the verb-noun code at entry to the display interface logic.

$MPAC_2$: Used to store the address for an "address to be specified" noun when that noun is used by a program (as opposed to use of that noun by the astronaut, in which case he would enter the address via the DSKY).

$MPAC_3$: Used as temporary storage for $CADRFLSH_i$ or COPINDEX.

$MPAC_4$: Used as temporary storage for $DSPFLG_i$.

MPAC2SAV: Storage for $MPAC_2$ while it is being used as working storage.

NOUNREG: See DATA section.

$NVWORD_i$ (i = 0,1,2): Set of three single precision erasable memory cells used to retain values of the verb-noun pattern (verb code in bits 14-8, noun code in bits 7-1) for priority, mark/extended verb, and normal displays, respectively. Program notations NVWORD, MARKNV and NVSAVE, respectively. $NVWORD_2$ is initialized as 0 in "DOFSTRT1".

NVWORD1: See BURN section.

OPTION1, OPTION2: See DATA section.

OUTHERE: Single precision index used to determine the return from the display interface routines based on the type of astronaut response. Scaled B14.

PRIOTIME: Single precision time when a priority display is activated, scaled B14 in units of centiseconds. Used to assure that a reply to an interrupted normal or mark/extended verb display is not interpreted as a response to the interrupting priority display.

R1SAVE: Single precision cell, initialized to 0 in "STARTSB2", used in "FLASHSUB" to control a special return to the calling routine.

REQRET: See DSKY section.

TIMENOW: See EXVB section.

USERPRIO: Single precision storage for the priority of the job using the display interface routines. (Actually equal to $MPAC_7$ and therefore maintained while the job is asleep.)

VERBREG: See DATA section.

DSKY

Display and Keyboard Interface Logic

KEYRUPT1 (Entered on program interrupt #5)

SAMPTIME = TIMENOW (for noun 65)

TS = low 5 bits of channel 15 (five bit key code)

Switch FLAGWRD5 bit 15 (DSKYFLAG) to 1

Establish "CHARIN" (pr30)

Set $MPAC_0$ of "CHARIN" job = TS

Resume

CHARIN 21d22REG = DSPLOCK

DSPLOCK = 1

If CADRSTOR ≠ +0 and $MPAC_0 \neq 22_8$: (flashing display; not error reset

Switch bit 5 of channel 11 to 1 (key release lamp)

CHAR = $MPAC_0$

If CHAR = 1,2,3,4,5,6 or 7: (numbers 1 through 7)

Proceed to "NUM"

If CHAR = 10_8 or 11_8: (numbers 8 and 9)

If DSPCOUNT ≤ -0, end job

If bit 2 or 1 of DECBRNCH = 1, proceed to "NUM"

Proceed to "CHARALRM" (decimal numbers not allowed)

If CHAR = 20_8: (the number zero)

CHAR = 0

Proceed to "NUM"

If CHAR = 21_8, proceed to "VERB" (verb)

If CHAR = 22_8, proceed to "ERROR" (error reset)

If CHAR = 31_8, proceed to "VBRELDSP" (key release)

If CHAR = 32_8, proceed to "POSGN" (+ sign)

If CHAR = 33_8, proceed to "NEGSGN" (- sign)

If CHAR = 34_8, proceed to "ENTER" (enter)

If CHAR = 36_8, proceed to "CLEAR" (clear)

If CHAR = 37_8, proceed to "NOUN" (noun)

CHARALRM Switch bit 7 of channel 11 to 1 (operator error lamp)

End job

VERB VERBREG = 0

DSPCOUNT = K:VD1 (19)

Skip next two steps

NOUN NOUNREG = 0

DSPCOUNT = K:ND1 (17)

Perform "2BLANK" (blank verb or noun register)

DECBRNCH = 00001_8 (to indicate + decimal)

REQRET = +0

ENTRET = "ENDOFJOB"

End job

NUM If DSPCOUNT $\leq$ -0, end job (numbers not called for)

Perform "GETINREL"

If CLPASS > 0, CLPASS = 0 (indicate that number inserted since last clear)

CODE = low 5 bits of $K{:}RELTAB_{CHAR}$

Perform "DSPIN" with COUNT = DSPCOUNT (insert number on DSKY)

If bits 1 and 2 of DECBRNCH both = 0: (octal)

TS = XREG, YREG or ZREG according to whether INREL = 2, 3 or 4 (INREL not expected to equal 0 or 1 if number is octal)

TS = 8 TS + CHAR (TS cycled left three places)

If bit 1 or bit 2 of DECBRNCH = 1: (+ or - decimal)

TS = VERBREG, NOUNREG, XREG, YREG or ZREG according to whether INREL = 0, 1, 2, 3 or 4

TS = 10 TS + CHAR

If TS $\geq$ 16384, proceed to "DECEND" (must be 5th character)

If INREL = 0, set VERBREG = TS and i = 18

If INREL = 1, set NOUNREG = TS and i = 16

If INREL = 2, set XREG = TS and i = 10

If INREL = 3, set YREG = TS and i = 5

If INREL = 4, set ZREG = TS and i = 0

If i $\neq$ DSPCOUNT: (more characters may be inserted)

DSPCOUNT = DSPCOUNT - 1

End job

If bit 1 or bit 2 of DECBRNCH = 1, proceed to "DECEND"

DSPCOUNT = - DSPCOUNT (block further numerical characters)

End job

DECEND If INREL = 0 or 1: (verb or noun registers)

DSPCOUNT = - DSPCOUNT (scaling already B14)

End job

If bit 1 of DECBRNCH = 1, $TS_{dp} = 10^{-5}$ TS

If bit 2 of DECBRNCH = 1, $TS_{dp} = - 10^{-5}$ TS

If INREL = 2, store TS_{dp} in XREG and XREGLP (LP = lower half)

If INREL = 3, store TS_{dp} in YREG and YREGLP

If INREL = 4, store TS_{dp} in ZREG and ZREGLP

DSPCOUNT = - DSPCOUNT

End job

GETINREL Set INREL in accordance with DSPCOUNT and the following table:

DSPCOUNT	INREL	Significance
0 thru 4	4	Register #3, digits 5 (lowest) thru 1
5 thru 9	3	Register #2, digits 5 thru 1
10 thru 14	2	Register #1, digits 5 thru 1
16 or 17	1	Noun register, digits 2 or 1
18 or 19	0	Verb register, digits 2 or 1
15, 20 or over meaningless		

Return

POSGN Perform "SIGNTEST"

Perform "GETINREL"

SGNOFF = 5, 3 or 0 according to whether INREL = 2, 3 or 4

SGNON = SGNOFF + 1

Perform "SGNCOM"

Switch bit 1 of DECBRNCH to 1 (to indicate + decimal)

Proceed to "BOTHSGN"

NEGSGN Perform "SIGNTEST"

Perform "GETINREL"

SGNON = 5, 3 or 0 according to whether INREL = 2, 3 or 4

SGNOFF = SGNON + 1

Perform "SGNCOM"

Switch bit 2 of DECBRNCH to 1 (to indicate - decimal)

Proceed to "BOTHSGN"

SGNCOM CODE = 0

Perform "11DSPIN" with TS = SGNOFF

Switch bit 11 of CODE to 1

Perform "11DSPIN" with TS = SGNON

Return

SIGNTEST — If bit 1 or bit 2 of DECBRNCH = 1, end job (sign already set)

If DSPCOUNT ≠ 14 (R1D1), 9 (R2D1) or 4 (R3D1), end job

Return

BOTHSGN — If INREL = 2, switch bit 5 of DECBRNCH to 1

If INREL = 3, switch bit 4 of DECBRNCH to 1

If INREL = 4, switch bit 3 of DECBRNCH to 1

If CLPASS > 0, CLPASS = 0

End job

REQDATX — DSPCOUNT = K:R1D1 (14)

Skip next 3 steps

REQDATY — DSPCOUNT = K:R2D1 (9)

Skip next step

REQDATZ — DSPCOUNT = K:R3D1 (4)

REQRET = - return address (to cause "ENTER" to branch to caller of "REQDATX" etc. when requested data is entered)

Perform "5BLANK"

Switch bit 6 of channel 11 to 1 (verb-noun flash)

Proceed to address specified by ENTRET

REQMM — REQRET = - return address

DSPCOUNT = K:ND1 (17)

NOUNREG = 0

Perform "2BLANK"

Switch bit 6 of channel 11 to 1 (verb-noun flash)

DECBRNCH = 00001_8 (bit 1 = 1 to indicate + decimal)

Proceed to address specified by ENTRET

ENTER CLPASS = 0

ENTRET = "ENDOFJOB"

If REQRET ≥ +0, proceed to "ENTPASO" (normal data entry)

If |REQRET| = "MMCHANG" + 1: (data requested by "MMCHANG")

REQRET = |REQRET|

Switch bit 6 of channel 11 to 0 (verb-noun flash off)

Return via REQRET (to second step of "MMCHANG")

If bit 1 or bit 2 of DECBRNCH = 1 and DSPCOUNT is positive:

Proceed to "DSPALARM" (decimal data requires all five characters before an enter)

REQRET = |REQRET|

Switch bit 6 of channel 11 to 0 (verb-noun flash off)

Return via REQRET (to program that performed "REQDATX" etc.)

CLEAR TS = DSPCOUNT

Perform "GETINREL" with DSPCOUNT = |DSPCOUNT|

DSPCOUNT = TS

If CLPASS ≤ 0: (first "clear" since the last enter)

If INREL ≤ 1, end job (the verb and noun registers are not cleared, they are changed)

Perform "5BLANK" skipping the first step (have INREL already)

CLPASS = CLPASS + 1 (enable multiple clears if needed)

End job

(Otherwise, it's a multiple component clear)

INREL = INREL - 1 (Register #n has already been cleared; decrement INREL to indicate that register #n-1 is to be cleared as well.)

If INREL ≤ 1, end job (Registers 1, 2 and 3 are already cleared; "CLEAR" can do no more.)

REQRET = REQRET + 3 (change return address to loading program to force it to back up to a previous step in the loading process; from "perform 'REQDATY'" to "perform 'REQDATX'" etc.)

TS1 = INREL

VERBREG = VERBREG - 1

Perform "UPDATVB" (Change verb display to reflect the change in the component to be loaded.)

INREL = TS1

Perform "5BLANK" skipping the first step

CLPASS = CLPASS + 1 (enable clear of register #1 after #3 and #2)

End job

ERROR DSPLOCK = 21d22REG (error reset leaves DSPLOCK unchanged)

Inhibit interrupts

Switch bit 10 of channel 11 to 1 (reset "Restart" lamp)

Switch $DSPTAB_{11}$ to $100\ 000\ 000\ x0x\ 000_2$

(reset "Program Alarm", "Tracker fail", "LR altitude fail", and "LR velocity fail"; bits 9, 8, 5, and 3. Leave bits 6 and 4 alone; "Gimbal lock" and "No attitude" lamps.)

Switch bits 13, 12 and 11 of IMODES33 to 1
(Set PIPA good, Downlink good and Uplink good bits)

Switch big 10 of IMODES30 to 1
(Set PIPA good bit)

Switch RADMODES bits 8 (LRVELFLG), 5 (LRALTFLG) and 4 (RRDATAFL) to 0

Switch RADMODES bit 7 (RCDUFAIL) to 1

Perform "C13STALL"

Switch bit 10 of channel 13 to 0 (reset "test lamps" bit)

Switch bits 7 and 3 of channel 11 to 0
(reset "Operator Error" and "Uplink Activity" lamps)

$DSPTAB_i$ = ($|DSPTAB_i|$ with bit 12 set to 1) sign$DSPTAB_i$ for all values of i from 10 through 0

Release interrupt inhibit

$FAILREG_i$ = 0 for i = 0, 1

SFAIL = 0

End job

VBRELDSP Switch bit 3 of channel 11 to 0 (Reset Uplink Activity lamp)

If 21d22REG > 0 and bit 14 of MONSAVE1 = 1: (monitor going)

DSPLOCK = 0

If CADRSTOR = +0, perform "RELDSP1"

End job

Perform "RELDSP"

If CADRSTOR ≠ +0, proceed to "PINBRNCH"

End job

DSPOCTWD WDRET = return address

Switch bit 14 of DSPCOUNT to 1 (to blank sign in "DSPIN")

WDCNT = 4

WDAGAIN TSwd = TSwd cycled left 3 places (bit 15 shifted into bit 3 position, 14 into 2, etc.)

i = low 3 bits of TSwd

CODE = low 5 bits of $K{:}RELTAB_i$

COUNT = DSPCOUNT

If DSPCOUNT > 0, DSPCOUNT = DSPCOUNT - 1

Perform "DSPIN" (display digit in position specified by COUNT)

If WDCNT > 0:

  WDCNT = WDCNT - 1

  Proceed to "WDAGAIN"

DSPCOUNT = -19

Return via WDRET

DSPDECWD WDRET = return address

Perform "DSPSIGN" (returns with $MPAC_{dp} = |MPAC_{dp}|$)

$MPAC_{dp} = MPAC_{dp}$ + K:DECROUND

If overflow ($MPAC_{dp} \geq 1$), $MPAC_{dp}$ = K:posmaxdp

WDCNT = 4

DSPDCWD1 i = integral part of (10 $MPAC_{dp}$) (highest digit first)

CODE = low five bits of $K{:}RELTAB_i$ (character code)

$MPAC_{dp}$ = fractional part of (10 $MPAC_{dp}$)

COUNT = DSPCOUNT

If DSPCOUNT > 0, DSPCOUNT = DSPCOUNT - 1

Perform "DSPIN"

If WDCNT > 0:

    WDCNT = WDCNT - 1

    Proceed to "DSPDCWD1"

DSPCOUNT = -19

Return via WDRET

DSPSIGN — If $MPAC_{dp} \geq +0$, perform "+ON"

If $MPAC_{dp} \leq -0$, perform "-ON"

$MPAC_{dp} = |MPAC_{dp}|$

Return

+ON — Perform "GETINREL"

SGNOFF = 5, 3 or 0 according to whether INREL = 2, 3 or 4

SGNON = SGNOFF + 1

CODE = 00000_8 (bit 11 = 0)

Perform "11DSPIN" with TS = SGNOFF

Switch bit 11 of CODE to 1

Perform "11DSPIN" with TS = SGNON

Return

-ON — Perform "GETINREL"

SGNON = 5, 3 or 0 according to whether INREL = 2, 3 or 4

SGNOFF = SGNON + 1

CODE = 00000_8 (bit 11 = 0)

Perform "11DSPIN" with TS = SGNOFF

Switch bit 11 of CODE to 1

Perform "11DSPIN" with TS = SGNON

Return

DSPDC2NR WDRET = return address

Perform "DSPSIGN"

WDCNT = 1

Proceed to "DSPDCWD1"

DSPDECVN WDRET = return address

$MPAC_{dp}$ = K:VNDSPCON TS

WDCNT = 1

Proceed to "DSPDCWD1"

DSP2DEC WDRET = return address

CODE = 00000_8 (bit 11 = 0)

Perform "11DSPIN" with TS = 3 (-R2S off)

Perform "11DSPIN" with TS = 4 (+R2S off)

Perform "DSPSIGN"

WDCNT = 9

Proceed to "DSPDCWD1"

BLANKSUB FREEDSKY = 0

If DSPLOCK > 0, return

If bit 14 of MONSAVE1 = 1, return (no blank for externally initiated monitor)

FREEDSKY = 1

If bits 3-1 of TSblank all = 0, return

TScnt = DSPCOUNT

If bit 1 of TSblank = 1, perform "5BLANK" with DSPCOUNT = K:R1D1

If bit 2 of TSblank = 1, perform "5BLANK" with DSPCOUNT = K:R2D1

If bit 3 of TSblank = 1, perform "5BLANK" with DSPCOUNT = K:R3D1

DSPCOUNT = TScnt

Return

<u>5BLANK</u> Perform "GETINREL"

CODE = 0

If INREL = 2:

XREG = 0

XREGLP = 0

Set bits 5, 2 and 1 of DECBRNCH = 0

Perform "DSPIN" with COUNT = 14 (R1D1)

Perform "2BLANK" twice, with DSPCOUNT = 13, then 11

DSPCOUNT = K:R1D1

If INREL = 3:

YREG = 0

YREGLP = 0

Set bits 4, 2 and 1 of DECBRNCH = 0

Perform "DSPIN" with COUNT = 5 (R2D5)

Perform "2BLANK" twice, with DSPCOUNT = 9, then 7

DSPCOUNT = K:R2D1

If INREL = 4:

ZREG = 0

ZREGLP = 0

Set bits 3, 2 and 1 of DECBRNCH = 0

Perform "DSPIN" with COUNT = 4 (R3D1)

Perform "2BLANK" twice, with DSPCOUNT = 3, then 1

DSPCOUNT = K:R3D1

Return

2BLANK i = integral part of $\frac{1}{2}$ DSPCOUNT

If the sign of $DSPTAB_i$ is positive, NOUT = NOUT + 1

$DSPTAB_i$ = - 04000_8 (minus to flag for output)

Return

11DSPIN i = TS

COUNT = 2

Proceed to "DSPIN1"

DSPIN i = integral part of $\frac{1}{2}$ (low 5 bits of COUNT)

If bit 1 of COUNT = 0: (COUNT is even)

COUNT = 0

Proceed to "DSPIN1"

CODE = CODE shifted left five places (to positions 6-10)

If bit 14 of COUNT = 0:

COUNT = 1

Proceed to "DSPIN1"

COUNT = 3 (sign to be changed as well as digit)

DSPIN1 Inhibit interrupts

TS = $|DSPTAB_i|$

If COUNT = 0, set bits 1-5 of TS = bits 1-5 of CODE

If COUNT = 1, set bits 6-10 of TS = bits 6-10 of CODE

If COUNT = 2, set bit 11 of TS = bit 11 of CODE (sign)

If COUNT = 3, set bits 6-11 of TS = bits 6-11 of CODE

If TS = $|DSPTAB_i|$, skip next two steps

If $DSPTAB_i$ not already flagged for output, NOUT = NOUT + 1

$DSPTAB_i$ = - TS (complemented to flag for output)

Release interrupt inhibit

Return

21d22REG: Single precision storage for the DSPLOCK indication so that an "error reset" may leave DSPLOCK unchanged.

CADRSTOR: See DINT section.

CHAR: The five bit octal keycode extracted from bits 5 through 1 of channel 15 when a keyboard interrupt is triggered by the depression of any of the DSKY keyboard buttons. The DSKY buttons and the binary codes gated into channel 15 by their respective depressions are:

10001	VERB	00010	2	01000	8
11111	NOUN	00011	3	01001	9
11010	+	00100	4	11110	CLR
11011	-	00101	5	11001	KEY REL
10000	0	00110	6	11100	ENTR
00001	1	00111	7	10010	RSET

CLPASS: A single precision register used to direct the logic enabling multiple-clearing of the three five-digit display registers. Set to zero whenever a valid numerical character or a sign is processed unless it has been previously set to a large negative number to disable multiple clears. See routine "CLEAR".

CODE: A five-bit, binary relay code to select the configuration of the DSKY illumination relays to form the proper character. The character codes are:

10101	0	01111	4	11101	8
00011	1	11110	5	11111	9
11001	2	11100	6	00000	blank
11011	3	10011	7		

COUNT: Single precision register used as working storage in the "DSPIN" routine, first to designate the DSPTAB word to be changed, and second to indicate which of the two digits controlled by one DSPTAB word is to be changed and whether a sign is to be changed or blanked.

DECBRNCH: A single precision flagword whose bits have the following significance:

Bit 1	Plus decimal	(not octal)
2	Minus decimal	
3	Register #3 contains a decimal number	
4	Register #2 contains a decimal number	
5	Register #1 contains a decimal number	
6-15	spare	

Bits 3-5 are used in multiple component loads to assure all components are decimal or octal.

DSPCOUNT: A single precision quantity which indicates the register and digit position on the DSKY into which a number is to be placed. See description of DSPTAB registers. DSPCOUNT is set negative to indicate that further depressions of numerical keys are to be ignored by the program until another "command" key is depressed (VERB, NOUN, ENTR, etc.). Scaled B14.

DSPLOCK: Single precision flag set to some non-zero value to indicate when the display and keyboard are being used by the astronaut (or uplink).

$DSPTAB_i$ (i=0-10): Computer storage of the DSKY illumination relay settings. Bits 15 thru 13 are zero and bit 12 is 1 except when the DSPTAB register is complemented to direct the "T4RUPT" routine to change the displays.

R1, R2 and R3 are the three digital display registers, with D5 the least significant digit and D1 the most significant. Each of these registers has an associated sign bit, indicated below by -RiS or +RiS. If no sign is to be indicated, neither sign bit is set.

The two-digit (decimal) "noun", "verb" and "mode" or "program" registers are indicated below by N, V and M respectively and again, D1 is the more significant of the two digits.

The portion of the display that is controlled by each DSPTAB register is indicated below.

Register	Bit 11	Bits 6-10	Bits 1-5
$DSPTAB_{10}$		MD1	MD2
$DSPTAB_9$		VD1	VD2
$DSPTAB_8$		ND1	ND2
$DSPTAB_7$			R1D1
$DSPTAB_6$	+R1S	R1D2	R1D3
$DSPTAB_5$	-R1S	R1D4	R1D5
$DSPTAB_4$	+R2S	R2D1	R2D2
$DSPTAB_3$	-R2S	R2D3	R2D4
$DSPTAB_2$		R2D5	R3D1
$DSPTAB_1$	+R3S	R3D2	R3D3
$DSPTAB_0$	-R3S	R3D4	R3D5

$DSPTAB_{11}$: See INTR section.

ENTRET: See DATA section.

$FAILREG_i$ (i=0,1): See PGSR section.

FREEDSKY: See DATA section.

IMODES30: See IMUC section.

IMODES33: See INTR section.

INREL: Single precision index to indicate whether numerical characters are to be placed into the verb register, the noun register or one of the five-digit registers R1, R2 or R3.

K:DECROUND: Double precision constant stored as 00000_8 02476_8. Equation value: 0.000005 at a scaling of B0. Used to round a double precision AGC quantity to five decimal digits for display.

K:ND1: Single precision constant stored as 17, scaled B14. Used to set DSPCOUNT to indicate that the next numerical character received is to be placed into the most significant digit position of the noun register. See description of $DSPTAB_0$-$DSPTAB_{10}$.

K:R1D1, K:R2D1, K:R3D1: Single precision constants stored as 14, 9 and 4. Used to set DSPCOUNT to indicate that the next numerical character received is to be placed into the most significant digit position of R1, R2 or R3. See description of $DSPTAB_0$-$DSPTAB_{10}$.

K:$RELTAB_i$ (i=0-11) : Set of twelve single precision octal constants stored as follows:

i		i	
0	04025	6	34034
1	10003	7	40023
2	14031	8	44035
3	20033	9	50037
4	24017	10	54000
5	30036	11	60000

K:VD1: Single precision constant stored as 19, scaled B14. Used to set DSPCOUNT to indicate that the next numerical character received is to be placed into the most significant digit position of the verb register. See description of $DSPTAB_0$-$DSPTAB_{10}$.

K:VNDSPCON: Single precision constant stored as 00244_8. Equation value: 0.01. Used to convert decimal verb, noun and mode numbers (stored as whole numbers scaled B14) into fractions in preparation for the decimal display routine. Scaled B0.

MONSAVE1: See DATA section.

$MPAC_0$: Multiple precision accumulator associated with a particular job. Used in "KEYRUPT1" to store the keycode in the accumulator of the "CHARIN" job.

NOUNREG: See DATA section.

NOUT: See INTR section.

RADMODES: See RADR section.

REQRET: Single precision storage for the return address to an internal routine that has requested an astronaut input. Used also as a flag to avoid the processing of verb/noun information normally initiated by the "enter" keycode. Set equal to +0 to indicate that data is not a response to an internal request and that verb/noun information should be processed. Incremented by +3 to step backwards in the process of a multiple register load (from "REQDATY" to "REQDATX" for example). A "return address" is always a positive arithmetic quantity.

SAMPTIME: Double precision value of TIMENOW loaded at the time of astronaut entry of most recent keycode, by waitlist call for a monitor, or by an uplink interrupt.

SFAIL: See TEST section.

SGNOFF: Single precision index designating the DSPTAB register whose bit 11 is to be cleared to prevent illumination of both signs together.

SGNON: Single precision index designating the DSPTAB register whose bit 11 is to be set to illuminate a plus or minus sign in R1, R2 or R3. See description of $DSPTAB_0$-$DSPTAB_{10}$.

TIMENOW: See EXVB section.

VERBREG: See DATA section.

WDCNT: Single precision index scaled B14 and unitless.

WDRET: Single precision octal return address storage.

XREG, XREGLP, YREG, YREGLP, ZREG, ZREGLP: See DATA section.

Extended Verbs

GOEXTVB Proceed to routine specified by the contents of TSextfan

TSextfan	(verb)	Starting address of routine	Function
0	40	VBZERO	zero IMU CDU or Rendezvous Radar CDU
1	41	VBCOARK	coarse align (ICDU or RRCDU)
2	42	IMUFINEK	fine align IMU
3	43	IMUATTCK	load IMU attitude error needles
4	44	RRDESEND	terminate continuous designate
5		ALM/END	not defined
6		ALM/END	not defined
7	47	V47TXACT	AGS initialization
8	48	DAPDISP	load autopilot data
9	49	CREWMANU	start automatic attitude maneuver
10	50	GOLOADLV	please perform
11		ALM/END	not defined
12	52	GOLOADLV	please mark X
13	53	GOLOADLV	please mark Y
14	54	GOLOADLV	please mark X or Y
15	55	ALINTIME	align time
16	56	TRMTRACK	terminate tracking
17	57	LRON	permit landing radar updates
18	58	LROFF	inhibit landing radar updates
19	59	LRPOS2K	LR to position 2
20	60	RATEDISP	display DAP estimated rates
21	61	DAPATTER	display DAP attitude error
22	62	TOTATTER	display total attitude error
23	63	R04	sample radar once/second
24	64	VB64	calculate and display S-band ant. angles
25	65	SNUFFOUT	docked U-V control disable
26	66	ATTACHED	attached
27	67	V67	W matrix monitor
28	68	ALM/END	not defined
29	69	VERB69	cause restart
30	70	V70UPDAT	update liftoff time
31	71	V71UPDAT	universal update - block address
32	72	V72UPDAT	universal update - single address
33	73	V73UPDAT	update AGC time (octal)
34	74	DNEDUMP	initialize downlink for erasable dump
35	75	OUTSNUFF	remove U-V control disable
36	76	MINIMP	minimum impulse mode
37	77	NOMINIMP	rate command mode
38	78	R77	start LR spurious return test
39	79	R77END	terminate LR spurious return test

TSextfan	(verb)	Starting address of routine	Function
40	80	LEMVEC	update LM state vector
41	81	CSMVEC	update CSM state vector
42	82	V82PERF	request orbit param display (R30)
43	83	V83PERF	request rend param display (R31)
44		ALM/END	not defined
45	85	VERB85	display RR LOS azimuth and elevation
46		ALM/END	not defined
47		ALM/END	not defined
48		ALM/END	not defined
49	89	V89PERF	align X or Z LM axis along LOS (R63)
50	90	V90PERF	out of plane rend display
51	91	GOSHOSUM	display bank sum
52	92	SYSTEST	operate IMU performance test
53	93	WMATRXNG	clear RENDWFLG
54		ALM/END	not defined
55	95	UPDATOFF	no state vector update allowed
56	96	VERB96	interrupt integration and go to POOH
57	97	GOLOADLV	please verify engine failure
58		ALM/END	not defined
59	99	GOLOADLV	please enable engine

<u>ALM/END</u> Switch bit 7 of channel 11 to 1 (operator error)

Proceed to "PINBRNCH"

<u>TESTXACT</u> If EXTVBACT $>$ 0, proceed to "ALM/END"

If FLAGWRD4 bit 14 (PRIODFLG), 12 (PDSPFLAG) or 7 (PRONVFLG) = 1:

Proceed to "ALM/END"

EXTVBACT = 00025_8 (set bits 1, 3, and 5)

TSvn = -2

Perform "NVSUB" with TSmonopt = 00000_8

Check for new job waiting, and perform it if required

Return

VBZERO If NOUNREG = 20 and IMUCADR = 0: (ICDU) (vb40)

Perform "IMUZERO"

Perform "IMUSTALL"

Proceed to "PINBRNCH"

If NOUNREG = 72: (RRCDU)

Perform "RDRUSECK"

Perform "RRZERO"

Perform "RADSTALL"

Proceed to "PINBRNCH"

Proceed to "ALM/END" (If NOUNREG ≠ 20 or 72)

VBCOARK If NOUNREG = 20 and IMUCADR = 0: (vb41)

Perform "TESTXACT"

Proceed to "GOXDSPF" with TS = K:V25N22 (load THETAD)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, continue at next step.)

Perform "EXDSPRET" with TS = K:V41N00 (coarse align verb)

Perform "IMUCOARS"

Perform "IMUSTALL"

Proceed to "ENDEXT"

If NOUNREG = 72:

Perform "RDRUSECK"

Perform "TESTXACT"

Switch RADMODES bits 15 (CDESFLAG) and 10 (DESIGFLG) to 0

(If NOUNREG = 72:)

Proceed to "GOXDSPF" with TS = K:V24N73 (load $TANG_{0,1}$)
(If terminate, proceed to "ENDEXT"; if proceed, repeat this step; if other response, continue at next step.)

$OPTIONX_1$ = 2

$OPTIONX_0$ = 6

Perform "GOXDSPFR" with TS = K:V04N12 ($OPTIONX_{0,1}$)
(If terminate, proceed to "ENDEXT"; if proceed, skip next two steps; if other response, repeat this step.)

Perform "BLANKET" with TS = 00004_8

End job

If bit 2 of $OPTIONX_1$ = 0: ($OPTIONX_1$ = 0,1,4,5,...)

Switch FLAGWRD0 bit 5 (LOKONSW) to 1

If bit 2 of $OPTIONX_1$ = 1: ($OPTIONX_1$ = 2,3,6,7,...)

Switch FLAGWRD0 bit 5 (LOKONSW) to 0

Switch RADMODES bit 15 (CDESFLAG) to 1

Switch FLAGWRD5 bit 4 (NORRMON) to 1

Release interrupt inhibit

Perform "EXDSPRET" with TS = K:V41N00

Establish "RRDESK2" (pr20)

Proceed to "ENDEXT"

Proceed to "ALM/END" (NOUNREG ≠ 20 or 72)

<u>IMUFINEK</u> If IMUCADR ≠ 0, proceed to "ALM/END" (vb42)

Perform "TESTXACT"

Proceed to "GOXDSPF" with TS = K:V25N93 (OGC, IGC, MGC)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, continue at next step.)

Perform "EXDSPRET" with TS = K:V42N00

Perform "IMUFINE"

Perform "IMUSTALL"

If ISSGOOD = 0, proceed to "ENDEXT"

Perform "IMUPULSE" with TS = address of (OGC, IGC, MGC)

Perform "IMUSTALL"

Proceed to "ENDEXT"

<u>IMUATTCK</u> If MODREG ≠ 0, proceed to "ALM/END" (vb43)

If bit 4 or 5 of channel 12 is 1, proceed to "ALM/END" (not allowed if in coarse align or zeroing mode)

If bit 13 or bit 14 of channel 31 = 0, proceed to "ALM/END" (DAP is on)

Perform "TESTXACT"

Proceed to "GOXDSPF" with TS = K:V25N22 (load THETAD)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, continue at next step.)

Perform "EXDSPRET" with TS = K:V43N00

Switch bit 6 of channel 12 to 1 (Enable ICDU error counters)

Call "ATTCK2" in 0.02 second (See IMUC section)

Proceed to "ENDEXT"

<u>RRDESK2</u> Perform "RRDESNB" (established by verb 41 routine)

Perform "RADSTALL"

If RADGOOD = 0, perform "ALARM" with TS = 00503_8

End job

<u>RRDESEND</u> If RADMODES bit 15 (CDESFLAG) = 1: (vb44)

Inhibit interrupts

Switch bits 15 (CDESFLAG) and 10 (DESIGFLG) of RADMODES to 0

Switch bit 2 of channel 12 to 0 (disable RR error counter)

Delay 1 second

Switch FLAGWRD5 bit 4 (NORRMON) to 0

Proceed to "PINBRNCH"

V47TXACT Perform "TESTXACT" (vb47)

Establish "AGSINIT" (this section) (pr04)

End job

DAPDISP Perform "TESTXACT" (vb48)

Proceed to "DAPDATA1" (DAPB section) (change to pr07)

CREWMANU If MODREG ≠ 0, proceed to "ALM/END" (vb49)

Perform "TESTXACT"

Establish "R62DISP" (ATTM section) (pr10)

End job

GOLOADLV Switch bit 6 of channel 11 to 0 (verb/noun flash) (vb50,52, 53,54, 97,99)

Proceed to "LOADLV" (DINT section)

ALINTIME Perform "TESTXACT" (vb55) (change to pr07)

Proceed to "GOXDSPF" with TS = K:V25N24 ($DSPTEMX_{dp}$)
(If terminate, proceed to "ENDEXT"; if proceed, proceed to "ENDEXT"; if other response, continue at next step.)

If TSverb ($MPAC_0$) ≠ 23, proceed to "ENDEXT"

Inhibit interrupts

$MPAC_{dp}$ = TIMENOW
(exchange)
TIMENOW = 0

$MPAC_{dp} = MPAC_{dp} + DSPTEMX_{dp}$

Force sign agreement in the two halves of $MPAC_{dp}$

TIMENOW = TIMENOW + $MPAC_{dp}$

Release interrupt inhibit

Proceed to "ENDEXT"

TRMTRACK If FLAGWRD0 bits 7 (RNDVZFLG) and 9 (P25FLAG) both = 0: (vb56)

Proceed to "PINBRNCH"

Switch FLAGWRD0 bits 7 (RNDVZFLG) and 9 (P25FLAG) to 0

Switch FLAGWRD2 bit 14 (SRCHOPTN) to 0

	If FLAGWRD1 bit 5 (TRACKFLG) = 0, proceed to "PINBRNCH"		
	Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 0		
	Switch FLAGWRD0 bit 8 (IMUSE) to 0		
	Perform "INTSTALL"		
	Clear P20, P25 restart logic and inhibit interrupts		
	Perform "STOPRATE"		
	Perform "RESTORDB"		
	Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0		
	Switch bits 14 and 2 of channel 12 to 0	(disable tracker and RR error counter)	
	Proceed to "GOPROG2"		
LRON	Perform "TESTXACT"		(vb57)
	Proceed to "DSP68"	(this section)	
LROFF	Switch FLGWRD11 bit 8 (LRINH) to 0		(vb58)
	Proceed to "PINBRNCH"		
LRPOS2K	If FLAGWRD7 bit 5 (AVEGFLAG) = 0:	(see Anomaly Report No. L-1C-03)	(vb59)
	Proceed to "LRPOS2K1"		
	If MODREG = 63, proceed to "V59GP63"		
LRPOS2K1	Perform "RDRUSECK"		
	Proceed to "LRP2COMM"		
RATEDISP	Switch FLAGWRD0 bit 15 (NEED2FLG) to 1		(vb60)
	Proceed to "PINBRNCH"		
DAPATTER	Switch FLAGWRD0 bits 4 (NEEDLFLG) and 15 (NEED2FLG) to 0		(vb61)
	Proceed to "PINBRNCH"		
TOTATTER	Switch FLAGWRD0 bit 4 (NEEDLFLG) to 1 and bit 15 (NEED2FLG) to 0		(vb62)
	Proceed to "PINBRNCH"		
R04	Perform "RDRUSECK"	(this section)	(vb63)

	Perform "TESTXACT"		
	Switch FLAGWRD3 bit 9 (RO4FLAG) to 1		
	Proceed to "RO4Z"	(RADR section)	
<u>VB64</u>	Perform "TESTXACT"		(vb64)
	Establish "SBANDANT"	(this section)	(pr04)
	End job		
<u>SNUFFOUT</u>	Switch FLAGWRD5 bit 13 (SNUFFER) to 1		(vb65)
	Proceed to "PINBRNCH"		
<u>ATTACHED</u>	If FLAGWRD8 bit 8 (SURFFLAG) = 1:		(vb66)
	Proceed to "ALM/END"	(this section)	
	Establish "ATTACHIT"	(this section)	(pr10)
	End job		
<u>V67</u>	Perform "TESTXACT"	(this section)	(vb67)
	Establish "V67CALL"	(RNAV section)	(pr05)
	End job		
<u>VERB69</u>	Cause a hardware restart	("GOPROG")	(vb69)
<u>V70UPDAT</u>	UPVERBSV = 0		(vb70)
	Skip next five steps		
<u>V71UPDAT</u>	UPVERBSV = 1		(vb71)
	Skip next three steps		
<u>V72UPDAT</u>	UPVERBSV = 2		(vb72)
	Skip next step		
<u>V73UPDAT</u>	UPVERBSV = 3		(vb73)
	Perform "TESTXACT"		

If MODREG ≠ 0:

Switch bit 7 of channel 11 to 1 (operator error)

Switch bit 3 of channel 11 to 0 (uplink activity)

Proceed to "ENDEXT"

Proceed to "UPUPDATE" (this section)

DNEDUMP DNTMGOTO = "DNDUMPI" (vb74)

Proceed to "PINBRNCH"

OUTSNUFF Switch FLAGWRD5 bit 13 (SNUFFER) to 0 (vb75)

Proceed to "PINBRNCH"

MINIMP Switch DAPBOOLS bit 15 (PULSES) to 1 (vb76)

Proceed to "PINBRNCH"

NOMINIMP Switch DAPBOOLS bit 15 (PULSES) to 0 (vb77)

Inhibit interrupts

Perform "ZATTEROR" (DAPA section)

Proceed to "PINBRNCH" (releases interrupt inhibit)

R77 Perform "RDRUSECK" (vb78)

If FLAGWRD3 bit 9 (R04FLAG) = 1, proceed to "ALM/END"

Switch FLAGWRD5 bit 11 (R77FLAG) to 1

Proceed to "R04Z" (RADR section)

R77END RSAMPDT = 0 (vb79)

Delay 0.32 second

Switch FLAGWRD5 bit 11 (R77FLAG) to 0

Proceed to "PINBRNCH"

LEMVEC Switch FLAGWRD1 bit 8 (VEHUPFLG) to 0 (vb80)

Skip next step

<u>CSMVEC</u>	Switch FLAGWRD1 bit 8 (VEHUPFLG) to 1		(vb81)
	Switch FLAGWRD1 bit 6 (NOUPFLAG) to 0		
	Proceed to "PINBRNCH"		
<u>V82PERF</u>	Perform "TESTXACT"		(vb82)
	Proceed to "V82CALL"	(this section)	(change to pr07)
<u>V83PERF</u>	Perform "TESTXACT"		(vb83)
	Call "R31CALL" in 0.02 seconds	(this section)	
	End job		
<u>VERB85</u>	Perform "TESTXACT"		(vb85)
	Proceed to "DSPRRLOS"	(RNAV section)	
<u>V89PERF</u>	If MODREG ≠ 0, proceed to "ALM/END"		(vb89)
	Perform "TESTXACT"		
	Establish "V89CALL"	(ATTM section)	(pr10)
	End job		
<u>V90PERF</u>	Perform "TESTXACT"		(vb90)
	Establish "R36"	(this section)	(pr07)
	End job		
<u>GOSHOSUM</u>	If MODREG ≠ 0, proceed to "ALM/END"		(vb91)
	Perform "TESTXACT"		
	Proceed to "SHOWSUM2"	(TEST section)	
<u>SYSTEST</u>	If MODREG ≠ 0, proceed to "ALM/END"		(vb92)
	Perform "TESTXACT"		
	Establish "REDO"	(TEST section)	(pr22)
	End job		
<u>WMATRXNG</u>	Inhibit interrupts		(vb93)
	Switch FLAGWRD5 bit 1 (RENDWFLG) to 0		

Proceed to "PINBRNCH" (releases interrupt inhibit)

UPDATOFF Switch FLAGWRD1 bit 6 (NOUPFLAG) to 1 (vb95)

Proceed to "PINBRNCH"

VERB96 Switch FLAGWRD9 bit 5 (QUITFLAG) to 1 (vb96)

MMNUMBER = 0

Proceed to "V37" (PGSR section)

AGSINIT — If FLAGWRD3 bit 13 (REFSMFLG) = 0: (vb47)

Perform "ALARM" with TS = 00220_8

Proceed to "ENDEXT"

DSPTEMX = AGSK

AGSDISPK — Proceed to "GOXDSPF" with TS = K:V06N16 (DSPTEMX)
(If terminate, proceed to "ENDEXT"; if proceed, proceed to "AGSVCALC"; if other response, continue at next step.)

If TSverb ($MPAC_0$) = 32: (recycle)

AGSK = SAMPTIME (time when enter button pushed)

If TSverb ≠ 32, AGSK = $DSPTEMX_{dp}$

DSPTEMX = AGSK

Proceed to "AGSDISPK"

AGSVCALC — Switch FLAGWRD2 bit 1 (NODOFLAG) to 1

Switch FLAGWRD4 bit 1 (XDSPFLAG) to 1

Perform "EXDSPRET" with TS = K:V06N16

TDEC1 = TIMENOW

Perform "LEMPREC"

$\underline{TS}v$ = [REFSMMAT] $\underline{V}ATT$ K:VSCALE rescaled to B15 or B13 feet per second rounded off to single precision and converted to 2's complement

$\underline{TS}r$ = [REFSMMAT] $\underline{R}ATT$ K:RSCALE rescaled to B25 or B23 feet, rounded off to single precision and converted to 2's complement form

$AGSBUFF_0 = TSr_x$

$AGSBUFF_1 = TSv_x$

$AGSBUFF_2 = TSr_y$

$AGSBUFF_3 = TSv_y$

$AGSBUFF_4 = TSr_z$

$AGSBUFF_5 = TSv_z$

TDEC1 = TAT

Perform "CSMPREC"

$\underline{TSv}$ = [REFSMMAT] $\underline{VATT}$ K:VSCALE rescaled to B15 or B13 feet per second, rounded off to single precision and converted to 2's complement

$\underline{TSr}$ = [REFSMMAT] $\underline{RATT}$ K:RSCALE rescaled to B25 or B23 feet, rounded off to single precision and converted to 2's complement form

$AGSBUFF_6 = TSr_x$

$AGSBUFF_7 = TSv_x$

$AGSBUFF_8 = TSr_y$

$AGSBUFF_9 = TSv_y$

$AGSBUFF_{10} = TSr_z$

$AGSBUFF_{11} = TSv_z$

TSt = (TAT - AGSK)/K:TSCALE

$AGSBUFF_{12}$ = most significant half of TSt

$AGSBUFF_{13}$ = least significant half of TSt

DNLSTCOD = 1

Delay 20 seconds (for downlink of AGS downlist)

DNLSTCOD = AGSWORD (restore downlist)

If FLAGWRD0 bit 8 (IMUSE) = 0:

 If IMUCADR ≠ +0, delay 0.1 second, then repeat this step

 Perform "IMUZERO"

 Perform "IMUSTALL"

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

Proceed to "GOMARK3" with TS = K:V50N16 (DSPTEMX)
 (If terminate, proceed to "ENDEXT"; if proceed, proceed to "ENDEXT"; if other response, proceed to "ENDEXT".)

UPUPDATE UPOLDMOD = MODREG (verbs 70-73)

UPVERB = UPVERBSV

UPCOUNT = 1

DNLSTCOD = 1

MODREG = 27

Establish "DSPMMJOB" (pr30)

If UPVERB = 1 or 2, proceed to "OHWELL1" (get # of components)

COMPNUMB = 2

Proceed to "OHWELL2" (verbs 70 and 73 have only 2 components)

OHWELL1 $MPAC_2$ = "$UPBUFF_0$" (address of $UPBUFF_0$)

Proceed to "GOXDSPF" with TS = K:V21N01
(If terminate, proceed to "UPOUT4"; if proceed, repeat this step; if other response, continue at next step.)

If TSverb ($MPAC_0$) = 32, repeat previous step (recycle)

If $UPBUFF_0 < 3$, proceed to second line of "OHWELL1"

If $UPBUFF_0 > 20$, proceed to second line of "OHWELL1"

COMPNUMB = $UPBUFF_0$

UPCOUNT = UPCOUNT + 1

OHWELL2 i = UPCOUNT - 1

$MPAC_2$ = "$UPBUFF_i$" (address of)

Proceed to "GOXDSPF" with TS = K:V21N01
(If terminate, proceed to "UPOUT4"; if proceed, repeat this step; if other response, continue at next step.)

If TSverb ($MPAC_0$) = 32, repeat previous step (recycle)

If UPCOUNT $\geq$ COMPNUMB, proceed to "UPVERIFY"

UPCOUNT = UPCOUNT + 1

Proceed to "OHWELL2"

UPVERIFY $MPAC_2$ = "UPTEMP"

Proceed to "GOXDSPF" with TS = K:V21N02
(If terminate, proceed to "UPOUT4"; if proceed, proceed to "UPSTORE"; if other response, continue at next step.)

If TSverb ($MPAC_0$) = 32, repeat previous step (recycle)

If UPTEMP $\leq$ 0, proceed to "UPVERIFY"

If UPTEMP $>$ COMPNUMB, proceed to "UPVERIFY"

i = UPTEMP - 1

Proceed to second step of "OHWELL2"

UPSTORE Inhibit interrupts

Invert bit 3 of FLAGWRD7 (VERIFLAG)

If UPVERB $>$ 2, proceed to "UPEND73"

Establish "UPJOB" (pr30)

End job

UPEND73 $UPBUFF_{8,9}$ = $UPBUFF_{0,1}$

Perform "TIMEDIDL"; if error return, continue; otherwise skip next line
Switch bit 7 of channel 11 to 1 (error return)

Proceed to "UPOUT4"

TIMEDIDL $UPBUFF_{18,19}$ = TIMENOW (exchange)
TIMENOW = 0

$MPAC_{dp}$ = $UPBUFF_{8,9}$

$UPBUFF_{8,9}$ = 0

$MPAC_{dp}$ = $MPAC_{dp}$ + $UPBUFF_{18,19}$

If overflow ($|MPAC_{dp}| \geq 2^{28}$):

TIMENOW = TIMENOW + $UPBUFF_{18,19}$

$UPBUFF_{18,19}$ = 0

(If overflow)

Return to caller + one line (indicating an error return)

Force sign agreement between components of $MPAC_{dp}$

TIMENOW = TIMENOW + $MPAC_{dp}$

Return to caller + two lines (indicating a non-error return)

<u>UPJOB</u> Perform "INTSTALL"

Switch FLGWRD10 bit 7 (REINTFLG) to 1

If UPVERB = 0, proceed to "UPEND70"

If UPVERB = 1, proceed to "UPEND71"

If UPVERB = 2, proceed to "UPEND72"

<u>UPEND70</u> $UPBUFF_{8,9}$ = $-UPBUFF_{0,1}$

Perform "TIMEDIDL"; if error return, continue; otherwise skip next line

Switch bit 7 of channel 11 to 1 and skip next four lines (error return)

TETCSM = TETCSM - $UPBUFF_{0,1}$

TETLEM = TETLEM - $UPBUFF_{0,1}$

$UPBUFF_i$ = 0 for i = 10 through 13

TEPHEM = TEPHEM + $UPBUFF_{0,1}$ ($UPBUFF_{0,1}$ zeroed in process)

Perform "INTWAKEU"

Proceed to "UPOUT4"

<u>UPEND71</u> EBANK = bits 11-9 of $UPBUFF_1$

UPTEMP = low 8 bits of $UPBUFF_1$

i = COMPNUMB - 3

If UPTEMP + i $\geq 00400_8$, proceed to "UPERROUT" (block of addresses extends beyond end of EBANK)

UPTEMP = UPTEMP + 1400_8

$E_{j+UPTEMP}$ = $UPBUFF_{j+2}$ for j = 0 through i

Perform "INTWAKEU"

Proceed to "UPOUT4"

UPEND72 — If COMPNUMB is an even number, proceed to "UPERROUT"

Perform indented steps for i = 1,3,5,... through COMPNUMB - 2

  EBANK = bits 11-9 of $UPBUFF_i$

  TSadr = low 8 bits of $UPBUFF_i + 1400_8$

  $E_{TSadr} = UPBUFF_{i+1}$ (load specified address with given data)

  (end of "indented steps")

Perform "INTWAKEU"

UPOUT4 — MODREG = UPOLDMOD

Establish "DSPMMJOB" (pr30)

DNLSTCOD = 0

Switch bit 3 of channel 11 to 0 (uplink activity lamp)

Proceed to "ENDEXT"

UPERROUT — Switch bit 7 of channel 11 to 1 (operator error)

Perform "INTWAKEU"

Proceed to "UPOUT4"

V82CALL — If FLAGWRD7 bit 5 (AVEGFLAG) = 1, proceed to "V82GON"

$OPTIONX_0 = 2$

$OPTIONX_1 = 1$

Proceed to "GOXDSPF" with TS = K:V04N12
  (If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, repeat this step.)

Call "TICKTEST" in 0.08 second

V82GOFLP — V82FLAGS = 00000_8

Establish "V82GOFF1" (pr07)

If bits 1 and 2 of V82FLAGS both = 0, delay one second and then try this test again (i.e. delay until either is set to one)

Proceed to "GOXDSPF" with TS = K:V16N44 (HAPOX, HPERX, TFF)
(If terminate, switch bit 5 of EXTVBACT to 0 and end job;
if proceed, switch bit 5 of EXTVBACT to 0 and end job;
if other response, proceed to "V82GOFLP".)

V82GOFF1 TDEC1 = TIMENOW

TSTART82 = TDEC1

If $OPTIONX_1 \neq 1$, perform "CSMPREC"

If $OPTIONX_1 = 1$, perform "LEMPREC"

$\underline{R}ONE = \underline{R}ATT$

$\underline{V}ONE = \underline{V}ATT$

If PBODY = 0:

TFFdRTMU = K:1dRTMUE

HPERMIN = K:MINPERE

Switch FLAGWRD7 bit 2 (V82EMFLG) to 0

TS = K:RPAD

Proceed to "BOTHPAD"

If PBODY = 2:

TFFdRTMU = K:1dRTMUM

HPERMIN = K:MINPERM

TS = $|\underline{R}LS|$

Switch FLAGWRD7 bit 2 (V82EMFLG) to 1

BOTHPAD RPADTEM = TS

Perform "SR30.1" (get HAPOX, HPERX, TFF, mTPER)

TSTART82 = TIMENOW - TSTART82

If mTPER $\neq$ 0:

mTPER = TSTART82 + mTPER

Set V82FLAGS to 00001_8

(If mTPER ≠ 0)

End job

TFF = TFF + TSTART82

Set V82FLAGS to 00002_8

End job

<u>TICKTEST</u> If bit 5 of EXTVBACT = 0:

Establish "ENDEXT" (pr25)

End task

Call "TICKTEST" in 1 second

If bit 1 of V82FLAGS = 1:

mTPER = mTPER + K:1SEC

If bit 2 of V82FLAGS = 1:

TFF = TFF + K:1SEC

End task

<u>V82GON</u> Establish "V82GON1" (VAC area required) (pr07)

If NEWJOB > 0, perform "CHANG1"

Proceed to "GOXDSPF" with TS = K:V16N44 (HAPOX, HPERX, TFF)
(If terminate, continue at next step; if proceed, continue at next step; if other response, repeat this step.)

Switch bit 5 of EXTVBACT to 0

End job

<u>V82GON1</u> RONE = RN (loaded so as to prevent update of state vector between storage of R and V)

VONE = VN

If FLAGWRD8 bit 11 (LMOONFLG) = 1: (moon)

Switch FLAGWRD7 bit 2 (V82EMFLG) to 1

TFFdRTMU = K:1dRTMUM

(If FLAGWRD8 bit 11 (LMOONFLG) = 1:)

HPERMIN = K:MINPERM

TS = |RLS|

Proceed to "V82GON2"

If FLAGWRD8 bit 11 (LMOONFLG) = 0: (earth)

Switch FLAGWRD7 bit 2 (V82EMFLG) to 0

TFFdRTMU = K:1dRTMUE

HPERMIN = K:MINPERE

TS = K:RPAD

V82GON2 RPADTEM = TS

Perform "SR30.1"

If bit 5 of EXTVBACT = 0, proceed to "ENDEXT"

Delay 1 second

Proceed to "V82GON1"

SR30.1 If FLAGWRD7 bit 2 (V82EMFLG) = 1: (moon)

RONE = RONE, shifted left 2 places (B27)

VONE = VONE, shifted left 2 places (B5)

RMAG1 = |RONE|

VONEPR = TFFdRTMU VONE

TFFNP = $|(\text{RMAG1 VONEPR}) * \text{unitRONE}|^2$

TFFVSQ = $-|\text{VONEPR}|^2$

TFFALFA = 2 / RMAG1 + TFFVSQ

TFFRTALF = $\sqrt{|\text{TFFALFA}|}$

TS = TFFRTALF^2 signTFFALFA

If TS = 0:

TFF1dALF = 0

Skip next line

TFF1dALF = 1 / TS

TS = $\sqrt{|1 - \text{TFFALFA} \quad \text{TFFNP}|}$

RPER = TFFNP / (1 + TS)

TS1 = (1 + TS) TFF1dALF

If TS1 > 0:

    If |TS1| ≤ K:posmaxdp

        RAPO = TS1

        Skip next line

RAPO = K:posmaxdp

TS = RAPO - RPADTEM

If FLAGWRD7 bit 2 (V82EMFLG) = 1:   (moon centered)

    TS = TS, shifted right 2 places

HAPOX = TS   (limited to ≤ K:MAXNM)

TS = RPER - RPADTEM

If FLAGWRD7 bit 2 (V82EMFLG) = 1:

    TS = TS, shifted right 2 places

HPERX = TS   (limited to ≤ K:MAXNM)

If HPERX < HPERMIN, mTPER = +0

If HPERX ≥ HPERMIN:

    TSr = RPER

    Perform "CALCTPER"

    mTPER = - TSt

TSr = HPERMIN + RPADTEM

Perform "CALCTFF"

TFF = -TSt

Return

CALCTPER   Switch FLAGWRD7 bit 1 (TFFSW) to 1

Skip next step

CALCTFF   Switch FLAGWRD7 bit 1 (TFFSW) to 0

RTERM = TSr

TSqsq = (2 - RTERM TFFALFA) RTERM - TFFNP

If FLAGWRD7 bit 1 (TFFSW) = 1, TSqsq = 0

If TSqsq $<$ 0:   (trajectory does not cross RTERM)

    TSt = K:posmaxdp

    Return

QTERM = $\sqrt{\text{TSqsq}}$

TFFQ1 = $\underline{\text{V}}$ONEPR · $\underline{\text{R}}$ONE, shifted left 3 places

If TFFQ1 $\geq$ 0:   (outbound)

    TSnum = - QTERM - TFFQ1   (meters$^{\frac{1}{2}}$)

    TSden = 2 - RTERM TFFALFA - TFFALFA RMAG1   (unitless)

    If |TSden| $<$ 2^{-19}, TSden = 0

If TFFQ1 $<$ 0:   (inbound)

    TSnum = RTERM - RMAG1   (meters)

    TSden = - QTERM + TFFQ1   (meters$^{\frac{1}{2}}$)

    If |TSden| $<$ 2^{-6} or 2^{-7}, TSden = 0   (2^{-6} earth, 2^{-7} moon)

If TSden = 0:

    If TFFALFA $\leq$ 0:

        TSt = 0

        Return

    TS1dz = 0

    Proceed to "TFFEL1"

If |TSnum TFFRTALF / TSden| $\geq$ 1:

TS1dz = TSden/TSnum

Proceed to "TFFEL1"

TSz = TSnum/TSden

TFFTEM = TSz^2 TFFNP signTSz (used only in CSM)

TFFX = TSz^2 TFFALFA

TSpoly = K:TFF0 + K:TFF1 TFFX + K:TFF2 $TFFX^2$ + ... + K:TFF5 $TFFX^5$

TStrtmu = TSz (RTERM - 2TSz^2 TSpoly + RMAG1)

If TStrtmu $\geq$ 0, proceed to "ENDTFF"

If TStrtmu signTFFQ1 $\geq$ 0, proceed to "ENDTFF"

If TFF1dALF $\leq$ 0, proceed to "ENDTFF"

TStrtmu = TStrtmu + TFF1dALF K:PId16 / TFFRTALF

Proceed to "ENDTFF"

TFFEL1 TFFDELQ = -QTERM - TFFQ1

TFFX = $TS1dz^2$ TFF1dALF

If |TFFX| $\geq$ 1, TFFX = K:posmaxdp sign TFFX

TSpoly = K:TFF0 + K:TFF1 TFFX + K:TFF2 $TFFX^2$ + ... + K:TFF5 $TFFX^5$

TS = TS1dz TFF1dALF (TFFX TSpoly - 1)

TFFTEM = TFFNP TFF1dALF sign(TS1dz RMAG1 + TFFQ1) (CSM only)

TStrtmu = TFF1dALF (2TS + $\frac{1}{2}\sqrt{\text{TFF1dALF}}$ K:PId16 - TFFDELQ)

ENDTFF TSt = TStrtmu TFFdRTMU

If overflow, TSt = K:posmaxdp

Return

R31CALL Establish "V83CALL" (pr03)

Delay 1 second

If bit 12 EXTVBACT = 0, proceed to second line of "R31CALL"

Establish "DISPN5X" (pr05)

End Task

DISPN5X Proceed to "GOXDSPF" with TS = K:V16N54 (RANGE, RRATE, RTHETA)
(If terminate or proceed, set bit 5 of EXTVBACT = 0 and end job; if other response, repeat this step.)

V83CALL If FLAGWRD7 bit 5 (AVEGFLAG) = 1:

If FLAGWRD6 bit 8 (MUNFLAG) = 1, proceed to "GETRVN"

Proceed to "DOCMBASE" with TS6 = TIMENOW

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "DOCMBASE" with TS6 = TIMENOW

TDEC1 = TIMENOW

Perform "LEMPREC"

BASETHP = RATT (scaled B29 or B27)

BASETHV = VATT (scaled B7 or B5)

TS6 = TAT

DOCMBASE BASETIME = TS6

TDEC1 = TS6

Perform "CSMPREC"

BASEOTP = RATT (scaled B29 or B27)

BASEOTV = VATT (scaled B7 or B5)

REV83 If FLAGWRD7 bit 5 (AVEGFLAG) = 1, proceed to "GETRVN"

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

TDEC1 = TIMENOW

Perform "LEMPREC"

Proceed to "OTHCONIC"

TDEC1 = TIMENOW

Perform "INTSTALL"

Switch FLAGWRD0 bit 12 (MOONFLAG) to 0

RCV = BASETHP

VCV = BASETHV

If FLAGWRD8 bit 11 (LMOONFLG) = 1:

    Switch FLAGWRDO bit 12 (MOONFLAG) to 1

Switch FLAGWRD3 bit 4 (INTYPFLG) to 1

TET = BASETIME

Perform "INTEGRVS"

OTHCONIC RONE = RATT

VONE = VATT

Perform "INTSTALL"

Switch FLAGWRD3 bit 4 (INTYPFLG) to 1

TS = TAT

OTHINT TDEC1 = TS

Switch FLAGWRDO bit 12 (MOONFLAG) to 0

RCV = BASEOTP

VCV = BASEOTV

If FLAGWRD8 bit 11 (LMOONFLG) = 1:

    Switch FLAGWRDO bit 12 (MOONFLAG) to 1

TET = BASETIME

Perform "INTEGRVS"

COMPDISP RANGE = |RATT - RONE|

RRATE = unit(RATT - RONE) • (VATT - VONE)

Perform "CDUTRIG"

Perform "NBTOSM"

ZNBrf = $[REFSMMAT]^T$ [NBSMMAT] K:UNITZ

TSp = unit(ZNBrf - (ZNBrf • unitRONE) unitRONE)

TSu = ((unitRONE * VONE) * unitRONE) • TSp

RTHETA = arccos(TSp • ZNBrf signTSu)

If unitRONE • ZNBrf < 0, RTHETA = 1 – RTHETA

If bit 5 of EXTVBACT = 0, proceed to "ENDEXT"

Set bit 12 of EXTVBACT to 1

Proceed to "REV83"

GETRVN RONE = RN (change to pr22)

VONE = VN

TS1 = VCSM

TS2 = RCSM

TS = PIPTIME (change to pr03)

If FLAGWRD6 bit 8 (MUNFLAG) = 0:

Perform "INTSTALL"

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0

Proceed to "OTHINT"

RATT = TS2 [REFSMMAT]

VATT = TS1 [REFSMMAT]

(note that RATT and VATT are equivalent to locations 0D and 6D, respectively, of the pushdown list)

Proceed to "COMPDISP"

| R36 $DSPTEMX_{dp}$ = TIG (vb90)

Proceed to "GOXDSPF" with TS = K:V06N16 (DSPTEMX)
(If terminate, proceed to "ENDEXT"; if proceed, continue at next step; if other response, repeat this step.)

TDEC1 = DSPTEMX

If DSPTEMX = 0, TDEC1 = TIMENOW

Perform "CSMPREC"

RPASS36 = RATT

UNP36 = unit(VATT * unitRATT)

TDEC1 = TAT

Perform "LEMPREC"

TS1o's = RPASS36 – RATT

RANGE = RATT • UNP36

RRATE = VATT • UNP36

TSuf = unit((unitRATT * VATT) * unitRATT)

TSuf = unit(1, $TSuf_y$, $TSuf_z$)

TSulos = unit(TSlos - (unitRATT • TSlos) unitRATT)

TSulos = unit(1, $TSulos_y$, $TSulos_z$)

RTHETA = arccos(TSulos TSuf)

If (TSulos * TSuf • RATT) < 0, RTHETA = 1 - RTHETA

Proceed to "GOXDSPF" with TS = K:V06N90 (RANGE, RRATE, RTHETA)
(If terminate, proceed to "ENDEXT"; if proceed, proceed to "ENDEXT"; if other response, proceed to "R36")

<u>SBANDANT</u> TDEC1 = TIMENOW

Perform "LEMCONIC"

If PBODY = 0: (means earth)

TS = RATT

Skip next three lines

TSt = TAT

Perform "LSPOS"

TS = (K:REMDIST VMOON) + RATT

TS = -unitTS

Perform "CDUTRIG"

TS = [REFSMMAT] TS (transform to stable member)

$PITCHANG_{dp}$ = +0

$YAWANG_{dp}$ = +0

Perform "SMTONB"

$\underline{\text{RLM}}$ = [SMNBMAT] $\underline{\text{TS}}$

RLMYTEMP = RLM$_y$

RLM$_y$ = (RLM$_y$ - RLM$_x$) K:1OVSQRT2

RLM$_x$ = (RLMYTEMP + RLM$_x$) K:1OVSQRT2

$\underline{\text{RLMTEMP}}$ = $\underline{\text{RLM}}$

$\underline{\text{TS2}}$ = $\underline{\text{RLM}}$ - ($\underline{\text{RLM}}$ • $\underline{\text{K}}$:UNITY) $\underline{\text{K}}$:UNITY

$\underline{\text{TS2}}$ = unit$\underline{\text{TS2}}$; if overflow, proceed to "SBANDEX"

$\underline{\text{RLM}}$ = -($\underline{\text{TS2}}$ * $\underline{\text{K}}$:UNITZ)

TS = $\underline{\text{RLM}}$ • $\underline{\text{K}}$:UNITY

PITCHANG = arcsin(signTS |$\underline{\text{RLM}}$|)

TS1 = $\underline{\text{TS2}}$ • $\underline{\text{K}}$:UNITZ

If TS1 $<$ 0:

PITCHANG = 0.5 - PITCHANG

$\underline{\text{RLM}}$ = unit$\underline{\text{RLMTEMP}}$ * $\underline{\text{TS2}}$

$\underline{\text{TS}}$ = ($\underline{\text{K}}$:UNITX cosPITCHANG) - ($\underline{\text{K}}$:UNITZ sinPITCHANG)

TS = $\underline{\text{TS}}$ • $\underline{\text{RLM}}$

YAWANG = arcsin(signTS |$\underline{\text{RLM}}$|)

<u>SBANDEX</u> If bit 5 of EXTVBACT = 0, proceed to "ENDEXT"

(change to pr05)

Perform "GOXDSPFR" with TS = K:V06N51 (PITCHANG, YAWANG)
(If terminate or proceed, set bit 5 of EXTVBACT = 0 and end job; if other response, end job.)

TS = 100_2

Perform "BLANKET" (change to pr04)

Proceed to "SBANDANT"

<u>ATTACHIT</u> Perform "INTSTALL"

Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

If FLAGWRD8 bit 11 (LMOONFLG) = 0:

Switch FLAGWRD8 bit 12 (CMOONFLG) to 0

Inhibit interrupts

XKEPCSM = XKEPLEM

TCCSM = TCLEM

VCVCSM = VCVLEM

RCVCSM = RCVLEM

NUVCSM = NUVLEM

DELTACSM = DELTALEM

TETCSM = TETLEM

VRECTCSM = VRECTLEM

RRECTCSM = RRECTLEM

Release interrupt inhibit

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "USEPIOS"

Perform "MOVEPLEM"

Set FLAGWRD0 bit 12 (MOONFLAG) = FLAGWRD8 bit 11 (LMOONFLG)

PBODY = 0

If FLAGWRD0 bit 12 (MOONFLAG) = 1, PBODY = 2

Perform "SVDWN1" (scaling controled by PBODY)

QPRET = "PINBRNCH"

Proceed to "INTWAKE"

RDRUSECK If FLAGWRD3 bit 11 (NOR29FLG) = 0, proceed to "ALM/END"

If FLAGWRD5 bit 11 (R77FLAG) = 1, proceed to "ALM/END"

If FLAGWRD7 bit 6 (V37FLAG) = 0, skip next line

If FLGWRD11 bit 15 (LRBYPASS) = 0, proceed to "ALM/END"

If FLAGWRD1 bit 5 (TRACKFLG) = 0, return

Proceed to "ALM/END"

DSP68 Perform "GOXDSPFR" with TS = K:V06N68 (RANGEDSP, TTFDISP, DELTAH)
(If terminate, set bits 5 and 1 of EXTVBACT = 0 and end job;
if proceed, proceed to "SET57"; if other response, end job.)

WAIT68 Delay two seconds

If bit 5 and bit 1 of EXTVBACT = 0, proceed to "ENDEXT"

If bit 5 of EXTVBACT = 1, proceed to "DSP68"

Perform "GOMARK3R" with TS = K:V50N68 (RANGEDSP, TTFDISP, DELTAH)
(If terminate or proceed, set bits 5 and 1 of EXTVBACT = 0
and end job; if other response, proceed to "RESET57".)
(TS is formed by adding 13000_8 to K:V06N68)

Proceed to "WAIT68"

SET57 Switch FLGWRD11 bit 8 (LRINH) to 1

Set bit 5 of EXTVBACT = 0

End job

RESET57 Switch FLGWRD11 bit 8 (LRINH) to 0

EXTVBACT = 00025_8

End job

LRP2COMM Perform "LRPOS2" (See Anomaly Report No. L-1C-03)

Perform "RADSTALL"

If RADGOOD = 0, perform "ALARM" with TS = 00523_8

Proceed to "PINBRNCH"

V59GP63 RPCRTIME = 37777_8

RPCRTQSW = -1

Proceed to "PINBRNCH"

$AGSBUFF_{0,2,4}$: Single precision X, Y and Z components of the LM position vector, scaled B25 (earth) **or** B23 (moon) in units of feet and in stable member coordinates.

$AGSBUFF_{1,3,5}$: Single precision X, Y and Z components of the LM velocity vector, scaled B13 (moon) or B15 (earth) in units of feet/second and in stable member coordinates.

$AGSBUFF_{6,8,10}$: Single precision X, Y and Z components of the CSM position vector, scaled B25 (earth) or B23 (moon) in units of feet and in stable member coordinates.

$AGSBUFF_{7,9,11}$: Single precision X, Y and Z components of the CSM velocity vector, scaled B15 (earth) or B13 (moon) in units of feet/second and in stable member coordinates.

$AGSBUFF_{12,13}$: Double precision difference between the timetag of the state vectors in $AGSBUFF_{0-11}$ and the time stored in AGSK, scaled B18 in units of seconds.

AGSK: Double precision time of AGS initialization, scaled B28 in units of centiseconds.

AGSWORD: Single precision storage for the value of DNLSTCOD when a list is temporarily interrupted to change to another list.

BASEOTP, BASEOTV: Double precision vector storage for position and velocity of the CSM at BASETIME, scaled B29 (earth) or B27 (moon) for position; B7 (earth) or B5 (moon) for velocity. Position is in units of meters, and velocity is in units of meters/centisecond.

BASETHP, BASETHV: Double precision vector storage for position and velocity of the LM at BASETIME, scaled B29 (earth) or B27 (moon) for position in units of meters; B7 (earth) or B5 (moon) for velocity in units of meters per centisecond.

BASETIME: Double precision reference time for verb 83 routines, scaled B28 in units of centiseconds.

COMPNUMB: Single precision number of components (each single precision octal) in a program 27 update, scaled B14 and unitless.

DAPBOOLS: See DAPA section.

DELTACSM, DELTALEM: See ORBI section.

DELTAH: See SERV **section**.

DNLSTCOD: See TELE section.

DNTMGOTO: See TELE section.

DSPTEMX: See DATA section.

E_{ADR}: Single precision erasable memory cell whose address is in ADR.

EBANK: See MATX section.

EXTVBACT: Single precision flagword indicating when extended verbs are in action.

HAPOX, HPERX: Double precision heights above the earth or moon at apogee and perigee, scaled B29 in units of meters; displayed by N44.

HPERMIN: Double precision minimum perigee altitude, scaled B29 (earth) or B27 (moon) in units of meters; used to define the entry interface altitude.

IGC: See COOR section.

IMUCADR, ISSOGGD: See IMUC section.

K:1dRTMUE: Double precision constant, program notation 1/RTMUE , scale factor B-17, value 0.50087529 E-5 x 2^{17}. Corresponding to the reciprocal of root of unmodified earth μ.

K:1dRTMUM: Double precision constant, program notation 1/RTMUM , scale factor B-14, value 0.45162595 E-4 x 2^{14}. Corresponding to the reciprocal of root of moon μ.

K:1OVSQRT2: Double precision constant stored as 0.7071067815, corresponding to $1/\sqrt{2}$. Equation value: 0.7071067815. Scaled B0 and unitless.

K:1SEC: Single precision constant stored as 100 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 100.

K:MAXNM: Double precision constant stored as 01065_8 05603_8, scaled B29 in units of meters. Equation value: 18519814. (Equivalent to about 9999.8995 nautical miles.)

K:MINPERE: Double precision constant stored as 91440 x 2^{-29}, scaled B29 in units of meters. Equation value: 91440.

K:MINPERM: Double precision constant stored as 10668 x 2^{-27}, scaled B27 in units of meters. Equation value: 10668.

K:PId16: Double precision constant stored as 3.141592653 x 2^{-4}, scaled B5 and unitless. Equation value: 6.2831853.

K:posmaxdp: See "Major Variables" section.

K:REMDIST: Double precision constant stored as 384402000 x 2^{-29}, scaled B29 in units of meters. Equation value: 384402000.

K:RPAD: Double precision constant stored as 6373338 x 2^{-29}, scaled B29 in units of meters. Equation value: 6373338.

K:RSCALE: Double precision constant stored as 3.280839 x 2^{-3}, scaled B3 in units of feet per meter. Equation value: 3.280839.

K:TFF0, K:TFF1, K:TFF2, K:TFF3, K:TFF4, K:TFF5: Six double precision constant coefficients of a polynomial approximation. Scaled B0 and unitless. Equation value:

0.3333333333
-0.1999819135
0.1418148467
-0.101310997
0.05609004986
-0.01536156925

K:TSCALE: Double precision constant stored as 100 x 2^{-10}, scaled B10 in units of centiseconds per second. Equation value: 100.

<u>K</u>:UNITX, <u>K</u>:UNITZ, <u>K</u>:UNITY: See SERV section.

K:VSCALE: Double precision constant stored as 328.0839 x 2^{-9}, scaled B9 in units of feet per second/meters per centisecond. Equation value: 328.0839.

MGC: See COOR section

MMNUMBER: See PGSR section.

MODREG: See DATA section.

MPAC: See DINT section.

mTPER: Double precision time to perigee, scaled B28 in units of centiseconds.

[NBSMMAT]: See COOR section.

NEWJOB: See MATX section.

NOUNREG: See DATA section.

<u>N</u>UVCSM, <u>N</u>UVLEM: See ORBI section.

OGC: See COOR section.

$OPTIONX_0$, $OPTIONX_1$: Display registers used by noun 12 with extended verbs. Similar to OPTION1, OPTION2. Same register as DSPTEMX.

PBODY: See ORBI section.

PIPTIME: See SERV section.

PITCHANG: Cell used to contain the pitch gimbal angle required to point the S-band antenna toward the center of the earth. Scaled B0, in units of revolutions.

QPRET: See ORBI section.

QTERM: Double precision product of the cotangent of flight path angle at RTERM and the square root of semi-latus rectum, scaled B16 (earth) or B15 (moon) in units of meters to the one-half power.

RADGOOD, RADMODES: See RADR section.

RANGE, RRATE: Double precision range and range-rate, scaled B29 for range in units of meters and B7 for range-rate in units of meters per centisecond. In "R36" contains out of plane position and velocity, same scaling and units.

RANGEDSP: See DESC section.

RAPO, RPER: Double precision radius at apogee and perigee, scaled B29 (earth) or B27 (moon) in units of meters.

RATT, VATT, TAT: See ORBI section.
RCSM: See SERV section.
RCV, VCV: See CONC section.

RCVCSM, RCVLEM: See ORBI section.

[REFSMMAT] : See COOR section.

RLM: Double precision vector defined by transforming the unit line-of-sight vector (reference coordinates) first into stable member then into navigation base coordinates, and finally rotated and compensated by the orientation of the S-band antenna mount with respect to the navigation base.

RLMTEMP: Temporary storage location for RLM to be used in later calculations.

RLMYTEMP: Temporary storage location for RLM_y.

RLS: See CONC section.

RMAG1: Double precision magnitude of RONE, scaled B29 (earth) or B27 (moon) in units of meters.

RN, VN: See SERV section.

RONE, VONE: Double precision position and velocity vectors at TSTART82; scaled B29 (earth) or B27 (moon) for position, and B7 (earth) or B5 (moon) for velocity. Position is in units of meters, with velocity in units of meters/centisecond.

RPADTEM: Double precision radius of launch site on earth or moon for use as a base for computing altitude, scaled B29 (earth) or B27 (moon) in units of meters.

RPASS36: Double precision vector storage for CSM position vector in routine 36, scaled B29 in units of meters.

RPCRTIME: See SERV section.

RPCRTQSW: See SERV section.

RRECTCSM, RRECTLEM: See ORBI section.

RSAMPDT: See RADR section.

RTERM: Double precision terminal radius for calculation of TFF, scaled B29 (earth) or B27 (moon) in units of meters.

RTHETA: Display angle information for R31 and R36, scaled B0 in units of revolutions.

SAMPTIME: See DSKY section.

SMNBMAT : See COOR section.

$TANG_{0,1}$: See RADR section.

TCCSM, TCLEM: See ORBI section.

TDEC1: See ORBI section.

TEPHEM: See COOR section.

TET, TETCSM, TETLEM: See ORBI section.

TFF: Double precision time of free fall to RTERM, scaled B28 in units of centiseconds.

TFF1dALF: Double precision semi-major axis, stored in units of meters with variable scaling.

TFFALFA: Double precision reciprocal of the semi-major axis, stored as $meters^{-1}$ with variable scaling.

TFFDELQ: Double precision difference between −QTERM and TFFQ1, scaled B16 (earth) or B15 (moon).

TFFdRTMU: Double precision reciprocal of the square root of mu of primary body; variable scaling.

TFFNP: Double precision semi-latus rectum, stored in units of meters with variable scaling.

TFFQ1: Intermediate quantity calculated in "CALCTFF", scaled B16 (earth) or B15 (moon).

TFFRTALF: Double precision square root of TFFALFA, stored as $meters^{-\frac{1}{2}}$ with variable scaling.

TFFTEM: Double precision intermediate variable used in "CALCTFF", stored in units of meters with variable scaling.

TFFVSQ: Double precision value of the complement of the square of the velocity divided by the root of mu; variable scaling.

TFFX: Double precision universal variable, scaled B0 and unitless.

THETAD: See IMUC section.
TIG: See BURN section.
TIMENOW: Double precision current time scaled B28 in units of centiseconds; a computer counter incremented every centisecond automatically, and modified by verbs 55, 70 and 73.

TSTART82: Double precision start time of the verb 82 routines, scaled B28 in centiseconds; used to update TFF from its value at the time of verb 82 initialization to a value corresponding to the time at which it is displayed. Also used to update mTPER.
TTFDISP: See DESC section.
UNP36: Double precision vector storage for normal to the CSM orbital plane, scaled B1 and unitless.

UPBUFF$_{0-19}$: Single precision buffer cells for P27 updates.

UPCOUNT: Single precision number of components received in a P27 update, scaled B14 and unitless.

UPOLDMOD: Single precision storage for the value of MODREG at the initialization of a P27 update.

UPTEMP: Single precision storage for the number of a P27 update component to be corrected or for an address of a cell to be updated.

UPVERBSV, UPVERB: Single precision indication of the verb that initiated a P27 update, scaled B14 and unitless.

V82FLAGS: Single precision flagword used in verb 82 routines. Bit two is set when only TFF is computed and bit one is set when mTPER is computed.

VCSM: See SERV section.

VCVCSM, VCVLEM: See ORBI section.

VMOON: See COOR section.

VONEPR: Double precision value of VONE   TFFdRTMU, scaled B-10 (earth) or B-9 (moon).

VRECTCSM, VRECTLEM: See ORBI section.

XKEPCSM, XKEPLEM: See ORBI section.

YAWANG: Cell used to contain the yaw gimbal angle required to point the S-band antenna toward the center of the earth, scaled B0 in units of revolutions.

ZNBrf: See COOR section.

IMUC

IMU Computations

SVCT3 (This task is used as part of the waitlist control and is entered every 81.93 seconds)

If FLAGWRD2 bit 15 (DRIFTFLG) = 1:

- If IMUCADR = +0, establish "NBDONLY" (pr35)
- If IMUCADR ≠ +0, call "SVCT3" in 5.0 seconds

End task

1/GYRO $\underline{G}$COMP = $\underline{G}$COMP rescaled to B21 pulses (truncated at 2^{-7} pulses)

TS = address of $\underline{G}$COMP

Perform "IMUPULSE"

Perform "IMUSTALL"

If ISSGOOD = 0, End job

$\underline{G}$COMP = fractional part of $\underline{G}$COMP rescaled to B14 pulses

End job

NBDONLY If GCOMPSW < 0, End job

Inhibit interrupts

If FLAGWRD2 bit 15 (DRIFTFLG) = 0, End job

TS = 0

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

- TS = 00200_8
- Perform "PIPASR" skipping first step

TS1 = 1dPIPADT

1dPIPADT = TIMENOW_{1s} (load present time)

Release interrupt inhibit

TSt = 1dPIPADT - TS1 (present time - previous time)

NBD2 TSt = TSt (corrected for possible overflow of TIMENOW counter)

GCOMPSW = 0

If TS > 0: (SURFFLAG set)

$$\underline{GCOMP} = \underline{GCOMP} + \begin{bmatrix} -ADIAX & ADSRAX & 0 \\ 0 & -ADIAY & ADSRAY \\ 0 & -ADSRAZ & -ADIAZ \end{bmatrix} \underline{DELV}$$

$$\underline{GCOMP} = \underline{GCOMP} - TSt \begin{bmatrix} NBDX \\ NBDY \\ -NBDZ \end{bmatrix}$$

If $|GCOMP_{x_{sp}}| > 2$, GCOMPSW = 2(integral part of $\frac{1}{2}(|GCOMP_{x_{sp}}| - 1)) - 1$

If $|GCOMP_{y_{sp}}| > 2$, GCOMPSW = 2(integral part of $\frac{1}{2}(|GCOMP_{y_{sp}}| - 1)) - 1$

If $|GCOMP_{z_{sp}}| > 2$, GCOMPSW = 2(integral part of $\frac{1}{2}(|GCOMP_{z_{sp}}| - 1)) - 1$

If GCOMPSW > 0, proceed to "1/GYRO"

End job

1/PIPA If GCOMPSW < 0, return

$$\underline{DELV}_{dp} = \underline{DELV}_{sp} + \begin{bmatrix} PIPASCF_x & 0 & 0 \\ 0 & PIPASCF_y & 0 \\ 0 & 0 & PIPASCF_z \end{bmatrix} \underline{DELV}_{sp} - 1dPIPADT \begin{pmatrix} PIPABIAS_x \\ PIPABIAS_y \\ PIPABIAS_z \end{pmatrix}$$

GCOMPSW = 0

$$\underline{GCOMP} = \underline{GCOMP} + \begin{bmatrix} -ADIAX & ADSRAX & 0 \\ 0 & -ADIAY & ADSRAY \\ 0 & -ADSRAZ & -ADIAZ \end{bmatrix} \underline{DELV} - 1dPIPADT \begin{pmatrix} NBDX \\ NBDY \\ -NBDZ \end{pmatrix}$$

If $|GCOMP_x| > 2$, GCOMPSW = $|GCOMP_x| - 2$

If $|GCOMP_y| > 2$, GCOMPSW = $|GCOMP_y| - 2$

If $|GCOMP_z| > 2$, GCOMPSW = $|GCOMP_z| - 2$

If GCOMPSW > 0, establish "1/GYRO" (pr21)

Return

LASTBIAS Perform "PIPUSE", skipping 1st step

If GCOMPSW < 0, End job

TS = 0

If FLAGWRD8 bit 8 (SURFFLAG) = 1, TS = 00200_8

TSt = $PIPTIME_{1s}$ - 1dPIPADT

1dPIPADT = K:pip2sec

Proceed to "NBD2"

PIPUSE PIPA = 0 (-0)

If bit 6 of IMODES30 = 1: (IMU caged)

Return

Inhibit interrupts

Switch bit 1 of IMODES30 to 0 (Enable PIPA fail monitor)

Perform "SETISSW"

Release interrupt inhibit

Return

PIPFREE Inhibit interrupts

Switch bit 1 of IMODES30 to 1 (Disable PIPA fail monitor)

If bit 10 of IMODES30 = 0: (PIPA failure)

Perform "ALARM" with TS = 00212_8

Perform "SETISSW"

Release interrupt inhibit

Return

IMUMON TS = 00000_8

For i = 15, 14, 13, 12, 11 and 9, set bit i of TS to 1 if bit i of IMODES30 is not equal to bit i of channel 30

If TS = 00000_8, proceed to "TNONTEST"
(no change in IMU related discretes on channel 30)

Set bits 15-11 and 9 of IMODES30 equal to bits 15-11 and 9 of channel 30

If bit 15 of TS = 1: (change in IMU temperature discrete)

Switch bit 15 of TS to 0

If bit 15 of IMODES30 = 1: (IMU temp exceeding limits)

Switch bit 4 of channel 11 to 1
(turn on temperature caution lamp)

If bit 15 of IMODES30 = 0: (temp returned within limits)

If bit 1 of IMODES33 = 0, switch bit 4 of channel 11 to 0
(turn lamp off unless lamp test in progress)

If TS = 00000_8, proceed to "TNONTEST" (no further changes)

If bit 14 of TS = 1: (change in ISS turn-on delay discrete)

Switch bit 14 of TS to 0

If bit 2 of IMODES30 = 0: (no turn-on sequence failure in effect)

If bit 14 of IMODES30 = 0, perform "ITURNON2"
(ISS turn-on initiate; start turn-on sequence)

If bit 14 of IMODES30 = 1: (ISS turn-on delay just terminated)

If bit 15 of channel 12 = 0: (ISS turn-on delay was not terminated by LGC; set bit 2 to indicate turn-on-sequence failure)

Switch bit 2 of IMODES30 to 1

Perform "ALARM" with TS = 00207_8

If TS = 00000_8, proceed to "TNONTEST"

If bit 13 of TS = 1: (change in status of IMU fail discrete)

Switch bit 13 of TS to 0

Perform "SETISSW"

If TS = 00000_8, proceed to "TNONTEST"

If bit 12 of TS = 1: (change in status of ICDU fail discrete)

Switch bit 12 of TS to 0

Perform "SETISSW"

If TS = 00000_8, proceed to "TNONTEST"

If bit 11 of TS = 1: (change in status of IMU cage discrete)

Switch bit 11 of TS to 0

If bit 11 of IMODES30 = 0: (IMU caged externally)

Switch bits 15-10 of channel 14 to 0
(stop all ICDU and RRCDU drive pulses)

Switch bits 8, 6, 5, 4 and 2 of channel 12 to 0
(disable inertial data display, disable ICDU Error Counters, reset ICDU zero discrete, remove coarse align enable discrete, disable RRCDU Error Counters)

Switch FLAGWRD5 bit 7 (ENGONFLG) to 0

Switch bit 13 of channel 11 to 0 and bit 14 of channel 11 to 1 (engine control discretes to off)

Perform "CAGESUB1"

Perform "RNDREFDR" (reset TRACK, DRIFT and REFSM flags)

CDU_iCMD = 0 for i = x, y and z (-0)

GYROCMD = 0 (-0)

Switch bits 9-6 of channel 14 to 0
(remove all gyro-torque logic discretes)

If bit 11 of IMODES30 = 1, proceed to "ISSZERO"

If TS = 00000_8, proceed to "TNONTEST"

If bit 9 of TS = 1: (IMU power on/off)

Switch bit 9 of TS to 0

If bit 9 of IMODES30 = 1: (IMU power off)

Switch bit 6 of IMODES33 to 1 (disable DAP)

Perform "RNDREFDR"

If FLAGWRDO bit 8 (IMUSE) = 1:

Perform "ALARM" with TS = 00214_8

Switch FLAGWRDO bits 8 (IMUSE) and 7 (RNDVZFLG) to 0

If bit 9 of IMODES30 = 0: (IMU power on)

If bit 2 of IMODES30 = 0, perform "ITURNON2"
(Start turn-on sequence if no turn-on-sequence failure indication present)

TNONTEST If bit 7 of IMODES30 = 0, proceed to "C33TEST"

If bit 8 of IMODES30 = 0; (Delay till next T4RUPT cycle)

Switch bit 8 of IMODES30 to 1

Proceed to "C33TEST"

Switch bits 7 and 8 of IMODES30 to 0

If bit 14 of IMODES30 = 1: (ISS power on without initiation of turn-on delay, e.g. verb 36)

If bit 4 of channel 12 = 1, proceed to "C33TEST"
(coarse align enabled; may be near gimbal lock)

If FLAGWRDO bit 8 (IMUSE) = 1, proceed to "C33TEST"

Perform "CAGESUB2"

Proceed to "ISSZERO"

If bit 9 of IMODES30 = 1, perform "ALARM" with TS = 00213_8
(Turn-on delay initiated without ISS power on)

Perform "CAGESUB"

Call "ENDTNON" in 90 seconds

Proceed to "C33TEST"

ISSZERO Switch bit 4 of $DSPTAB_{11}$ to 0 (turn off no attitude lamp)

Switch bit 15 of $DSPTAB_{11}$ to 1 (to flag for output)

Switch bit 5 of channel 12 to 1 (ICDU zero)

CDU = 0

Call "UNZ2" in 0.32 second

C33TEST TS = 00000_8

For i = 13, 12 and 11, switch bit i of TS to 1 if bit i of IMODES33 is not equal to bit i of channel 33

If TS = 00000_8, return (to T4RUPT routine)

Set bits 13-11 of IMODES33 = bits 13-11 of channel 33

(Channel 33 flip-flops reset by WAND instruction)

If bit 13 of TS = 1: (change in status of PIPA fail discrete)

Switch bit 13 of TS to 0

Set bit 10 of IMODES30 = bit 13 of IMODES33

Perform "SETISSW"

If bit 1 of IMODES30 = 1: (primary PIPA monitor inhibited)

If bits 10, 9, 8, 7 and 5 of IMODES30 all = 0:
(PIPA fail, IMU power on, turn-on delay complete, turn-on delay not just initialized, and secondary PIPA fail monitor enabled ("PFAILOK"))

Perform "ALARM" with TS = 00212_8

If TS = 00000_8, return (to T4RUPT routine)

If bit 12 of TS = 1: (Downlink)

Switch bit 12 of TS to 0

Perform "DNTMFAST"

If TS = 00000_8, return (to T4RUPT routine)

If bit 11 of TS = 1: (Uplink)

Switch bit 11 of TS to 0

Perform "UPTMFAST"

Return (to T4RUPT routine)

CAGESUB Switch bits 6 and 15 of channel 12 to 0
(Disable ICDU Error Counter and reset "Turn-on delay complete" discrete.)

Switch bits 4 and 5 of channel 12 to 1
(Set coarse align discrete and ICDU zero discrete)

CAGESUB1 Switch bit 4 of $DSPTAB_{11}$ to 1 and flag for output
(No attitude lamp on)

CAGESUB2 Switch bits 1, 3, 4, 5 and 6 of IMODES30 to 1
(Inhibit PIPA, CDU and IMU fail monitors, inhibit secondary PIPA fail monitor, and set IMU caged flag)

Switch bit 6 of IMODES33 to 1 (Disable DAP)

Return

SETISSW TS = 00000_8

If bits 13 and 4 of IMODES30 both = 0, TS = 10000_8 (IMU fail)

If bits 12 and 3 of IMODES30 both = 0, TS = 04000_8 + TS (ICDU)

If bits 10 and 1 of IMODES30 both = 0, TS = 01000_8 + TS (PIPA)

If TS $\neq$ 00000_8: (failure)

Perform "ALARM" with TS = TS - 00001_8

Switch bit 1 of channel 11 to 1 (ISS warning lamp)

Return

If bit 1 of IMODES33 = 0, switch bit 1 of channel 11 to 0
(Extinguish ISS warning lamp if lamp test not in progress)

Return

GLOCKMON TS = 00000_8

If $|CDU_z| \leq$ K:70degs, proceed to "SETGLOCK"

Switch bit 6 of TS to 1

If $|CDU_z| \leq$ K:85degs, proceed to "SETGLOCK"

If bit 4 of channel 12 = 1, proceed to "SETGLOCK"
(already in coarse align)

Perform "SETCOARS"

Call "CA+ECE" in 0.06 seconds

SETGLOCK If bit 6 of $DSPTAB_{11}$ ≠ bit 6 of TS:

If bit 6 of $DSPTAB_{11}$ = 1: (bit 6 of TS = 0)

If bit 1 of IMODES33 = 1, return (lamp test)

If bit 6 of $DSPTAB_{11}$ = 0: (bit 6 of TS = 1)

If bit 6 of IMODES30 = 1, return

Invert bit 6 of $DSPTAB_{11}$ and flag for output
(Gimbal lock warning light on or off)

Return (to T4RUPT routine)

CA+ECE Switch bit 6 of channel 12 to 1 (Enable ICDU Error Counters)

End task

UNZ2 $\underline{CDU}$ = 0

Switch bits 4 and 5 of channel 12 to 0
(Disable coarse align mode, reset ICDU zero discrete)

Delay 10.24 seconds

Switch bits 3, 4, and 6 of IMODES30 to 0
(Enable ICDU and IMU fail monitors and reset IMU caged flag)

Switch bit 6 of IMODES33 to 0 (Enable DAP)

If FLAGWRD2 bit 15 (DRIFTFLG) = 0:

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

1dPIPADT = less significant half of TIMENOW

Perform "SETISSW"

Switch bit 15 of channel 12 to 0
(Switchover to normal operate mode should be complete)

Call "PFAILOK" in 4 seconds

End task

ITURNON2 Switch bit 7 of IMODES30 to 1 (initiate IMU turn-on delay)

RADMODES = 00102_8

Return

ENDTNON If bit 2 of IMODES30 = 1: (turn-on sequence failure)

Switch bit 2 of IMODES30 to 0

If bit 14 of IMODES30 = 0: (turn-on delay still in effect)

Delay 90 seconds

Proceed to "ENDTNON"

If FLAGWRD0 bit 8 (IMUSE) = 1, proceed to "IMUBAD"

End task

Switch bit 15 of channel 12 to 1
(Switch ISS to normal operate mode)

Switch bit 4 of $DSPTAB_{11}$ to 0 and flag for output
(Turn off no attitude lamp)

Proceed to "UNZ2"

PFAILOK If bit 6 of IMODES30 = 1, end task (IMU caged)

Switch bit 10 of IMODES30 to 1 (reset PIPA fail bit)

Switch bit 13 of IMODES33 to 1 (reset PIPA fail bit)

Switch bit 5 of IMODES30 to 0 (Enable secondary PIPA fail mon.)

Perform "SETISSW"

End task

IMUPULSE $MPAC_5$ = TS

If bit 6 of IMODES30 = 1: (IMU caged)

IMUCADR = -0

Return

If LGYRO = 0: (gyro free for torquing)

Switch bit 6 of channel 14 to 1 (Enable gyro torquing)

TSt = 0.04

If LGYRO > 0: (gyro already being torqued)

Put this job to sleep

When awakened, continue at next step if LGYRO then = 0, or proceed to previous step if LGYRO is still > 0.

TSt = 0.01

Call "STRTGYRO" in TSt seconds

LGYRO = $MPAC_5$

GYRODEX = 0

Force sign agreement within each component of $\underline{E}_{LGYRO}$

Return

STRTGYRO Switch bits 7, 8, 9 and 10 of channel 14 to 0 (reset gyro select discretes, sign bit, gyro activity bit)

If bit 6 of IMODES30 = 1, proceed to "IMUBAD"

STRTGYR2 If GYRODEX = 3: (finished)

LGYRO = 0

Awaken any job put to sleep in "IMUPULSE"

Proceed to "IMUFINED"

If GYRODEX = 2: (torque X gyro about its output axis)

LGYRO = LGYRO - 4 (index X component)

If GYRODEX = 1: (torque Z gyro)

LGYRO = LGYRO + 2 (index Z component)

If GYRODEX = 0: (torque Y gyro first)

LGYRO = LGYRO + 2 (index Y component)

GYRODEX = GYRODEX + 1

$TS_{dp} = E_{LGYRO}$ (X, Y or Z component of vector specified at input)

If $|TS_{dp}| <$ K:gyromin, proceed to "STRTGYR2"

$TS_{dp} = TS_{dp}$ + K:GYROFRAC signTS_{dp}

If $TS_{dp} < 0$, switch bit 9 of channel 14 to 1 (negative torque)

If GYRODEX = 1, switch bit 8 of channel 14 to 1 (Y)

If GYRODEX = 2, switch bits 7 and 8 of channel 14 to 1 (Z)

If GYRODEX = 3, switch bit 7 of channel 14 to 1 (X)

RUPTREG2 = fractional part of $|TS_{dp}|$ sgnTS_{dp} (bits 1-7 of LS half)

TScmd = 8192 (fractional part of((TS_{dp} - RUPTREG2) / 8192))

RUPTREG1 = $|TS_{dp}$ - TScmd$|$

If RUPTREG1 $\leq$ 8192: (equals 0 or 8192)

If RUPTREG1 $\neq$ 0, TScmd = TScmd + 8192

E_{LGYRO} = RUPTREG2 (portion of command less than one pulse)

GYROCMD = TScmd

TSt = K:gyrtm GYROCMD + 0.03

Call "STRTGYRO" in TSt seconds

Proceed to "GYROEXIT"

E_{LGYRO} = RUPTREG1 - 16384 + RUPTREG2

GYROCMD = 8192 + TScmd

TSt = K:gyrtm GYROCMD - 0.03

Call "8192AUG" in TSt seconds

Proceed to "GYROEXIT"

8192AUG If bit 6 of IMODES30 = 1, proceed to "IMUBAD" (IMU caged)

If bit 4 of channel 12 = 1, proceed to "IMUBAD"
(coarse align enabled; disables gyro torquing)

TS = E_{LGYRO} rounded off to nearest multiple of 8192
(E_{LGYRO} contains multiples of 8192 plus a fraction of one pulse; fraction is ignored)

If TS = 0:

GYROCMD = GYROCMD + 8192

TSt = K:gyrtm GYROCMD + 0.03

Call "STRTGYRO" in TSt seconds

Proceed to "GYROEXIT"

$E_{LGYRO} = E_{LGYRO} - 8192$

GYROCMD = GYROCMD + 8192

TSt = K:gyrtm GYROCMD - 0.03

Call "8192AUG" in TSt seconds

GYROEXIT Switch bit 10 of channel 14 to 1 (send GYROCMD)

End task

IMUZERO Inhibit interrupts

If bits 4 and 6 of $DSPTAB_{11}$ both = 1:
(No attitude and "Gimbal Lock" lamps both on)

Perform "ALARM" with TS = 00206_8

IMUCADR = -0

Release interrupt inhibit

Return

If bit 6 of IMODES30 = 1: (IMU caged)

IMUCADR = -0

Release interrupt inhibit

Return

Switch bits 5 and 6 of IMODES33 to 1
(Indicate zeroing in progress; disable DAP)

Switch bits 3 and 4 of IMODES30 to 1
(Inhibit ICDU and IMU fail monitors)

Switch bits 4 and 6 of channel 12 to 0
(Disable Coarse align mode and ICDU Error Counters)

Switch bit 4 of $DSPTAB_{11}$ to 0 and flag for output
(Turn off "no attitude" lamp)

Switch bit 5 of channel 12 to 1
(Zero ICDU's)

CDU = 0

Call "IMUZERO2" in 0.32 second

If bit 9 of IMODES30 = 1: (IMU not operating)

Perform "ALARM" with TS = 00210_8

Release interrupt inhibit

Return

IMUZERO2 If bit 6 of IMODES30 = 1, proceed to "IMUBAD" (IMU caged)

CDU = 0

Switch bit 5 of channel 12 to 0 (Release ICDU's)

Delay 10.24 seconds

If bit 6 of IMODES30 = 1, proceed to "IMUBAD"

Switch bits 4 and 3 of IMODES30 to 0
(Remove IMU and ICDU fail monitor inhibit bits)

Switch bits 6 and 5 of IMODES33 to 0
(Enable DAP and reset zeroing indication)

Perform "SETISSW"

Proceed to "ENDIMU"

IMUCOARS Inhibit interrupts

If bit 6 of IMODES30 = 1: (IMU caged)

IMUCADR = -0

Release interrupt inhibit

Return

Perform "SETCOARS"

Call "COARS" in 0.06 second

Release interrupt inhibit

Return

SETCOARS — If bit 4 of channel 12 = 1, return (already in coarse align)

Switch bit 6 of channel 12 to 0 (disable ICDU error counters)

Switch bit 10 of channel 14 to 0 (disable gyro torque pulses)

GYROCMD = 0 (-0)

Switch bit 4 of channel 12 to 1 (Switch ICDU to coarse align)

Switch bit 4 of $DSPTAB_{11}$ to 1 and flag for output (turn on "No attitude" lamp)

Switch bit 6 of IMODES33 to 1 (disable DAP)

Switch bit 4 of IMODES30 to 1 (inhibit IMU fail monitor)

RNDREFDR — Switch FLAGWRD1 bit 5 (TRACKFLG) to 0

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 0

Switch FLAGWRD3 bit 13 (REFSMFLG) to 0

Return

COARS — If bit 6 of IMODES30 = 1, proceed to "IMUBAD" (caged)

Switch bit 6 of channel 12 to 1 (Enable ICDU Error Counters)

$COMMAND_z = THETAD_z - CDU_z$

$COMMAND_y = THETAD_y - CDU_y$ converted to one's complement form and rounded off

$COMMAND_x = THETAD_x - CDU_x$

Delay 0.02 second

COARS2 — If bit 6 of IMODES30 = 1, proceed to "IMUBAD" (caged)

i = 0

Perform the indented steps 3 times, for j = z, then y, then x

If $|COMMAND_j| = 0$, $CDU_jCMD = -0$

If $|COMMAND_j| \leq K{:}commax$, but $\neq 0$:

$CDU_jCMD = COMMAND_j$

$COMMAND_j = 0$

$i = i + 1$

If $|COMMAND_j| > K{:}commax$:

$CDU_jCMD = K{:}commax\ signCOMMAND_j$

$COMMAND_j = COMMAND_j - CDU_jCMD$

$i = i + 1$

(End of "indented steps")

If $i > 0$: (command is not zero)

Switch bits 13, 14 and 15 of channel 14 to 1
(Send output pulses to ICDU Error Counters from CDU_jCMD cells)

Delay 0.6 second

Proceed to "COARS2"

Delay 1.5 seconds

Perform the indented steps 3 times (or until "ALARM" situation is encountered), for j = z, then y, then x

$TS = CDU_j - THETAD_j$ converted to one's complement form

If $|TS| > K{:}COARSTOL$: (coarse align error)

Perform "ALARM" with $TS = 00211_8$

Proceed to "IMUBAD"

(End of "indented steps")

Proceed to "ENDIMU"

IMUFINE Inhibit interrupts

If bit 6 of IMODES30 = 1: (IMU caged)

IMUCADR = -0

Release interrupt inhibit

Return

Switch bits 4 and 5 of channel 12 to 0
(Reset ICDU coarse align and zeroing discretes)

Switch bit 6 of IMODES33 to 0 (Enable DAP)

Switch bit 4 of $DSPTAB_{11}$ to 0 and flag for output
(Turn off "No Attitude" lamp)

Call "IFAILOK" in 5.12 seconds

Call "IMUFINED" in 2 seconds

Release interrupt inhibit

Return

IMUFINED If bit 6 of IMODES30 = 1, proceed to "IMUBAD" (caged)

Proceed to "ENDIMU"

IFAILOK If bit 6 of IMODES30 = 1, end task

If bit 4 of channel 12 = 1, end task
(Coarse align mode)

Switch bit 13 of IMODES30 to 1 (reset IMU fail bit)

Switch bit 4 of IMODES30 to 0 (enable IMU fail monitor)

Perform "SETISSW"

End task

IMUSTALL Inhibit interrupts

If IMUCADR > 0 or if IMUCADR < -1:

TS1 = Return address of routine calling "IMUSTALL"

Proceed to "BAILOUT1" with TS = 31210_8

If IMUCADR = -1:   (operation already complete and good)

    IMUCADR = +0

    Release interrupt inhibit

    ISSGOOD = 1

    Return

If IMUCADR = -0:   (operation already complete and bad)

    IMUCADR = +0

    Release interrupt inhibit

    ISSGOOD = 0

    Return

IMUCADR = return address   (to caller of "IMUSTALL")

Put present job to sleep

When awakened, return via LOC

<u>ENDIMU</u> If bit 1 of channel 11 = 1, proceed to "IMUBAD"   (ISS bad)

<u>IMUGOOD</u> If IMUCADR = +0:   ("IMUSTALL" not entered yet)

    IMUCADR = -1

    End task

LOC = IMUCADR

ISSGOOD = 1

Wake job put to sleep in "IMUSTALL"

IMUCADR = +0

End task

<u>IMUBAD</u> If IMUCADR = +0:

    IMUCADR = -0

    End task

LOC = IMUCADR

ISSGOOD = 0

Wake job put to sleep in "IMUSTALL"

IMUCADR = +0

End task

ATTCK2 CDU_zCMD = $THETAD_z$ K:ONETENTH

CDU_yCMD = $THETAD_y$ K:ONETENTH

CDU_xCMD = $THETAD_x$ K:ONETENTH

Switch bits 13, 14 and 15 of channel 14 to 1 (send CDU_iCMD's)

End task

RO2BOTH If FLAGWRD3 bit 13 (REFSMFLG) = 1:

Switch FLAGWRD0 bit 8 (IMUSE) to 1

Return

If bit 9 of IMODES30 = 1: (IMU not operating)

Perform "ALARM" with TS = 00210_8

If bit 9 of IMODES30 = 0: (REFSMMAT invalid)

Perform "ALARM" with TS = 00220_8

Proceed to "GOTOPOOH"

1dPIPADT: Single precision time interval for application of PIPA biases and gyro drift compensation, scaled B8, or storage for present time for the purpose of computing that time interval, scaled B14, in units of centiseconds.

ADIAX: Single precision angular drift of the X gyro around its output axis caused by linear acceleration of the IMU in the direction of the X gyro input axis (+XSM), scaled B-6 in units of gyro pulses / centimeters per second. (One gyro pulse corresponds to 2^{-21} revolutions.) Pad loaded.

ADIAY: Single precision angular drift of the Y gyro around its output axis caused by linear acceleration of the IMU in the direction of the Y gyro input axis (+YSM), scaled B-6 in units of gyro pulses / centimeters per second. Part of the erasable load.

ADIAZ: Single precision angular drift of the Z gyro around its output axis caused by linear acceleration of the IMU in the direction of the Z gyro input axis (+ZSM), scaled B-6 in units of gyro pulses / centimeters per second. Part of the erasable load.

ADSRAX: Single precision angular drift of the X gyro around its output axis caused by linear acceleration of the IMU in the direction of the X gyro spin-reference axis (-YSM), scaled B-6 in units of gyro pulses / centimeters per second. Part of the erasable load.

ADSRAY: Single precision angular drift of the Y gyro around its output axis caused by linear acceleration of the IMU in the direction of the Y gyro spin-reference axis (-ZSM), scaled B-6 in units of gyro pulses / centimeters per second. Part of the erasable load.

ADSRAZ: Single precision angular drift of the Z gyro around its output axis caused by linear acceleration of the IMU in the direction of the Z gyro spin-reference axis (+YSM), scaled B-6 in units of gyro pulses / centimeters per second. Part of the erasable load.

CDU (CDU_x, CDU_y, CDU_z): Single precision vector containing the measured values of the IMU gimbal angles (outer, inner and middle gimbal in X, Y, and Z components, respectively), scaled B-1 in units of revolutions and stored in two's complement form. Each component is an LGC input counter incremented directly from the Coupling Data Unit in response to changes in the IMU gimbal angles.

CDU_iCMD: (i = x,y or z): **Three single precision counters scaled B1 in units of revolutions, gated to the ICDU Error Counters by setting bits 15, 14** and 13 of channel 14. Bits 15-13 reset when respective counters reach -0. Scaling is B14 in units of pulses.

COMMAND: Temporary storage for changes to the three gimbal angles during coarse alignment, scaled B1 in units of revolutions.

DELV: See SERV section.

$DSPTAB_{11}$: See INTR section.

$\underline{E}_{LGYRO}$: Double precision vector containing three desired gyro torque angles whose address is specified at entry to the IMUPULSE routine, scaled B0 in units of revolutions, or B21 in units of pulses. $\underline{E}_{LGYRO}$ = $\underline{G}$COMP or (OGC,IGC,MGC)

$\underline{G}$COMP: Double precision vector containing required gyro compensation angles, scaled B14 (or B21) in units of gyro pulses (1 gyro pulse = 2^{-21} revolutions).

GCOMPSW: Single precision switch indicating whether gyro compensation is required or inhibited, scaled B14 and unitless.

GYROCMD: Computer cell counted down as torquing pulses are sent to one of the gyros, scaled B14 in units of gyro pulses. Used for commands to all three gyros; the pulse train is initiated by setting bit 10 of Channel 14 and it is routed to the appropriate gyro torque motor by the setting in bits 7 and 8 of Channel 14.

GYRODEX: An index equivalent to that maintained by the program in bits 15-13 of LGYRO; used to indicate which gyro is being torqued and assigned a separate label merely for convenience in functional representation.

i,j: Single precision index registers, scaled B14 and unitless.

IMODES30: Single precision flagword whose individual bits have the following meanings:

Bit 15	(1) IMU temperature not within prescribed limits (0) IMU temperature within limits
Bit 14	(1) ISS turn-on delay not in effect (0) ISS turn-on delay initiated and in effect
Bit 13	(1) IMU good (0) IMU fail
Bit 12	(1) ICDU good (0) ICDU fail
Bit 11	(1) IMU not externally caged (0) IMU caged, externally
Bit 10	(1) PIPA good (0) PIPA fail (identical to bit 13 of IMODES33)
Bit 9	(1) IMU off (0) IMU operating
Bit 8	(1) IMU turn-on delay in progress (0) IMU turn-on delay complete or not initiated
Bit 7	(1) IMU turn-on delay initiate (0) IMU turn-on delay not initiated
Bit 6	(1) IMU caged (Internally) (0) IMU not caged

Bit 5 (1) Secondary PIPA fail monitor inhibited
(0) Secondary PIPA fail monitor enabled

Bit 4 (1) IMU fail monitor inhibited
(0) IMU fail monitor enabled

Bit 3 (1) ICDU fail monitor inhibited
(0) ICDU fail monitor enabled

Bit 2 (1) ISS turn-on sequence failure
(0) No ISS turn-on sequence failure in effect

Bit 1 (1) Primary PIPA fail monitor inhibited
(0) Primary PIPA fail monitor enabled

IMODES33: See INTR section.

IMUCADR: Single precision octal storage for address to return to program that is making use of the ISS and waiting for a particular operation to be accomplished.

ISSGOOD: Variable introduced as a substitute for variable return address; set to 1 or 0 to indicate whether an IMU mode switch was successfully completed (1) or not (0).

K:70degs: Single precision constant stored as -0.38888, program notation "-70DEGS," scaled B-1 in units of revolutions. Equation value: +0.19444. (Equivalent to +69.9984 degrees).

K:85degs: Single precision constant stored as -0.38888 + -0.08333, scaled B-1 in units of revolutions. Equation value: +0.23610. (Equivalent to +84.99 degrees.)

K:COARSTOL: Single precision constant stored as -0.01111, scaled B-1 in units of revolutions. Equation value: 0.005555. (Equivalent to +1.9998 degrees.)

K:commax: Single precision constant stored as -191×2^{-14} and -192×2^{-14}, program notations "-COMMAX" and "-COMMAX-", scaled B1 in units of revolutions. Equation value: +0.0234375. (Equivalent to +8.4375 degrees or half the mechanical limit of the ICDU Error Counter.)

K:GYROFRAC: Double precision constant stored as 0.215×2^{-21}, scaled B21 in units of gyro torque pulses. Equation value: 0.21875. (The closest approximation to 0.215 with a least increment of 0.0078125.)

K:gyromin: Single precision constant stored as 77601_8, program notation "-GYROMIN," scaled B7 in units of gyro torque pulses. Equation value: 1.0. (1 gyro pulse is equivalent to 2^{-21} revolutions.)

K:gyrtm: Single precision constant stored as 01000_8, program notation "BIT10," scaled B0 in units of centiseconds/gyro torquing pulse. Equation value: 1 / 3200 seconds/pulse.

K:ONETENTH: See DAPA section.

K:pip2sec: Single precision constant stored as 31000_8, program notation "PRIO31", scaled B8 in units of centiseconds. Equation value: 200.

LGYRO: Single precision octal address (positive) of cell containing gyro torquing command.

LOC: See MATX section.

MPAC: See DINT section.

NBDX, NBDY: Single precision angular drift around the output axes of the X and Y gyros caused by the passage of time, scaled B-5 in units of gyro pulses per centisecond.

NBDZ: Single precision complement of the drift around the output axis of the Z gyro caused by the passage of time, scaled B-5 in units of gyro pulses per centisecond.

$\underline{P}$IPA: Single precision sensed-change-in-velocity vector scaled B14 in units of centimeters per second, expressed in stable member (IMU) coordinates. The three components are incremented directly from the Pulse-Integrating, Pendulous Accelerometers on the stable member of the IMU.

$PIPABIAS_x$, $PIPABIAS_y$, $PIPABIAS_z$: Single precision bias factors for the X, Y and Z PIPA's, scaled B-3 in units of centimeters per second per centisecond (equivalent to PIPA counts per centisecond).

$PIPASCF_x$, $PIPASCF_y$, $PIPASCF_z$: Single precision scale factor errors associated with the X, Y and Z PIPA's respectively, scaled B-9 and unitless (accelerometer counts per accelerometer count).

PIPTIME: See SERV section.

RADMODES: See RADR section.

RUPTREG1: Single precision storage for portions of gyro torque commands greater than or equal to 2^{13} pulses, scaled B27 in units of gyro pulses (2^{-21} revolutions).

RUPTREG2: Single precision storage for portions of gyro torque command less than one pulse, scaled B7 in units of gyro torque pulses. Fractional values of commands are not issued. They are stored for addition to future commands.

THETAD: Single precision vector containing the gimbal angles that define a desired orientation between the IMU and the spacecraft for attitude maneuvers or IMU alignment, scaled B-1 in units of revolutions and stored in two's complement form.

TIMENOW: See EXVB section.

INTR

Program Interrupts

#1 Caused by underflow of counter TIME6, starting address 4004_8

Proceed to "DOT6RUPT"

#2 Caused by overflow of counter TIME5, starting address 4010_8

Proceed to "T5RUPT"

#3 Caused by overflow of counter TIME3, starting address 4014_8

Proceed to "T3RUPT" (start scheduled task)

#4 Caused by overflow of counter TIME4, starting address 4020_8

Proceed to "T4RUPT"

#5 Caused by depression of a key on the DSKY keyboard, starting address 4024_8

Proceed to "KEYRUPT1"

#6 Caused by depression of mark or reject buttons or crew indication of change in desired descent rate, starting address 4030_8

Proceed to "MARKRUPT"

#7 Caused by overflow of uplink serial input buffer, starting address 4034_8

Proceed to "UPRUPT"

#8 Caused by end pulse from the downlink system, starting address 4040_8

Proceed to "DODOWNTM"

#9 Caused by expiration of time delay (90-100 ms) after bit 4 of channel 13 is set, starting address 4044_8

Proceed to "RADAREAD"

#10 Caused by input from the rotational hand controller, starting address 4050_8

Proceed to "PITFALL"

#11 Caused by hardware difficulties, starting address 4000_8

Proceed to "GOPROG"

T4RUPT — If DSRUPTSW < 0, proceed to "QUIKDSP"

If DSRUPTSW = 0, DSRUPTSW = 8

DSRUPTSW = DSRUPTSW - 1

TSruptsw = DSRUPTSW

If bit 15 of $DSPTAB_{11}$ = 1: (flagged for output)

- Switch bits 15-12 of $DSPTAB_{11}$ to 0
- OUTO = $DSPTAB_{11} + 60000_8$
- DSRUPTSW = DSRUPTSW - 8192 - (5)(256)
- Set TIME4 to cause "T4RUPT" in 20 milliseconds
- Proceed to "PROCEEDE"

If FLAGWRD5 bit 15 (DSKYFLAG) = 0 or if NOUT = 0:

- OUTO = 00000_8
- Set TIME4 to cause "T4RUPT" in 120 milliseconds
- Proceed to "PROCEEDE"

NOUT = NOUT - 1

TS = -0

Perform "DSPSCAN"

If SENTCODE = 0:

- OUTO = 00000_8
- Set TIME4 to cause "T4RUPT" in 120 milliseconds
- Proceed to "PROCEEDE"

DSRUPTSW = DSRUPTSW - 8192 - (5)(256)

Set TIME4 to cause "T4RUPT" in 20 milliseconds

Proceed to "PROCEEDE"

PROCEEDE If bit 14 of channel 32 ≠ bit 14 of IMODES33: (proceed button)

Set bit 14 of IMODES33 = bit 14 of channel 32

If bit 14 of IMODES33 = 0: (proceed button just pushed)

Establish "PROCKEY" (pr30)

If TSruptsw = 0 or 4:

Perform "RCSMONIT"

If TSruptsw = 1 or 5:

Perform "RRAUTCHK"

Perform "DAPT4S"

If TSruptsw = 2 or 6:

Perform "IMUMON"

Perform "GLOCKMON"

If TSruptsw = 3 or 7:

Perform "DAPT4S"

Resume

PROCKEY If $|22 - \text{VERBREG}| \leq 1$:

Proceed to "CHARALRM"

REQRET = 0

DSPCOUNT = -19

Proceed to "VBPROC"

QUIKDSP If DSRUPTSW $<$ -8192: (bit 14 = 0)

  OUTO = 00000_8

  Set TIME4 to cause "T4RUPT" in 20 milliseconds

  DSRUPTSW = DSRUPTSW + 256 + 8192

  Resume

If NOUT = 0:

  OUTO = 00000_8

  Proceed to "SYNCT4"

NOUT = NOUT - 1

TS = -0

Perform "DSPSCAN"

If SENTCODE = 0: (no display to be changed)

  OUTO = 00000_8

  Proceed to "SYNCT4"

Set TIME4 to cause "T4RUPT" in 20 milliseconds

DSRUPTSW = DSRUPTSW - 8192 + 256

Resume

DSPSCAN If bit 15 of $DSPTAB_{DSPCNT}$ = 1: (negative to flag for output)

  $DSPTAB_{DSPCNT} = |DSPTAB_{DSPCNT}|$

  OUTO = bits 15-11 of $K{:}RELTAB_{DSPCNT}$ + bits 11-1 of $DSPTAB_{DSPCNT}$

  SENTCODE = 1

  Return

If DSPCNT $>$ 0:

  DSPCNT = DSPCNT - 1

  Proceed to "DSPSCAN"

```
        If TS = -0:      (first time through list)
            TS = +0
            DSPCNT = 10
            Proceed to "DSPSCAN"
        NOUT = 0
        SENTCODE = 0
        Return
SYNCT4  TS = 20
        DSRUPTSW = DSRUPTSW + 256
        If DSRUPTSW < 0:
            TS = TS + 20
            Proceed to second step of "SYNCT4"
        Set TIME4 to call "T4RUPT" in TS milliseconds
        Resume
```

DSPCNT: Single precision permanent index, scaled B14 and unitless.

DSPCOUNT: See DSKY section.

$DSPTAB_i$ (i = 0-10): See DSKY section.

$DSPTAB_{11}$: Single precision flagword whose bits designate relays to be set to illuminate lamps on the DSKY. Bit 9 lights the "program alarm" (PROG) lamp when set; bit 8 lights the "tracker fail" (TRACKER) lamp when set; bit 6 lights the "gimbal lock warning" (GIMBAL LOCK) lamp when set; bit 5 lights the "LR altitude fail" lamp when set; bit 4 lights the "no attitude" (NO ATT) lamp when set; and bit 3 lights the "LR velocity fail" lamp when set.

DSRUPTSW: Single precision index used to cycle through the display and monitor functions of "T4RUPT," scaled B14 and unitless.

IMODES33: Single precision flagword whose individual bits have the following significance: (Bits 15, 10, 9, and 4-2 have no significance.)

Bit 14	(1)	Proceed button not depressed during last "T4RUPT" cycle
	(0)	Proceed button just pushed
Bit 13	(1)	PIPA good
	(0)	PIPA fail
Bit 12	(1)	Downlink not too fast
	(0)	Downlink too fast
Bit 11	(1)	Uplink not too fast
	(0)	Uplink too fast
Bit 8	(1)	Inertial data just displayed
	(0)	Inertial data not displayed
Bit 7	(1)	Display altitude
	(0)	Display altitude rate
Bit 6	(1)	DAP disable
	(0)	DAP enabled
Bit 5	(1)	ICDU zeroing (See IMUC section)
	(0)	ICDU not zeroing
Bit 1	(1)	Lamp test in progress
	(0)	Lamp test not in progress

K:RELTAB$_i$ (i = 0-10): Table of **twelve** single precision constants containing the routing codes for OUTO in bits 15-12. See DSKY section.

NOUT: Single precision count of the number of DSPTAB registers (excluding DSPTAB$_{11}$) to be output as soon as possible, scaled B14 and unitless.

OUTO: Single precision output register which routes an eleven-binary-bit display relay code (in bits 11-1) according to the routing code in bits 15-12. OUTO remains set for only twenty milliseconds before being reset to 00000_8. (Equivalent to channel 10.)

REQRET: See DSKY section.

SENTCODE: Variable quantity introduced as a substitute for a variable return address, switched to 1 if a relay code is inserted in OUTO and switched to 0 if no code is inserted in OUTO.

TIME3: Single precision counter incremented every 10 milliseconds (every centisecond) in phase with the computer clock (TIMENOW). Set under program control (to 16384 minus the required time delay in centiseconds) to cause program interrupt #3 (when TIME3 overflows) and initiate the first task in the waitlist. (A task is a routine initiated at a specified time by a "Call" instruction.)

TIME4: Single precision counter incremented every 10 milliseconds (every centisecond), 7.5 milliseconds after the TIME3 increment. Set under program control (to 16384 minus the required time delay in centiseconds) to cause program interrupt #4.

TIME5: Single precision counter incremented every 10 milliseconds (every centisecond) in phase with TIME3. Set under program control (to 16384 minus the required time delay in centiseconds) to cause program interrupt #2 (see DAPA section).

TIME6: Single precision counter decremented every 0.625 millisecond when bit 15 of channel 13 is set. Set under program control (to required delay in units of 0.0625 centiseconds) to cause program interrupt #1 (see DAPA section).

VERBREG: See DATA section.

MATX

Sine, Cosine

The routines "COSINE" and "SINE" are those normally used to calculate the cosine or sine of an angle. The angle is input in the multiple precision accumulator (MPAC) in one's complement form, scaled B0 in units of revolutions. The output is a cosine or a sine in MPAC, scaled B1 and unitless.

COSINE $MPAC_{dp} = \frac{1}{4} - |MPAC_{dp}|$ ($\frac{1}{4}$ corresponds to 90 degrees)

SINE If $|MPAC_{dp}| \geq \frac{1}{2}$:

$$MPAC_{dp} = \frac{1}{2}\,\text{sign}MPAC_{dp} - MPAC_{dp}$$

If $|MPAC_{dp}| \geq \frac{1}{4}$:

$$MPAC_{dp} = \frac{1}{2}\,\text{sign}MPAC_{dp} - MPAC_{dp}$$

$x = MPAC_{dp}$ (rescaled to B-1)

$TS = K{:}sn1\ x + K{:}sn3\ x^3 + K{:}sn5\ x^5 + K{:}sn7\ x^7 + K{:}sn9\ x^9$

$MPAC_{dp} = TS$ (rescaled from B3 to B1)

Return

Constant Coefficients

	Stored Value	Scaled	Equation Value
K:sn1	0.3926990796	B4	2π (= 6.28318527)
K:sn3	-0.6459637111	B6	$-(2\pi)^3$ 0.166666570
K:sn5	0.318758717	B8	$(2\pi)^5$ 0.00833302539
K:sn7	-0.074780249	B10	$-(2\pi)^7$ 0.000198074150
K:sn9	0.009694988	B12	$(2\pi)^9$ 0.00000260188699

Single Precision Cosine and Sine ($\cos_{sp}$ and $\sin_{sp}$)

The routines "SPCOS" and "SPSIN" are used to calculate the cosine or the sine of an angle when only single precision accuracy is required. They make use of the trigonometric identities $\cos(x) = \sin(x + \pi/2)$ and $\sin(x) = \sin(\pi - x)$ and of the Taylor series

$$\sin(x) = \sum_{i=0}^{n} (-1)^{i} \frac{x^{i+1}}{(i+1)!}$$

modified for maximum accuracy using only three terms. The input to both of the routines is an angle scaled B-1 in units of revolutions. The output is a cosine or sine scaled B0. Input and output are both through the single precision accumulator, denoted by A. Entry to the routine is made at "SPCOS" to calculate the cosine and at "SPSIN" to calculate the sine.

<u>SPCOS</u> $A = A + \frac{1}{4}$

<u>SPSIN</u> $X = A$

If $|X| \geq \frac{1}{2}$, $X = \frac{1}{2}$ signX $-$ X

If $|X| \geq \frac{1}{4}$:

$X = \frac{1}{2}$ sign X $-$ X

If $|X| = \frac{1}{4}$:

A = K:posmaxsp signX (K:posmaxsp = almost 1)

Return

TS = K:sn1sp X + K:sn3sp X^3 + K:sn5sp X^5

If $|TS| \geq 1$, TS = K:posmaxsp signTS

A = TS

Return

<u>Constant Coefficients</u>

	Stored Value	Scaled	Equation Value
K:sn1sp	0.7853134	B3	2π (= 6.2824)
K:sn3sp	-0.3216147	B7	$-(2\pi)^3$ 0.16596
K:sn5sp	0.0363551	B11	$(2\pi)^5$ 0.0076032

Arcsine, Arccosine

The routines "ARCSIN" and "ARCCOS" are those used to calculate the angle corresponding to a given sine or cosine. The input to the routine "ARCSIN" is a sine in MPAC, scaled B1. The output from "ARCSIN" is an angle between $-\frac{1}{4}$ and $+\frac{1}{4}$ that corresponds to the sine. The input to the routine "ARCCOS" is a cosine in MPAC, scaled B1. The output from "ARCCOS" is an angle between 0 and $\frac{1}{2}$ that corresponds to the cosine. The angle output is in MPAC and is scaled B0 in units of revolutions (one's complement form). ($\frac{1}{4}$ in units of revolutions is equivalent to 90 in units of degrees, etc.)

ARCSIN   Perform "ARCSUB"

$MPAC_{dp} = \frac{1}{4} - TS$

Return

ARCCOS   Perform "ARCSUB"

$MPAC_{dp} = TS$

Return

ARCSUB   If $|MPAC_{dp}| < 2^{-27}$:

$TS = \frac{1}{4}$

Return

If $MPAC_{dp} > 0$, PorM = 0

If $MPAC_{dp} < 0$, PorM = 1

$X = |MPAC_{dp}|$

If $X \geq 1.000244\ (1 + 2^{-12})$, or if the less significant half of X is negative and $X > 1.000122\ (1 + 2^{-13})$:

$TS1_{dp} = (LOC_o,\ BANKSET_o)$

Perform "ALARM1" with $TS = 01301_8$

If $X \geq 1$:

$TS = \frac{1}{2}$ PorM

Return

$TS = \sqrt{(1 - X) / 2}$

$TS = TS\ (K{:}as0 + K{:}as1\ X + Kas2\ X^2 + \ldots + K{:}as7\ X^7)$

If PorM = 1, $TS = \frac{1}{2} - TS$

Return

Constant Coefficients

	Stored Value	Scaled	Equation Value
K:as0	0.353553385	B0	$1.570796302 / \sqrt{2}\pi$
K:as1	-0.0483017006×2	B-1	$-0.214598801 / \sqrt{2}\pi$
K:as2	0.0200273085×2^2	B-2	$0.088978987 / \sqrt{2}\pi$
K:as3	-0.0112931863×2^3	B-3	$-0.050174305 / \sqrt{2}\pi$
K:as4	0.00695311612×2^4	B-4	$0.030891881 / \sqrt{2}\pi$
K:as5	$-0.00384617957 \times 2^5$	B-5	$-0.017088126 / \sqrt{2}\pi$
K:as6	$0.001501297736 \times 2^6$	B-6	$0.006670090 / \sqrt{2}\pi$
K:as7	$-0.000284160334 \times 2^7$	B-7	$-0.001262491 / \sqrt{2}\pi$

The numbers in the last column (excluding the factor of $1 / \sqrt{2}\pi$) agree closely with published Hastings series values.

Square Root

The "SQRT" routine computes the double precision square root of a triple precision number with variable scale factor i. The input is a triple precision scalar in MPAC with scale factor i. The output is a double precision scalar in MPAC with scale factor $\frac{1}{2}$ i. If i is an odd number, the output will have to be divided by the square root of 2 to reduce its scale factor to an integer.

SQRT i = scale factor of $MPAC_{tp}$

$X = MPAC_{tp} / 2^i$

If $X \leq 0$:

If $X < -2^{-14}$, proceed to "POODOO1" with $TS = 21302_8$ and $TS1_{dp} = (LOC_o, BANKSET_o)$

Return with $MPAC_{dp} = 0$

If $X < \frac{1}{4}$, repeat the indented steps until $X \geq \frac{1}{4}$

$i = i - 2$

$X = 4X$

(X now double precision and between $\frac{1}{4}$ and 1)

TS_{sp} = more significant half of X

If $X \geq \frac{1}{2}$, $BUF_{sp} = 0.5884\ TS_{sp} + 0.4192$

If $X < \frac{1}{2}$, $BUF_{sp} = 0.8324\ TS_{sp} + 0.2974$

$BUF_{sp} = \frac{1}{2} BUF_{sp} + \frac{1}{2} TS_{sp} / BUF_{sp}$

$TS_{dp} = \frac{1}{2} BUF_{sp} + \frac{1}{2} X / BUF_{sp}$

If $TS_{dp} \geq 1$, $TS_{dp} = 1 - 2^{-28}$

$i = \frac{1}{2} i$

$MPAC_{dp} = 2^i\ TS_{dp}$

Return

Natural Logarithm (complemented)

The routine "LOGSUB" is used to calculate the natural logarithm of a scalar using the Taylor series approximation

$$\ln(x) = -\sum_{i=1}^{\infty} \frac{(1-x)^i}{i}$$

truncated at i = 7 and modified for maximum accuracy in the domain $\frac{1}{2} \leq x < 1$. The input to the routine is the argument of the function, scaled B0 and stored in MPAC (double precision). The output is the complement of the natural logarithm of the input, scaled B5 and stored in MPAC (double precision).

LOGSUB $i = 0$

$X = MPAC_{dp}$

If $X < \frac{1}{2}$, repeat the indented steps until $X \geq \frac{1}{2}$

$i = i - 1$

$X = 2X$

(X now between $\frac{1}{2}$ and 1)

$x = 1 - X - 2^{-28}$

$TS_{dp} = K{:}ln0 + K{:}ln1\ x + K{:}ln2\ x^2 + \ldots + K{:}ln7\ x^7 + i\ K{:}cln2$

$MPAC_{dp} = -TS_{dp}$

Return

Constant Coefficients

	Stored Value	Scaled	Equation Value
K:ln0	0.0000000060	B5	0.0000001920
K:ln1	-0.0312514377	B5	-1.0000460064
K:ln2	-0.0155686771	B5	-0.4981976672
K:ln3	-0.0112502068	B5	-0.3600066176
K:ln4	-0.0018545108	B5	-0.0593443456

	Stored Value	Scaled	Equation Value
K:ln5	-0.0286607906	B5	-0.9171452992
K:ln6	0.0385598563	B5	1.2339154016
K:ln7	-0.0419361902	B5	-1.3419580864
K:cln2	0.0216608494	B5	0.6931471808

Conversion from Two's Complement Form to One's Complement Form

Conversion from two's complement form into one's complement form is accomplished using the "CDULOGIC" routine. The input is a single precision angle in two's complement form, stored in the multiple precision accumulator MPAC and scaled B-1 in units of revolutions. The output is a double precision angle in one's complement form, stored in MPAC and scaled B0 in units of revolutions.

CDULOGIC If $MPAC_{sp} \geq 0$: (one's complement binary equivalent to two's complement binary)

$MPAC_{dp} = MPAC_{sp}$

(shifted right one to change scaling from B-1 to B0)

If $MPAC_{sp} < 0$: (one's complement binary equivalent to two's complement binary minus one least significant bit)

$MPAC_{dp} = MPAC_{sp}$

(shifted right one to change scaling from B-1 to B0; minus 2^{-15}, the value of one least significant bit)

Return

Conversion from One's Complement Form to Two's Complement Form

Conversion from one's complement form to two's complement form is accomplished using one of the three routines "1STO2S", "2V1STO2S" or "V1STO2S" and their common subroutine "1TO2SUB", depending on whether the quantity to be converted has one, two or three components. The input to "1TO2SUB" is a double precision angle in one's complement form, stored in MPAC and scaled B0 in units of revolutions. The output from "1TO2SUB" is a single precision angle in two's complement form, stored in MPAC and scaled B-1 in units of revolutions.

1STO2S Perform "1TO2SUB"

Return

2V1STO2S Perform "1TO2SUB" with $MPAC_{dp} = TS_x$

$TS_x = MPAC_{sp}$

Perform "1TO2SUB" with $MPAC_{dp} = TS_y$

$TS_y = MPAC_{sp}$

Change interpretive mode to double precision (two single precision components stored as one double precision scalar)

Return

V1STO2S Perform "1TO2SUB" with $MPAC_{dp} = TS_x$

$TS_x = MPAC_{sp}$

Perform "1TO2SUB" with $MPAC_{dp} = TS_z$

$TS_z = MPAC_{sp}$

Perform "1TO2SUB" with $MPAC_{dp} = TS_y$

$TS_y = MPAC_{sp}$

Change interpretive mode to triple precision (three single precision components stored as one triple precision scalar)

Return

1TO2SUB If $MPAC_{dp} \geq 0$: (two's complement binary equivalent to one's complement binary)

$MPAC_{sp} = MPAC_{dp}$

(shifted left one to change scaling from B0 to B-1)

If $MPAC_{dp} = -0$: (there is only one "zero" in two's complement form and it is equivalent to the +0 in one's complement form)

$MPAC_{sp} = +0$

If $MPAC_{dp} < 0$: (two's complement binary equivalent to one's complement binary plus one least significant bit)

$MPAC_{sp} = MPAC_{dp}$

(shifted left one to change scaling from B0 to B-1; plus 2^{-15}, the value of one least significant bit in single precision two's complement form scaled B-1)

If overflow ($|MPAC_{sp}| \geq \frac{1}{2}$):

$MPAC_{sp} = MPAC_{sp} - 1 \text{ sign} MPAC_{sp}$

Return

Single Precision Arcsine Routine ($\arcsin_{sp}$)

The routine "SPARCSIN" is used to calculate an angle from a given sine of the angle when only single precision accuracy is required. The input to the routine is the sine of an angle (or cosine of an angle if the complement of the angle is desired), scaled B1 and unitless. The output is the angle scaled B-1 in units of revolutions. Input and output are both through the single precision accumulator, denoted by A.

SPARCSIN $A = 2\ A$ (rescales to B0)

If overflow (i.e. $|A| \geq 1$):

$A = \text{sign}A\ \text{K:posmaxsp}$

$$A = \text{K:DPL1}\ A / 2 + \text{K:DPL3}\ A^3 / 2 + \text{K:DPL5}\ A^5 / 2 + \text{K:DPL7}\ A^7 / 2 + \text{K:DPL9}\ A^9 / 2$$

Return (A contains 2 times the angle in revolutions i.e., scaled B-1)

Constant Coefficients

	Stored Value	Scaled	Equation Value
K:DPL1	0.64099121	B0	10502×2^{-14}
K:DPL3	0.02636718	B0	432×2^{-14}

	Stored Value	Scaled	Equation Value
K:DPL5	0.44555664	B0	7300×2^{-14}
K:DPL7	-0.72039794	B0	-11803×2^{-14}
K:DPL9	0.51251221	B0	8397×2^{-14}

DUMMYJB2 Release interrupt inhibit

Switch bit 2 of channel 11 to 0 (COMP ACT off)

ADVAN If NEWJOB = -0: (no jobs; do self test)

SUPERBNK = 4

EBANK = 2 (bits 3-1 of BBANK)

FBANK = 33_8 (bits 15-11 of BBANK)

Proceed to address specified in SELFRET
(return to self test routines where they left off)

If NEWJOB > 0: (A job has been awakened or established with job core set other than zero. Before the next step, an interrupt may establish or awaken a job of higher priority with job core zero; in that case, NEWJOB would be reset to +0 before next step.)

Inhibit interrupts

If NEWJOB > 0:

Switch bit 2 of channel 11 to 1 (COMP ACT on)

$TS_0 = LOC_0$

$TS_1 = BANKSET_0$

Proceed to "CHANJOB4"

Release interrupt inhibit

(Otherwise, NEWJOB = +0)

Switch bit 2 of channel 11 to 1

SUPERBNK = bits 7-5 of $BANKSET_0$

EBANK = bits 3-1 of $BANKSET_0$ (via BBANK)

FBANK = bits 15-11 of $BANKSET_0$ (via BBANK)

Proceed to address specified in LOC_0

<u>FINDVAC</u> (Entered to "Establish" a job that requires working storage)

Inhibit interrupts

NEWPRIO = A (priority, octal, in bits 14-10 of accumulator)

$NEWLOC_{dp}$ = contents of double precision cell specified in Q-register
(stored in fixed memory after a "TC FINDVAC")

EXECTEM1 = FBANK

FBANK = 1 (note: SUPERBNK setting does not affect FBANK #1)

<u>FINDVAC2</u> Scan VACiUSE for i = 1-5 for an available VAC area
(If available, VACiUSE = its own address; if not, it = +0)

If none available, proceed to "BAILOUT1" with TS = 31201_8
and $TS1_{dp}$ = (Q-register, EXECTEM1)

NEWPRIO = NEWPRIO + VACiUSE + 1
(priority in bits 14-10; address of "1 + VACiUSE" in bits 9-1)

VACiUSE = +0

Proceed to "NOVAC2"

<u>SPVAC</u> (Entered with interrupts inhibited and priority in NEWPRIO to "Establish" a job that requires a working storage)

Q-register = Q-register - 2

$NEWLOC_{dp}$ = (A,L) (job starting address in accumulator instead of in fixed memory)

EXECTEM1 = FBANK

FBANK = 1

Proceed to "FINDVAC2"

<u>NOVAC</u> (Entered to "Establish" a job that requires no working storage)

Inhibit interrupts

NEWPRIO = A + "$MPAC_6$" - "QPRET"

$NEWLOC_{dp}$ = contents of double precision cell specified in Q-register

EXECTEM1 = FBANK

FBANK = 1

<u>NOVAC2</u> LOCCTR = 0

EXECTEM2 = 7

<u>NOVAC3</u> If $PRIORITY_{LOCCTR}$ = -0, proceed to "CORFOUND"

LOCCTR = LOCCTR + 12

If EXECTEM2 > 0:

EXECTEM2 = EXECTEM2 - 1

Proceed to "NOVAC3"

Proceed to "BAILOUT1" with TS = 31202_8 (No available job cores) and $TS1_{dp}$ = (Q-register, EXECTEM1)

<u>CORFOUND</u> $PRIORITY_{LOCCTR}$ = NEWPRIO

$PUSHLOC_{LOCCTR}$ = bits 9-1 of NEWPRIO

If LOCCTR > 0, proceed to "SETLOC"

$OVFIND = 0$

$FIXLOC = PUSHLOC_0$

SPECTEST If NEWJOB is negative non-zero or +0, proceed to "CCSHOLE"

If NEWJOB > 0, proceed to "SETLOC" (new job just established with job core zero, but job of higher priority awakened during same interrupt with job core other than zero)

$NEWJOB = +0$

LOC_0 = more significant half of NEWLOC

$BANKSET_0$ = less significant half of NEWLOC

FBANK = EXECTEM1

Return to 2 + address in Q-register

SETLOC LOC_{LOCCTR} = more significant half of NEWLOC

$BANKSET_{LOCCTR}$ = less significant half of NEWLOC

$TS = PRIORITY_{NEWJOB}$

If TS < NEWPRIO, NEWJOB = LOCCTR (this job will be started at the next entry to "CHANJOB4" if NEWJOB is set here)

FBANK = EXECTEM1

Return to 2 + address in Q-register

CHANG1 (Entered from jobs programmed in "basic" language to check for jobs of higher priority)

TS_0 = Q-register

TS_1 = FBANK + EBANK (FBANK in bits 15-11; EBANK in bits 3-1)

EBANK = 2 (bits 3-1 of BBANK)

FBANK = 1 (bits 15-11 of BBANK)

Inhibit interrupts

$TS_1 = TS_1 + SUPERBNK$ (SUPERBNK in bits 7-5)

Proceed to "CHANJOB4"

CHANG2 (Entered after the completion of a line of interpretive instructions to check for a job of higher priority)

$TS_0 = - LOC_0$

EBANK = 2 (bits 3-1 of BBANK)

FBANK = 1 (bits 15-11 of BBANK)

Inhibit interrupts

$TS_1 = BANKSET_0 + SUPERBNK$

CHANJOB4 $LOC_0 = LOC_{NEWJOB}$ (no change if NEWJOB = +0)

$BANKSET_0 = BANKSET_{NEWJOB}$

$LOC_{NEWJOB} = TS_0$

$BANKSET_{NEWJOB} = TS_1$

SUPERBNK = bits 7-5 of $BANKSET_0$

Exchange $MPAC_i$ of this job with $MPAC_i$ of new job for i = 0-7 ($MPAC_7$ = MODE)

If OVFIND $\neq$ 0, $PUSHLOC_0 = - PUSHLOC_0$

OVFIND = 0

Exchange $PUSHLOC_0$ with $PUSHLOC_{NEWJOB}$

Exchange $PRIORITY_0$ with $PRIORITY_{NEWJOB}$

FIXLOC = bits 9-1 of $PRIORITY_0$

If $PUSHLOC_0 < 0$:

$PUSHLOC_0 = - PUSHLOC_0$

OVFIND = 1

NEWJOB = +0

ENDPRCHG Release interrupt inhibit

If $LOC_0 \leq 0$:

$LOC_0 = - LOC_0 + 1$

EBANK = bits 3-1 of $BANKSET_0$ (via BBANK)

FBANK = bits 15-11 of $BANKSET_0$ (via BBANK)

Proceed to interpretive decoder

EBANK = bits 3-1 of $BANKSET_0$ (via BBANK)

FBANK = bits 15-11 of $BANKSET_0$ (via BBANK)

Proceed to job whose address is specified in LOC_0

PRIOCHNG Inhibit interrupts (Entered to "Change priority" of a job)

NEWPRIO = A (Priority in bits 14-10 of accumulator)

$BANKSET_0$ = FBANK + EBANK (FBANK in bits 15-11; EBANK in bits 3-1)

EBANK = 2

FBANK = 1

LOC_0 = Q-register

BUF_0 = +0

$PRIORITY_0$ = NEWPRIO + bits 9-1 of $PRIORITY_0$
(changing priority in bits 14-10 but leaving VAC address unchanged)

BUF_1 = - $PRIORITY_0$

Proceed to "EJSCAN"

JOBSLEEP (Entered with "address at which sleeping job is to begin when awakened" in accumulator)

LOC_0 = A

FBANK = 1

Inhibit interrupts

$PRIORITY_0$ = - $PRIORITY_0$ (to indicate that job is asleep)

$BANKSET_0$ = SUPERBNK + EBANK

BUF_1 = -0

Proceed to "EJSCAN"

<u>ENDOFJOB</u> FBANK = 1 (Entered to "End job")

Inhibit interrupts

BUF_1 = -0

TS = bits 9-1 of $PRIORITY_0$ (1 + address of VAC area if any)

$PRIORITY_0$ = -0

If TS > "$MPAC_6$" - "QPRET": (VAC area to be released)

  VACiUSE = TS - 1 (using VACiUSE +1 address in TS)

<u>EJSCAN</u> i = 12

If $PRIORITY_i$ > 0:

  If $PRIORITY_i$ - 1 > -BUF_1: (the "1" is insignificant)

    BUF_1 = - $PRIORITY_i$ + 1 ("1" does not change priority information in bits 14-10)

    BUF_0 = i (effectively)

If $PRIORITY_i$ = +0, proceed to "CCSHOLE"

If i < 84:

  i = i + 12

  Proceed to second step of "EJSCAN"

If $BUF_1 \geq +0$, proceed to "CCSHOLE"

If $BUF_1 = -0$: (all jobs completed)

NEWJOB = -0

Proceed to "DUMMYJB2"

If $BUF_0 = +0$: ("PRIOCHNG")

NEWJOB = +0

Proceed to "ENDPRCHG"

NEWJOB = BUF_0

$TS_0 = LOC_0$ (contents meaningless if end of job)

Inhibit interrupts

$TS_1 = BANKSET_0 + SUPERBNK$

Proceed to "CHANJOB4"

<u>JOBWAKE</u> Inhibit interrupts

TS = A (starting address of job when awakened)

Q-register = Q-register - 2

EXECTEM1 = FBANK

FBANK = 1

LOCCTR = 0

EXECTEM2 = 7

<u>JOBWAKE4</u> If $PRIORITY_{LOCCTR} < 0$ and $TS = LOC_{LOCCTR}$: (correct sleeper)

NEWPRIO = $- PRIORITY_{LOCCTR}$

$PRIORITY_{LOCCTR}$ = NEWPRIO

$NEWLOC_{ms} = 2000_8$ + bits 10-1 of TS

(If $PRIORITY_{LOCCTR} < 0$ and $TS = LOC_{LOCCTR}$:)

$NEWLOC_{1s}$ = bits 15-11 of TS + $BANKSET_{LOCCTR}$

If LOCCTR > 0, proceed to "SETLOC"

Proceed to "SPECTEST"

LOCCTR = LOCCTR + 12

If EXECTEM2 > 0:

EXECTEM2 = EXECTEM2 - 1

Proceed to "JOBWAKE4"

LOCCTR = -1 (indicating that no such sleeper was found)

FBANK = EXECTEM1

Return to 2 + address in Q-register

<u>DELAYJOB</u> Inhibit interrupts

TSt = A

RUPTREG1 = 2

<u>DELLOOP</u> If $DELAYLOC_{RUPTREG1} \neq 0$:

If RUPTREG1 = 0, proceed to "BAILOUT1" with TS = 31104_8 and $TS1_{dp}$ = ($BUF2_o$, $BUF2_1$)

RUPTREG1 = RUPTREG1 - 1

Proceed to "DELLOOP"

WAITEXIT = "TCGETCAD" - 2

L = FBANK + RUPTREG1 (FBANK = 0)

WAITADR = "WAKER"

Proceed to "DLY2" ("returns" to "TCGETCAD")

TCGETCAD DELAYLOC$_{RUPTREG1}$ = bits 10-1 of BUF2$_0$ + BUF2$_1$

Proceed to "JOBSLEEP" with A = DELAYLOC$_{RUPTREG1}$

WAKER (Must be in FBANK #0)

i = FBANK + EBANK (= RUPTREG1 of above)

A = DELAYLOC$_i$

DELAYLOC$_i$ = 0

Perform "JOBWAKE"

Proceed to "TASKOVER"

WAITLIST (Entered to "Call" a task with "delta-time to interrupt" in accumulator)

Inhibit interrupts

WAITDELT = A

WAITEXIT = Q-register ("Calling address" + 1)

TS$_{dp}$ = contents of double precision register specified in WAITEXIT

L = less significant half of TS

WAITADR = more significant half of TS

Proceed to "DLY2"

TWIDDLE (Entered to "Call" a task with the same FBANK, SUPERBNK and EBANK as the caller.)

Inhibit interrupts

Q-register = Q-register - 1

L = FBANK + SUPERBNK + EBANK

Inhibit interrupts

WAITDELT = A

WAITEXIT = Q-register

TSadr = Q-register + 1

WAITADR = contents of single precision register specified in TSadr

DLY2 WAITBANK = FBANK + EBANK (FBANK in bits 15-11; EBANK in bits 3-1)

EBANK = 3

FBANK = 1

If WAITDELT $\leq$ 0, proceed to "POODOO1" with TS = 21204_8 and $TS1_{dp}$ = (WAITEXIT, WAITBANK)

If TIME3 > 128, T1 = 16384 - TIME3 (delta-t to first task in list)

If TIME3 = 128, T1 = 1 (improper performance; unexpected)

If TIME3 < 128, T1 = - TIME3 (interrupt has occurred and is waiting to be processed; TIME3 continues to count (every centisecond) until it is reset)

TSdt = T1

If WAITDELT > TSdt - 1: (Call time greater than or equal to time to earliest task, task zero)

i = 0

Proceed to "WTLST5"

TSt = (8192 - WAITDELT) + 8192 modulo 2^{14}

Exchange TIME3 and TSt (Switch contents of one to the other)

$LST1_i = LST1_{i-1}$ for i = 7 thru 1 in that order

$LST1_0$ = 1 - (16384 - TSt - WAITDELT)

TS = more significant half of $LST2_{16}$

$LST2_{2i} = LST2_{2i-2}$ for i = 8 thru 0 in that order

$LST2_0$ = (WAITADR , L) (double precision)

If TS ≠ "SVCT3", proceed to "BAILOUT1" with TS = 31203_8 and $TS1_{dp}$ = (WAITEXIT, WAITBANK)

EBANK = bits 3-1 of WAITBANK (via BBANK)

FBANK = bits 15-11 of WAITBANK (via BBANK)

TSadr = 2 + WAITEXIT

Proceed to address specified in TSadr (return to caller)

<u>WTLST5</u> TSdt = TSdt - $LST1_i$ + 1 ($LST1_i$ = 1 - delta time between tasks i and i + 1 where time to task zero is counting down now)

If WAITDELT > TSdt - 1:

i = i + 1

If i = 8, proceed to "BAILOUT1" with TS = 31203_8 and $TS1_{dp}$ = (WAITEXIT, WAITBANK)

Proceed to "WTLST5"

WAITTEMP = TSdt - WAITDELT - 1

$LST1_i$ = $LST1_i$ + WAITTEMP + 1 (= 1 - "delta time from task i to new task")

$LST1_j$ = $LST1_{j-1}$ for j = 7 thru i+2 in that order for i 6.

$LST1_{i+1}$ = -WAITTEMP for i 7 (= 1 - "delta time from new task to task which was number i+1 but has been displaced")

TS = more significant half of $LST2_{16}$

$LST2_{2j}$ = $LST2_{2j-2}$ for j = 8 thru i+2 in that order for i 7

$LST2_{2i+2}$ = (WAITADR , L) (double precision)

If TS ≠ "SVCT3", proceed to "BAILOUT1" with TS = 31203_8 and $TS1_{dp}$ = (WAITEXIT, WAITBANK)

EBANK = bits 3-1 of WAITBANK (via BBANK)

FBANK = bits 15-11 of WAITBANK (via BBANK)

TSadr = 2 + WAITEXIT

Proceed to address specified in TSadr (return to caller)

<u>T3RUPT</u> BANKRUPT = FBANK + SUPERBNK + EBANK

QRUPT = Q-register

TS = $LST1_0$

$LST1_i = LST1_{i+1}$ for i = 0 thru 6 in that order

$LST1_7$ = 1 - 8193

RUPTAGN = -0

TIME3 =(16383 + TS + TIME3)modulo 16384

If overflow, RUPTAGN = 1 (two simultaneous tasks or tardy "T3RUPT")

TSadr = more significant half of $LST2_0$

TSbanks = less significant half of $LST2_0$

$LST2_{2i} = LST2_{2i+2}$ for i = 0 thru 7 in that order

$LST2_{16}$ = "SVCT3" (less significant half insignificant)

SUPERBNK = bits 7-5 of TSbanks

FBANK = bits 15-11 of TSbanks (via BBANK)

EBANK = bits 3-1 of TSbanks (via BBANK)

Proceed to address specified in TSadr

<u>TASKOVER</u> If RUPTAGN > 0:

EBANK = 3

FBANK = 1

Proceed to third step of "T3RUPT"

(Otherwise, RUPTAGN = -0)

SUPERBNK = bits 7-5 of BANKRUPT

Q-register = QRUPT

EBANK = bits 3-1 of BANKRUPT (via BBANK)

FBANK = bits 15-11 of BANKRUPT (via BBANK)

A = ARUPT

L = LRUPT

Release any interrupt inhibits

Resume

LONGCALL $LONGTIME_{dp}$ = (A , L)

LONGCADR = contents of double precision register specified in Q-register (stored in fixed at "calling address" + 1)

$LONGEXIT_1$ = FBANK + EBANK

EBANK = 3

FBANK = 1

$LONGEXIT_0$ = Q-register + 2

If $LONGTIME_{ms} > 0$, proceed to "LONGCYCL"

If $LONGTIME_{ms} = 0$ and $LONGTIME_{ls} \leq 0$ or if $LONGTIME_{ms} < 0$, proceed to "P00D001" with TS = 21204_8 and $TS1_{dp}$ = ($LONGEXIT_o$, $LONGEXIT_1$)

LONGCYCL LONGTIME = LONGTIME - 8192

If $LONGTIME_{dp} > 0$:

Call "LONGCYCL" in 81.92 seconds

If $LONGTIME_{dp} \leq 0$:

$TSt = LONGTIME_{1s} + 8192$

Call "GETCADR" in TSt centiseconds

EBANK = bits 3-1 of $LONGEXIT_1$

FBANK = bits 15-11 of $LONGEXIT_1$

TSadr = $LONGEXIT_0$

$LONGEXIT_0$ = "TASKOVER"

Proceed to address specified in TSadr (return or "End task")

<u>GETCADR</u> TSbanks = less significant half of LONGCADR

TSadr = more significant half of LONGCADR

EBANK = bits 3-1 of TSbanks

FBANK = bits 15-11 of TSbanks

Proceed to address specified in TSadr

<u>FIXDELAY</u> (Entered to "delay" during a task)

A = contents of single precision cell specified in Q-register

Q-register = Q-register + 1

<u>VARDELAY</u> WAITDELT = A

WAITADR = Q-register

L = FBANK + SUPERBNK + EBANK

WAITEXIT = "TASKOVER" - 2

Proceed to "DLY2"

Routines used for inter-bank communication
(not a complete list; included for example only)

BANKCALL $BUF2_0 = A$

$BUF2_1 = L$

A = contents of single precision cell specified in Q-register

Q-register = Q-register + 1

SWCALL FCADR = A

TS = FBANK

FBANK = bits 15-11 of FCADR

TSadr = 02000_8 + bits 10-1 of FCADR

$A = BUF2_0$

$L = BUF2_1$ (to preserve their contents during a "BANKCALL")

$BUF2_0$ = Q-register

$BUF2_1 = TS$

Proceed to address specified in TSadr

SWRETURN FBANK = $BUF2_1$ (without disturbing contents of A or L)

Proceed to address specified in $BUF2_0$

POSTJUMP TS = A

A = contents of single precision cell specified in Q-register

BANKJUMP FBANK = bits 15-11 of A

TSadr = 02000_8 + bits 10-1 of A

A = TS

Proceed to address specified in TSadr

MAKECADR FCADR = $BUF2_1$ + bits 10-1 of $BUF2_0$

Return

SUPDACAL TSmp = FBANK + SUPERBNK

FBANK = bits 15-11 of FCADR

TSadr = 02000_8 + bits 10-1 of FCADR

Inhibit interrupts

SUPERBNK = bits 7-5 of L

A = contents of cell whose address is specified in TSadr

SUPERBNK = bits 7-5 of TSmp

Release interrupt inhibit

FBANK = bits 15-11 of TSmp

Return

A: Single precision accumulator with overflow bit in addition to the usual sign bit and fourteen magnitude bits. (Stored in ARUPT during an interrupt.)

BANKRUPT: Single precision storage for current bank settings when a job is interrupted and the banks are reset to process the interrupt.

$BANKSET_i$ (i = 0, 12, 24, 36, 48, 60, 72, 84): Single precision storage for the FBANK, SUPERBNK, and EBANK settings required by each job; part of the job core assigned to each active job. (FBANK in bits 15-11; SUPERBNK in bits 7-5; EBANK in bits 3-1.)

BUF_i (i = 0,1): Single precision working storage cells used in "EJSCAN" to determine the job of highest priority if any are active.

$BUF2_i$ (i = 0,1): Single precision storage for the return address and FBANK setting during a temporary transfer to another FBANK.

$DELAYLOC_i$ (i = 0,1,2): Single precision address of one of three jobs being "delayed".

EBANK: Single precision register which controls erasable memory access; scaled B6 and expressed as an octal quantity between 0 and 7. Gated directly to bits 3-1 of BBANK.

EXECTEM1, EXECTEM2: Single precision temporary storage cells.

FBANK: Single precision register which controls fixed memory access in conjunction with the SUPERBNK register; scaled B4 and expressed as an octal quantity between 0 and 37_8. Gated directly to bits 15-11 of BBANK.

FCADR: Single precision address with FBANK setting used for inter-bank communication with no change in the SUPERBNK setting. Bits 15-11 contain the FBANK setting and bits 10-1 contain the address.

FIXLOC: Single precision address of the VAC area (or job core accumulator, $MPAC_6 - 42_8$) of the job being executed.

L: Single precision "less significant" half of the accumulator when it contains a double precision number. (Stored in LRUPT during an interrupt.)

LOC_i (i = 0, 12, 24, 36, 48, 60, 72, 84): Single precision storage for the S-register portion of the starting address of a job; part of the job core assigned to each active job. If an interrupted job is using the interpreter language, LOC is complemented.

LOCCTR: Single precision job core index, scaled B14 and unitless.

LONGCADR: Double precision storage for starting address and bank settings for a task in "LONGCALL".

$LONGEXIT_i$ (i=0,1): Single precision storage for address and bank settings of routine that is "calling" a task via "LONGCALL".

LONGTIME: Double precision time interval from "now" to the time of initiation of a task in "LONGCALL"; scaled B28 in units of centi-seconds.

$LST1_i$ (i=0-7): Single precision storage for "one minus the delta-time between tasks i and i+1," where task i=0 is the one for which TIME3 is counting down; scaled B14 in units of centiseconds.

$LST2_i$ (i=0, 2, 4, 6, 8, 10, 12, 14, 16): Double precision storage for address and bank settings for each task in the waitlist. Set equal to the address of "SVCT3" when not in use (to cause "SVCT3" to be executed every 81.93 seconds).

$MPAC_i$ (i=0-7): Multiple precision accumulator used automatically by jobs coded in the interpretive language (via the interpretive decoder) and sometimes by jobs coded in basic language. A set of eight single precision cells associated with a particular job and used exclusively by that job. When a job is put to sleep or is interrupted by a job of higher priority, $MPAC_{0-7}$ are saved as part of the "job core" reserved for that job, and they are reset exactly as they were when the interrupted job is re-established.

NEWJOB: Single precision index of the job core of the active job of highest priority, scaled B14 and unitless; set to -0 when no jobs are active.

NEWLOC: Double precision temporary storage for the LOC and BANKSET of a job being established.

NEWPRIO: Single precision temporary storage for the priority and VAC area address for a job being established.

OVFIND: Single precision overflow indicator associated with the job being executed.

$PRIORITY_i$ (i=0, 12, 24, 36, 48, 60, 72, 84): Single precision storage for the priority (bits 14-10) and VAC area address assigned to each active job; if a job is "sleeping", PRIORITY is negative; if the job core of which a particular $PRIORITY_i$ is a part is available, $PRIORITY_i$ is equal to -0. (Adjusted for a NOVAC job so that QPRET will equal $MPAC_6$.)

$PUSHLOC_i$ (i=0, 12, 24, 36, 48, 60, 72, 84): Single precision address of the next available position in the VAC area of a particular job, set equal to the first position in the list when a job is established. If a job is interrupted while the overflow indicator is set, PUSHLOC is complemented.

Q-register: Single precision return address register automatically set by a TC instruction (basic) equal to "calling address" + 1 (address of instruction immediately following the TC instruction).

QPRET: Single precision octal return address storage cell loaded by interpretive transfer instructions (equals VAC_{42} or $MPAC_6$).

QRUPT: Single precision storage for current value of Q-register when a job is interrupted, for reloading the Q-register when the interrupt is completed.

RUPTAGN: Single precision cell used in "T3RUPT" to determine if more than one task must be processed at a single interrupt.

RUPTREG1: Single precision temporary storage cell.

SELFRET: See TEST section.

SUPERBNK: Single precision LGC channel which controls fixed memory access in conjunction with the FBANK register; scaled B10 and expressed as an octal quantity between 0 and 4.

TIME3: See INTR section.

VACiUSE (i=1, 2, 3, 4, 5): Single precision register at the head of each of the five working storage areas (VAC areas) that may be assigned to jobs. Each VAC area contains 43 single precision cells plus VACiUSE.

WAITADR: Single precision "S-register" portion of the address of a task being inserted in the waitlist.

WAITBANK: Single precision storage for current EBANK and FBANK while these banks are switched to enter a task in the waitlist.

WAITDELT: Single precision time interval between "now" and time at which a task is to start, scaled B14 in units of centiseconds.

WAITEXIT: Single precision return address to routine that is "calling" a task; used to locate the task address when it is stored in fixed memory at the calling address +1 and 2.

WAITTEMP: Single precision delta-time between time-from-now at which task i + 1 will be executed (time to task zero is counting down "now") and time-from-now at which new task is to be executed.

ORBI

STATEINT Establish "STATINT1" (pr05)

End task

STATINT1 If FLAGWRD9 bit 5 (QUITFLAG) = 1:

Switch FLAGWRD9 bit 5 (QUITFLAG) to 0

End job

TDEC1 = TIMENOW

Perform "INTSTALL" (wait until orbital integration free)

Switch FLAGWRD2 bit 1 (NODOFLAG) to 1

Perform "SETIFLGS" (Set up for Encke without W-matrix)

Switch FLAGWRD3 bits 15 (POOHFLAG) and 3 (VINTFLAG) to 1

If FLAGWRD8 bit 8 (SURFFLAG) = 1 and FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 1 (6x6)

Switch FLAGWRD3 bit 8 (PRECIFLG) to 0

Perform "INTEGRV"

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

Proceed to "ENDINT"

TDEC1 = TETCSM

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

Perform "SETIFLGS"

If FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 1 (9x9)

Switch FLAGWRD3 bit 8 (PRECIFLG) to 1

Perform "INTEGRV"

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

Proceed to "ENDINT"

<u>SETIFLGS</u> Switch FLAGWRD3 bit 5 (STATEFLG) to 1 (store integrated state vector)

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0 (Specify Encke)

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 0

Return

<u>ENDINT</u> Switch FLAGWRD3 bit 5 (STATEFLG) to 0

Call "STATEINT" in 600 seconds

End job

<u>CSMPREC</u> Perform "INTSTALL"

IRETURN = return address to caller of "CSMPREC"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Switch FLAGWRD3 bit 8 (PRECIFLG) to 1

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 4 (INTYPFLG) to 0

Proceed to second step of "INTEGRV"

<u>LEMPREC</u> Perform "INTSTALL"

IRETURN = return address to caller of "LEMPREC"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

Switch FLAGWRD3 bit 8 (PRECIFLG) to 1

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 4 (INTYPFLG) to 0

Proceed to second step of "INTEGRV"

CSMCONIC Perform "INTSTALL"

IRETURN = return address to caller of "CSMCONIC"

Switch FLAGWRD3 bits 3 (VINTFLAG) and 4 (INTYPFLG) to 1

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 0

Proceed to second step of "INTEGRV"

LEMCONIC Perform "INTSTALL"

IRETURN = return address to caller of "LEMCONIC"

Switch FLAGWRD3 bits 3 (VINTFLAG) and 1 (DIMOFLAG) to 0

Switch FLAGWRD3 bit 4 (INTYPFLG) to 1

Proceed to second step of "INTEGRV"

INTEGRV IRETURN = return address (to caller of "INTEGRV")

Switch FLAGWRD8 bits 15 (RPQFLAG) and 13 (NEWIFLG) to 1

If FLAGWRD3 bit 3 (VINTFLAG) = 1:

Perform "MOVEPCSM"

Set FLAGWRD0 bit 12 (MOONFLAG) = FLAGWRD8 bit 12 (CMOONFLG)

If FLAGWRD3 bit 3 (VINTFLAG) = 0:

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "USEPIOS"

Perform "MOVEPLEM"

Set FLAGWRD0 bit 12 (MOONFLAG) = FLAGWRD8 bit 11 (LMOONFLG)

PBODY = 0

If FLAGWRD0 bit 12 (MOONFLAG) = 1, PBODY = 2

Proceed to "ALOADED"

INTEGRVS IRETURN = return address

Switch FLAGWRD3 bit 8 (PRECIFLG) to 1

PBODY = 0

If FLAGWRD0 bit 12 (MOONFLAG) = 1, PBODY = 2

TDELTAV = 0

TNUV = 0

Perform "RECTIFY"

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 0

Switch FLAGWRD8 bits 15 (RPQFLAG) and 13 (NEWIFLG) to 1

ALOADED TDEC = TDEC1

If FLAGWRD3 bit 4 (INTYPFLG) = 0, proceed to "TESTLOOP"

RVCON TAU = TDEC - TET

Perform "RECTIFY"

Perform "KEPPREP"

TET = TET + TC

RECTOUT Perform "RECTIFY"

RATT = RRECT

VATT = VRECT

TAT = TET

TSmu = $K:MU_{PBODY}$

MUDEX = 0 (-2 in index register 1, X1)

If FLAGWRD0 bit 12 (MOONFLAG) = 1, MUDEX = 8
(-10 in index register 1, X1)

INTEXIT   Switch FLAGWRD9 bit 1 (AVEMIDSW) to 0

Switch FLAGWRD3 bit 8 (PRECIFLG) to 0

Switch FLAGWRD3 bit 5 (STATEFLG) to 0

Perform "INTWAKE"   (awaken any jobs waiting to integrate)

Return via IRETURN   (with PBODY in index register 2, X2)

RECTIFY   RRECT = RCV + TDELTAV   (Scaling controlled by PBODY)

RCV = RRECT

VRECT = VCV + TNUV

VCV = VRECT

TDELTAV = 0

TNUV = 0

TC = 0

XPREV = 0

Return

TESTLOOP   If FLAGWRD9 bit 5 (QUITFLAG) = 1:

    Switch FLAGWRD3 bit 5 (STATEFLG) to 0

    Proceed to "INTEXIT"

i = PBODY

Switch FLAGWRD0 bit 13 (MIDFLAG) to 0   (MIDFLAG should remain zero in LUMINARY)

If $|RCV| \geq K{:}RM_i$:

    Switch FLAGWRD0 bit 13 (MIDFLAG) to 1

$TSstep = K{:}p3D \sqrt{|\underline{R}CV|^3 / K{:}MU_i}$ (truncated to a multiple of 128 centiseconds)

$TSstep = 2^7 [\text{integral part of } (TSstep / 2^7)]$

If overflow or if TSstep > 2 K:DTd2MAX, TSstep = 2 K:DTd2MAX

DTd2 = $\frac{1}{2}$ (TDEC - TET)

If overflow or if |DTd2| ≥ $\frac{1}{2}$ TSstep, DTd2 = $\frac{1}{2}$ TSstep signDTd2

If |DTd2| < K:DTd2MIN, proceed to "A-PCHK" (convergence)

If FLAGWRD3 bit 15 (POOHFLAG) = 1 and bit 8 (PRECIFLG)=0: ("STATEINT")

If DTd2 < $\frac{1}{2}$ TSstep, proceed to "A-PCHK" (don't integrate past even timestep)

If FLAGWRD8 bit 13 (NEWIFLG) = 1:

Switch FLAGWRD8 bit 13 (NEWIFLG) to 0

If TET > TDEC, proceed to "INTEXIT"
(don't integrate backwards)

If (TDEC - TET) < 8 DTd2, proceed to "INTEXIT"
(don't integrate unless more than 4 timesteps behind)

<u>TIMESTEP</u> If FLAGWRDO bit 13 (MIDFLAG) = 0, proceed to "RECTEST"
(MIDFLAG should remain zero in LUMINARY)

If DTd2 [$\underline{R}$CV · $\underline{V}$CV] < 0, proceed to "RECTEST"

If FLAGWRDO bit 12 (MOONFLAG) = 1, proceed to "LUNSPH"

If FLAGWRD8 bit 15 (RPQFLAG) = 1:

TSt = TET

Perform "LSPOS"

$\underline{R}$PQV = $\underline{T}$Ssun

i = PBODY

If |$\underline{R}$CV - $\underline{R}$PQV| < K:RSPHERE:

Perform "ORIGCHNG"

(If |RCV - RPQV| < K:RSPHERE:)

    Proceed to "INTGRATE"

Proceed to "RECTEST"

LUNSPH   If |RCV| < K:RSPHERE, proceed to "RECTEST"

If FLAGWRD8 bit 15 (RPQFLAG) = 1:

    TSt = TET

    Perform "LSPOS"

    RPQV = -TSsun

Perform "ORIGCHNG"

Proceed to "INTGRATE"

ORIGCHNG   Perform "RECTIFY"

RRECT = RCV - RPQV

RCV = RRECT

If FLAGWRDO bit 12 (MOONFLAG) = 1:

    $TS_{yyy} = -TS_{yyy}$ (note that TS_{yyy} is not defined)

VRECT = VCV - TS_{yyy}

VCV = VRECT

If FLAGWRDO bit 12 (MOONFLAG) = 1:

    PBODY = 0

    Convert RCV, RRECT, VCV and VRECT to earth scaling

If FLAGWRDO bit 12 (MOONFLAG) = 0:

    PBODY = 2

(If FLAGWRDO bit 12 (MOONFLAG) = 0:)

Convert RCV, RRECT, VCV and VRECT to moon scaling

Invert FLAGWRDO bit 12 (MOONFLAG)

Return

RECTEST If |TDELTAV| ≥ K:rectr, if |TDELTAV| / |RCV| ≥ K:RECRATIO, or if |TNUV| ≥ K:rectv:

Perform "RECTIFY"

INTGRATE ZV = TNUV

YV = TDELTAV

Switch FLAGWRDO bit 14 (JSWITCH) to 0

DIFEQCNT = 0

ALPHAV = YV

H = 0

If FLAGWRDO bit 14 (JSWITCH) = 1:

Proceed to "DOW.."

ACCOMP i = PBODY

FV = 0

BETAV = ALPHAV + RCV

If FLAGWRD3 bit 1 (DIMOFLAG) = 1:

j = - DIFEQCNT

$VECTAB_j$ = BETAV

ALPHAM = |ALPHAV|

Perform "GAMCOMP"

ALPHAV = BETAV

ALPHAM = BETAM

If FLAGWRDO bit 13 (MIDFLAG) = 0: (MIDFLAG should remain zero in LUMINARY)

Perform "OBLATE"

Proceed to "NBRANGH"

TSt = TET

Perform "LSPOS"

i = 2 (index MU of secondary body)

If FLAGWRDO bit 12 (MOONFLAG) = 1:

TSsun = - TSsun

i = 0

BETAV = TSsun

RPQV = BETAV

RPSV = TSxxx (note that TSxxx is not defined)

If MODREG = 23 or if FLAGWRD3 bit 1 (DIMOFLAG) = 1:

j = 6 - DIFEQCNT

$VECTAB_j$ = ALPHAV - BETAV

RQVV = $VECTAB_j$

Switch FLAGWRD8 bit 15 (RPQFLAG) to 0

If FLAGWRDO bit 12 (MOONFLAG) = 1, RPSV = RPSV + RPQV

Perform "GAMCOMP" (BETAV = RPQV)

BETAV = RPSV

i = 4 (to index MU of the sun)

Perform "GAMCOMP"

Perform "OBLATE"

Proceed to "NBRANCH"

GAMCOMP BETAM = |BETAV|

RHO = ALPHAM / BETAM

q = RHO (RHO - 2 unitALPHAV · unitBETAV)

DdBETA = $\sqrt{1 + q}$

$$Fq = q \frac{3 + 3\,q + q^2}{1 + (1+q)^{3/2}}$$

$$TSgam = - K{:}MU_i \frac{RHO\ (unitALPHAV + unitBETAV\ Fq\ /\ RHO)}{BETAM^2\ DdBETA^3}$$

FV = FV + TSgam

If overflow, proceed to "GOBAQUE"

Return

OBLATE i = PBODY

If ALPHAM ≥ $K{:}RD_i$, return

If FLAGWRDO bit 12 (MOONFLAG) = 0:

COSØ = Z component of unitALPHAV

UZ = K:UNITZ

If FLAGWRDO bit 12 (MOONFLAG) = 1:

TSt = TET

Perform "MOONMX"

URPV = [MOONMAT] (unitALPHAV - ($[MOONMAT]^T$ LM504) * unitALPHAV)

COSØ = Z component of URPV

TS = $[MOONMAT]^T$ (K:UNITZ - (K:UNITZ * LM504))

UX = $[MOONMAT]^T$ (K:UNITX - (K:UNITX * LM504))

UZ = TS

$P_2' = 3 \text{ COS}\phi$

$P_3' = \frac{1}{2} (15 \text{ COS}\phi^2 - 3)$

$P_4' = (1/3)(7 P_3' \text{ COS}\phi - 4 P_2')$

$P_5' = \frac{1}{4} (9 P_4' \text{ COS}\phi - 5 P_3')$

$$\underline{TS} = \left[P_3' + \frac{K{:}j3j2_i}{ALPHAM}\left(P_4' + \frac{K{:}j4j3_i}{ALPHAM} P_5'\right)\right] \text{ unit}\underline{ALPHAV}$$

$$\underline{TS} = \underline{TS} - \left[P_2' + \frac{K{:}j3j2_i}{ALPHAM}\left(P_3' + \frac{K{:}j4j3_i}{ALPHAM} P_4'\right)\right] \underline{UZ}$$

$\underline{TS} = \underline{TS}\ K{:}j2_i/ALPHAM^4$ (computed quasi floating point)

$\underline{FV} = \underline{FV} + \underline{TS}$

If overflow, proceed to "GOBAQUE"

If FLAGWRDO bit 12 (MOONFLAG) = 0, return

$\underline{TS1} = 5(URPV_y^2 - URPV_x^2) \text{ unit}\underline{ALPHAV} + 2\ URPV_x\ \underline{UX} + 2\ URPV_y\ (\underline{UX} * \underline{UZ})$

$\underline{TS} = 5\ URPV_x\ (1 - 7 \text{ COS}\phi^2) \text{ unit}\underline{ALPHAV} + (5 \text{ COS}\phi^2 - 1)\ \underline{UX} + (10\ URPV_x\ URPV_z)\ \underline{UZ}$

$\underline{TS2} = (E32C31RM/ALPHAM)\ \underline{TS} + E3J22R2M\ \underline{TS1}$

$\underline{TS} = \underline{TS2}/ALPHAM^4$

$\underline{FV} = \underline{FV} + \underline{TS}$

If overflow, proceed to "GOBAQUE"

i = PBODY

Return

GOBAQUE If | TDELTAV | = 0, proceed to "POODOO" with TS = 20430_8

TAU = TC - H

TET = TET - H

Perform "KEPPREP"

Perform "RECTIFY"

Switch FLAGWRD8 bit 15 (RPQFLAG) to 1

Proceed to "TESTLOOP"

NBRANCH If DIFEQCNT = -24, proceed to "DIFEQ+2"

If DIFEQCNT = -12:

PSIV = PHIV + 4 FV

PHIV = PHIV + 2 FV

If DIFEQCNT = 0:

PHIV = FV

H = H + DTd2

DIFEQCNT = DIFEQCNT - 12

ALPHAV = YV + H (ZV + $\frac{1}{2}$ H FV)

If FLAGWRDO bit 14 (JSWITCH) = 1, proceed to "DOW.."

TAU = TC + DTd2

TET = TET + DTd2 (DTd2 rounded to nearest centisecond if DIFEQCNT = -24)

Perform "KEPPREP"

Proceed to "ACCOMP"

DIFEQ+2 YV = YV + H (ZV + PHIV H / 6)

ZV = ZV + (PSIV + FV) H / 6

If FLAGWRDO bit 14 (JSWITCH) = 0, proceed to "ENDSTATE"

$(W_{4n}, W_{5n}, W_{6n}) = \underline{Z}V$

$(W_{1n}, W_{2n}, W_{3n}) = \underline{Y}V$

If overflow, proceed to "WMATEND"

If n $\leq$ 1:

TDEC1 = TDEC

Proceed to third step of "INTEGRV"

n = n - 1

Proceed to "NEXTCOL"

ENDSTATE If overflow, proceed to "GOBAQUE"

TNUV = ZV

TDELTAV = YV

If FLAGWRD9 bit 2 (MIDAVFLG) = 1, proceed to "CKMID2"

If FLAGWRD3 bit 1 (DIMOFLAG) = 0, proceed to "TESTLOOP"

Switch FLGWRD10 bit 7 (REINTFLG) to 1
(integration routine to be restarted)

If FLAGWRD3 bit 3 (VINTFLAG) = 1:

Perform "MOVEACSM"

Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

Perform "SVDWN1"

Set FLAGWRD8 bit 12 (CMOONFLG) = FLAGWRD0 bit 12 (MOONFLAG)

If FLAGWRD3 bit 3 (VINTFLAG) = 0:

Perform "MOVEALEM"

Switch FLAGWRD8 bit 11 (LMOONFLG) to 1

Perform "SVDWN2"

(If FLAGWRD3 bit 3 (VINTFLAG) = 0:)

Set FLAGWRD8 bit 11 (LMOONFLG) = FLAGWRDO bit 12 (MOONFLAG)

Switch FLAGWRDO bit 14 (JSWITCH) to 1

n = 6

If FLAGWRD3 bit 2 (D6OR9FLG) = 1, n = 9

NEXTCOL $\underline{YV} = (W_{1n}, W_{2n}, W_{3n})$

$\underline{ZV} = (W_{4n}, W_{5n}, W_{6n})$

j = 0

DIFEQCNT = 0

$\underline{ALPHAV} = \underline{YV}$

H = 0

If FLAGWRDO bit 14 (JSWITCH) = 0, proceed to "ACCOMP"

DOW.. i = PBODY

j = - DIFEQCNT

$BETAM = K:MU_i$

$\underline{TS} = 3\ (\underline{ALPHAV} \cdot \text{unit}\underline{VECTAB}_j)\ \ \text{unit}\underline{VECTAB}_j - \underline{ALPHAV}$

$\underline{FV} = \underline{TS}\ BETAM / |\underline{VECTAB}_j|^3$

If FLAGWRDO bit 13 (MIDFLAG) = 0, proceed to "NBRANCH"
(MIDFLAG should remain zero in LUMINARY)

j = 6 - DIFEQCNT

i = 2-PBODY (earth or moon)

$BETAM = K:MU_i$

$\underline{TS} = 3\ (\underline{ALPHAV} \cdot \text{unit}\underline{VECTAB}_j)\ \text{unit}\underline{VECTAB}_j - \underline{ALPHAV}$

$\underline{TS} = \underline{TS}\ BETAM / |\underline{VECTAB}_j|^3$

If FLAGWRDO bit 12 (MOONFLAG) = 0:

Shift $\underline{TS}$ right 6 places for scaling

$\underline{FV} = \underline{FV} + \underline{TS}$

Proceed to "NBRANCH"

KEPPREP   KEPRTN = return address

i = PBODY

ROOTMU = $\sqrt{K:MU_i}$

A5 = $\frac{1}{2}$ unit$\underline{RCV} \cdot \underline{VCV}$

Q = (TAU - TC) / $|\underline{RCV}|$

TS = $\frac{1}{6} Q^2 (K:MU_i - |\underline{VCV}|^2 |\underline{RCV}|) / |\underline{RCV}|$ $(= -\frac{1}{6} s^2 (\frac{1}{r} - \frac{1}{a}))$

XKEPNEW = XPREV + ROOTMU Q $(1 - A5\ Q + 2\ A5^2\ Q^2 + TS)$

MUDEX = 0   (-2 in index register 1, X1)

If FLAGWRDO bit 12 (MOONFLAG) = 1, MUDEX = 8   (-10 in index register 1, X1)

Proceed to "KEPLERN"   (return directly to calling program from "KEPLERN")

WMATEND   Switch FLAGWRD3 bits 1 (DIMOFLAG) and 6 (ORBWFLAG) to 0

Switch FLAGWRD5 bit 1 (RENDWFLG) to 0

Switch FLAGWRD3 bit 5 (STATEFLG) to 1

Perform "ALARM" with TS = 00421_8

Proceed to "TESTLOOP"

USEPIOS   TSt = TDEC1

TET = TDEC1

Perform "MOONMX"

RCV = $[\text{MOONMAT}]^T$ (RLS + LM504 * RLS)

Switch FLAGWRD0 bit 12 (MOONFLAG) to 1

Perform "MOONMX"

TS = $[\text{MOONMAT}]^T$ (K:UNITZ + LM504 * K:UNITZ)

VCV = K:OMEGMOON TS * RCV

TDELTAV = 0

PBODY = 2

TNUV = 0

A-PCHK If FLAGWRD3 bit 5 (STATEFLG) = 0, proceed to "RECTOUT"

Switch FLGWRD10 bit 7 (REINTFLG) to 1

If FLAGWRD3 bit 3 (VINTFLAG) = 1:

  Perform "MOVEACSM"

  Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

  Perform "SVDWN1"

  Set FLAGWRD8 bit 12 (CMOONFLG) = FLAGWRD0 bit 12 (MOONFLAG)

If FLAGWRD3 bit 3 (VINTFLAG) = 0:

  Perform "MOVEALEM"

  Switch FLAGWRD8 bit 11 (LMOONFLG) to 1

  Perform "SVDWN2"

  Set FLAGWRD8 bit 11 (LMOONFLG) = FLAGWRD0 bit 12 (MOONFLAG)

Proceed to "RECTOUT"

MOVEPCSM XPREV = XKEPCSM

TC = TCCSM

VCV = VCVCSM

RCV = RCVCSM

TNUV = NUVCSM

TDELTAV = DELTACSM

TET = TETCSM

VRECT = VRECTCSM

RRECT = RRECTCSM

Return.

MOVEPLEM XPREV = XKEPLEM

TC = TCLEM

VCV = VCVLEM

RCV = RCVLEM

TNUV = NUVLEM

TDELTAV = DELTALEM

TET = TETLEM

VRECT = VRECTLEM

RRECT = RRECTLEM

Return

MOVEACSM XKEPCSM = XPREV

TCCSM = TC

VCVCSM = VCV

RCVCSM = RCV

NUVCSM = TNUV

DELTACSM = TDELTAV

TETCSM = TET

VRECTCSM = VRECT

RRECTCSM = RRECT

Return

MOVEALEM XKEPLEM = XPREV

TCLEM = TC

VCVLEM = VCV

RCVLEM = RCV

NUVLEM = TNUV

DELTALEM = TDELTAV

TETLEM = TET

VRECTLEM = VRECT

RRECTLEM = RRECT

Return

AVETOMID EGRESS = return address

If FLAGWRD5 bit 1 (RENDWFLG) or FLAGWRD3 bit 6 (ORBWFLAG) = 1:

(Integrate W-matrix through burn)

Perform "INTSTALL"

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 1

Switch FLAGWRD9 bit 1 (AVEMIDSW) to 1

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

(If FLAGWRD5 bit 1 (RENDWFLG) or FLAGWRD3 bit 6 (ORBWFLAG) = 1:)

    TDEC1 = PIPTIME

    Perform "INTEGRV"

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Perform "SETIFLGS"

TDEC1 = PIPTIME

Perform "INTEGRV"

Perform "INTSTALL"

RRECT = RN (scaling controlled by LMOONFLG)

RCV = RN

TET = PIPTIME

VRECT = VN

VCV = VN

TDELTAV = 0

TNUV = 0

TC = 0

XPREV = 0

Perform "MOVEALEM"

TRKMKCNT = 0

Perform "INTWAKE"

Return via EGRESS

MIDTOAV1 IRETURN1 = return address

TSerror = 0

Switch FLAGWRD9 bit 3 (MID1FLAG) to 1

If TDEC1 < TIMENOW + K:TIMEDELT:

  Switch FLAGWRD9 bit 3 (MID1FLAG) to 0

  TSerror = 1

  Perform "ALARM" with TS = 01703_8

  TDEC1 = TIMENOW + K:TIMEDELT

Skip next three steps

MIDTOAV2 IRETURN1 = return address

Switch FLAGWRD9 bit 3 (MID1FLAG) to 0

TDEC1 = TIMENOW + K:TIMEDELT

Perform "INTSTALL"

Switch FLAGWRD3 bits 1 (DIMOFLAG), 3 (VINTFLAG) and 4 (INTYPFLG) to 0

Switch FLAGWRD9 bit 2 (MIDAVFLG) to 1

Perform "INTEGRV"

Switch FLAGWRD9 bit 2 (MIDAVFLG) to 0

$\underline{R}$N1 = $\underline{R}$ATT

$\underline{V}$N1 = $\underline{V}$ATT

PIPTIME1 = TAT

RTX2 = X2 (0 for earth, 2 for moon)

RTX1 = X1 (-2 for earth, -10 for moon)

Inhibit interrupts

TSt = PIPTIME1 - TIMENOW

Force sign agreement between the two halves of TSt

Return via IRETURN1

CKMID2 If FLAGWRD9 bit 3 (MID1FLAG) = 1:

If TDEC ≥ TIMENOW + K:TIMEDELT, proceed to "TESTLOOP"

Switch FLAGWRD9 bit 3 (MID1FLAG) to 0

TSerror = 1

Perform "ALARM" with TS = 01703_8

TDEC = TIMENOW + K:TIMEDELT

Proceed to "TESTLOOP"

If |TDEC - TET| < K:30SECS, proceed to "A-PCHK"

TDEC = TIMENOW + K:TIMEDELT

Proceed to "TESTLOOP"

INTSTALL QPRET = return address

If FLGWRD10 bit 14 (INTFLAG) or 7 (REINTFLG) = 1:

Put present job to sleep

When awakened, job will resume at second step of "INTSTALL" when jobs of higher priority have been completed

Switch FLGWRD10 bit 14 (INTFLAG) to 1

Return via QPRET

INTWAKE If FLGWRD10 bit 7 (REINTFLG) = 1: (means restarted)

TBASE2 = QPRET of present job

Set restart group 2 to resume computations at next step

QPRET (of present job) = TBASE2

If FLGWRD10 bit 7 (REINTFLG) = 0: (i.e. if got a restart)

Return via QPRET (of present job)

INTWAKE1 Awaken job or jobs put to sleep in "INTSTALL"

Switch FLGWRD10 bits 14 (INTFLAG) and 7 (REINTFLG) to 0

Return via QPRET (of present job)

INTWAKEU Release interrupt inhibit

If UPSVFLAG = 0: (UPSVFLAG is the third component of of a state vector update)

Perform "INTWAKE1"

Return

RCV = RRECT

VCV = VRECT

TDELTAV = 0

TNUV = 0

TC = 0

XPREV = 0

i = 0

Switch FLAGWRD0 bit 12 (MOONFLAG) to 0

If |UPSVFLAG| = 2:

i = 2

Switch FLAGWRD0 bit 12 (MOONFLAG) to 1

If UPSVFLAG > 0:

Perform "MOVEACSM"

Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

Perform "SVDWN1"

(If UPSVFLAG > 0:)

Set FLAGWRD8 bit 12 (CMOONFLG) = FLAGWRDO bit 12 (MOONFLAG)

Switch FLAGWRD3 bit 6 (ORBWFLAG) to 0

If UPSVFLAG < 0:

Perform "MOVEALEM"

Switch FLAGWRD8 bit 11 (LMOONFLG) to 1

Perform "SVDWN2"

Set FLAGWRD8 bit 11 (LMOONFLG) = FLAGWRDO bit 12 (MOONFLAG)

Switch FLAGWRD5 bit 1 (RENDWFLG) to 0

UPSVFLAG = 0

Perform "INTWAKE1"

Return

P76 Switch FLAGWRD1 bit 5 (TRACKFLG) to 1

$\underline{D}ELVOV = \underline{D}ELVLVC$

Proceed to "GOFLASH" with TS = K:V06N33 (TIG)
(If terminate, proceed to "ENDP76"; if proceed, continue at next step; if other response, repeat this step.)

Proceed to "GOFLASH" with TS = K:V06N84 ($\underline{D}ELVOV$)
(If terminate, proceed to "ENDP76"; if proceed, continue at next step; if other response, repeat this step.)

Switch FLAGWRD2 bit 1 (NODOFLAG) to 1

TDEC1 = TIG

Perform "CSMPREC"

$$\underline{T}Sv = \begin{bmatrix} \text{unit}(\text{unit}\underline{R}ATT * \underline{V}ATT) * \text{unit}\underline{R}ATT \\ \text{unit}(\underline{V}ATT * \text{unit}\underline{R}ATT) \\ - \text{unit}\underline{R}ATT \end{bmatrix}^T \underline{D}ELVOV + \underline{V}ATT$$

Perform "INTSTALL"

Set FLAGWRD0 bit 12 (MOONFLAG) = FLAGWRD8 bit 12 (CMOONFLG)

VCV = TSv (scaling controlled by MOONFLAG)

RCV = RATT

TET = TIG

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0

TDEC1 = TETLEM

Perform "INTEGRVS"

Perform "INTSTALL"

RRECT = RATT

RCV = RATT

TET = TAT

VRECT = VATT

VCV = VATT

TDELTAV = 0

TNUV = 0

TC = 0

XPREV = 0

Switch FLGWRD10 bit 7 (REINTFLG) to 1

Perform "MOVEACSM"

Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

Perform "SVDWN1"

Set FLAGWRD8 bit 12 (CMOONFLG) = FLAGWRD0 bit 12 (MOONFLAG)

Perform "INTWAKE1"

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

ENDP76 TRKMKCNT = 0

Proceed to "GOTOPOOH"

A5: Double precision intermediate quantity used in calculation of XKEPNEW, scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

ALPHAM: Double precision magnitude of ALPHAV, with identical scaling and units.

ALPHAV: Working storage for double precision position deviation vector, scaled B22 (earth) or B18 (moon) in units of meters, or for total position vector, scaled B29 (earth) or B27 (moon) in units of meters. Actually, ALPHAV is changed to a unit vector, but this was not shown for ease in presentation.

BETAM: Double precision magnitude of BETAV or magnitude of $K:MU_i$ with variable scaling and units.

BETAV: Double precision total position vector of the spacecraft in relation to the earth, moon or sun, expressed in reference coordinates with variable scaling in units of meters.

COSØ: Double precision argument for "OBLATE" equations, scaled B1 and unitless. Equivalent to the cosine of the angle between the unit polar vector (earth or moon) and the total position vector (earth or moon centered coordinates).

DdBETA: Double precision ratio of D divided by BETAM where D is the magnitude of the vector D = BETAV - ALPHAV; scaled B1 and unitless.

DELVLVC: Double precision value of velocity increment in local vertical coordinates, scale factor B7, units meters per centisecond.

DELVOV: Double precision, instantaneous delta-velocity vector to be added (at time specified in TIG) to the permanent CSM state vector maintained in the LGC to reflect a burn performed by the CSM; scaled B7 in units of meters per centisecond and expressed in local vertical coordinates of the CSM at TIG.

DIFEQCNT: Single precision cell used for program control purposes, having values of 0, -12 and -24 at the beginning, middle and end of each integration step, scaled B14 and unitless.

DTd2: Double precision time increment for precision integration corresponding to half of one integration time-step, scaled B19 in units of centiseconds.

E32C31RM: Single precision erasable memory constant, scale factor B80, giving information on the C_{31} term for the lunar gravity model. The cell contains $C_{31} \times 1.5 \times r_M^3 \times mu_m$. The value corresponds to $C_{31} \times 1.5 \times (1.73809 \times 10^6)^3 \times 0.4902778 \times 10^9 \times 2^{-80}$.

ORBI - 26

E3J22R2M: Single precision erasable memory constant, scale factor B58, giving information on the J_{22} term for the lunar gravity model. The cell contains $J_{22} \times 3 \times r_M^2 \times mu_m$. The value corresponds to $J_{22} \times 3 \times (1.73809 \times 10^6)^2 \times 0.4902778 \times 10^9 \times 2^{-58}$.

EGRESS: Single precision octal return address storage cell.

Fq: Special function used in Encke's method of integration to achieve greater accuracy, double precision, scaled B1. See pages 11 and 190 in Astronautical Guidance by R. H. Battin (McGraw-Hill, 1964).

FV: Double precision disturbing acceleration vector, scaled B-16 (earth) or B-20 (moon) in units of meters per centisecond squared. See pages 189-191 of Astronautical Guidance by R. H. Battin (McGraw-Hill, 1964)

H: Double precision time since beginning of integration step, scaled B19 in units of centiseconds.

i: Single precision index scaled B14.

IRETURN, IRETURN1: Single precision octal return address storage cells.

j: Single precision index scaled B14.

KEPRTN: See CONC section.

K:3CSECS: Double precision constant stored as 3×2^{-28}, scaled B28 in units of centiseconds. Equation value: 3.

K:DTd2MAX: Double precision constant stored as $4 \times 10^5 \times 2^{-20}$, scaled B19 in units of centiseconds. Equation value: 200,000.

K:DTd2MIN: Double precision constant stored as 3×2^{-20}, scaled B19 in units of centiseconds. Equation value: 1.5.

K:$j2_0$: Double precision constant stored as $1.75501139 \times 10^{21} \times 2^{-72}$, program notation "J2REQSQ", scaled B72 in units of $meters^5$/centiseconds squared. Equation value: $1.75501139 \times 10^{21}$. (Equivalent to $3.986032 \times 10^{10} \times (6.378165 \times 10^6)^2 \times 0.10823 \times 10^{-2}$.)

K:$j2_2$: Double precision constant stored as $3.067493316 \times 10^{17} \times 2^{-60}$, program notation "J2REQSQ-2" , scaled B60 in units of $meters^5$/centisecond squared. Equation value: $3.067493316 \times 10^{17}$. (Equivalent to $0.4902778 \times 10^9 \times (1.738090 \times 10^6)^2 \times 0.207108 \times 10^{-3}$.)

K:$j3j2_0$: Double precision constant stored as $-1.355426363 \times 10^4 \times 2^{-27}$, program notation "2J3RE/J2", scaled B27 in units of meters. Equation value: -1.355426363×10^4. (Equivalent to $-0.23 \times 10^{-5} \times 6.378165 \times 10^6 / 0.10823 \times 10^{-2}$.)

K:j3j2_2: Double precision constant stored as $-1.7623602 \times 10^5 \times 2^{-25}$ program notation "2J3RE/J2-2", scaled B25 in units of meters. Equation value: -1.7623602×10^5. (Equivalent to $(-2.1 \times 10^{-5}/0.207108 \times 10^{-3}) \times 1.73809 \times 10^6$.)

K:j4j3_0: Double precision constant stored as $4.991607391 \times 10^6\ 2^{-26}$, program notation "J4REQ/J3," scaled B26 in units of meters. Equation value: 4.991607391×10^6. (Equivalent to $-0.18 \times 10^{-5} \times 6.378165 \times 10^6/-0.23 \times 10^{-5}$.)

K:j4j3_2: Double precision constant stored as 0, program notation "J4REQ/J3-2." Equation value: 0.

K:MU_0: Double precision constant stored as $3.986032 \times 10^{10}\ 2^{-36}$, program notation "MUEARTH", scaled B36 in units of meters cubed/centisecond squared. Equation value: 3.986032×10^{10}.

K:MU_2: Double precision constant stored as $4.9027780 \times 10^8\ 2^{-30}$, program notation "MUM", scaled B30 in units of meters cubed/centisecond squared. Equation value: 4.9027780×10^8.

K:MU_4: Double precision constant stored as $1.32715445 \times 10^{16}\ 2^{-54}$, program notation "MUEARTH-4," scaled B54 in units of meters cubed/centisecond squared. Equation value: $1.32715445 \times 10^{16}$.

K:OMEGMOON: Double precision constant stored as $2.66169947 \times 10^{-8}\ 2^{23}$, scaled B-23 in units of radians per centisecond. Equation value: $2.66169947 \times 10^{-8}$.

K:p3D: Double precision constant stored as 0.3×2^{-2}, scaled B2 and unitless. Equation value: 0.3.

K:RD_0: Double precision constant stored as $80{,}467{,}200 \times 2^{-29}$, program notation "RDE," scaled B29 in units of meters. Equation value: 80,467,200.

K:RD_2: Double precision constant stored as $16{,}093{,}440 \times 2^{-27}$, program notation "RDM," scaled B27 in units of meters. Equation value: 16,093,440.

K:RECRATIO: Double precision constant stored as 0.01, scaled B0 and unitless. Equation value: 0.01.

K:rectr: Double precision constant stored as 0.75, program notation "3/4," scaled B22 (earth) or B18 (moon) in units of meters. Equation value: 3,145,728 (earth) or 196,608 (moon).

K:rectv: Double precision constant stored as 0.75, program notation "3/4," scaled B3 (earth) or B-1 (moon) in units of meters per centisecond. Equation value: 6 (earth) or 0.375 (moon).

K:RM$_0$: Double precision constant stored as $1 - 2^{-28}$, program notation "RME," scaled B29 in units of meters. Equation value: 536,870,910.

K:RM$_2$: Double precision constant stored as $1 - 2^{-28}$, program notation "RMM," scaled B27 in units of meters. Equation value: 134,217,727.5.

K:RSPHERE: Double precision constant stored as $64{,}373{,}760 \times 2^{-29}$, scaled B29 in units of meters. Equation value: 64,373,760.

K:TIMEDELT: Double precision constant stored as 2000×2^{-28}, scaled B28 in units of centiseconds. Equation value: 2000.

K:UNITX, K:UNITZ: See SERV section.

LM504: Double precision libration vector of the moon, scaled B0 in units of radians, and expressed in moon-centered, moon-fixed coordinates.

MODREG: Program number from DSKY; see DATA section.

[MOONMAT]: See COOR section.

MUDEX: See CONC section.

n: Single precision index scaled B14 and used to indicate the column of the W-matrix that is being integrated.

P2', P3', P4', P5': Double precision Legendre polynomial derivatives, scaled B6, B5, B7 and B10, respectively.

PBODY: Single precision index used to distinguish between constants pertaining to the earth (0) and the moon (2); scaled B14.

PHIV: Double precision intermediate storage for the disturbing acceleration vector, scaled B-13 (earth) or B-17 (moon) in units of meters per centisecond squared. Used to implement the second-order difference equation,

$$\underline{R}_{t+h} = \underline{R}_t + h \left[\underline{V}_t + \frac{h}{6} \left(2\, \underline{a}_{t+\frac{1}{2}h} + \underline{a}_t\right)\right]$$

which is valid to the fourth degree. ($\underline{R}_{t+h}$ is the position deviation vector at time t + h, etc.)

PIPTIME, PIPTIME1: See SERV section.

PSIV: Double precision vector storage for intermediate values of disturbing acceleration, scaled B-13 (earth) or B-17 (moon) in units of meters per centisecond squared. Used to implement the first-order difference equation,

$$\underline{V}_{t+h} = \underline{V}_t + \frac{h}{6} (\underline{a}_{t+h} + 4\,\underline{a}_{t+\frac{1}{2}h} + \underline{a}_t)$$

which is valid to the fourth degree. ($\underline{a}_{t+\frac{1}{2}h}$ is the second derivative of the position deviation at time $t + \frac{1}{2}h$, etc.)

q: Double precision argument for Fq, scaled B2 and unitless.

Q: Double precision intermediate quantity used in calculating an initial guess for the universal conic variable; scaling varies downward from B0 (earth) and B2 (moon); units of centiseconds per meter.

QPRET: Single precision octal return address associated with a particular job, saved when a job is put to sleep; actually one of several mutually exclusive cells, each addressed only within a particular job. "QPRET" for one job is independent of and not affected by the address stored in "QPRET" by another job. See MATX section.

RATT: Double precision position vector output from orbital integration, valid at time TAT, scaled B29 in units of meters. (Called RATT1 and scaled B29 (earth) or B27 (moon) for output to routines desiring segregated scaling.)

RCV: Double precision conic portion of the position vector at TET, computed from the osculating conic at rectification (TET - TC), scaled B29 (earth) or B27 (moon) in units of meters.

RCVCSM, RCVLEM: The permanent state vectors for the CSM and LM contain six double precision vectors and three double precision scalars - a total of twenty-one double precision components. They are listed below along with the name of the equivalent working variable used in precision integration of each.

CSM	LM	
RCVCSM	RCVLEM	RCV
VCVCSM	VCVLEM	VCV
TCCSM	TCLEM	TC

CSM	LM	
DELTACSM	DELTALEM	TDELTAV
NUVCSM	NUVLEM	TNUV
TETCSM	TETLEM	TET
RRECTCSM	RRECTLEM	RRECT
VRECTCSM	VRECTLEM	VRECT
XKEPCSM	XKEPLEM	XPREV

RHO: Ratio of position deviation to total distance from the primary body, or ratio of spacecraft radius to radius of secondary body (moon or sun, earth or sun), scaled B1 and unitless.

RLS: See CONC section.

RN, RN1: See SERV section.

ROOTMU: See CONC section.

RPQV: Double precision position vector of secondary body with respect to the primary body, scaled B29 in units of meters; expressed in reference coordinates.

RPSV: Double precision position vector of the sun with respect to the primary body, scaled B38 in units of meters and expressed in reference coordinates.

RQVV: Double precision spacecraft position vector with respect to the secondary body, scaled B29 in units of meters; expressed in reference coordinates.

RRECT: Double precision total position vector at the last rectification, scaled B29 (earth) or B27 (moon) in units of meters. See CONC section.

RTX1, RTX2: Single precision values of index registers X1 and X2 respectively at the end of integration to identify the nature of the origin of the state vector.

TAT: Double precision state vector time-tag output from orbital integration, scaled B28 in units of centiseconds.

TAU: See CONC section.

TBASE2: Single precision cell normally used to retain time base information for restart group 2, for waitlist restarts. It is used in "INTWAKE" to retain the value of QPRET for restart purposes.

TC: See CONC section.

TDEC: Double precision time at the end of the desired integration interval, scaled B28 in units of centiseconds.

TDEC1: Double precision storage for TDEC in individual job registers so that several jobs can maintain different values of TDEC simultaneously. (See description of QPRET).

TDELTAV: Double precision deviation-from-conic-position vector, scaled B22 (earth) or B18 (moon) in units of meters.

TET: Double precision time associated with the most recently computed state vector, scaled B28 in units of centiseconds.

TETCSM: See description of RCVCSM.

TIG: See BURN section.

TIMENOW: Current time, B28, centiseconds. See EXVB section.

TNUV: Double precision deviation-from-conic-velocity vector, scaled B3 (earth) or B-1 (moon) in units of meters per centisecond.

TRKMKCNT: See RNAV section.

TSsun: Double precision vector contents of the MPAC when return from "LSPOS", representing the unit position vector of the sun, scaled B1 and unitless. See COOR section. (It should be noted that TSsun is a unit vector scaled B1 but RPQV, into which it is stored, is scaled B29. However, the affected section should not be entered in LUMINARY).

UPSVFLAG: Single precision flag loaded with a state vector update (address of UPSVFLAG is just before that of RRECT) to indicate whether update is for LM state or CSM state and whether it is in moon-centered or earth-centered coordinates, scaled B14 and unitless. See ORBI - 22.

URPV: Double precision position vector in moon-centered, moon-fixed coordinates, scaled B1 and unitless.

UX: Double precision lunar X-axis expressed in reference coordinates for the oblateness calculations, scaled B1 and unitless.

UZ: Double precision polar vector for earth or moon expressed in reference coordinates for the oblateness calculations, scaled B1 and unitless.

VATT: Double precision velocity vector output from orbital integration, valid at time TAT, scaled B7 in units of meters per centisecond. (Called VATT1 and scaled B7 (earth) or B5 (moon) for output to routines desiring segregated scaling.)

VCV: Double precision conic portion of the velocity vector at TET, computed from the osculating conic at rectification (TET - TC), scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

$VECTAB_i$: Temporary storage for total position vectors with respect to primary and one secondary body at the beginning, middle and end of each integration time-step for use in W-matrix propagation.

VN, VN1: See SERV section.

VRECT: Double precision total velocity vector at last rectification scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

[W]: See RNAV section.

X1, X2: Index registers one (1) and two (2) respectively.

XKEPNEW, XPREV: See CONC section.

YV: Double precision position deviation vector scaled B22 (earth) or B18 (moon) in units of meters. (State vector or W-matrix.)

ZV: Double precision velocity deviation vector scaled B3 (earth) or B-1 (moon) in units of meters per centisecond. (State vector or W-matrix.)

SLAP1	Inhibit interrupts	
	Perform "STARTSUB"	
	Switch $DSPTAB_{11}$ to $100\ 000\ 000\ x0x\ 000_2$	(INTR)
	ERCOUNT = 0	(TEST)
	$FAILREG_i = 0$ for i = 0, 1, 2	(PGSR)
	REDOCTR = 0	(PGSR)
	DSRUPTSW = -5120	(INTR)
DOFSTART	Switch channel 11 to 20000_8 (bit 14 is engine off)	
	THRUST = -0	(DESC)
DOFSTRT1	RCSFLAGS = 00004_8	(DAPA)
	ABDELV = 0	(SERV)
	NVSAVE and $DSPFLG_2$ = 0	(PGSR)
	CH5MASK, CH6MASK, and PVALVTEST = 0	(DAPA)
	ERESTORE = 0	(TEST)
	SMODE = +0	(TEST)
	DNLSTCOD = 0	(TELE)
	AGSWORD = 0	(EXVB)
	UPSVFLAG = 0	(ORBI)
	Switch channels 5, 6, 12, 13, and 14 to 00000_8	
	If $DSPTAB_{11}$ bit 4 (no attitude) and bit 6 (gimbal lock warning) both = 1:	
	Switch channel 12 to 00050_8	
	Make all restart groups inactive	
	MODREG = -0	
	IMODES30 = 37411_8	(IMUC)
	DB = K:MAXDB	(DAPB)
	RATEINDX = 2	(DAPB)
	DAPBOOLS = 21322_8	(DAPA)

EBANK = K:EBANK6

STIKSENS = K:STIKSTRT (DAPA)

RATEDB = K:RATESTRT (DAPA)

HIASCENT = K:FULLAPS (DAPB)

DKTRAP = K:77001OCT (DAPA)

LMTRAP = K:77001OCT (DAPA)

DKKAOSN and LMKAOSN = 60 (DAPA)

LMOMEGAN = 0 (DAPA)

DKOMEGAN = 10 (DAPA)

DKDB = 00200_8 (DAPB)

IMODES33 = 16040_8 (INTR)

Switch $FLAGWRD_i$ to 00000_8 for i = 0, 1, 2, 4, 5, 6, 9

FLAGWRD7 = 00100_8

FLAGWRD8 = $000\ xx0\ 0x0\ 000\ 000_2$ (leave bit 12 (CMOONFLG), bit 11 (LMOONFLG), and bit 8 (SURFFLAG) alone)

FLAGWRD3 = $00x\ 010\ 000\ 000\ 000_2$ (leave bit 13 (REFSMFLG) alone)

FLGWRD10 = $00x\ 000\ 000\ 000\ 000_2$ (leave bit 13 (APSFLAG) alone)

FLGWRD11 = 40000_8 (bit 15 is LRBYPASS)

Proceed to "DUMMYJB2"

STARTSUB DNTMGOTO = "DNPHASE1" (TELE)

RADMODES = 00102_8 + bit 6 of channel 33 (LR pos) (RADR)

STARTSB1 Set TIME3 to cause program interrupt #3 in 10 milliseconds

Set TIME4 to cause program interrupt #4 in 30 milliseconds

Set TIME5 to cause program interrupt #2 in 40 milliseconds

EBANK = K:EBANK6

Switch RCSFLAGS bit 13 to 0 (DAPA)

$T6NEXTTM_0$ = 37777_8 (disable TIME6 clock) (DAPA)

Switch bit 15 of channel 13 to 0

NXT6AXIS = 0 (DAPA)

NEXTP = 00000_8 (DAPA)

Switch DAPBOOLS bit 3 (ACCSOKAY) to 0 (DAPA)

T5ADR = "DAPIDLER" (DAPA)

If FLGWRD11 bit 15 (LRBYPASS) = 1, proceed to "STARTSB2"

If STILBADH > 0, skip next line (SERV)

STILBADH = STILBADH + 1

If STILBADV > 0, proceed to "STARTSB2" (SERV)

STILBADV = STILBADV + 1

STARTSB2 Switch channel 11 to $0xx\ 000\ 000\ 000\ 00x_2$
(leave engine on/off and ISS warning alone)

Switch FLAGWRD3 bit 9 (READRFLG) to 0

If FLAGWRD3 bit 11 (NOR29FLG) ≠ 0:

RADMODES = $x00\ x00\ xxx\ xxx\ xx0_2$

Skip next line

RADMODES = $x00\ x0x\ xxx\ xxx\ xx0_2$

Switch channel 12 to $0x0\ xxx\ x00\ xxx\ 000_2$

Switch FLAGWRD5 bit 4 (NORRMON) and bit 11 (R77FLAG) to 0

Switch channel 13 to $xxx\ 100\ 00x\ xx0\ 000_2$

Switch channel 14 to $000\ 000\ 000\ x00\ 000_2$

EBANK = K:STARTEB

Set all 8 waitlist times to 81.93 seconds (MATX)

Set all 9 waitlist task addresses to "SVCT3" (MATX)

Make all 8 job register sets available (MATX)

DSRUPTSW = -0 (INTR)

NEWJOB = -0 (MATX)

Make all 5 VAC areas available	(MATX)
$DSPTAB_i = -04000_8$ for i = 0-10 (negative for output)	(DSKY)
$DELAYLOC_i = 0$ for i = 0, 1, 2	(MATX)
R1SAVE = 0	(DINT)
INLINK = 00000_8	(TELE)
DSPCNT = 0	(INTR)
CADRSTOR = +0	(DINT)
REQRET = +0	(DSKY)
CLPASS = 0	(DSKY)
DSPLOCK = 0	(DSKY)
MONSAVE and MONSAVE1 = 0	(DATA)
VERBREG and NOUNREG = 0	(DATA)
DSPLIST = +0	(DINT)
MARKSTAT = +0	(ALIN)
EXTVBACT = 0	(EXVB)
IMUCADR = +0	(IMUC)
OPTCADR = +0	(ALIN)
RADCADR = +0	(RADR)
ATTCADR = +0	(ATTM)
LGYRO = +0	(IMUC)
FLAGWRD4 = 00000_8 (kill display interface routine action)	
NOUT = 11	(INTR)
SAMPLIM = -1	(RADR)

IMODES33 = $001\ 110\ 000\ x00\ 000_2$ (set PIPA good, downlink good, uplink good bits; leave DAP disable bit alone)

SELFRET = "SELFCHK" (TEST)

DSPCOUNT = -19 (DSKY)

Return

__V37__ If IMODES30 bit 6 = 1: (IMU caged)

  Perform "ALARM" with TS = 01520_8

  Proceed to "V37BAD"

If MMNUMBER = 70, proceed to "P70"

If MMNUMBER = 71, proceed to "P71"

If MMNUMBER = 0:

  If FLAGWRD7 bit 6 (V37FLAG) = 0, proceed to "CANV37"

  (Otherwise, "SERVICER" is running; cause it to exit to "CANV37")

  Switch FLAGWRD7 bit 5 (AVEGFLAG) to 0

  End job

If FLAGWRD2 bit 1 (NODOFLAG) = 1:

  Perform "ALARM" with TS = 01520_8

  Proceed to "V37BAD"

If MMNUMBER ≠ low 7 bits of $K{:}PREMM1_i$ for some i from 0 to 24

  Switch bit 7 of channel 11 to 1 (operator error)

  Proceed to "V37BAD"

MINDEX = i for which MMNUMBER = low 7 bits of $K{:}PREMM1_i$

If FLAGWRD7 bit 6 (V37FLAG) = 0, proceed to "CANV37"

Switch FLAGWRD7 bit 5 (AVEGFLAG) to 0

End job

__V37BAD__ Perform "RELDSP"

Proceed to "PINBRNCH" (reinstitute any interrupted display)

__CANV37__ SUPERBNK = 0

Perform "INTSTALL" (wait until integration is free)

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Switch FLAGWRD3 bit 15 (POOHFLAG) to 0

Switch FLGWRD11 to 40000_8

Switch FLAGWRD3 bits 9 (RO4FLAG) and 14 (GLOKFAIL) to 0

Switch FLAGWRD6 bit 8 (MUNFLAG) to 0

Switch FLAGWRD9 bit 7 (ABTTGFLG) to 0

Switch DAPBOOLS bit 9 (XOVINHIB) to 0

If MMNUMBER = 0, proceed to "POOH"

If FLAGWRDO bit 7 (RNDVZFLG) and 9 (P25FLAG) both = 0:

    Switch FLAGWRDO bit 8 (IMUSE) to 0

DNLSTCOD = $K:DNLADMM1_{MINDEX}$

Inhibit interrupts

Proceed to "SEUDOPOO"

POOH Perform "RELDSP"

Inhibit interrupts

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0 (disable RR Error counter)

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

Clear P20, P25 restart logic and cause "GOPROG3" to establish "STATINT1" (pr05)

Switch FLAGWRDO bits 7 (RNDVZFLG), 8 (IMUSE) and 9 (P25FLAG) to 0

DNLSTCOD = 0

SEUDOPOO AGSWORD = DNLSTCOD

Perform "ENGINOF1"

Perform "ALLCOAST"

$DSPFLG_2 = 77657_8$

Switch FLAGWRD1 bit 5 (TRACKFLG) and 7 (UPDATFLG) to 0

If MMNUMBER = 0:

Maintain "STATINT1" restart logic and clear all other

MODREG = 0

Proceed to "GOPROG2"

If MODREG or MMNUMBER = 22:

RESET22 — Switch FLAGWRDO bit 7 (RNDVZFLG), bit 8 (IMUSE) and bit 9 (P25FLAG) to 0

Switch RADMODES bit 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch channel 12 bit 2 to 0

Clear P20, P25 restart logic

Cause "GOPROG3" to call "V37XEQ"

Proceed to "GOPROG2"

If MMNUMBER ≠ 20 or 25:

If FLAGWRDO bit 7 (RNDVZFLG) or bit 9 (P25FLAG) = 1:

Proceed to 5th step of "RESET22"

Proceed to "RESET22"

If MODREG = 20 or 25:

Proceed to 4th step of "RESET22"

If MMNUMBER = 20 and FLAGWRDO bit 7 (RNDVZFLG) = 1 and bit 9 (P25FLAG) = 0, or if MMNUMBER = 25 and FLAGWRDO bit 9 (P25FLAG) = 1 and bit 7 (RNDVZFLG) = 0:

Switch FLAGWRD1 bit 5 (TRACKFLG) and 7 (UPDATFLG) to 1

Maintain P20, P25 restart logic and clear all other

MODREG = MMNUMBER

Proceed to "GOPROG2"

Proceed to 4th step of "RESET22"

V37XEQ Inhibit interrupts

MMTEMP = K:PREMM1$_{MINDEX}$

ii = bits 15-11 of MMTEMP (priority)

BASETEMP = K:FCADRMM1$_{MINDEX}$

Establish job specified in BASETEMP (EBANK information in bits 8-10 of MMTEMP) (prii)

MODREG = MMNUMBER

Establish "DSPMMJOB" (pr30)

Perform "RELDSP"

Release interrupt inhibit

End job

VBRQEXEC (Entered from "VERBFAN" on verb 30; DSPTEM1$_{tp}$ must be loaded with three single precision components as follows:

DSPTEM1$_0$	0pp ppp 000 000 00v
DSPTEM1$_1$	000 sss sss sss sss
DSPTEM1$_2$	fff ff0 00x xx0 eee

where "p" represents one of the bits of the five-bit priority to be assigned to the job; "v" is a single binary bit indicating whether the job is to be assigned working storage (1) or not (0); "s" is one of the 12 bits of the "S-register" portion of the address; "f" is one of the bits of the five-bit fixed-memory-bank code; "x" is one of the bits in the three-bit fixed-bank-extension code; "e" is one of the bits in the three-bit erasable memory-bank code; and "0" is a binary zero.)

Perform "RELDSP"

Establish job whose starting address is contained in 2CADR form in DSPTEM1$_1$ and DSPTEM1$_2$, with priority and storage allocation according to the information in DSPTEM1$_0$. ("FINDVAC" or "NOVAC")

End job

VBRQWAIT (Entered from "VERBFAN" on verb 31; DSPTEM1$_{tp}$ must be preloaded with three single precision components as follows:

$DSPTEM1_0$ time in centiseconds scaled B14
$DSPTEM1_1$ 000 sss sss sss sss
$DSPTEM1_2$ fff ff0 00x xx0 eee

where "s, f, x" and "e" have the meanings described above.)

Perform "RELDSP"

Call the task whose starting address is contained in 2CADR form in $DSPTEM1_1$ and $DSPTEM1_2$, in $DSPTEM1_0$ centiseconds. ("WAITLIST")

End job

<u>GOPROG</u> REDOCTR = REDOCTR + 1 (hardware restart)

RSBBQ = address of step performed before restart occurred

If bit 4 of $DSPTAB_{11}$ = 1: (no attitude lamp on)

Switch bits 4 and 6 of channel 12 to 1
(coarse align discrete, ICDU Error Counter enable)

Perform "LIGHTSET"

If bits 15-11 of ERESTORE ≠ 0:

Perform "STARTSUB"

Proceed to "DOFSTRT1"

If ERESTORE ≠ 0:

If ERESTORE ≠ SKEEP7:

Perform "STARTSUB"
Proceed to "DOFSTRT1"

EBANK = bits 11-9 of SKEEP4

i = SKEEP7

E_i = SKEEP5

i = SKEEP7 + 1

E_i = SKEEP6

ERESTORE = 0

Perform "STARTSUB"

Switch FLGWRD10 bit 14 (INTFLAG) to 0

Switch $DSPTAB_{11}$ to $100\ 000\ x00\ x0x\ 000_2$
(bit 15 is flag for output; leave bits 9,6,4 alone)

Switch IMODES30 to 011 111 x00 0xx $x0x_2$
    (leave IMU, CDU, and both PIPA fail monitor bits alone)

DNLSTCOD = AGSWORD

Switch bit 4 of channel 14 to 1 (thrust drive discrete)

If FLAGWRD5 bit 7 (ENGONFLG) = 1:

    Switch bit 13 of channel 11 to 1 (engine on discrete)

    Skip next line

Switch bit 14 of channel 11 to 1 (engine off discrete)

Proceed to "GOPROG3"

LIGHTSET If bit 5 channel 16 = 1 and channel 15 = 22_8:

    Perform "STARTSUB"

    Proceed to "DOFSTART"

Return

ENEMA Inhibit interrupts

Perform "STARTSB1"

Skip next line

GOPROG2 Perform "STARTSB2"

GOPROG2A Perform "LIGHTSET"

Switch FLGWRD10 bit 7 (REINTFLG) and bit 14 (INTFLAG) to 0

GOPROG3 If restart information is stored improperly:

    Perform "ALARM" with TS = 01107_8

    Proceed to "DOFSTRT1"

Establish "DSPMMJOB" (pr30)

Release interrupt inhibit
(allows possibility of interrupt)
Inhibit interrupts

Switch FLAGWRD1 bit 14 (DIDFLAG) to 0

Switch FLAGWRD1 bit 12 (RODFLAG) to 0

Switch FLAGWRD0 bit 11 (P21FLAG) to 0

If all restart groups are inactive:

  If bit 15 of MODREG = 1, proceed to "ENDRSTRT"

  Proceed to "GOTOPOOH" ("GOFLASH" will put "DSPMMJOB" to sleep, leaving program number blank)

Restart all jobs and tasks indicated by active restart groups

<u>ENDRSTRT</u> Proceed to "DUMMYJB2"

<u>GOTOPOOH</u> Switch DAPBOOLS bit 9 (XOVINHIB) and bit 6 (ULLAGER) to 0

Inhibit interrupts

Switch FLAGWRD4 bit 1 (XDSPFLAG) to 0

Release interrupt inhibit

Proceed to "GOFLASH" with TS = K:V37N99 (noun not processed)
(If terminate, repeat this step; if proceed, repeat this step; if other response, repeat this step.)

<u>ALARM</u> Inhibit interrupts

$ALMCADR_0$ = "calling address + 1" (S-register portion)

<u>ALARM2</u> $ALMCADR_1$ = BBANK + SUPERBNK (or'ed into bits 7-5)

If $FAILREG_0$ = 0:

  $FAILREG_0$ = TS (TS contains alarm code)

  Proceed to "PROGLARM"

If $FAILREG_1$ = 0:

  $FAILREG_1$ = TS

  Proceed to "PROGLARM"

<u>PROGLARM</u> $FAILREG_2$ = TS

Switch bit 9 of $DSPTAB_{11}$ to 1 and flag for output

Release interrupt inhibit

Return

BAILOUT Inhibit interrupts

$ALMCADR_0$ = "calling address + 1" (S-register portion)

Perform "ALARM2"

WHIMPER Resume (after this Resume, return is to next line)

Proceed to "ENEMA"

POODOO Inhibit interrupts

$ALMCADR_0$ = "calling address + 1" (S-register portion)

Perform "ALARM2" (TS contains the alarm code)

Switch FLAGWRD3 bit 5 (STATEFLG) to 0

Switch FLGWRD10 bit 7 (REINTFLG) to 0

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

If FLAGWRD7 bit 6 (V37FLAG) = 1; proceed to "SERVIDLE"

Make all restart groups inactive

Proceed to "WHIMPER"

CURTAINS Inhibit interrupts

$ALMCADR_0$ = "calling address + 1"

Perform "ALARM2" with TS = 00217_8

Return

BAILOUT1 Inhibit interrupts

$ALMCADR_{dp}$ = $TS1_{dp}$

Perform "ALARM2" starting at second line

Inhibit interrupts

Proceed to "WHIMPER"

POODOO1 Inhibit interrupts

$ALMCADR_{dp}$ = $TS1_{dp}$

Perform "ALARM2" starting at second line

Proceed to 4th line of "POODOO"

ABORT   Proceed to "WHIMPER"

P06   Switch FLAGWRD2 bit 1 (NODOFLAG) to 1

Inhibit interrupts

TIME2SAV = TIMENOW

TSt_{dp} = channel 3, channel 4   (sampled with special precaution to assure that the two halves are consistent)

Release interrupt inhibit

If TIMENOW was incremented during the last 1/3200 of a second (channel 4 = xxx.5 centiseconds):   (TIMENOW incompatible with channel 4)

Proceed to second step of "P06"

TSt = TSt rounded to the nearest centisecond

SCALSAVE = TSt

Inhibit interrupts

Perform "RNDREFDR"

Switch FLAGWRDO bit 8 (IMUSE) and bit 7 (RNDVZFLG) to 0

Switch bit 11 of channel 13 to 1   (enable standby)

Set restart tables to establish "POSTAND" when a restart is triggered by the recovery from standby (pr20)

Proceed to "GOPERF1" with TS = 00062_8
(If terminate, repeat this step; if proceed, repeat this step; if other response, repeat this step.)

POSTAND   Switch bit 11 of channel 13 to 0   (disable standby)

Inhibit interrupts

TIMENOW = 0

TSt_{dp} = channel 3, channel 4   (sampled with special precaution to assure that the two halves are consistent)

Release interrupt inhibit

If TIMENOW was incremented during the last 1/3200 of a second (channel 4 = xxx.5 centiseconds):

Proceed to second step of "POSTAND"

TSt = TSt rounded to the nearest centisecond

TS = TSt - SCALSAVE (rescaled from B23 to B28)

Force sign agreement in TS

If TS $\leq$ -0, TS = 2^{23} + TS

TSt = TIME2SAV + TS

Force sign agreement in TSt

TIMENOW = TIMENOW + TSt

Switch FLAGWRD2 bit 1 (NODOFLAG) to 0

Proceed to "GOTOPOOH"

V37RET If FLAGWRDO bit 7 (RNDVZFLG) = 1:

Cause "GOPROG3" to call "P20LEMC1" in 15 seconds

Proceed to "CANV37"

If FLAGWRDO bit 9 (P25FLAG) = 1:

Cause "GOPROG3" to establish "P25LEM1" (pr14)

Proceed to "CANV37"

ALARM1 Inhibit interrupts

$ALMCADR_{dp}$ = $TS1_{dp}$

Proceed to second step of "ALARM2"

VARALARM Inhibit interrupts

$ALMCADR_0$ = "calling address + 1"

Perform "ALARM2"

Proceed to $ALMCADR_0$

CCSHOLE   Inhibit interrupts

$ALMCADR_0$ = "calling address + 1"   (S-register portion)

Proceed to third line of "POODOO" with TS = 21103_8

AGSWORD: See EXVB section.

ALMCADR: Double precision storage for return address (address in most significant half, bank information in least significant half) of the routine that generated the latest alarm.

BASETEMP: Single precision storage for octal address of program to be started by verb 37.

BBANK: A computer hardware cell containing in bits 15-11 the fixed memory bank (FBANK) currently being used and in bits 3-1 the erasable. memory bank number.

DNLSTCOD: See TELE section.

$DSPFLG_2$: See DINT section.

$DSPTAB_{11}$: See INTR section.

DSPTEM1: See DATA section.

E_i: Single precision memory cell whose address is in i.

EBANK: See MATX section.

ERESTORE: See TEST section.

$FAILREG_i$ (i = 0,1,2): Three single precision registers used for storage of alarm code information. $FAILREG_{0,1}$ are zeroed via an "error reset", $FAILREG_2$ is unaltered. All three registers are zeroed by a Verb 36 (fresh start). $FAILREG_0$ contains the first alarm code generated after the "Error Reset"; $FAILREG_1$ contains the second; and $FAILREG_2$ always contains the most recent.

IMODES30: See IMUC section.

IMODES33: See IMUC section.

K:77001OCT: Single precision constant stored as 77001_8, scaled B-3 in units of revolutions per second. Equation value: +0.00389. (Equation value: 1.4 degrees per second.)

$K:DNLADMM1_i$ (i = 0-24): Table of 25 single precision indexes which determine the downlist sent during each major mode. See table below.

K:EBANK6: Single precision constant stored as 03000_8, scaled B6 and unitless. Equation value: 6.

$K:FCADRMM1_i$ (i = 0-24): Table of 25 single precision addresses of the 25 major mode programs. See table below.

K:FULLAPS: Single precision constant stored as 5050×2^{-16}, scaled B16 in units of kilograms. Equation value: 5050.

K:MAXDB: Single precision constant stored as 03434_8; used to initialize the attitude deadband. Scaled B-3 in revolutions and corresponds to approximately 5 degrees.

K:PREMM1_i (i= 0-24): Table of 25 major mode numbers with associated EBANK settings and priorities.

K:RATESTRT: Single precision constant stored as 77445_8, scaled B-3 in units of revolutions/second. Used to initialize location -RATEDB (referred to as RATEDB in DAPA section) in "DOFSTRT1". Equation value: -218.

K:STARTEB: Single precision constant stored as 01400_8, scaled B6 and unitless. Equation value: 3.

K:STIKSTRT: Single precision constant stored as 32321_8, scaled B-15 in revolutions per second/RHC counts. Used to initialize location STIKSENS in "DOFSTRT1". Equation value: 0.825268.

MINDEX: Single precision register used to select the appropriate table entries for a V37 selected program change (loaded based upon equality of MMNUMBER and bits 7-1 of K:PREMM1_i with the value of i.)

MMNUMBER: Single precision storage for the desired value of the major mode register, scaled B14 and unitless.

MMTEMP: Single precision storage for the number of the program being started by verb 37 (bits 7-1, bits 10-8 for EBANK) and for the priority with which the program is to be started (bits 11-15).

MODREG: See DATA section.

NVSAVE: See NVWORD in the DINT section.

RADMODES: See RADR section.

REDOCTR: Single precision counter set to zero in a fresh start and incremented whenever a hardware restart occurs; scaled B14 and unitless.

RSBBQ: Storage for the value of the address where a hardware restart occurred. The most significant part contains the BBANK and SUPERBNK information; the least significant part contains the Q-register information.

SCALSAVE: Double precision value of the standby clock (channels 3 and 4) at the time program 06 enables standby, scaled B23 in units of centiseconds.

SKEEP4, SKEEP5, SKEEP6, SKEEP7: See TEST section.

STILBADH: See SERV section.

STILBADV: See SERV section.

SUPERBNK: See MATX section.

TIME3, TIME4, TIME5: See "Major Variables" section.

TIME2SAV: Double precision value of TIMENOW when program 06 enables standby, scaled B28 in units of centiseconds.

TIMENOW: See EXVB section.

MAJOR MODE CHANGE CONSTANTS

i	K:PREMM1			K:FCADRMM1	K:DNLADMM1
	prio	EBANK	MM	(octal address)	
0	13	7	76	P76	2 (RENDEZVU)
1	13	7	75	P75	2 (RENDEZVU)
2	13	7	74	P74	2 (RENDEZVU)
3	13	7	73	P73	2 (RENDEZVU)
4	13	7	72	P72	2 (RENDEZVU)
5	13	7	68	LANDJUNK	4 (DESASCNT)
6	13	7	63	P63LM	4 (DESASCNT)
7	13	5	57	P57	5 (LUNRSALN)
8	13	5	52	PROG52	0 (COSTALIN)
9	13	5	51	P51	0 (COSTALIN)
10	13	7	47	P47LM	3 (ORBMANUV)
11	13	7	42	P42LM	3 (ORBMANUV)
12	13	7	41	P41LM	3 (ORBMANUV)
13	13	7	40	P40LM	3 (ORBMANUV)
14	13	7	35	P35	2 (RENDEZVU)
15	13	7	34	P34	2 (RENDEZVU)
16	13	7	33	P33	2 (RENDEZVU)
17	13	7	32	P32	2 (RENDEZVU)
18	13	7	30	P30	2 (RENDEZVU)
19	13	7	25	PROG25	2 (RENDEZVU)
20	13	7	22	PROG22	5 (LUNRSALN)
21	13	7	21	PROG21	2 (RENDEZVU)
22	13	7	20	PROG20	2 (RENDEZVU)
23	13	7	12	P12LM	4 (DESASCNT)
24	13	4	06	P06	0 (COSTALIN)

RADR

Radar Control Routines

RRAUTCHK If bit 2 of channel 33 = bit 2 (AUTOMODE) of RADMODES:

Proceed to "RRCDUCHK" (ON/OFF status of RR unchanged)

Set RADMODES bit 2 (AUTOMODE) = bit 2 of channel 33

Switch bits 15 (CDESFLAG), 14 (REMODFLG), 13 (RCDUOFLG), 11 (REPOSMON) and 1 (TURNONFL) of RADMODES to 0

If RADMODES bit 2 (AUTOMODE) = 1: (RR just turned off)

Switch bit 2 of channel 12 to 0 (disable RRCDU Error Counters)

Proceed to "RRCDUCHK"

Switch RADMODES bits 1 (TURNONFL) and 13 (RCDUOFLG) to 1

Call "RRTURNON" in 0.01 second

Proceed to "NORRGMON"

RRCDUCHK If bit 7 of channel 30 = bit 7 (RCDUFAIL) of RADMODES:

Proceed to "RRGIMON" (RR CDU fail discrete unchanged)

If RADMODES bit 2 (AUTOMODE) = 1, proceed to "NORRGMON" (RR not in auto mode: might be reading LR data)

Set RADMODES bit 7 (RCDUFAIL) = bit 7 of channel 30

If RADMODES bits 13 (RCDUOFLG), 7 (RCDUFAIL) and 2 (AUTOMODE) = 0, and FLAGWRD0 bit 7 (RNDVZFLG) = 1:

Perform "ALARM" with TS = 00515_8

Perform "SETTRKF"

RRGIMON If FLAGWRD5 bit 4 (NORRMON) = 1, proceed to "NORRGMON"

If FLAGWRD7 bit 5 (AVEGFLAG) = 1:

If FLAGWRD6 bit 8 (MUNFLAG) = 1, proceed to "NORRGMON"

If RADMODES bit 14 (REMODFLG), 13 (RCDUOFLG), 11 (REPOSMON) or 2 (AUTOMODE) = 1, proceed to "NORRGMON"

Perform "RRLIMCHK" with $TS_0 = CDU_t$ and $TS_1 = CDU_s$

If TSchk = 0: (RR positioned out of limits)

Switch RADMODES bit 11 (REPOSMON) to 1

Switch bits 14 and 2 of channel 12 to 0
(Disable RR tracker and RRCDU Error Counter)

Call "DORREPOS" in 0.02 second

NORRGMON If RADMODES bit 2 (AUTOMODE) = 1:

Switch bit 1 of channel 12 to 1 (Zero RR CDUs)

Return (to caller of "RRAUTCHK" in the T4RUPT routine)

RRTURNON Perform "RRZEROSB"

Delay 1 second

Switch RADMODES bit 1 (TURNONFL) to 0 (Turn-on complete)

End task

RRZEROSB Switch bit 1 of channel 12 to 1 (Drive RR CDUs to zero)

Delay 0.02 second

$CDU_t = 0$

$CDU_s = 0$

Switch bit 1 of channel 12 to 0 (Release RR CDUs)

Delay 10 seconds (Allow time for CDUs to match gimbal angles)

Switch RADMODES bit 13 (RCDUOFLG) to 0

If $|CDU_t| \leq$ K:p25, switch RADMODES bit 12 (ANTENFLG) to 0

If $|CDU_s| >$ K:p25, switch RADMODES bit 12 (ANTENFLG) to 1

Perform "SETTRKF"

Return

SETTRKF If bit 1 of IMODES33 = 1, return (lamp test in progress)

$TS = 00200_8$ (bit 8 = 1)

If RADMODES bit 13 (RCDUOFLG), 7 (RCDUFAIL) or 2 (AUTOMODE) = 1:

    If RADMODES bit 4 (RRDATAFL) = 0, TS = 00000_8 (bit 8 = 0)

If bit 8 of $DSPTAB_{11}$ ≠ bit 8 of TS:

    Set bit 8 of $DSPTAB_{11}$ = bit 8 of TS (tracker fail light)

    Switch bit 15 of $DSPTAB_{11}$ to 1 (flag for output)

Return

DORREPOS Perform "SETRRECR"

Delay 0.02 second

RDES = 0

If RADMODES bit 12 (ANTENFLG) = 1, RDES = $-\frac{1}{2}$ (-180° for Mode II)

Perform "RRTONLY"

RDES = 0

If RADMODES bit 12 (ANTENFLG) = 1, RDES = − K:1s$\frac{1}{4}$ (-90°)

Perform "RRSONLY"

Switch RADMODES bit 11 (REPOSMON) to 0 (repositioning complete)

If RADMODES bit 10 (DESIGFLG) = 1, proceed to "BEGDES"
    (someone waiting to designate)

Switch bit 2 of channel 12 to 0 (disable RR Error Counter)

End task

SETRRECR If bit 2 of channel 12 = 0: (RR Error Counters disabled)

    Switch bit 2 of channel 12 to 1

    LASTTCMD = 0

    LASTSCMD = 0

Return

RRTONLY RRRET = return address

RRINDEX = 0

Proceed to "RR1AX2"

RRSONLY RRRET = return address

RRINDEX = 1

RR1AX2 If RADMODES bits 14 (REMODFLG) and 11 (REPOSMON) both = 1:
(Remode requested while repositioning in progress)

Switch RADMODES bit 11 (REPOSMON) to 0

If RADMODES bit 10 (DESIGFLG) = 1, proceed to "BEGDES"
(Someone waiting to designate)

Switch bit 2 of channel 12 to 0 (disable RRCDU Error Counter)

End task

If RRINDEX = 0, $TS = RDES - CDU_t$ (one's complement difference of two's complement angles)

If RRINDEX = 1, $TS = RDES - CDU_s$

If $|TS| \leq$ K:0.00555 or RADMODES bit 2 (AUTOMODE) = 1, return via RRRET

$TRUNNCMD_i = 0$ for i = 0 and 1

$TRUNNCMD_{RRINDEX}$ = K:RRSPGAIN TS

Perform "RROUT"

Delay 0.5 second

Proceed to "RR1AX2"

RROUT $TS = TRUNNCMD_1$

If $|TS| >$ K:rrlimit, TS = K:rrlimit signTS

CDUSCMD = TS - LASTSCMD

LASTSCMD = TS

$TS = TRUNNCMD_0$

If $|TS| >$ K:rrlimit, TS = K:rrlimit signTS

CDUTCMD = TS - LASTTCMD

LASTTCMD = TS

Switch bits 12 and 11 of channel 14 to 1
(Send contents of CDUTCMD and CDUSCMD to RR CDU's)

Return

RRDESSM DESRET = return address

Switch FLAGWRD0 bit 6 (RRNBSW) to 0

Perform "CDUTRIG"

Perform "SMTONB"

$\underline{TS}los$ = [SMNBMAT] $\underline{R}RTARGET$

Perform "RRANGLES"

Inhibit Interrupts

Perform "RRLIMCHK" with TS_0 = $MODEPRES_0$ and TS_1 = $MODEPRES_1$

If TSchk = 0: (not within limits in present mode)

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

RADCADR = +0

RADLIMCK = 0

Proceed to "DESRETRN"

Perform "RMODINV"

Perform "RRLIMCHK" with TS_0 = $MODEALT_0$ and TS_1 = $MODEALT_1$

If TSchk = 0: (not within limits of either mode)

Perform "RMODINV"

RADLIMCK = 1

RADCADR = +0

Proceed to "DESRETRN"

Perform "RMODINV"

Switch RADMODES bit 14 (REMODFLG) to 1 (request remode)

RADLIMCK = 2

Proceed to "STARTDES"

RRDESNB DESRET = return address

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

DESCOUNT = K:MAXTRYS

Inhibit Interrupts

Perform "RRLIMNB" with TS_0 = $TANG_0$ and TS_1 = $TANG_1$

If TSchk = 0: (not within limits of present mode)

Perform "RMODINV"

Perform "RRLIMNB" with TS_0 = $TANG_0$ and TS_1 = $TANG_1$

If TSchk = 0: (not within limits in either mode)

Perform "RMODINV"

Perform "ALARM" with TS = 00502_8

Inhibit Interrupts

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0 (disable RRCDU Error Counters)

End job

Perform "RMODINV"

Switch RADMODES bit 14 (REMODFLG) to 1

Release interrupt inhibit

$TANGNB_i$ = $TANG_i$ for i = 0 and 1

Perform "RRNB" (get LOS vector in NB coordinates)

RRTARGET = TS

Switch FLAGWRDO bit 6 (RRNBSW) to 1

Inhibit interrupts

STARTDES Switch RADMODES bit 10 (DESIGFLG) to 1

If RADMODES bit 11 (REPOSMON) = 0: (see "DORREPOS")

Perform "SETRRECR"

Call "BEGDES" in 0.02 second

DESRETRN If RADCADR $\neq$ 0, end job

Release interrupt inhibit

Return via DESRET

RRLIMCHK TSchk = 1 (Entered with angles in two's complement form)

If RADMODES bit 12 (ANTENFLG) = 0: (Mode I)

If $|TS_0| >$ K:0.30555, TSchk = 0

If $|TS_1 +$ K:5.5DEGS$| >$ K:0.35833, TSchk = 0

Return

MODE2CHK If $|TS_1 +$ K:82DEGS$| >$ K:0.31667, TSchk = 0

If $|TS_0| \leq$ K:0.69444, TSchk = 0

Return

RRANGLES TSta = arcsin($-TSlos_y$)

TStb = $\frac{1}{2}$ - arcsin($-TSlos_y$)

$TSlos_y$ = 0

TS = unitTSlos

If $|TSlos| < 2^{-20}$: (unit vector poorly defined)
(trunnion angle nearly 90 degrees)

Proceed to "LUNDESCH"

SINTH = TS_x

COSTH = TS_z

Perform "ARCTRIG" (get THETA from SINTH and COSTH)

If RADMODES bit 12 (ANTENFLG) = 0: (Mode I)

$MODEALT_0$ = TStb converted to two's complement form

$MODEALT_1$ = $\frac{1}{2}$ + THETA similarly converted

$MODEPRES_0$ = TSta converted to two's complement form

$MODEPRES_1$ = THETA similarly converted

If RADMODES bit 12 (ANTENFLG) = 1:

$MODEALT_0$ = TSta converted to two's complement form

$MODEALT_1$ = THETA similarly converted

$MODEPRES_0$ = TStb converted to two's complement form

$MODEPRES_1$ = $\frac{1}{2}$ + THETA similarly converted

Return

RRNB TStr = $TANGNB_0$ converted to one's complement form

TSsh = $TANGNB_1$ converted to one's complement form

Skip **next two** lines

RRNBMPAC TStr = TS_1 converted to one's complement form

TSsh = TS_2 converted to one's complement form

TS_y = - sinTStr

TS_z = cosTSsh cosTStr

TS_x = sinTSsh cosTStr

Return

RMODINV If RADMODES bit 12 (ANTENFLG) = 1:

Switch RADMODES bit 12 (ANTENFLG) to 0 (change to Mode I)

Return

Switch RADMODES bit 12 (ANTENFLG) to 1 (change to Mode II)

Return

BEGDES If RADMODES bit 14 (REMODFLG) = 0, proceed to "STDESIG"

REMODE RDES = 0

If RADMODES bit 12 (ANTENFLG) = 1, RDES = $-\frac{1}{2}$

Perform "RRTONLY"

RDES = K:m45DEGSR

Perform "RRSONLY"

RDES = K:m50DEGSR (Mode II)

If RADMODES bit 12 (ANTENFLG) = 0, RDES = RDES + K:m80DEGSR

Perform "RRTONLY"

RDES = 0

If RADMODES bit 12 (ANTENFLG) = 0, RDES = $-\frac{1}{2}$

Perform "RRTONLY"

RDES = 0

If RADMODES bit 12 (ANTENFLG) = 0, RDES = -K:1s$\frac{1}{4}$ (-90°)

Perform "RRSONLY"

Perform "RMODINV"

Switch RADMODES bit 14 (REMODFLG) to 0

If RADMODES bit 10 (DESIGFLG) = 0, proceed to "RGOODEND"

STDESIG If RADMODES bit 11 (REPOSMON) = 1:

Switch RADMODES bit 10 (DESIGFLG) to 0

Proceed to "RDBADEND"

If RADMODES bit 15 (CDESFLAG) = 1, proceed to "MOREDES"

If RADMODES bit 10 (DESIGFLG) = 0, proceed to "ENDRADAR"

If DESCOUNT > 0:

DESCOUNT = DESCOUNT -1

Proceed to "MOREDES"

Switch bits 14 and 2 of channel 12 to 0
(disable RR tracker and RR CDU Error Counters)

Switch RADMODES bit 10 (DESIGFLG) to 0

Proceed to "RDBADEND"

MOREDES Establish "DODES" (pr26)

Call "STDESIG" in 0.5 second

End task

DODES $TANG_0 = CDU_t$

$TANG_1 = CDU_s$

$\underline{T}Slos = \underline{R}RTARGET$

If FLAGWRD0 bit 6 (RRNBSW) = 0:

Perform "CDUTRIG"

Perform "SMTONB"

$\underline{T}Slos = [SMNBMAT]\ unit(MLOSV\ \underline{R}RTARGET + K{:}MCTOMS\ \underline{L}OSVEL)$

$TStr = TANG_0$ converted to one's complement form

$TSsh = TANG_1$ converted to one's complement form

$TANG_1 = \cos TSsh\ TSlos_x - \sin TSsh\ TSlos_z$

$$TRUNNCMD_0 = -\ K{:}RDESGAIN \begin{pmatrix} \sin TStr\ \sin TSsh \\ \cos TStr \\ \sin TStr\ \cos TSsh \end{pmatrix} \cdot \underline{TS}los$$

If RADMODES bit 12 (ANTENFLG) = 1, $TANG_1 = -\ TANG_1$
(A relay in the RR reverses polarity of the shaft commands in mode II; compensate by changing polarity of command.)

$TRUNNCMD_1$ = K:RDESGAIN $TANG_1$

$$\underline{TS} = \begin{pmatrix} \sin TSsh\ \cos TStr \\ -\sin TStr \\ \cos TSsh\ \cos TStr \end{pmatrix}$$

If FLAGWRDO bit 6 (RRNBSW) = 0:

Perform "NBTOSM"

$\underline{TS} = [NBSMMAT]\ \underline{TS}$

TSchk = 0

$TS_{dp} = \underline{TS} \cdot \underline{RR}TARGET$ - K:COS1d2DG

$TS_{sp} < 0$, TSchk = 1 (single precision check of double precision quantity)

If RADMODES bit 15 (CDESFLAG) = 1, proceed to "DORROUT"

If TSchk = 0 and FLAGWRDO bit 5 (LOKONSW) = 0, proceed to "RRDESDUN"

If TSchk = 0 and FLAGWRDO bit 5 (LOKONSW) = 1:

Switch bit 14 of channel 12 to 1 (enable RR tracker)

If bit 4 of channel 33 = 1, proceed to "DORROUT"
(designate until data good discrete is present)

RRDESDUN TS = RADMODES (with bit 10 forced to 0)

Inhibit interrupts

RADMODES = TS

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

Switch bit 2 of channel 12 to 0 (Disable RR Error Counter)

Release interrupt inhibit

End job

DORROUT If FLAGWRD2 bits 12 (LOSCMFLG) or 14 (SRCHOPTN) = 1:

  TS = MLOSV RRTARGET + $\frac{1}{2}$ K:MCTOMS LOSVEL

  RRTARGET = unitTS

  MLOSV = |TS|

Inhibit Interrupts

If RADMODES bit 11 (REPOSMON) = 0, perform "RROUT"

If FLAGWRD2 bit 12 (LOSCMFLG) = 0, proceed to "ENDOFJOB"

If LOSCOUNT = 0:

  Remove "STDESIG" from waitlist if it is there

  Release interrupt inhibit

  If NEWJOB > 0, perform "CHANG1"

  If NEWJOB = +0, proceed to "R21LEM2"

LOSCOUNT = LOSCOUNT - 1

Proceed to "ENDOFJOB"

RRZERO If RADMODES bit 1 (TURNONFL) or bit 11 (REPOSMON) = 1:

  Call "RGOODEND" in 0.01 second

  Return

Inhibit interrupts

Switch RADMODES bit 13 (RCDUOFLG) to 1

Call "RRZ2" in 0.01 second

If RADMODES bit 2 (AUTOMODE) = 1, perform "ALARM" with TS = 00510_8

Release interrupt inhibit

Return

RRZ2 Perform "RRZEROSB"

Proceed to "ENDRADAR"

R04Z RSAMPDT = K:1SECp1

RTSTLOC = 0

RFAILCNT = 0

Inhibit interrupts

Set RADMODES bits 9 (ALTSCALE), 6 (LRPOSFLG), and 3 (RRRSFLAG) = bits 9, 6, and 3 of channel 33

Release interrupt inhibit

If FLAGWRD3 bit 9 (R04FLAG) = 0: (R77)

RSAMPDT = K: 250MSp1

RTSTBASE = 2

RTSTMAX = 6

Call "RADSAMP" in 0.06 second

Proceed to "PINBRNCH"

$OPTIONX_1$ = 1

$OPTIONX_0$ = 4

Perform "GOXDSPFR" with TS = K:V04N12 ($OPTIONX_0$, $OPTIONX_1$) (If terminate, proceed to "R04END"; if proceed, skip next two steps; other response, repeat this step)

Perform "BLANKET" with TS = 00004_8 (blank R3)

End job

RTSTDEX = $OPTIONX_1$

R04X If RTSTDEX ≤ 1: (Bits 3 and 2 are both 0; RR)

RTSTBASE = 0

If bit 2 of channel 33 = 1: (RR not in AUTO mode)

DSPTEM1 = 00201_8

(If bit 2 of channel 33 = 1:)

Proceed to "GOMARK4" with TS = K:V50N25 (DSPTEM1)
(If terminate, proceed to "R04END"; if proceed, repeat at second previous step (to assure that RR has been switched to AUTO); other response, repeat at previous step.)

Switch bit 14 of channel 12 to 1 (enable RR tracker)

RTSTMAX = 2

If RTSTDEX > 1: (Bit 3 or 2 is 1; LR)

RTSTBASE = 2

RTSTMAX = 6

Call "RADSAMP" in RTSTMAX centiseconds

Release interrupt inhibit

If RTSTDEX ≤ 1: (RR)

Proceed to "GOXDSPF" with TS = K:V16N72 (CDU_t and CDU_s)
(If terminate, proceed to "R04END"; if proceed, continue at next step; other response, repeat this step.)

Proceed to "GOXDSPF" with TS = K:V16N78 (**DNRRANGE**, DNRRDOT, TTOTIG)
(If terminate, proceed to "R04END";if proceed, continue at next step; other response, repeat at previous step.)

If RTSTDEX > 1: (LR)

Proceed to "GOXDSPF" with TS = K:V16N66 ($RSTACK_6$,channel 33)
(If terminate, proceed to "R04END"; if proceed, continue at next step; other response, repeat this step.)

Proceed to "GOXDSPF" with TS = K:V16N67 ($RSTACK_{0,2,4}$)
(If terminate, proceed to "R04END"; if proceed; continue at next step; other response, repeat at previous step.)

RSAMPDT = +0 (to stop "RADSAMP")

Delay 2 seconds

RSAMPDT = K:1SECp1

RTSTLOC = 0

If RTSTBASE = +0, TS = 2

If RTSTBASE > 0, TS = 1

RTSTDEX = TS

Proceed to "R04X"

R04END RSAMPDT = +0

Delay 1.28 seconds

Inhibit interrupts

Switch bit 14 of channel 12 to 0 (disable RR tracker)

Switch FLAGWRD3 bit 9 (R04FLAG) to 0

Proceed to "ENDEXT"

RADSAMP If RSAMPDT = +0, end task

Call "RADSAMP" in (|RSAMPDT| - 1) centiseconds

Establish "DORSAMP" (pr25)

RTSTDEX = RTSTBASE + RTSTLOC / 2

End task

DORSAMP If RTSTDEX = 0, perform "RRRANGE" with TSn = 1

If RTSTDEX = 1, perform "RRRDOT" with TSn = 1

If RTSTDEX = 2, perform "LRVELX" with TSn = 1

If RTSTDEX = 3, perform "LRVELY" with TSn = 1

If RTSTDEX = 4, perform "LRVELZ" with TSn = 1

If RTSTDEX = 5, perform "LRALT" with TSn = 1

Perform "RADSTALL"

If RADGOOD = 0, RFAILCNT = RFAILCNT + 1

Inhibit interrupts

If FLAGWRD5 bit 11 (R77FLAG) = 0:

  $RSTACK_{RTSTLOC}$ = $SAMPLSUM_{dp}$

  If RADMODES bit 6 (LRPOSFLG) ≠ bit 6 of channel 33:

    Perform "ALARM" with TS = 00522_8

    RFAILCNT = RFAILCNT + 1

If RTSTLOC ≠ RTSTMAX:

  RTSTLOC = RTSTLOC + 2

  End job

RTSTLOC = 0

End job

RRRANGE TSset = 00011_8 (bits 1 and 4 = 1)

Perform "INITREAD" with TSn = 1

Return

RRRDOT TSset = 00012_8 (bits 2 and 4 = 1)

Perform "INITREAD" with TSn = 1

Return

LRVELX TSset = 00014_8 (bits 3 and 4 = 1)

Perform "INITREAD"

Return

LRVELY TSset = 00015_8 (bits 1, 3 and 4 = 1)

Perform "INITREAD"

Return

LRVELZ TSset = 00016_8 (bits 2, 3 and 4 = 1)

Perform "INITREAD"

Return

LRALT TSset = 00017_8 (bits 1, 2, 3 and 4 = 1)

Perform "INITREAD" with TSn = 1

Return

INITREAD Inhibit interrupts

TIMEHOLD = TSn K:40ms

NSAMP = TSn - 1

SAMPLIM = TSn

OLDATAGD = bits 4, 5 and 8 of channel 33
(RR data good, LR pos data good, LR vel data good discretes)
Switch bits 1 - 4 of channel 13 to 0

TS = TSset

Perform "RADSTART"

TIMEHOLD = TIMEHOLD + TIMENOW

SAMPLSUM = 0

Release interrupt inhibit

Return

<u>RADSTALL</u> Inhibit interrupts

If RADCADR > 0 or if RADCADR < -1:

  $TS1_{dp}$ = return address of routine calling "RADSTALL"

  Proceed to "BAILOUT1" with TS = 31210_8

If RADCADR = -1: (operation already complete and good)

  RADCADR = +0

  Release interrupt inhibit

  RADGOOD = 1

  Return

If RADCADR = -0: (operation already complete and bad)

  RADCADR = +0

  Release interrupt inhibit

  RADGOOD = 0

  Return

RADCADR = return address (to caller of "RADSTALL")

Put present job to sleep

When awakened, return via LOC

<u>ENDRADAR</u> If RADMODES bit 7 (RCDUFAIL) = 0, proceed to "RDBADEND"

<u>RGOODEND</u> If RADCADR = +0:

  RADCADR = -1

  End task

```
          If RADCADR = -0 or RADCADR = -1: (should never be true)
               RADCADR = +0
               End task
          LOC = RADCADR
          RADGOOD = 1
          Wake job put to sleep in "RADSTALL"
          RADCADR = +0
          End task
RDBADEND  If RADCADR = +0:
               RADCADR = -0
               End task
          If RADCADR = -0 or RADCADR = -1:  (should never be true)
               RADCADR = +0
               End task
          LOC = RADCADR
          RADGOOD = 0
          Wake job put to sleep in "RADSTALL"
          RADCADR = +0
          End task
RADAREAD  (entered on interrupt 85-95 ms after bit 4 of channel 13 is set)
          TTOTIG = TTOGO
          DNINDEX = bits 3-1 of channel 13  (radar selection bits)
          If DNINDEX ≠ 0:  (If radar select bits zero, do not store data
                            for downlist (erasable problems))
               DNRADATA_DNINDEX = RNRAD  (radar data)
```

If SAMPLIM < 0:

Perform "ALARM" with TS = 00520_8

Resume

If SAMPLIM = 0:

If FLGWRD11 bit 15 (LRBYPASS) = 0, proceed to "BADRAD"

If FLAGWRD3 bit 9 (R04FLAG) = 0, perform "ALARM" with TS = 00521_8

Proceed to "BADRAD"

SAMPLIM = SAMPLIM - 1

If bit 3 of channel 13 = 0, proceed to "RENDRAD"

Perform "R77CHECK"

If bits 1 and 2 channel 13 are both 1, proceed to "LRHEIGHT" (LR range/altitude measurement)

TS_{dp} = RNRAD +K:LVELBIAS

i = 8

Perform "DGCHECK" (returns only if data is good)

If NSAMP > 0:

NSAMP = NSAMP - 1

Proceed to "RESAMPLE"

Proceed to "GOODRAD"

<u>LRHEIGHT</u> i = 5

If bit 9 of RADMODES (ALTSCALE) ≠ bit 9 of channel 33:

Proceed to "SCALCHNG" with j = 9

TS_{dp} = RNRAD

If bit 9 of channel 33 = 1: (LR range high scale)

$TS_{dp} = 5\ TS_{dp}$

Perform "DGCHECK"

Proceed to "GOODRAD"

RENDRAD If RADMODES bit 7 (RCDUFAIL) = 0 or if RADMODES bit 11 (REPOSMON) = 1, proceed to "BADRAD"

i = 4

If bit 1 of channel 13 = 0: (RR range rate measurement)

TS_{dp} = RNRAD - K:RDOTBIAS

Perform "DGCHECK"

Proceed to "GOODRAD"

If bit 3 of RADMODES (RRRSFLAG) ≠ bit 3 of channel 33:

Proceed to "SCALCHNG" with j = 3

TS_{dp} = RNRAD

If bit 3 of channel 33 = 1: (RR range high scale)

$TS_{dp} = 8\ TS_{dp}$

Perform "DGCHECK"

Proceed to "GOODRAD"

DGCHECK If bit i of channel 33 = 0 and bit i of OLDATAGD = 0: (bit 8 is LR velocity data good, bit 5 is LR altitude data good, bit 4 is RR data good, if zero)

SAMPLSUM = SAMPLSUM + TS_{dp}

Return (with good sample)

Set bit i of OLDATAGD = bit i of channel 33

RESAMPLE If SAMPLIM > 0:

TS = 00010_8

Perform "RADSTART"

Resume

Switch bit i of RADMODES to 1
(bit 8 is LRVELFLG, bit 5 LRALTFLG, bit 4 RRDATAFL)

SAMPLSUM = TS_{dp} (return with bad sample rather than none because no more tries are available)

Perform "RADLITES"

If FLGWRD11 bit 15 (LRBYPASS) = 0, proceed to "BADRAD"

If FLAGWRD3 bit 9 (RO4FLAG) = 0, perform "ALARM" with TS = 00521_8

Proceed to "BADRAD"

SCALCHNG Invert RADMODES bit j
(bit 9 is ALTSCALE, bit 3 is RRRSFLAG)

OLDATAGD = bits 4, 5 and 8 of channel 33 (data good bits)

Switch FLAGWRD5 bit 10 (RNGSCFLG) to 1

BADRAD SAMPLIM = -1

Cause "End task" instruction in "RDBADEND" to perform the same function as a "Resume" instruction

Proceed to "RDBADEND"

GOODRAD SAMPLIM = -1

Switch bit i of RADMODES to 0
(bit 8 is LRVELFLG, bit 5 is LRALTFLG, bit 4 is RRDATAFL)

Perform "RADLITES"

Cause "End task" instruction in"RGOODEND" to perform the same function as a "Resume" instruction

Proceed to "RGOODEND"

LRPOS2 Inhibit Interrupts

Switch RADMODES bit 6 (LRPOSFLG) to 1

If bit 7 of channel 33 = 0: (already in position two)

    Call "RGOODEND" in 0.01 second

    Release interrupt inhibit

    Return

Switch bit 13 of channel 12 to 1 (command LR to position 2)

Call "LRPOSCAN" in 6 seconds

Release interrupt inhibit

Return

LRPOSCAN SAMPLIM = 16_8

Delay 1 second

If bit 7 of channel 33 = 0: (position 2 achieved)

    Delay 2 seconds

    Switch bit 13 of channel 12 to 0

    Proceed to "RGOODEND"

If SAMPLIM > 0:

    SAMPLIM = SAMPLIM - 1

    Proceed to second step of "LRPOSCAN"

Switch bit 13 of channel 12 to 0 (terminate positioning commands

Proceed to "RDBADEND"

RRLIMNB TSchk = 1

If RADMODES bit 12 (ANTENFLG) = 1, proceed to "MODE2CHK"

If $|TS_0| >$ K:0.30555, TSchk = 0

If $TS_1 > 0$ and $|TS_1 +$ K:5.5DEGS$| >$ K:0.35833, TSchk = 0

If $TS_1 \leq 0$ and $|TS_1 +$ K:20.5DEGS$| >$ K:0.35833, TSchk = 0

Return

<u>LUNDESCH</u> If FLAGWRD8 bit 8 (SURFFLAG) = 1:

RADLIMCK = 0

RADCADR = + 0

Proceed to "DESRETRN"

If FLAGWRD0 bit 7 (RNDVZFLG) = 0, end job

Perform "RMODINV"

RADLIMCK = 1

RADCADR = + 0

Proceed to "DESRETRN"

<u>R77CHECK</u> If FLAGWRD5 bit 11 (R77FLAG) = 0, return

Set bits 5 (LRALTFLG) and 8 (LRVELFLG) of RADMODES = bits 5 and 8 of channel 33

Cause "End task" instruction in "RGOODEND" to perform the same function as a "Resume" instruction

Proceed to "RGOODEND"

<u>RADSTART</u> TS_3 = LOSCALAR

TS_1 = low 5 bits of TS_3

RADDEL = low 5 bits of ($00040_8 - TS_1$)

If RADDEL $\leq 00002_8$, proceed to "RADSTART" (If a T5 interrupt is scheduled to occur in 1/16 th of a centisecond, then these operations repeat until the interrupt has occurred)

Set bits TS of channel 13 to 1

RADTIME = $- TS_3$

Return

C13STALL If bit 4 of channel 13 = 0, return

TS_4 = LOSCALAR + RADTIME (LOSCALAR may have overflowed once since RADTIME was last loaded, but this occurrence has been compensated for in the coding)

If $TS_4 \geq$ K:90MSCALR + RADDEL, return

If $TS_4 <$ (K:mDTSCALR + K:90MSCALR + RADDEL), return

Proceed to second step of "C13STALL"

RADLITES If i < 5, proceed to second step of "SETTRKF"

If i = 8: (LR velocity data good)

k = +2

TS = 00004_8

j = 3

If i = 5: (LR altitude data good)

k = +1

TS = 00020_8

j = 5

If bit i of RADMODES = 1, proceed to "ONLITES"

If bit k of FLGWRD11 = 1, proceed to second step of "SETTRKF"

TS = 00000_8

LITIT If bit j of $DSPTAB_{11} \neq$ bit j of TS: (bit 5 is LR altitude fail lamp) (bit 3 is LR velocity fail lamp)

Set bit j of $DSPTAB_{11}$ = bit j of TS

Switch bit 15 of $DSPTAB_{11}$ to 1

Return

ONLITES Switch bit k of FLGWRD11 to 0

Proceed to "LITIT"

CDU_t, CDU_s: LGC input counters incremented directly from the CDU to maintain LGC knowledge of the RR trunnion and shaft angles, respectively. Single precision angles stored in two's complement form and scaled B-1 in units of revolutions.

CDUSCMD, CDUTCMD: LGC output counters connected to the RR shaft and trunnion channels of the CDU. The contents of each of these counters is a rate command scaled B14 in units of RR pulses (one's complement form) and is sent to its respective CDU Error Counter by setting the appropriate enabling discrete in channel 14. See definition of K:RRSPGAIN.

COSTH: See COOR section.

DESCOUNT: Single precision counter defining the maximum amount of time allowed for an attempt to designate, scaled B14 and unitless.

DESRET: A single precision octal return address storage cell.

DNINDEX: A single precision index for selection of appropriate downlink buffer cell for radar data. DNINDEX = 1, 2, 4, 5, 6 or 7 for RR range, RR range-rate, LR X-velocity, LR Y-velocity, LR Z-velocity and LR altitude data, respectively; scaled B14 and unitless.

$DNRADATA_i$: Special storage for downlink of radar data. i = 1, 2, 4, 5, 6, and 7 to index six single precision cells (consecutive except between i = 2 and 4) alternately labelled DNRRANGE, DNRRDOT, DNLRVELX, DNLRVELY, DNLRVELZ and DNLRALT, respectively.

$DSPTAB_{11}$: See INTR section.

DSPTEM1: Temporary storage cell used mainly for display interface purposes.

IMODES33: See INTR section.

K:0.00555: Single precision constant stored as -.00555, scaled B-1 in units of revolutions. Equation value: +0.002775. (Equivalent to +1.0 degree.)
Note: Because a constant is stored in one's complement form, its equation value changes if it is compared with the absolute value of a negative two's complement number. In such a comparison, a single precision constant assumes an equation value of "$A + 2^{b-14}$" where b is the scale factor of the constant and A is the stated equation value of the constant. In the case of the constant "K:p25" for instance, A is 0.25 and b is -1.

K:0.30555. K:5.5DEGS, K:0.35833: Three single precision constants for checking whether the radar position angles are within the bounds of RR Mode I. Stored as -0.30555, 0.03056, and -0.35833, scaled B-1 in units of revolutions. Equation values: +0.152775, 0.01528 and +0.179165, respectively. See note above. (Equivalent to +55 degrees, 5.5 degrees, and +64.5 degrees.)

K:0.31667, K:82DEGS, K:0.69444: Three single precision constants for checking whether the radar position angles are within the bounds of RR Mode II. Stored as -0.31667, 0.45556 and -.69444, scaled B-1 in units of revolutions. Equation values: +0.158335, 0.22778, and +0.34722. See note above. (Equivalent to +57 degrees, 82 degrees and +125 degrees.)
The limits of Mode I are: $-55^\circ \leq$ trunnion $\leq 55^\circ$ and $-70^\circ \leq$ shaft $\leq 59^\circ$. The limits of Mode II are: $125^\circ <$ trunnion $< -125^\circ$ and $-139^\circ <$ shaft $< -25^\circ$. The latter corresponds to limits on the LOS angle of 41° and 155°.

K:1s¼: Single precision constant, program notation "HALF" stored as 0.5, scaled B-1 in units of revolutions. Equation value when used as a two's complement, negative number: $0.25 - 2^{-15}$. (Equivalent to about 90 degrees.)

K:1SECp1: Single precision constant stored as 101×2^{-14}, scaled B14 in units of centiseconds. Equation value: 101.

K:20.5DEGS: Single precision constant stored as 0.11389, scaled B-1 in units of revolutions. Equation value: 0.05695. (Equivalent to 20.5 degrees.)

K:250MSp1: Single precision constant stored as 26×2^{-14}, scaled B14 in units of centiseconds. Equation value: 26.

K:40ms: Single precision constant, program notation "BIT3", stored as 00004_8, scaled B14 in units of centiseconds. Equation value: 4. (Equivalent to half the nominal value of one radar sampling cycle.)

K:90MSCALR: Single precision constant stored as 9×2^{-9}, scaled B9 in units of centiseconds. Equation value 9. (Equivalent to 90 milliseconds.)

K:COS1d2DG: Double precision constant stored as $0.999961923 \times 2^{-2}$, scaled B2 and unitless. Equation value: 0.99996192.

K:LVELBIAS: Single precision constant stored as -12288×2^{-14}, scaled B14 in units like those of RNRAD. Equation value: -12288. (Program comment states, "Landing radar bias for 153.6 kc.")

K:m45DEGSR: Single precision constant stored as 70000_8, scaled B-1 in units of revolutions. Equation value when used as a two's complement number: -0.125. (Equivalent to - 45 degrees.)

K:m80DEGSR: Single precision constant stored as -0.44444, scaled B-1 in units of revolutions. Equation value when used as a two's complement number: -0.22225. (Equivalent to -80.01 degrees.)

K:mDTSCALR: Single precision constant stored as $-.59375 \times 2^{-9}$, scaled B9 in units of centiseconds. Equation value -.59375. (Equivalent to -5.9375 milliseconds)

K:MAXTRYS: Single precision constant stored as 60×2^{-14}, scaled B14 and unitless. Equation value: 60. (Equivalent to 30 seconds of time for RR designate.)

K:MCTOMS: Double precision constant stored as 100×2^{-13}, scaled B13 in units of centiseconds. Equation value: 100.

K:p25: Single precision constant stored as -0.5, scaled B-1 in units of revolutions. Equation value: +0.25. (Equivalent to +90 degrees)

K:RDESGAIN: Single precision constant stored as 0.53624, scaled B12 in units of RR drive pulses per radian of error. Equation value 2196.5. (Equivalent to $0.5 \times 2\ \text{sec}^{-1} \times 360\ \text{deg/rev} \times (10/384)^{-1}$ pulses per degree per second. The first two terms null 0.5 of the error in $\frac{1}{2}$ second, and the fourth is derived from the fact that a saturated error counter causes a drive rate of 10 degrees per second.)

K:RDOTBIAS: Double precision constant stored as 17000×2^{-28}, scaled B28 in units of radar counts (same as RNRAD). Equation value: 17000.

K:rrlimit: Single precision constant, program notation "-RRLIMIT", stored as -384×2^{-14}, scaled B14 in units of RR drive pulses. Equation value: 384. (See K:RRSPGAIN for explanation.)

K:RRSPGAIN: Single precision constant stored as 0.59062, scaled B15 in units of RR drive pulses per revolution of error. Equation value: 19353. (Equivalent to 0.7 x $2 sec^{-1}$ x 360 deg/rev x $(10/384)^{-1}$ pulses per degree per second. The first two terms "null 0.7 of the error in $\frac{1}{2}$ second," and the fourth is derived from the fact that a saturated error counter causes a drive rate of 10 degrees per second.)

LASTSCMD, LASTTCMD: Storage for the previous value of total RR shaft and trunnion CDU error counters; scaled B14 in units of RR drive pulses. Used to convert present position deviation into a desired rate command to be inserted into CDUSCMD or CDUTCMD.

LOC: See MATX section.

LOSCALAR: A 14 bit register corresponding to channel 4 of the computer. It is incremented 1 bit every 1/3200 of a second and is driven by a 102.4 kc signal from the computer oscillator. It overflows (and is propagated to channel 3) every 5.12 seconds. This register is 0.005 second out of phase with the TIMENOW registers. It is equivalent to a single precision time cell scaled B9 in units of centiseconds.

LOSCOUNT: Single precision counter defining the interval between computation of a new line-of-sight vector, scaled B14 and unitless.

LOSVEL: See RNAV section.

MLOSV: See RNAV section.

$MODEALT_0$,$MODEALT_1$: Value of the trunnion and shaft angles, respectively, which are necessary to point the RR along the desired line-of-sight in the alternate mode of the RR. Single precision angles in two's complement form, scaled B-1 in units of revolutions. Program notation "MODEB".

$MODEPRES_0$, $MODEPRES_1$: Value of the trunnion and shaft angles, respectively, which are necessary to point the RR along the desired line-of-sight in the present mode of the RR. Single precision angles in two's complement form, scaled B-1 in units of revolutions. Program notation "MODEA".

[NBSMMAT]: See COOR section.

NEWJOB: See MATX section.

NSAMP: A single precision counter, scaled B14. Controls the number of sampling cycles in the total sampling interval for LR velocity measurements.

OLDATAGD: A single precision cell containing the status of radar "data good" discretes at the beginning of a sample. (See RADMODES)

$OPTIONX_0$, $OPTIONX_1$: See EXVB section.

RADCADR: Single precision octal storage for address to return to program that is sampling the radar and waiting for sample to be completed.

RADDEL: Computed in "RADSTART" it is a single precision time until the next T5 interrupt, scaled B9 in units of centiseconds. Derived from the low five bits of LOSCALAR.

RADGOOD: Temporary variable introduced as a substitute for variable return address; set to 1 or 0 to indicate whether a radar operation was successfully completed or not.

RADLIMCK: Temporary variable indication which of three return options from "RRDESSM" is taken.

RADMODES: A flagword whose bits have the following significance when set (1).

Bit	Mnemonic	Meaning
15	CDESFLAG	Continuous designate; used in conjunction with bit 10
14	REMODFLG	RR remode required
13	RCDUOFLG	RR zeroing in progress
12	ANTENFLG	RR in Mode II (in Mode I if zero)
11	REPOSMAN	RR repositioning in progress; RR was outside of prescribed limits.
10	DESIGFLG	RR designation in progress
9	ALTSCALE	LR range high scale (low scale if bit is zero)
8	LRVELFLG	LR velocity data bad (LR vel data good if bit is zero)

RADMODES: (continued.)

Bit	Mnemonic	Meaning
7	RCDUFAIL	RR CDU operative (RR CDU failed if bit is zero)
6	LRPOSFLG	LR commanded to and presumed to be in position #2
5	LRALTFLG	LR position data bad (LR pos data good if bit is zero)
4	RRDATAFL	RR data bad (if zero, the RR data is "good"; the RR tracker has acquired a target, hopefully the CSM and hopefully not a side lobe).
3	RRRSFLAG	RR range high scale (low scale if bit is zero)
2	AUTOMODE	RR not turned on or not in automatic mode of operation (if zero, the RR is on and it is in the automatic mode)
1	TURNONFL	RR turn-on in progress

RADTIME: Single precision storage for the complement of the value of LOSCALAR at the time bit 4 of channel 13 was set, scaled B9 in units of centiseconds.

RDES: Desired RR position angle (shaft or trunnion); a single precision variable in two's complement form, scaled B-1 in units of revolutions.

RFAILCNT: Single precision counter scaled B14 and used to keep track of the number of unsuccessful attempts to read the radar data.

RNRAD: Single precision LGC counter advanced directly by whichever radar circuit is enabled for sampling, scaled B14 in units of counts.

Sample Type	Value of 1 count		
RRRDOT	-0.19135344	meters/second	-0.6278 fps
RRRANGE			
low scale	2.859024	meters	9.38 feet
high scale	22.872192	meters	75.04 feet
LRVELX	-0.1962912	meters/second	-0.6440 fps
LRVELY	0.3694176	meters/second	1.212 fps
LRVELZ	0.2642006	meters/second	0.8668 fps
LRALT			
low scale	0.3288792	meters	1.0790 feet
high scale	1.644296	meters	5.3950 feet

RRINDEX: Single precision index to indicate whether the content of RDES is a desired shaft angle (1) or a desired trunnion angle (0), scaled B14.

RRRET: Single precision octal return address storage.

RRTARGET: Desired line-of-sight vector, a double precision unit vector scaled B1 in stable member or navigation base coordinates (see FLAGWRD0 bit 6).

RSAMPDT: A cell used for storage of the low-speed sampling interval, in centiseconds scaled B14.

$RSTACK_i$: A series of 4 double precision cells loaded with radar sample data for display in nouns 66 and 67. (i = 0, 2, 4, 6)

RTSTBASE: Single precision quantity scaled B14 used to set RTSTDEX for LR or RR sampling in an automatic sampling mode.

RTSTDEX: An option loaded by the astronaut or set by the program to designate the specific radar data to be sampled single precision, scaled B14.

RTSTLOC: A single precision index used to postion sampled data in the downlink communication cells $RSTACK_{RTSTLOC}$.

RTSTMAX: A single precision limit on the number of $RSTACK_i$ cells to be loaded, scaled B14.

SAMPLIM: A limit on the number of sampling cycles that may be executed in a given sampling interval to limit the number of bad samples that may be attempted before the sampling is abandoned, scaled B14.

SAMPLSUM: Double precision total of radar data accumulated in n sampling intervals (n always 1 for RR and LRALT), scaled B28 in units of radar input counts.

Sample Type	Value for 1 sample	
RRRDOT	-0.19135344	meters/second
RRRANGE	2.859024	meters
LRVELX	-0.1962912	meters/second
LRVELY	0.3694176	meters/second
LRVELZ	0.2642006	meters/second
LRALT	0.3288792	meters

SINTH: See COOR section.

[SMNBMAT]: See COOR section.

$TANG_0$, $TANG_1$: Single precision storage for desired values of or desired changes in the RR trunnion and shaft angles, respectively. Scaled B-1 in units of revolutions, two's complement form in the "RRDESNB" and "R21LEM" routines. Scaled B2 in units of radians (one's complement form) in the "DODES" routine.

$TANGNB_0$, $TANGNB_1$: Temporary two's complement storage (astronaut desired or radar marked) radar position angles (trunnion and shaft, respectively), scaled B-1 in units of revolutions.

THETA: See COOR section.

TIMEHOLD: Double precision time at the middle of the total sampling interval (assuming no bad samples), scaled B28 in units of centiseconds.

TIMENOW: See EXVB section.

$TRUNNCMD_{0,1}$: Single precision storage for the desired values of the RR trunnion and shaft angle drive rates, respectively. Scaled B14 in units of RR drive pulses. An alternate mnemonic for $TRUNNCMD_1$ is SHAFTCMD.

TTOGO: See BURN section.

TTOTIG: Double precision time to TIG, loaded in the radar read routine "RADAREAD" at the time of the radar reading, scaled B28 in units of centiseconds.

RNAV

PROG20 (PROG22) If FLAGWRD8 bit 8 (SURFFLAG) = 0:

Switch FLAGWRD1 bit 8 (VEHUPFLG) to 0

Proceed to "PROG20A"

ORBCHGO Switch FLAGWRD1 bit 8 (VEHUPFLG) to 1

OPTION2 = 1 (CSM orbit option)

Proceed to "GOPERF4" with OPTION1 = 00012_8
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

If OPTION2 = 1, proceed to "PROG20A"

Proceed to "GOFLASH" with TS = K:V06N33
(If terminate, proceed to "GOTOPOOH"; if proceed, continue with next step; other response, repeat this step.)

Perform "INTSTALL"

TS_{lnchtm} = TIG

TDEC1 = TIG

Switch FLAGWRD3 bits 4 (INTYPFLG), 3 (VINTFLAG), 2 (D6OR9FLG) and 1 (DIMOFLAG) to 0

Perform "INTEGRV"

$\underline{R}$SUBL = $\underline{R}$ATT

TDEC1 = TAT

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Switch FLAGWRD3 bits 4 (INTYPFLG) and 1 (DIMOFLAG) to 0

If FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 1

Perform "INTEGRV"

$\underline{V}SUBC = \underline{V}ATT$

$\underline{R}SUBC = \underline{R}ATT$

$\underline{U}CSM = unit(\underline{R}SUBL*unit(\underline{R}SUBC*\underline{V}SUBC))$

$CSTH = unit(\underline{R}SUBC)\cdot\underline{U}CSM$

$SNTH = \sqrt{1 - CSTH^2}$

$\underline{R}VEC = \underline{R}SUBC$

$\underline{V}VEC = -\underline{V}SUBC$

Switch FLAGWRD7 bit 9 (RVSW) to 0

Perform "TIMETHET"

$\underline{N}EWVEL = -\underline{T}S_v$

$\underline{N}EWPOS = \underline{T}S_r$

$TS_{transtm} = T$

$\underline{N}CSMVEL = |\underline{N}EWVEL|\ unit(unit(\underline{N}EWPOS*\underline{R}SUBL)*\underline{N}EWPOS)$

Perform "INTSTALL"

$TET = TS_{lnchtm} - TS_{transtm}$

$\underline{R}RECT = \underline{N}EWPOS$

$\underline{R}CV = \underline{N}EWPOS$

$\underline{V}RECT = \underline{N}CSMVEL$

$\underline{V}CV = \underline{N}CSMVEL$

$\underline{T}DELTAV = 0$

TNUV = 0

TC = 0

XPREV = 0

PBODY = 2

Perform "MOVEACSM"

Switch FLAGWRD8 bit 12 (CMOONFLG) to 1

Perform "SVDWN1"

Set FLAGWRD8 bit 12 (CMOONFLG) = FLAGWRD0 bit 12 (MOONFLAG)

Perform "INTWAKE1"

PROG20A Perform "RO2BOTH" (IMU status check routine)

Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 1

Switch FLAGWRD0 bit 7 (RNDVZFLG) to 1

Switch FLAGWRD2 bits 14 (SRCHOPTN) and 13 (ACMODFLG) to 0

Switch FLAGWRD3 bit 9 (R04FLAG) to 0

Switch FLAGWRD5 bit 4 (NORRMON) to 0

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 channel 12 ~~to~~ 0

P20LEM1 TRKMKCNT = 0

TDEC1 = TIMENOW

Perform "LPS20.1" (get RR target vector)

If MLOSV ≥ K:FHNM:

Perform "PRIOLARM" with TS = 00526_8
(If terminate, proceed to "TRMTRACK"; if proceed, repeat this step; other response, proceed to "P20LEM1".)

End job

P20LEMA If FLAGWRD8 bit 8 (SURFFLAG) = 0, perform "R61LEM"

P20LEMB Change priority to 26_8

If FLAGWRD1 bit 5 (TRACKFLG) = 0, proceed to "P20LEMWT"

P20LEMB7 If bit 2 of channel 33 = 0, proceed to "P20LEMB3" (RR auto mode)

If MODREG ≠ 20 and if MODREG ≠ 22:

Perform "PRIOLARM" with TS = 00514_8
(If terminate, proceed to "TRMTRACK"; if proceed, proceed to "P20LEMB"; other response, proceed to "P20LEMB".)

End job

Proceed to "GOPERF1" with TS = 00201_8
(If terminate, proceed to "TRMTRACK"; if proceed, proceed to "P20LEMB"; other response, continue at next step.)

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Switch bit 7 of channel 11 to 1 (operator error lamp)

Proceed to "P20LEMB"

Proceed to "R23LEM" (manual acquisition monitor)

P20LEMB1 Switch FLAGWRD2 bit 13 (ACMODFLG) to 1

Proceed to "P20LEMB"

P20LEMB3 If RADMODES bit 13 (RCDUOFLG) = 1: (RR just turned on and still zeroing)

Delay 2.5 seconds

Proceed to "P20LEMB3"

If FLAGWRD2 bit 14 (SRCHOPTN) or 13 (ACMODFLG) = 1:

Switch FLAGWRD2 bits 14 (SRCHOPTN) and 13 (ACMODFLG) to 0

Proceed to "P20LEMWT"

P20LEMC3 TDEC1 = TIMENOW

Perform "UPPSV"

P20LEMC If FLAGWRD0 bit 7 (RNDVZFLG) = 0, end job

If FLAGWRD1 bit 5 (TRACKFLG) = 1, proceed to "P20LEMF"

Call "P20LEMD1" in 15 seconds

End job

P20LEMD1 If FLAGWRD1 bit 5 (TRACKFLG) = 0: (stall until TRACKFLG is set again, then continue at automatic acquisition)

Call "P20LEMD1" in 15 seconds

End task

Establish "P20LEMC3" (pr26)

End task

P20LEMF Proceed to "R21LEM"

P20LEMWT Call "P20LEMC1" in 2.5 seconds

End job

P20LEMC1 If FLAGWRD0 bit 7 (RNDVZFLG) = 0, end task

If FLAGWRD1 bit 5 (TRACKFLG) = 0:

Call "P20LEMC1" in 15 seconds

End task

Establish "R22LEM42" (pr26)

End task

PROG21 OPTION2 = 1

Proceed to "GOPERF4" with OPTION1 = 2
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

$DSPTEM1_{dp} = 0$

Proceed to "GOFLASH" with TS = K:V06N34 (get time in DSPTEM1)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at the next step; other response, repeat this step.)

$TS = DSPTEM1_{dp}$

If TS = 0, TS = TIMENOW

TDEC1 = TS

Perform "INTSTALL"

If FLAGWRDO bit 11 (P21FLAG) = 1:

Proceed to "P21CONT"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

If |OPTION2| ≥ 2: (CSM option)

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 4 (INTYPFLG) to 0

Perform "INTEGRV"

P21VSAVE P21TIME = TAT

P21BASER = RATT (scaled B29 or B27)

P21BASEV = VATT (scaled B7 or B5)

P21VEL = |VATT| (insured scaling of B7 for N91 display)

P21GAM = arcsin (unit RATT · VATT / P21VEL) (RATT-B29, VATT-B7)

P21ORIG = PBODY (PBODY in index register 2)

If |OPTION2| ≤ 1:

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Skip next line

Switch FLAGWRDO bit 11 (P21FLAG) to 1

Switch FLAGWRD3 bit 12 (LUNAFLAG) to 0

If P21ORIG ≠ 0 :

    Switch FLAGWRD3 bit 12 (LUNAFLAG) to 1

ALPHAV = RATT

TS_t = TAT

Switch FLAGWRD1 bit 13 (ERADFLAG) to 0

Perform "LAT-LONG"

P21ALT = ALT K:K.01   (ALT/100 for N91 display)

Proceed to "GOFLASH" with TS = K:V06N43 (LAT, LONG, ALT)
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to "GOTOPOOH"; other response, continue at next step.)

$DSPTEM1_{dp}$ = P21TIME + K:600SEC

Proceed to the fourth step of "PROG21"

P21CONT   RCV = P21BASER

VCV = P21BASEV

TET = P21TIME

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 0

Switch FLAGWRD0 bit 12 (MOONFLAG) to 0

If P21ORIG ≠ 0:

    Switch FLAGWRD0 bit 12 (MOONFLAG) to 1

Perform "INTEGRVS"

Proceed to "P21VSAVE"

PROG25   Perform "R02BOTH"

Switch FLAGWRD1 bit 5 (TRACKFLG) to 1

Switch FLAGWRD0 bit 9 (P25FLAG) to 1

Switch FLAGWRD0 bit 7 (RNDVZFLG) to 0

P25LEM1 — If FLAGWRD0 bit 9 (P25FLAG) = 0, end job

If FLAGWRD1 bit 5 (TRACKFLG) = 0:

- Delay 60 seconds
- Establish "P25LEM1" (pr14)
- End job

R65CNTR = 7

Perform "R65LEM"

Proceed to "P25LEM1"

LPS20.1 — LS21X = return address

If FLAGWRD2 bit 12 (LOSCMFLG) = 1:

- If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "CSMINT"

Perform "LEMCONIC"

LMPOS = RATT

LMVEL = VATT

TDEC1 = TAT

CSMINT — Perform "CSMCONIC"

TS = [REFSMMAT] (VATT - LMVEL)

Remove "STDESIG" from waitlist if it is there

LOSVEL = TS

TS = RATT - LMPOS

If FLAGWRD0 bit 7 (RNDVZFLG) = 1:

- If OVFIND $\neq$ 0, OVFIND = 0
- TS_1 = unit$\left(TS \times 2^9\right)$

(If FLAGWRDO bit 7 (RNDVZFLG) = 1:)

If OVFIND ≠ 0: (OVFIND reset to 0)

Perform "PRIOLARM" with TS = 00526_8
(If terminate, proceed to "TRMTRACK"; if proceed, repeat this step; other response, proceed to "P20LEM1".)

End job

If FLAGWRDO bit 7 (RNDVZFLG) = 0:

$\underline{TS}$ = unit $\underline{TS}$

If OVFIND ≠ 0: (OVFIND reset to 0)

Perform "PRIOLARM" with TS = 00526_8
(If terminate, proceed to "TRMTRACK"; if proceed, repeat this step; other response, proceed to "P20LEM1".)

End job

$\underline{RR}$TARGET = [REFSMMAT] unit $\underline{TS}$

MLOSV = $|\underline{TS}|$

Switch FLAGWRDO bit 6 (RRNBSW) to 0

Proceed to LS21X

UPPSV LS21X = return address

Perform "INTSTALL"

Perform "SETIFLGS"

If FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 1

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

Switch FLAGWRD3 bit 2 (D6OR9FLG) to 1

If FLAGWRD1 bit 8 (VEHUPFLG) = 0:

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

Perform "INTEGRV"

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

(If FLAGWRD1 bit 8 (VEHUPFLG) = 0:)

TDEC1 = TETLEM

Proceed to "UPPSV4"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Perform "INTEGRV"

Perform "INTSTALL"

TDEC1 = TETCSM

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

UPPSV4 Perform "SETIFLGS"

Perform "INTEGRV"

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

Proceed to "P20LEMC"

$\underline{L}$MPOS = $\underline{R}$CVLEM (rescaled by 2^{-2})

$\underline{L}$MVEL = $\underline{V}$CVLEM (rescaled by 2^{-2})

Return via LS21X

R61LEM GENRET = return address

Switch FLAGWRD1 bit 10 (R61FLAG) to 1

Proceed to "R61C+L02"

R65LEM GENRET = return address

Switch FLAGWRD1 bit 10 (R61FLAG) to 0

R61C+L01 If bit 4 of channel 33 = 1, proceed to "R61C+L02"

Switch FLAGWRD3 bit 9 (R04FLAG) to 1

Perform "RRRDOT"

Perform "RADSTALL"

Perform "RRRANGE"

Perform "RADSTALL"

Switch FLAGWRD3 bit 9 (R04FLAG) to 0

R61C+L02 If FLAGWRD1 bit 5 (TRACKFLG) = 0:

    Proceed to "P20LEMWT"

SCAXIS = K:UNITZ

TDEC1 = TIMENOW + K:3SECONDS

Perform "LPS20.1"   (get RR target vector)

POINTVSM = RRTARGET

Inhibit interrupts

TScdu = CDUD

Release interrupt inhibit

Perform "VECPNT1"

THETAD = TS

If FLAGWRD1 bit 5 (TRACKFLG) = 0:

    Proceed to "P20LEMWT"

If bit 10 of channel 30 or bit 14 of channel 31 = 1:

    Perform "BALLANGS"

    Proceed to "R61C+L06"

TSref = RRTARGET

Perform "CDUTRIG"

Perform "SMTONB"

$\underline{TS}_{ref} = [SMNBMAT] \underline{TS}_{ref}$

If TS_{refz} - K:COS15DEG $<$ 0:

Inhibit interrupts

Perform "ZATTEROR"

Perform "SETMINDB"

Switch FLAGWRD5 bit 6 (3AXISFLG) to 0

Switch FLAGWRD4 bit 12 (PDSPFLAG) to 1

Perform "R60LEM"

Inhibit interrupts

Perform "RESTORDB"

Switch FLAGWRD4 bit 12 (PDSPFLAG) to 0

Proceed to "R61C+L06"

Inhibit interrupts

$\underline{C}DUD = \underline{T}HETAD$

Release interrupt inhibit

R61C+L06 If FLAGWRD1 bit 10 (R61FLAG) = 1 or R65CNTR = 0, return via GENRET

R65CNTR = R65CNTR - 1

Delay 6 seconds

Establish "R61C+L01" (pr 26)

Proceed to "ENDOFJOB"

R21LEM Switch bit 14 channel 12 to 0

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

- $TANG_0 = 0$
- $TANG_1 = 0$

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

- If RADMODES bit 12 (ANTENFLG) = 1:
  - Proceed to "R21LEM10"
- $TANG_0 = -\frac{1}{2}$ (-180°)
- $TANG_1 = -\frac{1}{4}$ (-90°)

Switch FLAGWRD0 bit 5 (LOKONSW) to 0

Perform "RRDESNB"

Perform "RADSTALL"

If RADGOOD = 0:

- Perform "PRIOLARM" with TS = 00503_8
  (If terminate, proceed to "TRMTRACK"; if proceed, proceed to "R24LEM"; other response, proceed to "P20LEMC3".)
- End job

R21LEM10 Switch FLAGWRD2 bit 12 (LOSCMFLG) to 1

DESCOUNT = K:MAXTRIES

R21LEM2 LOSCOUNT = 3

R21LEM1 TDEC1 = TIMENOW + K:HALFSEC

Perform "LPS20.1" (get RR target vector)

Switch FLAGWRD0 bit 5 (LOKONSW) to 1

Switch FLAGWRD5 bit 4 (NORRMON) to 0

Perform "RRDESSM"

If RADLIMCK = 0: (not within mode 2 limits on lunar surface)

Proceed to "R21LEM4"

If RADLIMCK = 1: (not within limits in either mode)

Proceed to "P20LEMA"

(Otherwise, RADLIMCK = 2)

Perform "RADSTALL"

If RADGOOD = 0 : (lock-on not achieved)

Perform "PRIOLARM" with TS = 00503_8
(If terminate, proceed to "TRMTRACK"; if proceed, proceed to "R24LEM"; other response, proceed to "P20LEMC3".)

End job

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

If FLAGWRD8 bit 8 (SURFFLAG) = 1, proceed to "P20LEMWT"

Proceed to "R21DISP"

R21LEM4 REPOSCNT = K:MAXTRIES

Switch FLAGWRD0 bit 10 (FSPASFLG) to 1

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

REPOSTM = TIMENOW + K:TENSEC

TDEC1 = TIMENOW + K:TENSEC

60TIMES Perform "LPS20.1"

Perform "RRDESSM"

If RADLIMCK = 0:

  If REPOSCNT = 0: (looked 600 seconds ahead did not find)

    Perform "PRIOLARM" with TS = 530_8
    (All responses go to "TRMTRACK")

    End job

  REPOSCNT = REPOSCNT - 1

  REPOSTM = REPOSTM+K:TENSEC

  TDEC1 = REPOSTM (old designate time plus 10 seconds)

  Proceed to "60TIMES"

If RADLIMCK = 1:

  End of job

If RADLIMCK = 2:

  Remove "BEGDES" from waitlist

  If FLAGWRDO bit 10 (FSPASFLG) = 0:

    Proceed to "R21LEM8"

  Switch FLAGWRDO bit 10 (FSPASFLG) to 0

  REPOSTM = REPOSTM+K:TENSEC

  TDEC1 = REPOSTM (old designate time plus 10 seconds)

  Proceed to "60TIMES"

R21LEM8 TDEC1 = REPOSTM

Perform "UPPSV"

Switch RADMODES bit 15 (CDESFLAG) to 1

Switch FLAGWRD0 bit 5 (LOKONSW) to 0

Switch FLAGWRD5 bit 4 (NORRMON) to 1

Perform "RRDESNB"

Call "R21LEM9" in (REPOSTM-TIMENOW seconds)

End of job

R21LEM9 Remove "STDESIG" from waitlist

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0

Establish "R21LEM10" (pr26)

End task

R21DISP Perform "GOPERF2R" with TS = K:V06N72 (CDU_t , CDU_s)
(If terminate, proceed to "TRMTRACK"; if proceed, proceed to "P20LEMWT"; other response, repeat this step.)

Perform "BLANKET" with TS = 00100_8

End of job

R22LEM42 If FLAGWRD8 bit 8 (SURFFLAG) = 0:

R65CNTR = 2

Perform "R65LEM"

Proceed to "R22LEM"

Delay 2 seconds

Proceed to "R22LEM"

<u>R22LEM</u> If FLAGWRD0 bit 7 (RNDVZFLG) = 0, end job

If FLAGWRD1 bit 5 (TRACKFLG) = 0, proceed to "R22WAIT"

If bit 14 of channel 12 = 0: (RR tracker disabled)

    Proceed to "P20LEMA"

If bit 2 of channel 33 = 1: (RR AUTO mode switch not set)

    Proceed to second step of "P20LEMB7"

If RADMODES bit 13 (RCDUOFLG) = 1:

    Proceed to "R22LEM42"

Perform "LRS22.1"

If TSerror = 1, proceed to "P20LEMC"

If TSerror = 2: (actual LOS differs from computed LOS)

    Perform "PRIOLARM" with TS = 00525_8
        (If terminate, proceed to "TRMTRACK"; if proceed, skip next step; other response, repeat this step.)

    End job

    Proceed to "PRIODSP" with TS = K:V06N05 (display deviation)
        (If terminate, proceed to "TRMTRACK"; if proceed, continue at next step; other response, proceed to "P20LEMC".)

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

    If FLAGWRD1 bit 5 (TRACKFLG) = 0, proceed to "R22WAIT"

    If $\arccos RRBORSIT_z \geq$ K:30DEG:

        Perform "R61LEM"

        Proceed to "R22WAIT"

If FLAGWRD1 bit 6 (NOUPFLAG) = 1:

    Proceed to "R22LEM42"

If FLAGWRD1 bit 7 (UPDATFLG) = 0:

    Proceed to "R22LEM42"

Perform "LSR22.3"

TRKMKCNT = TRKMKCNT + 1

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

    R65CNTR = 5

    Perform "R65LEM"

    Proceed to "R22LEM"

Delay 2 seconds

Proceed to "R22LEM"

<u>R22LEM96</u> N49FLAG = 0

Establish "N49DSP" (pr27)

If N49FLAG = 0, repeat this step (delay until N49FLAG ≠ 0)

If N49FLAG < 0:

    If N49FLAG = - 2, proceed to "R22LEM"

    Proceed to "ASTOK"

Return via LGRET

<u>R22WAIT</u> Call "P20LEMC1" in 15 seconds

Proceed to second step of "P20LEMWT"

R22RSTRT Perform "RRRDOT" (If a restart occurs while reading radar come here)

Perform "RADSTALL"

If RADGOOD = 0, proceed to "P20LEMC"
(could not read radar)

Proceed to "R22LEM"

N49DSP Proceed to "PRIODSP" with TS= K:V06N49
(If terminate, set N49FLAG to - 2 and end job; if proceed, set N49FLAG = -1 and end job; other response, set N49FLAG to + value and end job.)

R23LEM Switch FLAGWRD5 bit 4 (NORRMON) to 1

Perform "SETMINDB" with interrupts inhibited

Switch bit 14 of channel 12 to 1 (RR track enable)

Proceed to "GOPERF1" with TS = 00205_8 (request manual acquisition)
(If terminate, proceed to "R23LEM2"; if proceed, continue at next step; other response, proceed to "R23LEM3".)

Inhibit interrupts

Perform "RRLIMCHK" with $TS_0 = CDU_t$ and $TS_1 = CDU_s$

If TSchk = 0: (Manual acquisition not within limits)

Perform "PRIOLARM" with TS = 00501_8
(If terminate, proceed to "R23LEM2"; if proceed, repeat this step; other response, proceed to "R23LEM3".)

End job

Perform "RESTORDB"

Release interrupt inhibit

Switch FLAGWRD5 bit 4 (NORRMON) to 0

Proceed to "P20LEMB1"

R23LEM2 Switch FLAGWRD5 bit 4 (NORRMON) to 0

Proceed to "TRMTRACK"

R23LEM3 Perform "R61LEM"

Proceed to third step of "R23LEM"

R24LEM Switch FLAGWRD2 bit 14 (SRCHOPTN) to 1

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

DATAGOOD = 0

OMEGDISP = 0

Perform "PRIODSPR" with TS = K:V16N80 (monitor DATAGOOD, OMEGDISP)
(If terminate, proceed to "TRMTRACK"; if proceed, proceed to "R24END"; other response, proceed to "R24LEM3".)

Proceed to "LRS24.1"

R24END Remove "CALLDGCH" from waitlist (kill it)

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0 (disable RR CDU error counters)

Proceed to "P20LEM1"

R24LEM3 Remove "CALLDGCH" from waitlist (kill it)

Switch RADMODES bits 10 (DESIGFLG) and 15 (CDESFLAG) to 0

Switch bit 2 of channel 12 to 0 (disable RRCDU Error Counters)

Delay 0.5 second

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

Perform "R61LEM"

RADCADR = +0

Proceed to fifth step of "R24LEM"

LRS22.1 Switch FLAGWRD5 bit 10 (RNGSCFLG) to 0

Inhibit interrupts

Set RADMODES bit 3 (RRRSFLAG) = bit 3 of channel 33 (RR range scale)

Release interrupt inhibit

READRDOT Perform "RRRDOT" (read RR range-rate)

Perform "RADSTALL"

If RADGOOD = 0:

  TSerror = 1

  Return

Inhibit interrupts

(Save for downlink storage)

$TS_{5\&6}$ = TIMEHOLD

RDOTMSAV = SAMPLSUM

$TS_3 = CDU_y$

$TS_4 = CDU_z$

$TS_2 = CDU_x$

TS_t = TIMENOW

$TANG_0 = CDU_t$

$TANG_1 = CDU_s$

Release interrupt inhibit

Perform "RRRANGE" (read RR range)

Perform "RADSTALL"

If RADGOOD = 0:

If FLAGWRD5 bit 10 (RNGSCFLG) = 1:

Proceed to "READRDOT"

TSerror = 1

Return

Inhibit interrupts

RANGRDOT = $DNRADATA_{1\&2}$

MKTIME = $TS_{5\&6}$

AIG = TS_3

AMG = TS_4

$TANGNB_0$ = $TANG_0$

$TANGNB_1$ = $TANG_1$

AOG = TS_2

release interrupt inhibit

RDOTM = K:RDOTCONV RDOTMSAV (scaled to (meters/centisecond)/2^7)

RRTRUN = $TANG_0$ converted to one's complement form

RRSHAFT = $TANG_1$ similarly converted

RM = K:RANGCONV SAMPLSUM

Perform "RRNB" (determine actual LOS from radar position angles)

$\underline{R}$RBORSIT = $\underline{T}$S

TDEC1 = TS_t

Perform "LPS20.1" (get estimate of LOS based of present state vector information)

Perform "CD*TR*GS" with ANG = (AOG, AIG, AMG)

Perform "SMTONB"

TS = [SMNBMAT] RRTARGET

DSPTEM1 = arccos (TS · RRBORSIT) (angular error between the two)

If DSPTEM1 ≥ K:THREEDEG:

TSerror = 2

Return

TSerror = 0

Return

LSR22.3 If FLAGWRD8 bit 8 (SURFFLAG) = 1:

Proceed to "LSR22.4"

Switch FLAGWRD5 bit 9 (DMENFLG) to 1

If FLAGWRD1 bit 8 (VEHUPFLG) = 1:

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

Perform "SETIFLGS"

TDEC1 = MKTIME

Perform "INTEGRV" (LM)

Perform "INTSTALL"

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 0

If FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 1

Switch FLAGWRD3 bits 3 (VINTFLAG)and 5 (STATEFLG) to 1

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0

TDEC1 = MKTIME

(If FLAGWRD1 bit 8 (VEHUPFLG) = 1:)

Perform "INTEGRV"

If FLAGWRD5 bit 1 (RENDWFLG) = 0:

Perform "WLINIT"

Proceed to "RANGEBQ"

Perform "INTSTALL"

Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

Perform "SETIFLGS"

TDEC1 = MKTIME

Perform "INTEGRV"

Perform "INTSTALL"

Switch FLAGWRD3 bit 1 (DIMOFLAG) to 0

If FLAGWRD5 bit 1 (RENDWFLG) = 1:

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 2 (D6OR9FLG) to 1

Switch FLAGWRD3 bits 3 (VINTFLAG) and 4 (INTYPFLG) to 0

Switch FLAGWRD3 bit 5 (STATEFLG) to 1

TDEC1 = MKTIME

Perform "INTEGRV"

If FLAGWRD5 bit 1 (RENDWFLG) = 0:

Perform "WLINIT"

Proceed to "RANGEBQ"

LSR22.4 Perform "INTSTALL"

Switch FLAGWRD3 bit 5 (STATEFLG) to 1 (Only two flag settings necessary because of SURFFLAG = 1)

Switch FLAGWRD3 bit 3 (VINTFLAG) to 0

TDEC1 = MKTIME

Perform "INTEGRV"

Switch FLAGWRD5 bit 9 (DMENFLG) to 0

Perform "INTSTALL"

If TRKMKCNT = 0:

    Perform "WLINIT"

    Switch FLAGWRD3 bit 3 (VINTFLAG) to 1

    Perform "SETIFLGS" (Standard flag setting for integration)

    TDEC1 = MKTIME

    Perform "INTEGRV"

    Proceed to "RANGEBQ"

Switch FLAGWRD3 bits 1 (DIMOFLAG) and 3 (VINTFLAG) to 1

Switch FLAGWRD3 bits 2 (D6OR9FLG) and 4 (INTYPFLG) to 0

TDEC1 = MKTIME

Perform "INTEGRV"

Proceed to "RANGEBQ"

RANGEBQ If FLAGWRD8 bit 8 (SURFFLAG) = 0:

R65CNTR = 0

Perform "R65LEM"

WHCHREAD = 1 (RANGE code in N49)

$\underline{TSrlc} = \underline{DELTACSM} + \underline{RCVCSM} - \underline{DELTALEM} - \underline{RCVLEM}$ (scaling controlled by LMOONFLG)

$\underline{ULC} = \text{unit}\underline{TSrlc}$ (quasi-floating point)

$\underline{BVECTOR}_0 = \underline{ULC}$

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, $\underline{BVECTOR}_0 = -\underline{ULC}$

$\underline{BVECTOR}_1 = 0$

$\underline{BVECTOR}_2 = 0$

$DELTAQ = RM - |\underline{TSrlc}|$

$VARIANCE = RANGEVAR\ |\underline{TSrlc}|^2$

If VARIANCE ≤ RVARMIN, VARIANCE = RVARMIN

Perform "LGCUPDTE"

WHCHREAD = 2 (R-RATE code in N49)

$\underline{TSrlc} = \underline{DELTACSM} + \underline{RCVCSM} - \underline{DELTALEM} - \underline{RCVLEM}$

$\underline{ULC} = \text{unit}\underline{TSrlc}$ (quasi-floating point)

$\underline{BVECTOR}_1 = \underline{TSrlc}$

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, $\underline{BVECTOR}_1 = -\underline{TSrlc}$

$\underline{TSvlc} = \underline{NUVCSM} + \underline{VCVCSM} - \underline{NUVLEM} - \underline{VCVLEM}$

$TSrdot = \underline{ULC} \cdot \underline{TSvlc}$

$VARIANCE = RATEVAR\ TSrdot^2$

If VARIANCE < VVARMIN, VARIANCE = VVARMIN

$DELTAQ = |\underline{TSrlc}|\ (RDOTM - TSrdot)$

$\underline{BVECTOR}_0 = (\underline{ULC} * \underline{TSvlc}) * \underline{ULC}$

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, $\underline{B}VECTOR_0$ = - $\underline{B}VECTOR_0$

$\underline{B}VECTOR_2$ = 0

appropriate scaling change made (See definition of $\underline{B}VECTOR_0$)

VARIANCE = VARIANCE $|\underline{T}Srlc|^2$

Perform "LGCUPDTE"

If FLAGWRD8 bit 8 (SURFFLAG) = 1, return (to caller of "LSR22.3")

$\underline{A}NG$ = (AOG, AIG, AMG)

Perform "CD*TR*GS"

Perform "NBTOSM"

$\underline{X}NB_{ref}$ = [REFSMMAT]^T [NBSMMAT] $\underline{K}$:UNITX

$\underline{Y}NB_{ref}$ = [REFSMMAT]^T [NBSMMAT] $\underline{K}$:UNITY

$\underline{Z}NB_{ref}$ = [REFSMMAT]^T [NBSMMAT] $\underline{K}$:UNITZ

$\underline{T}Srlc$ = $\underline{D}ELTACSM$ + $\underline{R}CVCSM$ - $\underline{D}ELTALEM$ - $\underline{R}CVLEM$

$\underline{U}LC$ = unit$\underline{T}Srlc$

SINTHETA = - $\underline{U}LC$ · $\underline{Y}NB_{ref}$

RXZ = $|\underline{T}Srlc|\sqrt{1 - SINTHETA^2}$

WHCHREAD = 3 (shaft code in N49)

SINTH = $\underline{U}LC$ · $\underline{X}NB_{ref}$

COSTH = $\underline{U}LC$ · $\underline{Z}NB_{ref}$

Perform "ARCTAN"

DELTAQ = RXZ (K:2PId8 RRSHAFT - K:2PId8 THETA - $X789_x$)

$\underline{B}VECTOR_0$ = - unit($\underline{U}LC$*$\underline{Y}NB_{ref}$)

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, $\underline{B}VECTOR_0$ = - $\underline{B}VECTOR_0$

$\underline{B}VECTOR_1$ = 0

$\underline{B}VECTOR_2$ = (RXZ , 0 , 0)

$VARIANCE = RXZ^2 (SHAFTVAR + K{:}IMUVAR)$

Perform "LGCUPDTE"

$\underline{T}Srlc = \underline{D}ELTACSM + \underline{R}CVCSM - \underline{D}ELTALEM - \underline{R}CVLEM$

$\underline{U}LC = unit\underline{T}Srlc$

$SINTHETA = - \underline{U}LC \cdot \underline{Y}NB_{ref}$

$RXZ = |\underline{T}Srlc| \sqrt{1 - SINTHETA^2}$

WHCHREAD = 4 (trunnion code in N49)

$\underline{B}VECTOR_0 = - (\underline{U}LC * \underline{Y}NB_{ref}) * \underline{U}LC$

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, $\underline{B}VECTOR_0 = - \underline{B}VECTOR_0$

$\underline{B}VECTOR_1 = 0$

$\underline{B}VECTOR_2 = (0, RXZ, 0)$

$VARIANCE = RXZ^2 (TRUNVAR + K{:}IMUVAR)$

$DELTAQ = RXZ (K{:}2PId8\ RRTRUN - K{:}2PId8\ arcsinSINTHETA - X789_y)$

Perform "LGCUPDTE"

Return (to caller of "LSR22.3")

LRS24.1 NSRCHPNT = 0

Switch bit 14 of channel 12 to 1 (enable RR tracker)

If FLAGWRD2 bit 14 (SRCHOPTN) = 0, end job

Inhibit interrupts

Call "CALLDGCH" in 6 seconds

Release interrupt inhibit

If RADMODES bit 14 (REMODFLG) = 1, end job

TDEC1 = TIMENOW + K:1.5SECS

Perform "LEMCONIC"

$\underline{R}LMSRCH = \underline{R}ATT$

TSv = VATT

TDEC1 = TAT

Perform "CSMCONIC"

LOSDESRD = unit(RATT - RLMSRCH)

VXRCM = unit(unitVATT * RATT)

TS_v = [REFSMMAT] (VATT - TSv)

If NSRCHPNT = 0:

    RRTARGET = [REFSMMAT] LOSDESRD

If NSRCHPNT = 1:

    UXVECT = unit(VXRCM * LOSDESRD)

    UYVECT = unit(LOSDESRD * UXVECT)

    RRTARGET = [REFSMMAT] unit(K:OFFSTFAC UYVECT + LOSDESRD)

If NSRCHPNT > 1:

    UXVECTPR = UXVECT

    UYVECTPR = UYVECT

    UXVECT = unit(K:SIN60DEG UYVECTPR + K:COS60DEG UXVECTPR)

    UYVECT = unit(K:COS60DEG UYVECTPR - K:SIN60DEG UXVECTPR)

    RRTARGET = [REFSMMAT] unit(K:OFFSTFAC UYVECT + LOSDESRD)

LOSVEL = TS_v

Inhibit interrupts

Remove "STDESIG" from waitlist if it is there

Switch RADMODES bit 15 (CDESFLAG) to 1

Perform "RRDESSM"

If RADLIMCK $\neq$ 2: (not within limits)

Perform "ALARM" with TS = 00527_8

Inhibit interrupts

Remove a call to "CALLDGCH" from waitlist if it is there

Proceed to "ENDOFJOB"

OMEGCALC $TANGNB_0 = CDU_t$

$TANGNB_1 = CDU_s$

Perform "RRNB" (get LOS vector in nav. base coordinates)

OMEGDISP = arccos TS_z

Proceed to "ENDOFJOB"

CALLDGCH If FLAGWRDO bit 7 (RNDVZFLG) = 0, end task

Establish "DATGDCHK" (pr25)

End task

DATGDCHK If bit 4 of channel 33 = 0: (RR data good)

DATAGOOD = K:ALL1S

Inhibit interrupts

Remove "STDESIG" from waitlist if it is there

Proceed to "ENDOFJOB"

If NSRCHPNT = 6, proceed to "LRS24.1"

NSRCHPNT = NSRCHPNT + 1

Proceed to second step of "LRS24.1"

R29 If RADMODES bit 10 (DESIGFLG) = 1, proceed to "R29.LOS"

Inhibit interrupts

Switch RADMODES bit 10 (DESIGFLG) to 1

Switch bit 14 of channel 12 to 0 (disable tracker)

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

Switch FLAGWRD0 bit 1 (OLDESFLG) to 0

Perform "SETRRECR"

If RADMODES bit 12 (ANTENFLG)= 1:

> Call "PREPOS29" in 0.01 second
>
> Switch RADMODES bit 11 (REPOSMON) to 1
>
> Proceed to "NOR29NOW"

Establish "R29REMOJ" (pr21)

Switch RADMODES bit 10 (DESIGFLG) to 0

Switch RADMODES bit 14 (REMODFLG) to 1

Proceed to "NOR29NOW"

R29.LOS TS_t = TIMENOW - PIPTIME

$\underline{TS} = \underline{R}CSM - \underline{R} + TS_t\ (\underline{V}CSM - \underline{V})$

If FLAGWRD2 bit 12 (LOSCMFLG) = 1, proceed to "NOR29NOW"

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 1

$\underline{L}OSSM = \underline{TS}$

$\underline{L}OSVDTd4$ = K:.5SECB17 ($\underline{V}CSM - \underline{V}$)

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

If FLAGWRD0 bit 1 (OLDESFLG) = 1, proceed to "NOR29NOW"

Inhibit interrupts

Switch FLAGWRD0 bit 1 (OLDESFLG) to 1

TS = 100

If PIPCTR > 0, TS = 4

Call "BEGDES29" in TS centiseconds

Release interrupt inhibit

Proceed to "NOR29NOW"

R29REMOJ Call "REMODE" in 0.01 second

Perform "RADSTALL"

End job

PREPOS29 RDES = $-\frac{1}{2}$ (-180°)

Perform "RRTONLY"

Switch RADMODES bit 11 (REPOSMON) to 0

End task

R29READ Establish "R29RDJOB" (pr26)

Delay 2 seconds

If FLAGWRD3 bit 9 (READRFLG) = 1, proceed to "R29READ"

End task

R29RDJOB If FLAGWRD3 bit 11 (NOR29FLG) = 1, proceed to "ENDR29RD"

If RADMODES bit 2 (AUTOMODE) = 1, proceed to "ENDRRD29"

Perform "RRRDOT"

Perform "RADSTALL"

If RADGOOD = 0, proceed to "ENDRRD29"

TS_t = TIMEHOLD

Inhibit interrupts

$TS_{cdut} = CDU_t$

$TS_{cdus} = CDU_s$

$TS_{cduy} = CDU_y$

$TS_{cduz} = CDU_z$

$TS_{cdux} = CDU_x$

R29RANGE Perform "RRRANGE"

Perform "RADSTALL"

If RADGOOD = 0:

If FLAGWRD5 bit 10 (RNGSCFLG) = 0, proceed to "ENDRRD29"

Switch FLAGWRD5 bit 10 (RNGSCFLG) to 0

Proceed to "R29RANGE"

Inhibit interrupts

$RM_0 = DNRADATA_1$

$RM_1 = DNRADATA_2$

$MKTIME = TS_t$

$TANGNB_0 = TS_{cdut}$

$TANGNB_1 = TS_{cdus}$

$AIG = TS_{cduy}$

$AMG = TS_{cduz}$

$AOG = TS_{cdux}$

TRKMKCNT = 1

Release interrupt inhibit

End job

ENDRRD29 Switch channel 12 bit 14 to 0

ENDR29RD TRKMKCNT = 0

Switch FLAGWRD3 bit 9 (READRFLG) to 0

End job

BEGDES29 Establish "R29DODES" (pr21)

Delay 0.5 second

If RADMODES bit 10 (DESIGFLG) = 0, end task

If FLAGWRD2 bit 12 (LOSCMFLG) = 1:

  Delay 0.01 second

  Proceed to third line of "BEGDES29"

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 1

Proceed to "BEGDES29"

R29DODES TANG = 1

If TANG > 0, $\underline{T}Ssm = \underline{L}OSSM$

If TANG = 0:

  $\underline{L}OSSM = \underline{L}OSSM + \underline{L}OSVDTd4$

  $\underline{T}Ssm = \underline{L}OSSM + \underline{L}OSVDTd4$

$\underline{T}S_{LOSSM}$ = unit $\underline{T}Ssm$

If TANG > 0:

  Inhibit interrupts

  $TS_{cdut} = CDU_t$

  $TS_{cdus} = CDU_s$

  $\underline{A}NG = \underline{C}DU$

  Perform "QUICTRIG"

Perform "SMTONB"

$\underline{ULOSNB} = [SMNBMAT] \; \underline{TS}_{LOSSM}$

If TANG = 0, proceed to "R29DPAS2"

Inhibit interrupts

TANG = 0

$TS_{cost} = cos_{sp} \; TS_{cdut}$

$TS_{sint} = sin_{sp} \; TS_{cdut}$

$TS_{sins} = sin_{sp} \; TS_{cdus}$

$TS_{coss} = cos_{sp} \; TS_{cdus}$

$TSm = TS_{coss} \; TS_{cost} \; ULOSNB_z - TS_{sint} \; ULOSNB_y + TS_{cost} \; TS_{sins} \; ULOSNB$

TSm = 2 TSm (cosine of angle between actual LOS and radar LOS)

If overflow, TS = + 1 (overflow will occur when radar LOS converges with actual LOS)

If TS = +1:

Switch bit 14 of channel 12 to 1 (self track enable)

Release interrupt inhibit

Proceed to second step of "R29DODES"

R29DPAS2 $TANG_0 = TS_{coss} \; ULOSNB_x - TS_{sins} \; ULOSNB_z$

$TSm = TS_{sint} \; TS_{sins} \; ULOSNB_x + TS_{cost} \; ULOSNB_y + TS_{coss} \; TS_{sint} \; ULOSNB$

$SHAFTCMD = K{:}RR29GAIN \; TANG_0$

$TRUNNCMD_0 = K{:}RR29GAIN \; TSm$

If bit 4 of channel 33 = 1: (RR tracker not locked on)

Perform "RROUT"

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

End job

Switch RADMODES bit 10 (DESIGFLG) to 0

Switch bit 2 of channel 12 to 0 (disable RRCDU Error Counters)

Switch FLAGWRD3 bit 9 (READRFLG) to 1

TS = 100

If PIPCTR > 0, TS = 4

Call "R29READ" in TS centiseconds

Switch FLAGWRD2 bit 12 (LOSCMFLG) to 0

End job

<u>WLINIT</u> p = WRENDPOS

v = WRENDVEL

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

  p = WSURFPOS

  v = WSURFVEL

s = WSHAFT

t = WTRUN

$$[W] = \begin{bmatrix} p & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & p & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & p & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & v & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & v & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & v & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Switch FLAGWRD5 bit 1 (RENDWFLG) to 1

TRKMKCNT = 0

Return

<u>LGCUPDTE</u> LGRET = return address

Perform "INCORP1"

R22DISPR = $|\underline{DELTAX}_0|$

R22DISPV = $|\underline{DELTAX}_1|$

If R22DISPR > RMAX or if R22DISPV > VMAX:

  Proceed to "R22LEM96" (get astronaut OK)

ASTOK — Perform "INCORP2"

Return via LGRET

INCORP1 — EGRESS = return address

$$\underline{Z}I_0 = \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \end{bmatrix} \underline{B}VECTOR_0 + \begin{bmatrix} W_{41} & W_{51} & W_{61} \\ W_{42} & W_{52} & W_{62} \\ W_{43} & W_{53} & W_{63} \end{bmatrix} \underline{B}VECTOR_1 + \begin{bmatrix} W_{71} & W_{81} & W_{91} \\ W_{72} & W_{82} & W_{92} \\ W_{73} & W_{83} & W_{93} \end{bmatrix} \underline{B}VECTOR_2$$

$$\underline{Z}I_1 = \begin{bmatrix} W_{14} & W_{24} & W_{34} \\ W_{15} & W_{25} & W_{35} \\ W_{16} & W_{26} & W_{36} \end{bmatrix} \underline{B}VECTOR_0 + \begin{bmatrix} W_{44} & W_{54} & W_{64} \\ W_{45} & W_{55} & W_{65} \\ W_{46} & W_{56} & W_{66} \end{bmatrix} \underline{B}VECTOR_1 + \begin{bmatrix} W_{74} & W_{84} & W_{94} \\ W_{75} & W_{85} & W_{95} \\ W_{76} & W_{86} & W_{96} \end{bmatrix} \underline{B}VECTOR_2$$

$$\underline{Z}I_2 = \begin{bmatrix} W_{17} & W_{27} & W_{37} \\ W_{18} & W_{28} & W_{38} \\ W_{19} & W_{29} & W_{39} \end{bmatrix} \underline{B}VECTOR_0 + \begin{bmatrix} W_{47} & W_{57} & W_{67} \\ W_{48} & W_{58} & W_{68} \\ W_{49} & W_{59} & W_{69} \end{bmatrix} \underline{B}VECTOR_1 + \begin{bmatrix} W_{77} & W_{87} & W_{97} \\ W_{78} & W_{88} & W_{98} \\ W_{79} & W_{89} & W_{99} \end{bmatrix} \underline{B}VECTOR_2$$

If FLAGWRD5 bit 9 (DMENFLG) = 0, $\underline{Z}I_2 = 0$

$$TS = VARIANCE + \underline{Z}I_0 \cdot \underline{Z}I_0 + \underline{Z}I_1 \cdot \underline{Z}I_1 + \underline{Z}I_2 \cdot \underline{Z}I_2$$

$$GAMMA = 1 \,/\, (\sqrt{VARIANCE\ TS} + TS)$$

Proceed to "NEWZCOMP"

INCOR2-3

$$\underline{O}MEGA_0 = \underline{Z}I_0 \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \end{bmatrix} + \underline{Z}I_1 \begin{bmatrix} W_{14} & W_{24} & W_{34} \\ W_{15} & W_{25} & W_{35} \\ W_{16} & W_{26} & W_{36} \end{bmatrix} + \underline{Z}I_2 \begin{bmatrix} W_{17} & W_{27} & W_{37} \\ W_{18} & W_{28} & W_{38} \\ W_{19} & W_{29} & W_{39} \end{bmatrix}$$

$$\underline{O}MEGA_1 = \underline{Z}I_0 \begin{bmatrix} W_{41} & W_{51} & W_{61} \\ W_{42} & W_{52} & W_{62} \\ W_{43} & W_{53} & W_{63} \end{bmatrix} + \underline{Z}I_1 \begin{bmatrix} W_{44} & W_{54} & W_{64} \\ W_{45} & W_{55} & W_{65} \\ W_{46} & W_{56} & W_{66} \end{bmatrix} + \underline{Z}I_2 \begin{bmatrix} W_{47} & W_{57} & W_{67} \\ W_{48} & W_{58} & W_{68} \\ W_{49} & W_{59} & W_{69} \end{bmatrix}$$

$$\underline{O}MEGA_2 = \underline{Z}I_0 \begin{bmatrix} W_{71} & W_{81} & W_{91} \\ W_{72} & W_{82} & W_{92} \\ W_{73} & W_{83} & W_{93} \end{bmatrix} + \underline{Z}I_1 \begin{bmatrix} W_{74} & W_{84} & W_{94} \\ W_{75} & W_{85} & W_{95} \\ W_{76} & W_{86} & W_{96} \end{bmatrix} + \underline{Z}I_2 \begin{bmatrix} W_{77} & W_{87} & W_{97} \\ W_{78} & W_{88} & W_{98} \\ W_{79} & W_{89} & W_{99} \end{bmatrix}$$

If FLAGWRD5 bit 9 (DMENFLG) = 0, $\underline{O}MEGA_2 = 0$

$$\underline{D}ELTAX_0 = \underline{O}MEGA_0\ DELTAQ \,/\, TS$$

$$\underline{D}ELTAX_1 = \underline{O}MEGA_1\ DELTAQ \,/\, TS$$

$$\underline{D}ELTAX_2 = \underline{O}MEGA_2\ DELTAQ \,/\, TS$$

Return via EGRESS

NEWZCOMP Set TS_1 to the largest $|\underline{ZI}_i|$ (i = 0, 1, 2)

TS_2 = (number of leading zeros in TS_1) - 2

Shift each $\underline{ZI}_i$ left by TS_2 places (i = 0, 1, 2)

Proceed to "INCOR2-3" (effect of TS_2 taken into account in subsequent scaling shifts)

INCORP2 EGRESS = return address

Perform "INTSTALL" (wait until orbital integration is free)

$\underline{OMEGAM}_0$ = GAMMA $\underline{OMEGA}_0$

$\underline{OMEGAM}_1$ = GAMMA $\underline{OMEGA}_1$

$\underline{OMEGAM}_2$ = GAMMA $\underline{OMEGA}_2$

Switch FLGWRD10 bit 7 (REINTFLG) to 1
(Assure that this job retains control of the integration routines even in case of a restart)

$$\begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \end{bmatrix} = \begin{bmatrix} W_{11} & W_{21} & W_{31} \\ W_{12} & W_{22} & W_{32} \\ W_{13} & W_{23} & W_{33} \end{bmatrix} - \begin{bmatrix} ZI_{0_x} \\ ZI_{0_y} \\ ZI_{0_z} \end{bmatrix} (OMEGAM_{0_x}, OMEGAM_{0_y}, OMAEGAM_{0_z})$$

$$\begin{bmatrix} W_{41} & W_{51} & W_{61} \\ W_{42} & W_{52} & W_{62} \\ W_{43} & W_{53} & W_{63} \end{bmatrix} = \begin{bmatrix} W_{41} & W_{51} & W_{61} \\ W_{42} & W_{52} & W_{62} \\ W_{43} & W_{53} & W_{63} \end{bmatrix} - \begin{bmatrix} ZI_{0_x} \\ ZI_{0_y} \\ ZI_{0_z} \end{bmatrix} (OMEGAM_{1_x}, OMEGAM_{1_y}, OMEGAM_{1_z})$$

If FLAGWRD5 bit 9 (DMENFLG) = 0, skip next step

$$\begin{bmatrix} W_{71} & W_{81} & W_{91} \\ W_{72} & W_{82} & W_{92} \\ W_{73} & W_{83} & W_{93} \end{bmatrix} = \begin{bmatrix} W_{71} & W_{81} & W_{91} \\ W_{72} & W_{82} & W_{92} \\ W_{73} & W_{83} & W_{93} \end{bmatrix} - \begin{bmatrix} ZI_{0_x} \\ ZI_{0_y} \\ ZI_{0_z} \end{bmatrix} (OMEGAM_{2_x}, OMEGAM_{2_y}, OMEGAM_{2_z})$$

$$\begin{bmatrix} W_{14} & W_{24} & W_{34} \\ W_{15} & W_{25} & W_{35} \\ W_{16} & W_{26} & W_{36} \end{bmatrix} = \begin{bmatrix} W_{14} & W_{24} & W_{34} \\ W_{15} & W_{25} & W_{35} \\ W_{16} & W_{26} & W_{36} \end{bmatrix} - \begin{bmatrix} ZI_{1_x} \\ ZI_{1_y} \\ ZI_{1_z} \end{bmatrix} (OMEGAM_{0_x}, OMEGAM_{0_y}, OMEGAM_{0_z})$$

$$\begin{bmatrix} W_{44} & W_{54} & W_{64} \\ W_{45} & W_{55} & W_{65} \\ W_{46} & W_{56} & W_{66} \end{bmatrix} = \begin{bmatrix} W_{44} & W_{54} & W_{64} \\ W_{45} & W_{55} & W_{65} \\ W_{46} & W_{56} & W_{66} \end{bmatrix} - \begin{bmatrix} ZI_{1_x} \\ ZI_{1_y} \\ ZI_{1_z} \end{bmatrix} (OMEGAM_{1_x}, OMEGAM_{1_y}, OMEGAM_{1_z})$$

If FLAGWRD5 bit 9 (DMENFLG) = 0, proceed to "FAZC"

$$\begin{bmatrix} W_{74} & W_{84} & W_{94} \\ W_{75} & W_{85} & W_{95} \\ W_{76} & W_{86} & W_{96} \end{bmatrix} = \begin{bmatrix} W_{74} & W_{84} & W_{94} \\ W_{75} & W_{85} & W_{95} \\ W_{76} & W_{86} & W_{96} \end{bmatrix} - \begin{bmatrix} ZI_{1_x} \\ ZI_{1_y} \\ ZI_{1_z} \end{bmatrix} (OMEGAM_{2_x}, OMEGAM_{2_y}, OMEGAM_{2_z})$$

$$\begin{bmatrix} W_{17} & W_{27} & W_{37} \\ W_{18} & W_{28} & W_{38} \\ W_{19} & W_{29} & W_{39} \end{bmatrix} = \begin{bmatrix} W_{17} & W_{27} & W_{37} \\ W_{18} & W_{28} & W_{38} \\ W_{19} & W_{29} & W_{39} \end{bmatrix} - \begin{bmatrix} ZI_{2_x} \\ ZI_{2_y} \\ ZI_{2_z} \end{bmatrix} (OMEGAM_{0_x}, OMEGAM_{0_y}, OMEGAM_{0_z})$$

$$\begin{bmatrix} W_{47} & W_{57} & W_{67} \\ W_{48} & W_{58} & W_{68} \\ W_{49} & W_{59} & W_{69} \end{bmatrix} = \begin{bmatrix} W_{47} & W_{57} & W_{67} \\ W_{48} & W_{58} & W_{68} \\ W_{49} & W_{59} & W_{69} \end{bmatrix} - \begin{bmatrix} ZI_{2_x} \\ ZI_{2_y} \\ ZI_{2_z} \end{bmatrix} (OMEGAM_{1_x}, OMEGAM_{1_y}, OMEGAM_{1_z})$$

$$\begin{bmatrix} W_{77} & W_{87} & W_{97} \\ W_{78} & W_{88} & W_{98} \\ W_{79} & W_{89} & W_{99} \end{bmatrix} = \begin{bmatrix} W_{77} & W_{87} & W_{97} \\ W_{78} & W_{88} & W_{98} \\ W_{79} & W_{89} & W_{99} \end{bmatrix} - \begin{bmatrix} ZI_{2_x} \\ ZI_{2_y} \\ ZI_{2_z} \end{bmatrix} (OMEGAM_{2_x}, OMEGAM_{2_y}, OMEGAM_{2_z})$$

FAZC $\underline{TX789} = \underline{X789} + \underline{DELTAX}_2$

If FLAGWRD1 bit 8 (VEHUPFLG) = 1, perform "MOVEPCSM"

If FLAGWRD1 bit 8 (VEHUPFLG) = 0, perform "MOVEPLEM"

$\underline{TS} = \underline{TDELTAV} + \underline{DELTAX}_0$

PBODY = 0 (scaling controlled by PBODY)

If FLAGWRD8 bit 11 (LMOONFLG) = 1, PBODY = 2

If overflow:

$\underline{RCV} = \underline{DELTAX}_0 + \underline{RCV}$

$\underline{VCV} = \underline{DELTAX}_1 + \underline{VCV}$

```
        (If overflow:)
            Perform "RECTIFY"
            Proceed to "FAZAB3"
        TDELTAV = TS
        TS = TNUV + DELTAX1
        If overflow:
            VCV = DELTAX1 + VCV
            Perform "RECTIFY"
            Proceed to "FAZAB3"
        TNUV = TS
FAZAB3  If FLAGWRD1 bit 8 (VEHUPFLG) = 1:
            Perform "MOVEACSM"
            Perform "SVDWN1"
        If FLAGWRD1 bit 8 (VEHUPFLG) = 0:
            Perform "MOVEALEM"
            Perform "SVDWN2"
        If FLAGWRD5 bit 9 (DMENFLG) = 1:
            X789 = TX789
        Perform "INTWAKE"
        Return via EGRESS
V67CALL If OVFIND = 1, Switch OVFIND to 0
        Switch FLAGWRD7 bit 8 (V67FLAG) to 0
        Perform "INTSTALL"
```

$WWBIAS = \sqrt{\sum(W_{7i}^2 + W_{8i}^2 + W_{9i}^2)}$ for i = 1-9 (rescaled for display)

$WWPOS = \sqrt{\sum(W_{1i}^2 + W_{2i}^2 + W_{3i}^2)}$ for i = 1-9 (if one overflows, all limited to posmax)

$WWVEL = \sqrt{\sum(W_{4i}^2 + W_{5i}^2 + W_{6i}^2)}$ for i = 1-9

If OVFIND ≠ 0:

OVFIND = 0

WWPOS = K:DPPOSMAX

WWVEL = K:DPPOSMAX

WWBIAS = K:DPPOSMAX

If WWPOS > K:FT99999:

WWPOS = K:FT99999

Perform "INTWAKE"

TS_{WPOS} = WWPOS

TS_{WVEL} = WWVEL

TS_{WBIAS} = WWBIAS

Proceed to "GOXDSPF" with TS = K:V06N99 (WWPOS, WWVEL, WWBIAS)
(If terminate, proceed to "ENDEXT"; if proceed, proceed to next step; other response, repeat this step.)

If TS_{WVEL} - WWVEL + TS_{WPOS} - WWPOS + TS_{WBIAS} - WWBIAS ≠ 0 (crew input)

Switch FLAGWRD7 bit 8 (V67FLAG) to 1

If FLAGWRD7 bit 8 (V67FLAG) = 0, proceed to "ENDEXT"

TS_0 = WWPOS (double precision - rescaled for internal use)

TS_1 = WWVEL (double precision)

TS_2 = WWBIAS (double precision - rescaled for internal use)

If FLAGWRD8 bit 8 (SURFFLAG) = 1:

WSURFPOS = TS_0 single precision

WSURFVEL = TS_1 single precision

WTRUN = TS_2 single precision

WSHAFT = TS_2 single precision

Switch FLAGWRD5 bit 1 (RENDWFLG) to 0

Proceed to "ENDEXT"

WRENDPOS = TS_0 single precision

WRENDVEL = TS_1 single precision

WTRUN = TS_2 single precision

WSHAFT = TS_2 single precision

Switch FLAGWRD5 bit 1 (RENDWFLG) to 0

Proceed to "ENDEXT"

DSPRRLOS Establish "RRLOSDSP" (pr05)

Change priority to 04_8

Perform "GOXDSPFR" with TS = K:V16N56 (RR-AZ,RR-ELEV)
(Any response, switch bit 5 of EXTVBACT to 0 and end job.)

Perform "BLANKET" with TS = 00004_8 (Blank R3)

End job

RRLOSDSP $TS_1 = CDU_t$

$TS_2 = CDU_s$

Perform "RRNBMPAC"

$\underline{TS}_e$ = unit (TS_x, 0, TS_z)

$\underline{TS}_a = \underline{TS}$

COSTH = $\underline{TS}_e \cdot \underline{K}$:UNITZ

SINTH = $\underline{K}$:UNITX $\cdot \underline{TS}_e$

Perform "ARCTRIG"

If THETA < 0, THETA = 1 + THETA

RR-ELEV = THETA

SINTH = $\underline{TS}_a \cdot \underline{K}$:UNITY

COSTH = $\underline{TS}_a \cdot \underline{TS}_e$

Perform "ARCTRIG"

If THETA < 0, THETA = 1 + THETA

RR-AZ = THETA

Delay 1 second

If bit 5 of EXTVBACT = 1, proceed to "RRLOSDSP"

Proceed to "ENDEXT"

ALPHA: See CONC section.

ALT: See COOR section.

ANG: See COOR section.

AOG, AIG, AMG: Single precision storage for ICDU angles, scaled B-1 in units of revolutions and stored in two's complement form.

BVECTOR$_0$: Double precision vector defining the resolution of a navigation measurement into corrections to the position components of the state vector and the W-matrix; scaled B1 and unitless or scaled in units of meters per centisecond. Scaling is changed depending on whether BVECTOR$_0$ or BVECTOR$_1$ has the largest unscaled magnitude. That term is then normalized and BVECTOR$_0$, BVECTOR$_1$, DELTAQ and VARIANCE are rescaled to reflect the change in BVECTOR$_0$

BVECTOR$_1$: Double precision vector defining the resolution of a navigation measurement into corrections to the velocity components of the state vector and the W-matrix; scaled B24 in units of meters. Scaling change is described in BVECTOR$_0$.

BVECTOR$_2$: Double precision vector defining the resolution of a navigation measurement into corrections to the Rendezvous Radar position biases (shaft and trunnion) and the RR components of the W-matrix; scaled B25 in units of meters.

$CDU_{t,s}$: See RADR section.

CDU: See COOR section.

CDUD: See DAPA section.

COSTH: See COOR section.

CSTH: See CONC section.

DATAGOOD: Single precision display register, scaled B14 and unitless. Used in Routine 24 to indicate to the astronaut that the RR has acquired a target.

[illegible]

DELTAQ: Double precision difference between measured and predicted values of a navigation measurement; range measurement scaled B29 (earth) or B27 (moon) in units of meters; range-rate measurement scaled B33 (earth) or B31 (moon) in units of meters squared per centisecond; RR position angle measurement scaled B29 (earth) or B27 (moon) in units of meters. Scaling change described in $\underline{B}VECTOR_0$.

$\underline{D}ELTAX_0$: Double precision position correction vector scaled B29 (earth) or B27 (moon) in units of meters.

$\underline{D}ELTAX_1$: Double precision velocity correction vector, scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

$\underline{D}ELTAX_2$: Double precision Rendezvous Radar position bias correction vector (shaft and trunnion with third component zero) scaled B5 (earth) or B3 (moon) in units of radians.

DESCOUNT: See RADR section.

$DNRADATA_i$: See RADR section.

DSPTEM1: Temporary storage cell used mainly for display interface purposes.

EGRESS: Single precision octal return address storage cell.

EXTVBACT: See EXVB section.

GAMMA: Double precision weighting factor in update of W-matrix, scaled B-40 in units of $meters^{-2}$ or B-48 in units of centiseconds squared per $meter^4$.

GENRET: Single precision octal return address storage cell.

K:.5SECB17: Double precision constant stored as 50×2^{-17}, scaled B17 in units of centiseconds. Equation value 50.

K:1.5SECS: Double precision constant stored as 150×2^{-28}, scaled B28 in units of centiseconds.

K:2PId8: Double precision constant stored as $3.141592653 \times 2^{-2}$, scaled B3 and unitless. Equation value: 6.283185306.

K:30DEG: Double precision constant stored as 0.083333333, scaled B0 in units of revolutions. Equation value: 0.083333333. (Equivalent to 30 degrees.)

K:3SECONDS: Double precision constant stored as 300×2^{-28} scaled B28 in units of centiseconds. Equation value; 300. (Equivalent to 3.0 seconds.)

K:600SEC: Double precision constant stored as 60000×2^{-28}, scaled B28 in units of centiseconds. Equation value: 60000. (Equivalent to 10 minutes.)

K:ALL1S: Single precision constant stored as 11111×2^{-14}, scaled B14 and unitless. Equation value: 11111.

K:COS15DEG: Double precision constant stored as 0.96593×2^{-1}, scaled B1 and unitless. Equation value: cosine of 15 degrees.

K:COS60DEG: Double precision constant stored as 0.5, scaled B0 and unitless. Equation value: 0.5.

K:DPPOSMAX: Double precision constant stored as 37777_8 37777_8.

K:FHNM: Double precision constant stored as 740800×2^{-20}, scaled B20 in units of meters. Equation value: 740800. (Equivalent to 400 nautical miles.)

K:FT99999: Double precision constant stored as 30479×2^{-19}, scaled B19 in units of meters.

K:HALFSEC: Double precision constant stored as 50×2^{-28}, scaled B28 in units of centiseconds. Equation value: 50.

K:IMUVAR: Double precision constant stored as $10^{-6} \times 2^{12}$, scaled B-12 in units of radians squared. Equation value: 10^{-6}.

K:K.01: Double precision constant stored as 0.01, scaled B0. Used to convert output from "LAT-LONG" from units of meters to meters/100.

K:MAXTRIES: Single precision constant stored as 60×2^{-14}, scaled B14 and unitless. Equation value: 60.

K:OFFSTFAC: Double precision constant stored as 0.05678, scaled B0 and unitless. Equation value: 0.05678.

K:RANGCONV: Double precision constant stored as 2.859024×2^{-3}, scaled B3 in units of meters per count. Equation value: 2.859024. (Equivalent to 9.38 feet per count or 2.859 meters per count.)

K:RDOTCONV: Double precision constant stored as -0.0019135344 x 2^7, scaled B-7 in units of meters per centisecond per count. Equation value: -0.0019135344. (Equivalent to -0.6278 fps or -0.19135 meters per second per count.)

K:RR29GAIN: Single precision constant stored as -0.53624, scaled B12 in units of RR drive pulses per radian of error. Equation value: 2196.5. (Equivalent to K:RDESGAIN as defined in RADR section.)

K:SIN60DEG: Double precision constant stored as 0.86603, scaled B0 and unitless. Equation value: 0.86603.

K:TENSEC: Double precision constant stored as 1000 x 10^{-28} scaled B28 in units of centiseconds. Equation value: 1000. (Equivalent to 10 seconds.)

K:THREEDEG: Double precision constant stored as 0.008333333, scaled B0 in units of revolutions. Equation value: 0.008333333. (Equivalent to 3.0 degrees.)

K:UNITX, K:UNITY, K:UNITZ: Double precision constant unit vectors, scaled B1 and unitless. Equation value: (1, 0, 0), (0, 1, 0) and (0, 0, 1), respectively.

LAT: See COOR section.

LGRET: Single precision octal return address storage cell.

LMPOS: Double precision temporary storage vector of the LM position, scaled B29 in units of meters.

LMVEL: Double precision temporary storage vector of the LM velocity, scaled B7 in units of meters per centisecond.

LONG: See COOR section.

LOSCOUNT: See RADR section.

LOSDESRD: Double precision desired line-of-sight vector from the LM to the CSM, a unit vector scaled B1 and unitless.

LOSSM: Double precision LOS vector, scaled B24 in units of meters.

LOSVDTd4: Double precision change in LOS vector derived from $\frac{1}{2}$ second of LOS velocity, scaled B24 in units of meters.

LOSVEL: Double precision velocity vector of CSM with respect to LM, scaled B7 in units of meters per centisecond.

LS21X: Single precision octal return address storage cell.

MKTIME: Double precision time at which the RR range rate measurement is made, considered as "time of the mark" for incorporation purposes, scaled B28 in units of centiseconds.

MLOSV: Double precision magnitude of the LOS vector, scaled B29 in units of meters if FLAGWRDO bit 7 = 0, scaled B20 otherwise. (See RNAV - 8)

MODREG: See DATA section.

N49FLAG: Single precision flag to control the delay in "R22LEM96".

[NBSMMAT]: See COOR section.

NCSMVEL: Double precision velocity vector for the velocity after a plane change in routine "ORBCHGO", scaled B7 (earth) or B5 (moon) in units on meters per centisecond.

NEWPOS: Double precision position vector for the state prior to a plane change maneuver, scaled B29 (earth) or B27 (moon) in units of meters.

NEWVEL: Double precision velocity vector for the state prior to a plane shange maneuver, scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

NSRCHPNT: Single precision counter to direct the search pattern in routine 24 to one of its six options, scaled B14 and unitless.

NUVCSM, NUVLEM: See RCVCSM in ORBI section.

$OMEGA_0$: Double precision vector containing part of the weighting factors to determine the impact of a navigation measurement on each of the position components of the state vector and W-matrix; scaled B39 in units of meters squared or B43 in units of meters cubed per centisecond, for no "NEWZCOMP".

$OMEGA_1$: Double precision vector containing part of the weighting factors to determine the impact of a navigation measurement on each of the velocity components of the state vector and W-matrix; scaled B20 in units of meters squared per centisecond or B24 in units of meters cubed per centisecond squared, for no "NEWZCOMP".

$\underline{O}MEGA_2$: Double precision vector containing part of the weighting factors to determine the impact of a navigation measurement on each of the RR position biases and the RR components of the W-matrix; scaled B15 in units of meters or B19 in units of meters^2/centisecond for no "NEWZCOMP"

$\underline{O}MEGAM_0$, $\underline{O}MEGAM_1$, $\underline{O}MEGAM_2$: Double precision product of GAMMA with $\underline{O}MEGA_0$, $\underline{O}MEGA_1$ and $\underline{O}MEGA_2$; with variable scaling and units.

OMEGDISP: Double precision angle between line-of-sight vector and LM +Z axis, scaled B0 in units of revolutions.

OPTION1, OPTION2: See DATA section.

OVFIND: Single precision cell which is set to some non-zero value if an overflow occurs.

P21ALT: Value of ALT K:K.01 computed in "P21VSAVE" for (optional) display in R1 of N91, scaled factor B29, units (meters/100): see K:K.01.

$\underline{P}$21BASER: Value of P21 "base" vector for position, scale factor B29 (earth) or (moon), units meters (earth/moon). Loaded after completion of integration to specified input time, and used to initialize the integration if bit 11 (P21FLAG) of FLAGWRD0 = 1, thus permitting computation time to be saved if it is desired to iterate about a point which is a number of orbital integration time steps removed from the "permanent" CSM/LM state vector.

$\underline{P}$21BASEV: Value of P21 "base" vector for velocity, scale factor B7 (earth) or B5 (moon), units meters/centisecond. See $\underline{P}$21BASER.

P21GAM: Value of flight path angle computed in "P21VSAVE" for (optional) display in R3 of N91, scale factor B0, units revolutions.

P21ORIG: Single precision cell used to determine scaling pertaining to either earth or moon centered vectors; value of 0 (earth), 2(moon) scaled B14.

P21TIME: Cell used to retain time information, scale factor B28, units centiseconds. Used in P21 to contain the time tag of $\underline{P}$21BASER and $\underline{P}$21BASEV, and to permit the incrementing of the time for which the N34 display is generated

P21VEL: Double precision storage for the magnitude of the predicted velocity of the vehicle (crew option CSM or LM) at the time specified in Noun 34, scaled B7 in units of meters/centisecond.

PBODY: See ORBI section.

PIPCTR: See SERV section.

PIPTIME: See SERV section.

POINTVSM: See ATTM section.

R: See DESC section.

R22DISPR, R22DISPV: Double precision registers for display of navigation update to position and velocity, scaled B29 and B7 in units of meters and meters per centisecond respectively. (Listing mnemonics R22DISP and R22DISP+2).

R65CNTR: Single precision number of passes at six second intervals to be accomplished of routine 65, scaled B14 and unitless.

RADCADR: See RADR section.

RADGOOD: See RADR section.

RADLIMCK: See RADR section.

RADMODES: See RADR section.

RANGEVAR: Double precision, pad loaded variance expected in measured range, scaled B-12 and unitless. Previously a fixed memory constant with an equation value of 0.0033333 squared.

RANGRDOT: Double precision word used to store $DNRADATA_{1,2}$ for telemetry use.

RATEVAR: Double precision, pad loaded variance expected in measured range rate, scaled B-12 and unitless. Previously a fixed memory constant with an equation value of 0.0043333 squared.

RATT: See ORBI section.

RCSM: See SERV section.

RCV: See CONC section.

RCVCSM, RCVLEM: See ORBI section.

RDES: See RADR section.

RDOTM: Double precision measured range rate, scaled B7 in units of meters per centisecond.

RDOTMSAV: Double precision temporary storage location for SAMPLSUM, scaled B28 in units of radar counts (where one count represents -0.19135344 meters/second).

[REFSMMAT]: See COOR section.

REPOSCNT: Single precision cell used as a counter for limiting the number of calls to integration in R26 scaled B14.

REPOSTM: Double precision storage used to save the time of the state vectors used in the previous RR target vector calculation, scaled B28 in units of centiseconds.

RLMSRCH: Double precision LM position vector scaled B29 in units of meters.

RM: Double precision magnitude of measured range, scaled B29 in units of meters. Also used in routine 29 as two single precision storage cells (RM_0 and RM_1) for downlink. They are identical to $DNRADATA_1$ and $DNRADATA_2$, respectively.

RMAX: Pad loaded maximum value of position update allowed without astronaut approval, scaled B19 in units of meters. (Single precision)

RR-AZ: Double precision angle measured from the X-Z nav. base plane to the RR LOS vector, scaled B0 in units of revolutions.

RRBORSIT: Double precision measured line-of-sight vector, a unit vector scaled B1, unitless and expressed in nav. base coordinates.

RRECT: See CONC section.

RR-ELEV: Double precision angle measured from the Z nav. base axis to the projection of the RR LOS vector in the X-Z nav. base plane, scaled B0 in units of revolutions.

RRSHAFT: Double precision measured value of Rendezvous Radar shaft angle, scaled B0 in units of revolutions.

RRTARGET: See RADR section.

RRTRUN: Double precision measured value of Rendezvous Radar trunnion angle, scaled B0 in units of revolutions.

RSUBC: Double precision CSM position storage vector used in the "ORBCHGO" routine, scaled B29 (earth) or B27 (moon) in units of meters.

RSUBL: Double precision LM position storage vector used in the "ORBCHGO" routine, scaled B29 (earth) or B27 (moon) in units of meters.

RVARMIN: Single precision, pad loaded minimum expected value of VARIANCE in a range measurement, scaled B12 in units of meters squared. Changed by the program to a triple precision value scaled B40 in units of meters squared. Previously a fixed memory constant with an equation value of 8.1 squared meters squared or 27 squared feet squared.

RVEC: See CONC section.

RXZ: Double precision component of the LOS vector in the LM X-Z plane, before normalization scaled B29(earth) or B27(moon) in units of meters.

SAMPLSUM: See RADR section.

SCAXIS: See ATTM section.

SHAFTCMD: An alternate mnemonic for TRUNNCMD$_1$, see RADR section.

SHAFTVAR: Single precision pad loaded variance associated with the measured value of the RR shaft angle, scaled B-12 in units of radians squared.

SINTH: See COOR section.

SINTHETA: Double precision sine of the RR trunnion angle, scaled B1 and unitless.

[SMNBMAT]: See COOR section.

SNTH: See CONC section.

T: See CONC section.

TANG$_{0,1}$: See RADR section.

TANGNB$_{0,1}$: See RADR section.

TAT: See ORBI section.

TC: See CONC section.

TDEC1: See ORBI section.

TDELTAV: See ORBI section.

TET: See ORBI section.

TETCSM, TETLEM: See RCVCSM in ORBI section.

THETA: See COOR section.

THETAD: See ATTM section.

TIG: See BURN section.

TIMEHOLD: See RADR section.

TIMENOW: See EXVB section.

TNUV: See ORBI section.

TRKMKCNT: Single precision count of number of navigation updates made during P20 or P22, scaled B14 and unitless. Cell also used in R29 to indicate data storage for down telemetry; 1 - data stored, 0 - data not stored.

$TRUNNCMD_0$: See RADR section.

TRUNVAR: Single precision pad loaded variance associated with the measured value of the RR trunnion angle, scaled B-12 in units of radians^2.

TX789: Temporary storage for updated X789 vector.

UCSM: Double precision unit vector of the extimated CSM position at the orbit change maneuver point, scaled B1 and unitless.

ULC: Computed line-of-sight vector, a unit vector scaled B1.

ULOSNB: Double precision unit vector of the RR LOS in nav. based coordinates, scaled B1 and unitless.

UXVECT: Double precision unit vector used in defining UYVECT for the RR search routine (R 24), scaled B1 and unitless.

UXVECTPR: Double precision storage for the UXVECT of the previous position of the RR in the search routine (R 24), scaled B1 and unitless.

UYVECT: Double precision unit vector used to give the direction of the offset factor from the computed LOS in determining the desired RR boresight vector for the RR search routine (R 24), scaled B1 and unitless.

UYVECTPR: Double precision storage for the UYVECT of the previous position of the RR in the search routine (R 24), scaled B1 and unitless.

V: See DESC section.

VARIANCE: Triple precision variance associated with a navigation measurement, scaled B40 in units of meters squared or in units of meters^4 per centisecond squared. Scaling changed as described in $BVECTOR_0$ in this section.

VATT: See ORBI section.

VCSM: See SERV section.

VCV: See CONC section.

VCVCSM, VCVLEM: See RCVCSM description in ORBI section.

VMAX: Pad loaded maximum value of velocity update allowed without astronaut approval, scaled B7 in units of meters/centisecond, single precision.

VRECT: See CONC section.

VSUBC: Double precision CSM velocity storage vector used in the "ORBCHGO" routine scaled B7 (earth) or B5 (moon) in units of meters per centisecond.

VVARMIN: Single precision, pad loaded minimum expected value of range rate measurement variance, scaled B-12 in units of meters squared per centisecond squared. Changed by the program to a double precision value scaled B4 in the same units. Previously a fixed memory constant with an equation value of 0.0013 squared meters squared per centisecond squared per second squared.

VVEC: See CONC section.

VXRCM: Double precision unit vector defining the CSM orbital plane for the RR search routine, scaled B1 and unitless.

[W]: Double precision "error transition matrix," a 9x9 matrix (whose last three rows and columns are not always maintained) defined such that the covariance matric E equals W W^T. The individual elements of the matrix are denoted by two subscripts, the first indicating the row, the second indicating the column. The first three rows are scaled B19 in units of meters; the middle three rows are scaled B0 in units of meters per centisecond; the last three rows are scaled B-5 in units of radians. Because of LGC limitations on vector manipulations, the elements of the W-matrix are stored in the following order:

Address	Standard	Address	Standard	Address	Standard
W+0	W_{11}	W+54	W_{41}	W+108	W_{71}
W+2	W_{21}	W+56	W_{51}	W+110	W_{81}
W+4	W_{31}	W+58	W_{61}	W+112	W_{91}
W+6	W_{12}	W+60	W_{42}	W+114	W_{72}
W+8	W_{22}	W+62	W_{52}	W+116	W_{82}
W+10	W_{32}	W+64	W_{62}	W+118	W_{92}
W+12	W_{13}	W+66	W_{43}	W+120	W_{73}
⋮	⋮	⋮	⋮	⋮	⋮
W+46	W_{38}	W+100	W_{68}	W+154	W_{98}
W+48	W_{19}	W+102	W_{49}	W+156	W_{79}
W+50	W_{29}	W+104	W_{59}	W+158	W_{89}
W+52	W_{39}	W+106	W_{69}	W+160	W_{99}

In other words:

$$[W] = W+ \begin{bmatrix} 0 & 6 & 12 & 18 & 24 & 30 & 36 & 42 & 48 \\ 2 & 8 & 14 & 20 & 26 & 32 & 38 & 44 & 50 \\ 4 & 10 & 16 & 22 & 28 & 34 & 40 & 46 & 52 \\ 54 & 60 & 66 & 72 & 78 & 84 & 90 & 96 & 102 \\ 56 & 62 & 68 & 74 & 80 & 86 & 92 & 98 & 104 \\ 58 & 64 & 70 & 76 & 82 & 88 & 94 & 100 & 106 \\ 108 & 114 & 120 & 126 & 132 & 138 & 144 & 150 & 156 \\ 110 & 116 & 122 & 128 & 134 & 140 & 146 & 152 & 158 \\ 112 & 118 & 124 & 130 & 136 & 142 & 148 & 154 & 160 \end{bmatrix}$$

WHCHREAD: Single precision code to indicate which navigation measurement is being incorporated into a state vector scaled B14 (1-range, 2-range rate, 3-shaft, and 4-trunnion).

WRENDPOS, WRENDVEL, WSURFPOS, WSURFVEL, WSHAFT, WTRUN: Single precision initial estimates for the uncorrelated variance in spacecraft position, spacecraft velocity in flight and on the surface, and Rendezvous Radar position estimates, scaled B14, B0, B14, B0, B-5, B-5 in units of meters, meters per centisecond, meters, meters per centisecond, radians and radians respectively, pad loaded value.

WWPOS, WWVEL, WWBIAS: Double precision square roots of the sums of the squares of the position, velocity, and shaft elements of the W-matrix, scaled B19, B0, and B5, respectively, in units of meters, meters per centisecond, and radians.

$\underline{X}789$: Double precision vector containing the best estimate of bias necessary to offset RR position error, scaled B5 (earth) or B3 (moon) in units of radians, pad loaded value.

$\underline{X}NB_{ref}$, $\underline{Y}NB_{ref}$, $\underline{Z}NB_{ref}$: Double precision unit vectors in the directions of the LM + X, +Y and +Z navigation base axes respectively, scaled B1 and unitless; expressed in reference coordinates.

XPREV: See CONC section.

$\underline{Z}I_0$, $\underline{Z}I_1$, $\underline{Z}I_2$: Double precision intermediate vector quantities in the navigation measurement updates of the state vector and the W-matrix, scaled B20 in units of meters or B24 in units of meters squared per centisecond, before "NEWZCOMP" shifts.

SERV

Servicer

PREREAD Establish "LASTBIAS" (pr21)

Perform "PIPASR" skipping first step (PIPTIME1 unchanged)

Switch FLAGWRD7 bits 6 (V37FLAG) and 5 (AVEGFLAG) to 1

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 0

Establish "NORMLIZE" (pr22)

Delay 2 seconds

READACCS TS = TIME5 - 16377

If TS ≠ 0, TIME5 = TIME5 - 1 sign(TS)

Perform "PIPASR"

Establish "SERVICER" (pr20)

Switch bit 9 of channel 11 to 1 (test connector output)

If FLAGWRD7 bit 5 (AVEGFLAG) = 0:

AVEGEXIT = "AVGEND"

End task

If FLAGWRD6 bit 8 (MUNFLAG) = 1:

PIPCTR = 7

Call "R10,R11" in 0.2 second

Call "READACCS" in 2 seconds

End task

PIPASR PIPTIME1 = TIMENOW

$DELV_z$ = 0

$DELV_y$ = 0

$DELV_{x_{ls}}$ = 0

TEM = - PIPA

DELV = PIPA (single precision into more significant halves of DELV components)

PIPA = 0 (-0)

PGUIDE = PIPTIME1 - PIPTIME

CDUTEMP = CDU

PIPATMP = DELV

Return

NORMLIZE If FLAGWRD6 bit 8 (MUNFLAG) = 1:

R = [REFSMMAT] RN1

Perform "MUNGRAV" with TSr = R

V = [REFSMMAT] VN1

UHYP = unit(VCSM * RCSM)

If FLAGWRD6 bit 8 (MUNFLAG) = 0:

Perform "CALCGRAV" with TSr = RN1

Inhibit interrupts

Perform "COPYCYC" skipping first two steps (MASS unchanged)

Release interrupt inhibit

End job

SERVICER 1dPIPADT = K:PRIO31

Perform "1/PIPA"

ABDELV = |DELV|

ABDVCONV = K:KPIP ABDELV

MASS1 = MASS

If FLAGWRD8 bit 8 (SURFFLAG) = 0:

TSv = K:DPSVEX

If FLGWRD10 bit 13 (APSFLAG) = 1, TSv = K:APSVEX

MASS1 = MASS1 + MASS ABDVCONV / TSv

DVTOTAL = DVTOTAL + K:KPIP1 |DELV|

Perform "QUICTRIG" with ANG = CDUTEMP

XNBPIP = (COSIGA COSMGA, SINMGA, - SINIGA COSMGA)

$ZNBPIP_z$ = COSIGA COSOGA - SINIGA SINOGA SINMGA

$ZNBPIP_y$ = - SINOGA COSMGA

$ZNBPIP_x$ = COSOGA SINIGA + COSIGA SINOGA SINMGA

$\underline{Y}$NBPIP = $\underline{Z}$NBPIP * $\underline{X}$NBPIP

AVERAGEG If FLAGWRD6 bit 8 (MUNFLAG) = 1, perform "RVBOTH"

If FLAGWRD6 bit 8 (MUNFLAG) = 0, perform "CALCRVG"

Perform "COPYCYC"

$\underline{P}$IPATMP = 0

Switch FLAGWRD2 bit 11 (STEERSW) to 0

If FLAGWRD7 bit 7 (IDLEFLAG) = 0:

  If FLAGWRD6 bit 2 (AUXFLAG) = 1:

    Proceed to "DVMON"

  Switch FLAGWRD6 bit 2 (AUXFLAG) to 1

  Switch DAPBOOLS bit 14 (USEQRJTS) to 1

  Proceed to "SERVOUT"

Switch FLAGWRD6 bit 2 (AUXFLAG) to 0

Switch DAPBOOLS bit 14 (USEQRJTS) to 1

Proceed to "SERVOUT"

DVMON If ABDELV $\leq$ DVTHRUSH:

  If DVCNTR = 0:

    If "COMFAIL" is already active: (determined from restart tables)

      Proceed to "SERVOUT"

    Establish "COMFAIL" (pr25)

    Proceed to "SERVOUT"

```
          (If ABDELV ≤ DVTHRUSH:)
               DVCNTR = DVCNTR - 1
               Inhibit interrupts
               Perform "STOPRATE"
               Switch DAPBOOLS bit 14 (USEQRJTS) to 1
               Proceed to "SERVOUT"
          Switch FLAGWRD2 bit 11 (STEERSW) to 1
          DVCNTR = 1
          If FLGWRD10 bit 13 (APSFLAG) = 0:
               If bit 9 of channel 32 = 1:
                    Switch DAPBOOLS bit 14 (USEQRJTS) to 0
                    Proceed to "SERVOUT"
          Switch DAPBOOLS bit 14 (USEQRJTS) to 1
SERVOUT   Release interrupt inhibit
          Perform "1/ACCS"
          Proceed to AVEGEXIT
COPYCYC   Inhibit interrupts
          MASS = MASS1
          GDT = GDT1
          PIPTIME = PIPTIME1
          VN = VN1
          RN = RN1
          Return
SERVEXIT  End job
```

AVGEND 1dPIPADT = less significant half of PIPTIME

Switch FLAGWRD2 bit 15 (DRIFTFLG) to 1

Perform "PIPFREE"

Switch bit 9 of channel 11 to 0 (test connector output)

Switch FLAGWRD3 bit 11 (NOR29FLG) to 1

Switch FLAGWRD7 bit 11 (SWANDISP) to 0

Switch FLAGWRD6 bit 8 (MUNFLAG) to 0

Perform "AVETOMID"

Switch FLAGWRD7 bit 6 (V37FLAG) to 0

Proceed to "V37RET"

SERVIDLE AVEGEXIT = "SERVEXIT"

Switch FLAGWRD7 bit 7 (IDLEFLAG) to 1

Switch FLGWRD11 to 40000_8 (bypass all LR updates)

If FLAGWRD6 bit 8 (MUNFLAG) = 1:

Maintain Group 2 in restart logic

Maintain Group 5 in restart logic

Establish "GOTOPOOH" in Group 4 of restart logic with priority 31_8

Clear all other restart logic

Proceed to "WHIMPER"

CALCRVG $\underline{DELVREF}$ = K:KPIP1 $\underline{DELV}$ [REFSMMAT]

$\underline{RN1} = \underline{RN} + PGUIDE\ (\underline{VN} + \frac{1}{2}\ \underline{DELVREF} + \frac{1}{2}\ \underline{GDT})$

Perform "CALCGRAV" with $\underline{TSr} = \underline{RN1}$

$\underline{VN1} = \underline{VN} + \underline{DELVREF} + \frac{1}{2}\ (\underline{GDT} + \underline{GDT1})$

Return

CALCGRAV RMAGSQ = $\left|\underline{T}Sr\right|^2$

$\underline{U}$NITR = unit$\underline{T}$Sr

If RTX2 = 0: (earth orbit)

TSsel = $\underline{K}$:UNITZ · $\underline{U}$NITR

TS = (1 - 5 $TSsel^2$) / 20

TSrdr = K:RESQ / RMAGSQ

$\underline{U}$NITGOBL = TSrdr K:20J TS $\underline{U}$NITR + TSrdr K:2J TSsel $\underline{K}$:UNITZ

$\underline{G}$DT1 = $K{:}mMUDT_{RTX2}$ ($\underline{U}$NITGOBL + $\underline{U}$NITR) / RMAGSQ

If RTX2 = 2: (lunar orbit)

$\underline{G}$DT1 = $K{:}mMUDT_{RTX2}$ $\underline{U}$NITR / RMAGSQ

Return

RVBOTH $\underline{R}$1S = $\underline{R}$CSM + PGUIDE ($\underline{V}$CSM + $\frac{1}{2}$ $\underline{G}$CSM)

Perform "MUNGRAV" with $\underline{T}$Sr = $\underline{R}$1S

$\underline{V}$1S = $\underline{V}$CSM + $\frac{1}{2}$ ($\underline{G}$CSM + $\underline{G}$DT1)

$\underline{G}$CSM = $\underline{G}$DT1

$\underline{R}$CSM = $\underline{R}$1S

$\underline{V}$CSM = $\underline{V}$1S

$\underline{T}$Sdv = K:KPIP2 $\underline{D}$ELV

$\underline{R}$1S = $\underline{R}$ + PGUIDE ($\underline{V}$ + $\frac{1}{2}$ $\underline{T}$Sdv + $\frac{1}{2}$ $\underline{G}$DT)

Perform "MUNGRAV" with $\underline{T}$Sr = $\underline{R}$1S

$\underline{V}$1S = $\underline{V}$ + $\underline{T}$Sdv + $\frac{1}{2}$ ($\underline{G}$DT + $\underline{G}$DT1)

ABVEL = $\left|\underline{V}1S\right|$

HDOTDISP = $\underline{U}$NITR · $\underline{V}$1S

$\underline{D}$ELVS = $\underline{R}$1S * $\underline{W}$M

HCALC = $\left|\underline{R}1S\right|$ - LANDMAG

Proceed to "MUNRETRN"

MUNGRAV UNITR = unit TSr

RMAGSQ = $|\underline{TS}r|^2$

GDT1 = K:mMUDT$_2$ UNITR / RMAGSQ

Return

MUNRETRN If FLGWRD11 bit 15 (LRBYPASS) = 1, proceed to "COPYCYC1"

If FLGWRD11 bit 9 (XORFLG) = 0:

  If HCALC < K:30kft:

    Switch DAPBOOLS bit 9 (XOVINHIB) to 1

    Switch FLGWRD11 bit 9 (XORFLG) to 1

If FLGWRD11 bit 10 (NOLRREAD) = 1, proceed to "CONTSERV"

If FLGWRD11 bit 3 (NO511FLG) = 1, proceed to "UPDATCHK"

If FLGWRD11 bit 11 (PSTHIGAT) = 1:

  If bit 7 of channel 33 = 0: (LR in position #2)

    Proceed to "UPDATCHK"

  Perform "ALARM" with TS = 00511_8

  Proceed to "CONTSERV"

If TTF $\leq$ - RPCRTIME or XNBPIP$_x$ < RPCRTQSW:

  If bit 6 of channel 33 = 0: (LR in position #1)

    Proceed to "UPDATCHK"

  Perform "ALARM" with TS = 00511_8

  Proceed to "CONTSERV"

Establish "HIGATJOB" (pr32)

Switch FLGWRD11 bits 11 (PSTHIGAT) and 10 (NOLRREAD) to 1

Proceed to the second step of "CONTSERV"

HIGATJOB Perform "LRPOS2"

Perform "RADSTALL"

If RADGOOD = 0: (bad return from "RADSTALL")

POSALARM Perform "PRIOLARM" with TS = 00523_8
(If terminate, end job; if proceed, proceed to "P1CHK";
if other response, proceed to "P2CHK".)

End job

POSGOOD Change job priority to 23 (pr23)

Perform "SETPOS2"

Switch FLGWRD11 bit 6 (LPOS2FLG) to 1

Switch FLGWRD11 bit 10 (NOLRREAD) to 0

End job

P1CHK Switch FLGWRD11 bit 3 (NO511FLG) to 1

If bit 6 of channel 33 = 0: (LR in position #1)

Switch FLGWRD11 bit 10 (NOLRREAD) to 0

End job

Proceed to "POSGOOD"

P2CHK If bit 7 of channel 33 = 0: (LR in position #2)

Proceed to "POSGOOD"

Proceed to "POSALARM"

UPDATCHK If FLGWRD11 bit 10 (NOLRREAD) = 1, proceed to "CONTSERV"

If FLGWRD11 bit 4 (RNGEDATA) = 0, proceed to "VMEASCHK"

TSh = HBEAMNB [XNBPIP]

TS = RADSKAL TSh · (V1S + DELVS)

If RADMODES bit 9 (ALTSCALE) = 0, TS = SKALSKAL TS

DELTAH = K:HSCAL (TS + HMEAS) TSh · UNITR - HCALC

If FLGWRD11 bit 11 (PSTHIGAT) = 0, proceed to "NOREASON"

TS = |DELTAH| - DELQFIX - HCALC / 8

LRLCTR = LRLCTR + 1

If TS $\geq$ 0: (DELTAH too large)

If LRRCTR $\neq$ 0 and LRLCTR - LRRCTR $<$ 4:

Switch FLGWRD11 bit 1 (HFLSHFLG) to 1

LRRCTR = LRLCTR

Proceed to "VMEASCHK"

Switch FLGWRD11 bit 1 (HFLSHFLG) to 0

NOREASON If FLGWRD11 bit 8 (LRINH) = 0, proceed to "VMEASCHK"

If HCALC $<$ HLROFF:

Switch FLGWRD11 bit 8 (LRINH) to 0

Proceed to "VMEASCHK"

If HCALC $\geq$ LRHMAX, proceed to "VMEASCHK"

TS = DELTAH LRWH (LRHMAX - HCALC) / LRHMAX

TSr = R1S + TS UNITR

Perform "MUNGRAV"

R1S = TSr

VMEASCHK If FLGWRD11 bit 7 (VELDATA) = 0, proceed to "VALTCHK"

ANG = LRCDU

Perform "QUICTRIG"

If VSELECT = 0, TSuv = VZBEAMNB

If VSELECT = 1, TSuv = VYBEAMNB

If VSELECT = 2, TSuv = VXBEAMNB

Perform "NBTOSM"

j = 2 VSELECT

VBEAM = [NBSMMAT] TSuv

TSgv = GDT (LRVTIME - PIPTIME) / K:2SECb28

TS = TSgv + V + K:KPIP1 PIPTEM + DELVS

VEST = TS $\cdot$ VBEAM

$DELTAV = K{:}VSCAL_j \; \underline{VMEAS} - \underline{VEST}$

$TS = |DELTAV| - (|\underline{TS}| / 8 + VELBIAS)$

LRMCTR = LRMCTR + 1

If TS $\geq$ 0: (DELTAV too large)

  If LRSCTR $\neq$ 0 and LRMCTR - LRSCTR < 4:

    Switch FLGWRD11 bit 2 (VFLSHFLG) to 1

  LRSCTR = LRMCTR

  If VSELECT = 0, switch FLGWRD11 bit 12 (VXINH) to 1

  Proceed to "VALTCHK"

Switch FLGWRD11 bit 2 (VFLSHFLG) to 0

If FLGWRD11 bit 12 (VXINH) = 1:

  Switch FLGWRD11 bit 12 (VXINH) to 0

  If VSELECT = 2, proceed to "VALTCHK"

If FLGWRD11 bit 8 (LRINH) = 0, proceed to "VALTCHK"

If ABVEL $\leq$ LRVF:

  $TS = LRWVF_{VSELECT}$

  Proceed to "WSTOR"

If LRVMAX $\leq$ ABVEL:

  TS = 0

  Proceed to "WSTOR"

$TS = LRWV_{VSELECT} \, (LRVMAX - ABVEL) / LRVMAX$

<u>WSTOR</u> If MODREG > 64:

  TS = LRWVFF

$TS_{dp} = (TS, 0)$

$\underline{TS}v = \underline{V}1S + TS_{dp} \; DELTAV \; \underline{V}BEAM$

$\underline{V}1S = \underline{TS}v$

Proceed to "VALTCHK"

VALTCHK If FLGWRD11 bit 5 (READVEL) = 0:

If ABVEL $\geq$ K:6KFTdSEC, proceed to "CONTSERV"

Switch FLGWRD11 bit 5 (READVEL) to 1

Establish "LRVJOB" (pr32)

CONTSERV Inhibit interrupts

Switch FLGWRD11 bits 4 (RNGEDATA) and 7 (VELDATA) to 0

COPYCYC1 If FLAGWRD3 bit 9 (READRFLG) = 1 or FLAGWRD3 bit 11 (NOR29FLG) = 1 or RADMODES bit 13 (RCDUOFLG) = 1 or RADMODES bit 2 (AUTOMODE) = 1:

Switch RADMODES bit 10 (DESIGFLG) to 0

Proceed to "NOR29NOW"

If RADMODES bit 14 (REMODFLG) and bit 11 (REPOSMON) = 0:

Proceed to "R29"

NOR29NOW Release interrupt inhibit

HCALC = $|\underline{R}1S|$ - LANDMAG

HCALC1 = HCALC

ALTBITS = K:ALTCONV HCALC

$\underline{UHZP}$ = unit($\underline{UNITR}$ * $\underline{UHYP}$)

$\underline{RN}1$ = $\underline{R}1S$ [REFSMMAT] (= $[REFSMMAT]^T$ $\underline{R}1S$)

$\underline{VN}1$ = $\underline{V}1S$ [REFSMMAT]

TS = K:ARCONV1 $|\underline{UNITR} * \underline{V}1S|^2 / |\underline{R}1S|$

Inhibit interrupts

$\underline{RUNIT}_{sp}$ = $\underline{UNITR}$

$DALTRATE_{sp}$ = TS

$\underline{R}$ = $\underline{R}1S$

$\underline{V}$ = $\underline{V}1S$

Return (to caller of "RVBOTH")

LRHJOB Perform "LRALT"

Perform "RADSTALL"

If RADGOOD = 0:

  If FLAGWRD5 bit 10 (RNGSCFLG) = 1:

    Switch FLAGWRD5 bit 10 (RNGSCFLG) to 0

    End job

  STILBADH = 2

  End job

If STILBADH > 0:

  STILBADH = STILBADH - 1

  End job

Inhibit interrupts

HMEAS = SAMPLSUM

MKTIME = PIPTIME1

AIG = $CDUTEMP_y$

AMG = $CDUTEMP_z$

AOG = $CDUTEMP_x$

Switch FLGWRD11 bit 4 (RNGEDATA) to 1

Release interrupt inhibit

End job

LRVJOB Call "RDGIMS" in 0.17 second

TSn = 5

If VSELECT = 0, perform "LRVELX"

```
          If VSELECT = 1, perform "LRVELZ"
          If VSELECT = 2, perform "LRVELY"
          Perform "RADSTALL"
          If RADGOOD = 0:
               STILBADV = 2
               Proceed to "ENDLRV"
          If STILBADV > 0:
               STILBADV = STILBADV - 1
               Proceed to "ENDLRV"
          Inhibit interrupts
          VMEAS = SAMPLSUM
          LRVTIMDL = LRVTIME
          LRCDUDL = LRCDU
          Switch FLGWRD11 bit 7 (VELDATA) to 1
ENDLRV    If VSELECT = 0, VSELECT = 3
          VSELECT = VSELECT - 1
          End job
RDGIMS    LRVTIME = TIMENOW
          LRCDU = CDU
          PIPTEM = PIPA
          End task
```

R10,R11 If FLAGWRD7 bit 5 (AVEGFLAG) = 0, end task

If PIPCTR = 0: (PIPTIME + 1.95 seconds)

If FLGWRD11 bit 15 (LRBYPASS) = 0
and bit 10 (NOLRREAD) = 0:

Establish "LRHJOB" (pr32)

Skip next two (2) steps

PIPCTR1 = PIPCTR - 1

Call "R10,R11" in 0.25 second

If FLGWRD11 bit 1 (HFLSHFLG) = 1:

Invert bit 5 of $DSPTAB_{11}$

Switch bit 15 of $DSPTAB_{11}$ to 1 (flag for output)

If FLGWRD11 bit 2 (VFLSHFLG) = 1:

Invert bit 3 of $DSPTAB_{11}$

Switch bit 15 of $DSPTAB_{11}$ to 1 (flag for output)

If FLAGWRD9 bit 9 (LETABORT) = 0:

Proceed to "LANDISP"

If MODREG = 71, proceed to "LANDISP"

If bit 4 of channel 30 = 0: (abort stage)

Proceed to "P71A"

If MODREG = 70, proceed to "LANDISP"

If bit 1 of channel 30 = 0: (abort)

Proceed to "P70A"

Proceed to "LANDISP"

LANDISP PIPCTR = PIPCTR1

If FLAGWRD7 bit 11 (SWANDISP) = 0, proceed to "DISPRSET"

LADQSAVE = "ALTROUT1"

If IMODES33 bit 7 = 1, LADQSAVE = "ALTOUT1"

If bit 6 of channel 30 = 1, proceed to "DISPRSET"
(inertial data display discrete is reset)

If FLAGWRD1 bit 14 (DIDFLAG) = 1, proceed to "SPEEDRUN"

Switch FLAGWRD1 bit 14 (DIDFLAG) to 1

Switch IMODES33 bit 7 to 0 (display rate first)

If FLAGWRD0 bit 2 (R10FLAG) = 1, end task

Switch bit 8 of channel 12 to 1 (set inertial data display moding discrete)

TRAKLATV = 0

TRAKFWDV = 0

LATVMETR = 0

FORVMETR = 0

Call "INTLZE" in 0.08 second

End task

INTLZE Switch bit 2 of channel 12 to 1 (enable RRCDU error counter)

Switch IMODES33 bit 8 to 1

End task

SPEEDRUN DT_{sp} = TIMENOW - PIPTIME

$\underline{V}VECT = \frac{1}{2}\ \underline{G}DT\ DT\ /\ K{:}1SEC$

$\underline{V}VECT_{sp} = \underline{V}VECT + \underline{V} + K{:}KPIP1b5\ (\underline{P}IPA + \underline{P}IPATMP)$

Delay 0.04 second

If FLAGWRD0 bit 2 (R10FLAG) = 1, proceed to LADQSAVE

If bit 2 of channel 12 = 0, proceed to "DISPRSET"

$\underline{T}S = \underline{V}VECT + \underline{D}ELVS_{ms}$

```
VHY = TS · UHYP

VHZ = TS · UHZP

LATVEL = K:VELCONV (M32 VHY + M22 VHZ)

FORVEL = K:VELCONV (M32 VHZ - M22 VHY)

If |FORVEL| < K:MAXVBITS:

     If TRAKFWDV FORVEL ≥ 0:

          TS = FORVEL - FORVMETR

          If TRAKFWDV = 0 and FORVEL FORVMETR < 0:

               If |TS| > K:MAXVBITS, TS = K:MAXVBITS signTS

     If TRAKFWDV FORVEL < 0:

          TS = - FORVMETR

     TRAKFWDV = 0

If |FORVEL| ≥ K:MAXVBITS:

     If FORVMETR FORVEL < 0:

          TS = K:MAXVBITS signFORVEL

          i = 0

     If FORVMETR FORVEL ≥ 0:

          If TRAKFWDV FORVEL ≥ 0:

               TS = K:MAXVBITS signFORVEL - FORVMETR

               i = 1 signFORVEL

          If TRAKFWDV FORVEL < 0:

               TS = K:MAXVBITS signFORVEL

               i = 0

          TRAKFWDV = i

CDUSCMD = TS

FORVMETR = FORVMETR + TS
```

If $|LATVEL| < K:MAXVBITS$:

If TRAKLATV LATVEL $\geq$ 0:

TS = LATVEL - LATVMETR

If TRAKLATV = 0 and LATVEL LATVMETR $<$ 0:

If $|TS| > K:MAXVBITS$, TS = K:MAXVBITS signTS

If TRAKLATV LATVEL $<$ 0:

TS = - LATVMETR

TRAKLATV = 0

If $|LATVEL| \geq K:MAXVBITS$:

If LATVMETR LATVEL $<$ 0:

TS = K:MAXVBITS signLATVEL

i = 0

If LATVMETR LATVEL $\geq$ 0:

If TRAKLATV LATVEL $\geq$ 0:

TS = K:MAXVBITS signLATVEL - LATVMETR

i = 1 signLATVEL

If TRAKLATV LATVEL $<$ 0:

TS = K:MAXVBITS signLATVEL

i = 0

TRAKLATV = i

CDUTCMD = TS

LATVMETR = LATVMETR + TS

Switch bits 11 and 12 of channel 14 to 1

Proceed to LADQSAVE

ALTROUT1 Switch IMODES33 bit 7 to 1

Switch bit 2 of channel 14 to 1 (select altitude rate display)

ALTRATE = DT DALTRATE + K:ARCONV $\underline{R}$UNIT · $\underline{V}$VECT

ALTM = - ALTRATE

If ALTM $\leq$ 0, ALTM = ALTRATE with bit 15 switched to 1

Switch bit 3 of channel 14 to 1 (altitude meter activity bit)

End task

ALTOUT1 Switch IMODES33 bit 7 to 0

Switch bit 2 of channel 14 to 0 (select altitude display)

TS = K:ARTOA

If ALTBITS $\geq$ 0:

ALTSAVE = ALTBITS

ALTBITS = - 1

TS = K:ARTOA2 DT

ALTSAVE = ALTSAVE + TS ALTRATE

If ALTSAVE < 0, ALTSAVE = 0

TS = ALTSAVE

If ALTSAVE $\geq$ K:altlim, TS = 2^{14} (fractional part of (ALTSAVE/2^{14})) + 2^{14}

ALTM = TS

Switch bit 3 of channel 14 to 1 (altitude meter activity bit)

End task

DISPRSET If FLAGWRDO bit 2 (R1OFLAG) = 0:

If IMODES33 bit 8 = 1:

Switch bit 2 of channel 12 to 0

Switch bit 8 of channel 12 to 0

Switch bits 7 and 8 of IMODES33 to 0

Switch FLAGWRD1 bit 14 (DIDFLAG) to 0

End task

```
SETPOS1     STILBADH = 2
            STILBADV = 2
            LRLCTR = 0
            LRMCTR = 0
            LRRCTR = 0
            LRSCTR = 0
            VSELECT = 0
            ANG = (LRALPHA_1, LRBETA_1, 0)
            Perform "SETPOS"
            Return

SETPOS2     ANG = (LRALPHA_2, LRBETA_2, 0)
            Perform "SETPOS"
            Return

SETPOS      Perform "CD*TR*GS"
            Perform "SMTONB"
            VYBEAMNB = [SMNBMAT] K:UNITY
            VXBEAMNB = [SMNBMAT] K:UNITX
            VZBEAMNB = VXBEAMNB * VYBEAMNB
            HBEAMNB = [SMNBMAT] K:HBEAMANT
            Return
```

1dPIPADT: See IMUC section.

ABDELV: Single precision magnitude of sensed change in velocity (DELV), scaled B14 in units of centimeters per second.

ABDVCONV: Double precision magnitude of DELV converted to units of meters per centisecond and scaled B5.

ABVEL: Double precision magnitude of velocity for display, scaled B7 in units of meters per centisecond.

AIG, AMG, AOG: **Single precision storage for** $CDUTEMP_y$, $CDUTEMP_z$ **and** $CDUTEMP_x$ **respectively for downlink purposes.**

ALTBITS: Double precision altitude computed for display on the tape-drive altitude meter, scaled B28 in units of Analog-altitude-display bits. Set to -1 to indicate that it has not been updated since the last time it was sampled to drive the display.

ALTM: Single precision cell used to provide altitude and altitude rate information to the tape-drive altitude and altitude rate meters, scaled B14 in units of Analog-altitude-display bits or Analog-altitude-rate-display bits. One Analog-altitude-display bit is equivalent to 0.714756 meters (2.345 feet), and one Analog-altitude-rate-display bit is equivalent to 0.1524 meters per second (0.5 feet per second). Data is provided to the meters in serial binary form at a 3200 pps rate when bit 3 of channel 14 is set, and bit 2 of channel 14 is set or reset by the program to distinguish between altitude-rate (1) and altitude (0) information.

ALTRATE: Single precision altitude rate calculated for display on the tape-drive altitude rate meter, scaled B14 in units of Analog-altitude-rate-display bits.

ALTSAVE: Double precision storage for previous altitude value for altitude meter display, scaled B28 in units of Analog-altitude-display bits.

ANG: See COOR section.

AVEGEXIT: Double precision variable address (program notation also AVGEXIT) to branch to the guidance routines specified by whichever program is controlling a burn.

CDU: See IMUC section.

CDUSCMD, CDUTCMD: Single precision cells loaded with values to be transmitted to the Error Counters in the two Rendezvous Radar channels of the Coupling Data Unit (RRCDU), for use in controlling the shaft and trunnion angles of the RR or for positioning the forward and lateral velocity meters. Information is gated out of the cells if bits 11 and 12 respectively of channel 14 are set, and the RRCDU Error Counters recognize the information if bit 2 of channel 12 is set. (Error Counters reset to zero whenever bit 2 of channel 12 is reset.) A saturated Error Counter (384 pulses) corresponds to an RR drive rate of 10 degrees per second; each bit represents 0.1698 meters per second (0.5571 feet per second) when used to drive the velocity meters.

CDUTEMP: Single precision vector storage for the reading of the ICDU at the time of a PIPA read, scaled B-1 in units of revolutions and stored in twos complement form.

COSIGA, COSMGA, COSOGA: See COOR section.

DALTRATE: Single precision expected rate of change of ALTRATE, scaled B0 in units of Analog-altitude-rate-display bits per centisecond.

DAPBOOLS: See DAPA section.

DELQFIX: Double precision Landing Radar Data reasonableness test parameter, scaled B24 in units of meters; part of the erasable load.

DELTAH: Double precision difference between the calculated altitude and that measured by the Landing Radar, scaled B24 in units of meters.

DELTAV: Double precision difference between the calculated velocity component and that measured by the Landing Radar, scaled B6 in units of meters per centisecond.

DELV: Double precision sensed-change-in-velocity vector, scaled B14 in units of centimeters per second (one PIPA pulse represents one centimeter per second on the LM) and expressed in Platform coordinates.

DELVREF: Double precision sensed-change-in-velocity vector converted to a scaling of B7 in units of meters per centisecond and expressed in the Reference coordinate system.

DELVS: Double precision vector difference between velocity relative to the rotating moon and inertial velocity, scaled B5 in units of meters per centisecond and expressed in the Platform coordinate system.

$DSPTAB_{11}$: See INTR section.

DT: Single precision time interval from beginning of navigation interval to the time of the generation of the display on the tape-drive meters, scaled B14 in units of centiseconds.

DVCNTR: Single precision counter set to determine the length of the thrust monitor, scaled B14 in units of navigation cycles.

DVTHRUSH: Single precision delta-v threshold, scaled B14 in units of centimeters per second; set according to the engine in use.

DVTOTAL: Double precision sum of velocity gained, scaled B7 in units of meters per centisecond.

FORVEL: Single precision forward velocity component (Body coordinates) of the LM relative to the rotating moon, scaled B14 in forward velocity display units.

FORVMETR: Single precision storage for the total value of velocity displayed on the Forward velocity meter, scaled B14 in forward velocity display units (see definition of CDUSCMD)

GCSM: Double precision gravity vector at the CSM, scaled B8 in units of meters per centisecond and expressed in the Platform coordinate system.

GDT, GDT1: Double precision gravity vector , scaled B8 in units of meters per centisecond and expressed in the Platform coordinate system.

HBEAMNB: Double precision unit vector in the direction of the Landing Radar measurement of altitude, scaled B1 and expressed in the Body coordinate system.

HCALC, HCALC1: Double precision calculated altitude above the landing site radius, scaled B24 in units of meters. HCALC1 is for display purposes.

HDOTDISP: Double precision calculated value of altitude rate, scaled B7 in units of meters per centisecond.

HLROFF: Double precision quantity representing the altitude at which LR altitude updates are inhibited during the final phase of descent, scaled B24 in units of meters.

HMEAS: Double precision Landing Radar measurement of altitude, scaled B28 in units of Landing Radar low scale altitude bits.

IMODES33: See INTR section.

K:1SEC: Single precision constant stored as 100 x 2^{-14}, scaled B14 in units of centiseconds. Equation value: 100.

K:2J: Double precision constant stored as 3.24692010 E-3, scaled B0 and unitless. Equation value: 3 x 0.0010823067.

K:20J: Double precision constant stored as 3.24692010 E-2, scaled B0 and unitless. Equation value: Ten times K:2J.

K:2SECb28: Double precision constant stored as 200 x 2^{-28}, program notation 2SEC(28), scaled B28 in units of centiseconds. Equation value: 200.

K:30kft: Double precision constant stored as 1.6768072 E7 x 2^{-24}, program notation "1-30KFT", scaled B24 in units of meters. Represents 2^{24} - 9144 meters (K:posmaxdp plus one least increment minus 9144 meters). Used to check current altitude against 9144 meters. Equation value: 9144 (Equivalent to 30,000 feet).

K:6KFTdSEC: Single precision constant stored as 18.288 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 18.288. (Equivalent to 6000 feet per second.)

K:ALTCONV: Double precision constant stored as 1.399078846 x 2^{-4}, scaled B4 in units of Analog-altitude-display bits per meter. Equation value: 1.399078846. (Equivalent to 0.714756 meters, or 2.345 feet, per bit.)

K:altlim: Single precision constant value of bit 16 in a quantity scaled B14 in units of Analog-altitude-display bits. Equation value: 32768.

K:APSVEX: Single precision constant stored as -3030. E-2 x 2^{-5}, scaled B5 in units of meters per centisecond. Equation value: -30.30.

K:ARCONV: Single precision constant stored as 24402_8, scaled B10 in units of Analog-altitude-rate-display bits / meter per centisecond. Equation value: 656.125. (Equivalent to 0.1524 meters per second, or 0.5 feet per second, per bit.)

K:ARCONV1: Double precision constant stored as 656.167979 x 2^{-10}, scaled B10 in units of Analog-altitude-rate-display bits / meter per centisecond. Equation value: 656.167979. (Equivalent to 0.1524 meters per second, or 0.5 feet per second, per bit.)

K:ARTOA: Single precision constant stored as 0.1066098 x 2^{-1}, scaled B1 in units of seconds times Altitude bits / Altitude rate bits. Equation value: 0.1066098. (Equivalent to 0.5 seconds x 0.5 / 2.345.)

K:ARTOA2: Single precision constant stored as 0.0021322 x 2^{8}, scaled B-8in units of Altitude bits per centisecond / Altitude rate bits per second. Equation value: 0.0021322. (Equivalent to 0.01 x 0.5 / 2.345.)

K:DPSVEX: Single precision constant stored as - 29.5588868 x 2^{-5}, scaled scaled B5 in units of meters per centisecond. Equation value: - 29.5588868

K:HBEAMANT: Double precision vector constant stored as (-0.4687018041, 0, -0.1741224271), scaled B1 and unitless. Equation value: (-0.9374036082, 0, -0.3482448542). (Altitude beam direction expressed in the LR coordinate system.)

K:HSCAL: Double precision constant stored as -0.3288792, scaled B0 in units of meters per bit. Equation value: -0.3288792. (Equivalent to 1.0790 feet per bit.)

K:KPIP: Single precision constant stored as 0.0512, scaled B-9 in units of meters per centisecond / centimeters per second. Equation value: 0.0001.

K:KPIP1: Double precision constant stored as 0.0128, scaled B-7 in units of meters per centisecond / centimeters per second. Equation value: 0.0001.

K:KPIP1b5: Double precision constant stored as 0.0512, scaled B-9 in units of meters per centisecond / centimeters per second. Equation value: 0.0001.

K:KPIP2: Double precision constant stored as 0.0064, scaled B-6 in units of meters per centisecond / centimeters per second. Equation value: 0.0001.

K:MAXVBITS: Single precision constant stored as 00547_8, scaled B14 in forward/lateral velocity display units. Equation value: 359. (Equivalent to 61.0 meters per second or 200.0 feet per second.)

K:mMUDT$_0$: Double precision constant stored as -7.9720645 E12 x 2^{-44}, scaled B44 in units of meters cubed per centisecond. Equation value: -7.9720645 E12. (Equivalent to -200 cs x 0.3986032 E11 m^3/cs^2.)

K:mMUDT$_2$: Double precision constant stored as -9.8055560 E10 x 2^{-44}, scaled B44 in units of meters cubed per centisecond. (Also called -MUDTMUN with a scale factor of B38.) Equation value: -9.8055560 E10. (Equivalent to -200 cs x 0.4902778 E9 meters cubed per centisecond squared.)

K:PRIO31: Single precision constant stored as 31000_8, scaled B8 in units of centiseconds. Equation value: 200.

K:RESQ: Double precision constant stored as 40.6809913 E12 x 2^{-58}, scaled B58 in units of meters squared. Equation value: 6,378,165 squared.

K:UNITX, K:UNITY, K:UNITZ: Three double precision vector constants stored as (0.5, 0, 0), (0, 0.5, 0) and (0, 0, 0.5), scaled B1 and unitless. Equation values: (1, 0, 0), (0, 1, 0) and (0, 0, 1).

K:VELCONV: Single precision constant stored as 22316_8, scaled B10 in forward/lateral velocity display units / meter per centisecond. Equation value: 588.875. (Equivalent to 0.1698 meters per second, or 0.5571 feet per second, per bit.)

K:VSCAL$_0$: Double precision constant stored as 0.5410829105, program notation VZSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: 0.002642006 / 5. (Equivalent to 0.8668 fps per bit; the "5" averages the sum of five samples.)

K:VSCAL$_2$: Double precision constant stored as 0.7565672446, program notation VYSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: 0.003694176 / 5. (Equivalent to 1.212 fps per bit; the "5" averages the sum of five samples.)

K:VSCAL$_4$: Double precision constant stored as -0.4020043770, program notation VXSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: -0.001962912 / 5. (Equivalent to -0.644 fps per bit; the "5" averages the sum of five samples.)

LADQSAVE: Single precision octal return address storage.

LANDMAG: see DESC section.

LATVEL: Single precision lateral velocity component (Body coordinates; positive to the right when looking forward) of the LM relative to the rotating moon, scaled B14 in forward/lateral velocity units.

LATVMETR: Single precision storage for the total value of velocity displayed on the lateral velocity meter, scaled B14 in forward/lateral velocity display units.

$LRALPHA_1$, $LRALPHA_2$: Single precision angle from the Z spacecraft axis to the Z LR coordinate axis measured in a right hand rotation around the -X spacecraft axis, for LR positions 1 and 2 respectively, scaled B-1 in units of revolutions and stored in twos complement form. Part of the erasable load.

$LRBETA_1$, $LRBETA_2$: Single precision angle from the +X spacecraft axis to the +X LR coordinate axis measured in a right hand rotation around the -Z LR coordinate axis, for LR positions 1 and 2 respectively, scaled B-1 in units of revolutions and stored in twos complement form. Part of the erasable load.

LRCDU, LRCDUDL: Single precision vector storage for the value of the three ICDU angles at the estimated midpoint of an LR velocity reading, scaled B-1 in units of revolutions and stored in twos complement form. LRCDUDL is for downlink purposes.

LRHMAX: Single precision maximum limit for altitude calculations that are allowed to be updated by the Landing Radar measurement, scaled B14 in units of meters. Part of the erasable load.

LRLCTR: Single precision count of the number of comparisons made between HMEAS and HCALC, scaled B14 and unitless.

LRMCTR: Single precision count of the number of comparisons made between measured velocity and calculated velocity, scaled B14 and unitless.

LRRCTR: Single precision counter used in conjunction with LRLCTR to determine if at least four good comparisons between HMEAS and HCALC have been made since the last unreasonable one, scaled B14 and unitless.

LRSCTR: Single precision counter used in conjunction with LRMCTR to determine if at least four good comparisons between measured velocity and calculated velocity have been made since the last unreasonable one, scaled B14 and unitless.

LRVF: Single precision erasable memory constant representing the velocity at which the velocity update coefficients are changed, scaled B7 in units of meters per centisecond. Part of the erasable load.

LRVMAX: Single precision maximum limit for velocity calculations that are allowed to be updated by the LR measurement, scaled B7 in units of meters per centisecond. Part of the erasable load.

LRVTIME, LRVTIMDL: Double precision time at the estimated midpoint of the LR velocity sample, scaled B28 in units of centiseconds. LRVTIMDL is for downlink purposes.

LRWH: Single precision weighting factor for the incorporation of LR altitude measurements into the LM state vector, scaled B0 and unitless. Part of the erasable load.

$LRWV_i$ (i = 0,1,2): Single precision weighting factors for LR Z, Y and X axis velocity updates, scaled B0 and unitless. Part of the erasable load.

$LRWVF_i$ (i = 0,1,2): Single precision weighting factors for LR Z, Y and X axis velocity updates, scaled B0 and unitless. Part of the erasable load.

LRWVFF: Single precision weighting factor for LR velocity updates for P65 and P66 , scaled B0 and unitless. Part of the erasable load.

M22, M32: See DAPA section.

MASS, MASS1: Double precision mass of the vehicle, scaled B16 in units of kilograms. Loaded by the astronaut (routine 03) and updated during average-g navigation.

MKTIME: Double precision time of PIPA readings which are associated with the Landing Radar altitude measurement for downlink purposes, scaled B28 in units of centiseconds.

MODREG: See DATA section.

[NBSMMAT]: See COOR section.

PGUIDE: Double precision length of the navigation-guidance period, scaled B28 in units of centiseconds.

PIPA: Single precision sensed-change-in-velocity vector, scaled B14 in units of centimeters per second and expressed in the Platform coordinate system. The three components are incremented directly from the Pulse-Integrating, Pendulous Accelerometers on the stable member of the Inertial Measurement Unit.

PIPATMP: Single precision vector storage for the current PIPA reading for use by the analog display routines, reset to zero after the current reading is incorporated into the navigation state vector; scaled B14 in units of centimeters per second and expressed in the Platform coordinate system.

PIPCTR: Single precision counter scaled B14 and unitless; used to determine time elapsed from the beginning of the navigation cycle in "R10,R11" and routine 29.

PIPCTR1: Single precision temporary storage for PIPCTR.

PIPTEM: Single precision vector storage for the sensed change in velocity between the beginning of the navigation cycle and the mean time of the LR velocity sample, scaled B14 in units of centimeters per second and expressed in the Platform coordinate system.

PIPTIME: Double precision time of the most recent PIPA read cycle, scaled B28 in units of centiseconds; time at which the state vector is valid.

PIPTIME1: Temporary storage for PIPTIME to avoid changing the downlink state vector until it is updated homogeneously.

R: see DESC section.

R1S: Temporary storage for R to avoid changing the state vector on the downlink until it is updated homogeneously, scaled B24 in units of meters and expressed in the Platform coordinate system.

RADGOOD, RADMODES: See RADR section.

RADSKAL: Double precision erasable memory quantity representing the LR scale information for high scale radar output, scaled B21 in units of low-scale altitude bits per meter per centisecond; part of erasable load.

RCSM: Double precision position vector of the CSM measured from the center of the earth or moon, program notations R-OTHER and R(CSM), scaled B29 or B24 (descent guidance) in units of meters and expressed in the Reference or the Platform (descent) coordinate system.

[REFSMMAT]: see COOR section.

RMAGSQ: Double precision square of the magnitude of the position vector, scaled B58 (CALCGRAV) or B48 (MUNGRAV) in units of meters squared.

RN: Double precision vector position of the LM measured from the center of the earth or moon, scaled B29 in units of meters and expressed in the Reference coordinate system.

RN1: Temporary storage for RN to avoid changing the state vector on the downlink until it is updated homogeneously.

RPCRTQSW: Double precision required X component of the X-body axis in Platform coordinates at the time of LR reposition to position 2, scaled B1 and unitless; part of the erasable load and may be altered by V59.

RPCRTIME: Single precision value of TTF at which the LR may be repositioned to position 2, scaled B17 in units of centiseconds; part of the erasable load and may be altered by V59.

RTX1, RTX2: See ORBI section.

RUNIT: Single precision unit vector along the position vector of the LM with respect to the center of the moon, scaled B1, unitless, and expressed in the Platform coordinate system.

SAMPLSUM: See RADR section.

SINIGA, SINMGA, SINOGA: See COOR section.

SKALSKAL: Single precision erasable memory factor by which the correction to the LR data is reduced if the LR is on low range scale, scaled B0 and unitless; part of the erasable load.

[SMNBMAT]: see COOR section.

STILBADH, STILBADV: Single precision counters, scaled B14 and unitless.

TEM: Single precision storage for -PIPA, scaled B14 in units of centimeters per second. TEM is used in the R.O.D. computations of the DESC section.

TIME5: see DAPA section.

TIMENOW: see EXVB section.

TRAKFWDV: Single precision flag set to 1, 0 or - 1 to indicate whether the previously computed value of FORVEL was limited to K:MAXVBITS or not, scaled B14 and unitless.

TRAKLATV: Single precision flag set to 1, 0 or - 1 to indicate whether the previously computed value of LATVEL exceeded K:MAXVBITS or not, scaled B14 and unitless.

TTF: see DESC section.

UHYP: Double precision unit vector normal to the CSM orbital plane, scaled B1 and unitless.

UHZP: Double precision unit local vertical vector in the forward direction, scaled B1 and unitless.

UNITGOBL: See BURN section.

UNITR: Double precision unit vector along the vector from the center of the moon or the earth to the LM, program notation UNIT/R/, scaled B1, unitless and expressed in the Platform or Reference coordinate system.

V: See DESC section.

V1S: Temporary storage for V to avoid changing state vector on the downlink until it is updated homogeneously, scaled B7 in units of meters per centisecond and expressed in the Platform coordinate system.

VBEAM: Double precision unit vector along one of the three Landing Radar velocity measurement directions, scaled B1 and expressed in Platform coordinates at LRVTIME.

VCSM: Double precision inertial velocity vector of the CSM, program notations V-OTHER and V(CSM), scaled B7 in units of meters per centisecond and expressed in the Reference or the Platform (for Descent) coordinate system.

VELBIAS: Double precision erasable constant representing the Landing Radar velocity reasonability test limit, scaled B6 in units of meters per centisecond; part of the erasable load.

VEST: Double precision projection of calculated velocity onto the particular LR velocity component direction being processed, scaled B6 in units of meters per centisecond.

VHY, VHZ: Single precision lateral and forward components of velocity relative to the rotating moon expressed in the Platform coordinate system (lateral velocity positive to the right when looking forward); scaled B5 in units of meters per centisecond.

VMEAS: Double precision velocity measurement from the LR sampling, scaled B28 in units of Landing Radar velocity bits.

VN: Double precision inertial velocity vector of the LM, scaled B7 in units of meters per centisecond and expressed in the Reference coordinate system.

VN1: Temporary storage for VN to avoid changing the state vector on the downlink until it is updated homogeneously.

VSELECT: Single precision index used to distinguish among the Z (0), Y (1) and X (2) axes of the Landing Radar coordinate system, scaled B14 and unitless.

VVECT: Single precision velocity vector used in the calculation of forward, lateral and vertical velocity components of the analog display, scaled B5 in units of meters per centisecond and expressed in the Platform coordinate system.

VXBEAMNB, VYBEAMNB, VZBEAMNB: Double precision unit vectors along the X, Y and Z Landing Radar velocity measurement directions (orthogonal), scaled B1 and expressed in the Body coordinate system.

WM: see DESC section.

XNBPIP, YNBPIP, ZNBPIP: Double precision unit vectors along the X, Y and Z spacecraft axes, scaled B1 and expressed in the Platform coordinate system at PIPTIME.

[XNBPIP] : Double precision matrix with the first row equal to the components of XNBPIP, the second row equal to the components of YNBPIP, and the third row equal to the components of ZNBPIP, scaled B1 and unitless.

TELE

Up and Down Telemetry

UPRUPT SAMPTIME = TIMENOW (for noun 65)

TS = INLINK

INLINK = 00000_8

Switch bit 3 of channel 11 to 1 (uplink activity lamp)

TS should be of the form xxxxx kkkkk $xxxxx_2$ where five x's represent the five bit keycode and five k's represent the complement of that keycode. If TS is not of this form:

    Switch FLAGWRD7 bit 4 (UPLOCKFL) to 1

    Resume

TScode = low 5 bits of TS

If TScode = 22_8: (error reset)

    Switch FLAGWRD7 bit 4 (UPLOCKFL) to 0

If TScode ≠ 22_8:

    If FLAGWRD7 bit 4 (UPLOCKFL) = 1, resume

Establish "CHARIN" (pr30)

Set $MPAC_0$ of "CHARIN" job = TScode

Resume

UPTMFAST If bit 11 of IMODES33 = 0:

    Perform "ALARM" with TS = 01106_8

Return

DNTMFAST If bit 12 of IMODES33 = 0:

    Perform "ALARM" with TS = 01105_8

Return

DODOWNTM If bit 7 of channel 13 = 1: (word order code)

    Proceed to address specified in DNTMGOTO

Perform "C13STALL"

Set bit 7 of channel 13 to 1

Proceed to address specified in DNTMGOTO

<u>DNPHASE1</u> SUBLIST = -1

DNECADR = -1

DNTMGOTO = "DNPHASE2"

CTLIST = K:DNTABLE$_{DNLSTCOD}$

Perform "WOZERO"

Channel 34 = -DNLSTCOD

Channel 35 = K:LOWIDCOD

Resume

<u>DNPHASE2</u> If DNECADR and SUBLIST are both <0: (control list)

    If CTLIST ≤ 0:

        Proceed to 4th line of "DNPHASE1"

    ADR = E$_{CTLIST}$

    If ADR > 0, CTLIST = CTLIST + 1

    If ADR < 0: (end of list)

        CTLIST = -CTLIST

        ADR = -ADR

    DNECADR = ADR

    If DNECADR = K:timeadr, perform "WOZERO"

    DOWNTYPE = bits 14-12 of DNECADR

    If DOWNTYPE < 6, proceed to "FETCH2WD"

    If DOWNTYPE = 6, proceed to "DODNPTR"

    Proceed to "DODNCHAN"

If DNECADR > 0, proceed to "FETCH2WD"

Proceed to "NEXTINSL"

<u>DODNCHAN</u> i = low 8 bits of DNECADR

$j = i + 1$

DNECADR = -1

Channel 34 = channel i

Channel 35 = channel j

Resume

<u>FETCH2WD</u> EBANK = bits 11-9 of DNECADR

TS = low 8 bits of DNECADR

DOWNTYPE = DOWNTYPE - 1

DNECADR = 2^{11} DOWNTYPE + 2^{8} EBANK + TS + 2
(putting DOWNTYPE into bits 14-12, EBANK into bits 11-9; making DNECADR negative after DOWNTYPE is reduced below zero)

ADR1 = TS + 1400_8

ADR2 = TS + 1401_8

Channel 34 = E_{ADR1}

Channel 35 = E_{ADR2}

Resume

<u>DODNPTR</u> SUBADR = $E_{DNECADR} - 30000_8$

If SUBADR < 0: (snapshot)

SUBLIST = DNECADR (address of sublist)

$i = 0$

SUBADR = $|SUBADR| - 00001_8$

Proceed to "SNAPLOOP"

SUBLIST = DNECADR

<u>NEXTINSL</u> SUBADR = $E_{SUBLIST} - 30000_8$

If SUBADR > 0, SUBLIST = SUBLIST + 1

If SUBADR < 0, SUBLIST = -1

DNECADR = |SUBADR|

DOWNTYPE = bits 14-12 of DNECADR

If DOWNTYPE < 6, proceed to "FETCH2WD"

If DOWNTYPE = 6, proceed to "DODNPTR"

Proceed to "DODNCHAN"

SNAPLOOP EBANK = bits 11-9 of SUBADR

ADR1 = 1401_8 + low 8 bits of SUBADR (1401 and 1402 to compensate for the 5th step of "DODNPTR" and the 11th step of "SNAPLOOP")

ADR2 = 1402_8 + low 8 bits of SUBADR

$DNTMBUFF_i = E_{ADR1}$

j = i + 1

$DNTMBUFF_j = E_{ADR2}$

i = i + 2

SUBLIST = SUBLIST + 1

SUBADR = $E_{SUBLIST} - 30000_8$

If SUBADR > 0: (continue snapshot)

SUBADR = |SUBADR| - 1

Proceed to "SNAPLOOP"

SUBLIST = |SUBADR| - 1

DNECADR = -1

SUBADR = SUBLIST

SUBLIST = -1

EBANK = bits 11-9 of SUBADR

ADR1 = 1401_8 + low 8 bits of SUBADR

ADR2 = 1402_8 + low 8 bits of SUBADR

Channel 34 = E_{ADR1}

Channel 35 = E_{ADR2}

Resume

<u>DNDUMPI</u> DUMPLOC = 00000_8

DNTMGOTO = "DNDUMPI3"

Perform "WOZERO"

Channel 34 = K:ERASID

Channel 35 = K:LOWIDCOD

Resume

<u>DNDUMPI3</u> DNTMGOTO = "DNDUMP1"

Channel 34 = DUMPLOC

Channel 35 = least significant half of TIMENOW

Resume

<u>DNDUMP1</u> DNTMGOTO = "DNDUMP"

Proceed to "DNDUMP2"

<u>DNDUMP</u> DUMPLOC = DUMPLOC + 2

TS = low 8 bits of DUMPLOC

If TS > 0, proceed to "DNDUMP2"

(Otherwise, TS = 0 and dump is changing banks)

If bit 13 of DUMPLOC = 0:

    Proceed to second line of "DNDUMPI"

Proceed to "DNPHASE1"

<u>DNDUMP2</u> EBANK = bits 11-9 of DUMPLOC

TS = low 8 bits of DUMPLOC

ADR2 = 1401_8 + TS

ADR1 = 1400_8 + TS

Channel 34 = E_{ADR1}

Channel 35 = E_{ADR2}

Resume

SVDWN1 R-OTHER = RCV + TDELTAV

V-OTHER = VCV + TNUV

Return

SVDWN2 If FLAGWRD9 bit 1 (AVEMIDSW) = 0:

RN = RCV + TDELTAV

VN = VCV + TNUV

PIPTIME = TET

Return

WOZERO Perform "C13STALL"

Switch bit 7 of channel 13 to 0 (word order code)

Return

ADR: Single precision temporary storage for the address taken from the control list. If it is negative this indicates the end of a downlist.

ADR1, ADR2: Single precision addresses (without EBANK information) of the two consecutive registers to be transmitted on the downlink.

CTLIST: Single precision address of the next entry in the downlink control list. When the final downlist quantity is read, CTLIST is complemented to cause downlist to be started again.

DNECADR: Single precision octal address of the first of two consecutive registers to be transmitted on the downlink (in bits 8-1). Bits 14-12 contain a code indicating the type of sample to be taken (see DOWNTYPE). Bit 15 is set (DNECADR made negative) to indicate that control is to be returned to the control list.

DNLSTCOD: Single precision index (range 0-5) indicating which of the downlists is to be telemetered, scaled B14 and unitless. Loaded by various programs to select the proper downlist. See K:DNTABLE.

DNTMBUFF: A series of single precision buffer cells used to store a simultaneous "snapshot" of a series of E-memory cells all sampled at the same downlink interrupt, thereby making the data time homogeneous.

DNTMGOTO: Single precision octal address controlling the phase of operation of the downlink program.

DOWNTYPE: Variable describing bits 14-12 of DNECADR in the normal downlink mode. If DOWNTYPE = 7, the address in DNECADR is interpreted as that of a channel. If DOWNTYPE is less than 6, the address in DNECADR is interpreted as that of a series of N consecutive registers (N = 2 (DOWNTYPE + 1)). If DOWNTYPE = 6, the address in DNECADR is interpreted as that of a sub-list and control is transferred to the sub-list decoder ("DODNPTR").

DUMPLOC: Single precision E-memory register which contains the counter and ECADR for each dump-word being sent. Bits 8-1 provide the relative address within the EBANK; bits 11-9 define the EBANK (propagated from bit 8); bits 13-12 serve as a counter of the number of complete dumps which have occurred. When bit 13 becomes 1 then memory has been dumped twice and the dumping stops.

E_{ADR}: Single precision register whose address is in ADR.

EBANK: See MATX section.

IMODES33: See INTR section.

INLINK: Single precision serial input register for receipt of uplink data. When the required sixteen bits of data are received from the uplink decoder, program interrupt #7 is generated.

K:DNTABLE$_i$ (i = 0-5): Table of six octal starting addresses of the downlink lists.

i	List address
0	"LMCSTADL"
1	"LMAGSIDL"
2	"LMRENDDL"
3	"LMORBMDL"
4	"LMDSASDL"
5	"LMLSALDL"

K:ERASID: Single precision octal constant stored as 01776_8. Used as E-memory octal dump downlist I.D. word, loaded in downlink word 1a.

K:LOWIDCOD: Single precision octal constant stored as 77340_8. Loaded into downlist word 1b of each of the six downlink lists. Sometimes referred to as the "sync" bits.

K:timeadr: Single precision constant stored as 77753_8, program notation MINTIME2. Equation value: 00024_8, address of TIMENOW.

$MPAC_0$: See DSKY section.

PIPTIME: See SERV section.

$\underline{R}$-OTHER, $\underline{V}$-OTHER: Double precision navigation state vectors of the CSM, scaled B29 and B7 respectively in units of meters and meters/centiseconds. See $\underline{R}$CSM, $\underline{V}$CSM in SERV section.

$\underline{R}$CV, $\underline{V}$CV: See CONC section.

$\underline{R}$N: See SERV section.

SAMPTIME: See DSKY section.

SUBADR: Single precision address code word like DNECADR but taken from a sub-list.

SUBLIST: Single precision address of the next entry in a downlink sub-list.

TDELTAV, TNUV: See ORBI section.

TET: See ORBI section.

TIMENOW: See EXVB section.

VN: See SERV section.

General Downlink Information

The LM Guidance Computer (LGC) downlink takes the form of 40-bit words transmitted as a basic rate of 50 words/second (part of a telemetry stream at 51.2 kbps). A "low bit rate" of one-fifth of this transmission rate also exists, although no computer words are included in low bit rate data. Each 40-bit word is divided into four parts:

a) The first bit is the word order code bit, set zero for the first word pair and the fifty-first word pair and one for the other 98 pairs in the standard telemetry cycle.

b) Bits #2 - #17 contain the first word of the word pair (bit 15 is in bit #2, bit 14 in bit #3, ... bit 1 in bit #16, and an odd parity bit in bit #17). The odd parity bit makes the total number of binary "ones" in bits #2 - #17 an odd number.

c) Bits #18 - #33 contain the second word of the word pair (bit 15 is in bit #18, bit 14 in bit #19, ... bit 1 in bit #32, and an odd parity bit for bits #18 - #33 in bit #33).

d) Bits #34 - #40 are the same as bits #2 - #8 (bits 15-9 of first word of the word pair), so that the total number of bits in the digital downlink from the computer is a multiple of 8 bits (i.e., 5 x 8 = 40).

The computer hardware monitors the period of the telemetry interrupts received from the telemetry system, and rejects the interrupt (takes no action) if the interrupts occur too rapidly. The mechanization requires a computer 100 pps pulse to occur between each accepted telemetry interrupt (which, under normal conditions, only occur once every 20 ms, or at a 50 pps rate). A channel bit (channel 33 bit 12) is set to a binary zero if an interrupt is rejected, and an alarm pattern (1105_8) is also produced.

The convention is established in the program that "bit 15" is the sign bit (a binary one if quantity negative) and "bit 1" is the least significant magnitude bit. Using "a" for the first word of a pair and "b" for the second word, the bit stream would appear as follows:

WOC 15a 14a 13a 12a 11a 10a 9a 8a 7a 6a 5a 4a 3a 2a 1a Parity_a

15b 14b 13b 12b 11b 10b 9b 8b 7b 6b 5b 4b 3b 2b 1b Parity_b

15a 14a 13a 12a 11a 10a 9a

(WOC is the "word order code", discussed in item (a) above.)

Most telemetered words have negative numbers expressed in ones complement form, and in general the signs of the most significant and least significant portions will not agree, since the individual portions of a multiple precision quantity are considered generally as separate "words" in the computer arithmetic unit. Several words in the downlist are used as control quantities in the program and have explicit meanings assigned to their individual bits. These words have their sign bits set separately from the rest of the word, and hence are not subject to the same conversion process as other "negative" quantities.

Scaling, units, and definition references for each parameter are contained in the tables.

The downlink program has the capability to sample a selected set of erasable locations essentially at the same time with the contents of these locations stored in a set of unshared erasable to be downlinked when individual telemetry interrupts are received. By this means a time-homogeneous set of downlink information can be obtained provided the program loading the cells involved observes proper restrictions. These selected buffered areas are commonly called "snapshots". The LUMINARY downlists have the following snapshots:

Coast and Align List	Words 2-8, 52-58
AGS Initialization and Update List	Words 52-58
Rendezvous and Prethrust List	Words 2-8, 9-13, 52-58
Orbital Maneuvers List	Words 2-8, 52-58
Descent and Ascent List	Words 2-13, 52-58
Lunar Surface Align List	Words 2-8, 9-13, 52-58

Downlink Sequences

There are six different 100-word lists which can be sent by the program during the flight (plus the special erasable memory dump). For convenience, these lists have been assigned serial numbers, #0 through #5 (listed in order of increasing identification words). During a given program, a certain list is transmitted as defined below:

List #0 The Coast and Align List is transmitted during:

P00 LGC Idling

P51 IMU Orientation Determination

P52 IMU Realignment

P06 LGC Power Down

List #1 The AGS Initialization and Update List is transmitted during:

P27 LGC Update

R47 AGS Initialization

List #2 The Rendezvous and Prethrust List is transmitted during:

P20 Rendezvous Navigation

P21 Ground Track Determination

P25 Preferred Tracking Attitude

P30 External Delta V Maneuver Guidance

P32 Coelliptic Sequence Initiation (CSI)

P33 Constant Differential Altitude (CDH)

P34 Transfer Phase Initiation (TPI)

P35 Transfer Phase Midcourse (TPM)

P72 CSM CSI Targeting

P73 CSM CDH Targeting

P74 CSM TPI Targeting

P75 CSM TPM Targeting

P76 Target DELTA V

List #3 The Orbital Maneuvers List is transmitted during:

P40 DPS Thrust

P41 RCS Thrust

P42 APS Thrust

P47 Thrust Monitor

List #4 The Descent and Ascent List is transmitted during:

P12 Powered Ascent Guidance

P63 Braking Phase Guidance

P64 Approach Phase Guidance

P66 Rate of Descent (ROD) Landing Phase Guidance

P68 Confirm Lunar Landing

P70 DPS Abort Guidance

P71 APS Abort Guidance

List #5 The Lunar Surface Align List is transmitted during:

P22 RR Lunar Surface Navigation

P57 Lunar Surface Alignment

Several cells are identified as "Spare": by program assembly technique, the word on the downlink has the first half as 00000_8 and the second half as a quantity of negligible usefulness for post-flight processing (the contents of the computer accumulator register, cell 0000, when the telemetry interrupt was recognized).

Words marked with an asterisk in the list were transmitted as a result of a double precision pickup in the downlink program and have no known importance.

COAST and IMU ALIGNMENT LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77777_8)	SYNC PATTERN (77340_8)	
2	R-OTHER$_0$ (X comp.)	R-OTHER$_1$ (X comp.)	
3	R-OTHER$_2$ (Y comp.)	R-OTHER$_3$ (Y comp.)	1717 - 1724
4	R-OTHER$_4$ (Z comp.)	R-OTHER$_5$ (Z comp.)	
5	V-OTHER$_0$ (X comp.)	V-OTHER$_1$ (X comp.)	
6	V-OTHER$_2$ (Y comp.)	V-OTHER$_3$ (Y comp.)	1725 - 1732
7	V-OTHER$_4$ (Z comp.)	V-OTHER$_5$ (Z comp.)	
8	TETCSM (T-OTHER)	TETCSM (T-OTHER)	1570-1
9	AGSK (K-FACTOR)	AGSK (K-FACTOR)	2020-1
10	TALIGN	TALIGN	2774-5
11	DOWNTORK$_2$ (POSTORKU)	DOWNTORK$_3$ (NEGTORKU)	3115, 3116
12	DOWNTORK$_4$ (POSTORKV)	DOWNTORK$_5$ (NEGTORKV)	3117, 3120
13	DNRADATA$_1$ (DNRRANGE)	DNRADATA$_2$ (DNRRDOT)	1330, 1331
14	TEVENT	TEVENT	1341-2
15	REFSMMAT$_0$ (R_1C_1)	REFSMMAT$_1$ (R_1C_1)	
16	REFSMMAT$_2$ (R_1C_2)	REFSMMAT$_3$ (R_1C_2)	
17	REFSMMAT$_4$ (R_1C_3)	REFSMMAT$_5$ (R_1C_3)	1733 - 1746
18	REFSMMAT$_6$ (R_2C_1)	REFSMMAT$_7$ (R_2C_1)	
19	REFSMMAT$_8$ (R_2C_2)	REFSMMAT$_9$ (R_2C_2)	
20	REFSMMAT$_{10}$ (R_2C_3)	REFSMMAT$_{11}$ (R_2C_3)	
21	AOTCODE	*SINIGA	0734, 0735
22	RLS$_0$ (X-comp)	RLS$_1$ (X-comp)	
23	RLS$_2$ (Y-comp)	RLS$_3$ (Y-comp)	2022 - 2027
24	RLS$_4$ (Z-comp)	RLS$_5$ (Z-comp)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
25	DNRADATA$_4$ (DNLRVELX)	DNRADATA$_5$ (DNLRVELY)	1333, 1334
26	DNRADATA$_6$ (DNLRVELZ)	DNRADATA$_7$ (DNLRALT)	1335, 1336
27	VGPREV$_0$ (VGTIG-X comp)	VGPREV$_1$ (VGTIG-X comp)	
28	VGPREV$_2$ (VGTIG-Y comp)	VGPREV$_3$ (VGTIG-Y comp)	3700 - 3705
29	VGPREV$_4$ (VGTIG-Z comp)	VGPREV$_5$ (VGTIG-Z comp)	
30	REDOCTR	THETAD$_0$ (X-ANGLE)	0320, 0321
31	THETAD$_1$ (Y-ANGLE)	THETAD$_2$ (Z-ANGLE)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	0074 - 0107
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	
45	DSPTAB$_0$	DSPTAB$_1$	
46	DSPTAB$_2$	DSPTAB$_3$	
47	DSPTAB$_4$	DSPTAB$_5$	1022 - 1035
48	DSPTAB$_6$	DSPTAB$_7$	
49	DSPTAB$_8$	DSPTAB$_9$	
50	DSPTAB$_{10}$	DSPTAB$_{11}$	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	
53	RN_2 (Y comp.)	RN_3 (Y comp.)	1217 - 1224
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	
56	VN_2 (Y comp.)	VN_3 (Y comp.)	1225 - 1232
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	*ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	OGC	OGC	2737 - 2740
66	IGC	IGC	2741-2
67	MGC	MGC	2743-4
68	BESTI (STAR ID 1)	BESTJ (STAR ID 2)	2755, 2756
69	$STARSAV1_0$ (X comp.)	$STARSAV1_1$ (X comp.)	
70	$STARSAV1_2$ (Y comp.)	$STARSAV1_3$ (Y comp.)	2760 - 2765
71	$STARSAV1_4$ (Z comp.)	$STARSAV1_5$ (Z comp.)	
72	$STARSAV2_0$ (X comp.)	$STARSAV2_1$ (X comp.)	
73	$STARSAV2_2$ (Y comp.)	$STARSAV2_3$ (Y comp.)	2766 - 2773
74	$STARSAV2_4$ (Z comp.)	$STARSAV2_5$ (Z comp.)	
75	$DNRADATA_4$ (DNLRVELX)	$DNRADATA_5$ (DNLRVELY)	1333, 1334
76	$DNRADATA_6$ (DNLRVELZ)	$DNRADATA_7$ (DNLRALT)	1335, 1336

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
77	CDUS	PIPA (X)	0036, 0037
78	PIPA (Y)	PIPA (Z)	0040, 0041
79	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	TIG	TIG	3441-2
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	*DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	DOWNTORK$_0$ (POSTORKP)	DOWNTORK$_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	
95	DSPTAB$_0$	DSPTAB$_1$	
96	DSPTAB$_2$	DSPTAB$_3$	
97	DSPTAB$_4$	DSPTAB$_5$	1022 - 1035
98	DSPTAB$_6$	DSPTAB$_7$	
99	DSPTAB$_8$	DSPTAB$_9$	
100	DSPTAB$_{10}$	DSPTAB$_{11}$	

AGS INITIALIZATION & UPDATE LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77776_8)	SYNC PATTERN (77340_8)	
2	$AGSBUFF_0$ (LM X POS)	*$AGSBUFF_1$ (not used by AGS)	2200, 2201
3	$AGSBUFF_2$ (LM Y POS)	*$AGSBUFF_3$ (not used by AGS)	2202, 2203
4	$AGSBUFF_4$ (LM Z POS)	*$AGSBUFF_5$ (not used by AGS)	2204, 2205
5	$AGSBUFF_{12}$ (Vector time MSB)	*$AGSBUFF_{13}$ (not used by AGS)	2214, 2215
6	$AGSBUFF_1$ (LM X VEL)	*$AGSBUFF_2$ (not used by AGS)	2201, 2202
7	$AGSUBFF_3$ (LM Y VEL)	*$AGSBUFF_4$ (not used by AGS)	2203, 2204
8	$AGSBUFF_5$ (LM Z VEL)	*$AGSBUFF_6$ (not used by AGS)	2205, 2206
9	$AGSBUFF_{13}$ (Vector time LSB)	*$VONE_2$ (not used by AGS)	2215, 2216
10	$AGSBUFF_6$ (CSM X POS)	*$AGSBUFF_7$ (not used by AGS)	2206, 2207
11	$AGSBUFF_8$ (CSM Y POS)	*$AGSBUFF_9$ (not used by AGS)	2210, 2211
12	$AGSBUFF_{10}$ (CSM Z POS)	*$AGSBUFF_{11}$ (not used by AGS)	2212, 2213
13	$AGSBUFF_{12}$ (Vector time MSB)	*$AGSBUFF_{13}$ (not used by AGS)	2214, 2215
14	$AGSBUFF_7$ (CSM X VEL)	*$AGSBUFF_8$ (not used by AGS)	2207, 2210
15	$AGSBUFF_9$ (CSM Y VEL)	*$AGSBUFF_{10}$ (not used by AGS)	2211, 2212
16	$AGSBUFF_{11}$ (CSM Z VEL)	*$AGSBUFF_{12}$ (not used by AGS)	2213, 2214
17	$AGSBUFF_{13}$ (Vector time LSB)	*$VONE_2$ (not used by AGS)	2215, 2216
18	COMPNUMB	UPOLDMOD	1167, 1170
19	UPVERB	UPCOUNT	1171, 1172

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
20	UPBUFF$_{0}$	UPBUFF$_{1}$	
21	UPBUFF$_{2}$	UPBUFF$_{3}$	
22	UPBUFF$_{4}$	UPBUFF$_{5}$	
23	UPBUFF$_{6}$	UPBUFF$_{7}$	
24	UPBUFF$_{8}$	UPBUFF$_{9}$	1173 - 1216
25	UPBUFF$_{10}$	UPBUFF$_{11}$	
26	UPBUFF$_{12}$	UPBUFF$_{13}$	
27	UPBUFF$_{14}$	UPBUFF$_{15}$	
28	UPBUFF$_{16}$	UPBUFF$_{17}$	
29	UPBUFF$_{18}$	UPBUFF$_{19}$	
30	REDOCTR	THETAD$_{0}$ (X-angle)	0320, 0321
31	THETAD$_{1}$ (Y-angle)	THETAD$_{2}$ (Z-angle)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	0074 - 0107
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
45	$DSPTAB_0$	$DSPTAB_1$	
46	$DSPTAB_2$	$DSPTAB_3$	
47	$DSPTAB_4$	$DSPTAB_5$	1022 - 1035
48	$DSPTAB_6$	$DSPTAB_7$	
49	$DSPTAB_8$	$DSPTAB_9$	
50	$DSPTAB_{10}$	$DSPTAB_{11}$	
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	
53	RN_2 (Y comp.)	RN_3 (Y comp.)	1217 - 1224
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	
56	VN_2 (Y comp.)	VN_3 (Y comp.)	1225 - 1232
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	*ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	$DOWNTORK_2$ (POSTORKU)	$DOWNTORK_3$ (NEGTORKU)	3115, 3116
66	$DOWNTORK_4$ (POSTORKV)	$DOWNTORK_5$ (NEGTORKV)	3117, 3120
67	SPARE	SPARE	
68	SPARE	SPARE	
69	AGSK (K-FACTOR)	AGSK (K-FACTOR)	2020-1

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
70	$UPBUFF_0$	$UPBUFF_1$	
71	$UPBUFF_2$	$UPBUFF_3$	
72	$UPBUFF_4$	$UPBUFF_5$	
73	$UPBUFF_6$	$UPBUFF_7$	
74	$UPBUFF_8$	$UPBUFF_9$	1173 - 1216
75	$UPBUFF_{10}$	$UPBUFF_{11}$	
76	$UPBUFF_{12}$	$UPBUFF_{13}$	
77	$UPBUFF_{14}$	$UPBUFF_{15}$	
78	$UPBUFF_{16}$	$UPBUFF_{17}$	
79	$UPBUFF_{18}$	$UPBUFF_{19}$	
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	SPARE	SPARE	
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	* DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	$DOWNTORK_0$ (POSTORKP)	$DOWNTORK_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
95	$DSPTAB_0$	$DSPTAB_1$	1022 - 1035
96	$DSPTAB_2$	$DSPTAB_3$	
97	$DSPTAB_4$	$DSPTAB_5$	
98	$DSPTAB_6$	$DSPTAB_7$	
99	$DSPTAB_8$	$DSPTAB_9$	
100	$DSPTAB_{10}$	$DSPTAB_{11}$	

RENDEZVOUS & PRETHRUST LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77775$_8$)	SYNC PATTERN (77340$_8$)	
2	R-OTHER$_0$ (X comp.)	R-OTHER$_1$ (X comp.)	
3	R-OTHER$_2$ (Y comp.)	R-OTHER$_3$ (Y comp.)	1717 - 1724
4	R-OTHER$_4$ (Z comp.)	R-OTHER$_5$ (Z comp.)	
5	V-OTHER$_0$ (X comp.)	V-OTHER$_1$ (X comp.)	
6	V-OTHER$_2$ (Y comp.)	V-OTHER$_3$ (Y comp.)	1725 - 1732
7	V-OTHER$_4$ (Z comp.)	V-OTHER$_5$ (Z comp.)	
8	TETCSM (T-OTHER)	TETCSM (T-OTHER)	1570-1
9	RANGRDOT (DNRRANGE)	RANGRDOT+1	3760, 3761
10	AIG	AMG	3457, 3460
11	AOG	TRKMKCNT	3461, 3462
12	TANGNB$_0$ (RR trunnion)	TANGNB$_1$ (RR shaft)	3752, 3753
13	MKTIME	MKTIME	3754-5
14	DELLT4 (T$_F$ Lambert)	DELLT4 (T$_F$ Lambert)	3451-2
15	RTARG$_0$ (X comp.)	RTARG$_1$ (X comp.)	
16	RTARG$_2$ (Y comp.)	RTARG$_3$ (Y comp.)	3443 - 3450
17	RTARG$_4$ (Z comp.)	RTARG$_5$ (Z comp.)	
18	DELVLVC$_0$ (DELVSLV-X)	DELVLVC$_1$ (DELVSLV-X)	
19	DELVLVC$_2$ (DELVSLV-Y)	DELVLVC$_3$ (DELVSLV-Y)	3433 - 3440
20	DELVLVC$_4$ (DELVSLV-Z)	DELVLVC$_5$ (DELVSLV-Z)	
21	TCSI (CSI time)	TCSI (CSI time)	3633-4
22	DELVEET1$_0$ (X comp.)	DELVEET1$_1$ (X comp.)	
23	DELVEET1$_2$ (Y comp.)	DELVEET1$_3$ (Y comp.)	2266 - 2273
24	DELVEET1$_4$ (Z comp.)	DELVEET1$_5$ (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
25	SPARE	SPARE	
26	TPASS4 (TPF time)	TPASS4 (TPF time)	3630-1
27	$X789_x$ (RR shaft bias)	$X789_x$ (RR shaft bias)	1700-1
28	$X789_y$ (RR trunnion bias)	$X789_y$ (RR trunnion bias)	1702-3
29	LASTTCMD(LASTYCMD)	LASTSCMD (LASTXCMD)	0112 - 0113
30	REDOCTR	$THETAD_0$ (X-angle)	0320, 0321
31	$THETAD_1$ (Y-angle)	$THETAD_2$ (Z-angle)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	0074 - 0107
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	
45	$DSPTAB_0$	$DSPTAB_1$	1022 - 1035
46	$DSPTAB_2$	$DSPTAB_3$	
47	$DSPTAB_4$	$DSPTAB_5$	
48	$DSPTAB_6$	$DSPTAB_7$	
49	$DSPTAB_8$	$DSPTAB_9$	
50	$DSPTAB_{10}$	$DSPTAB_{11}$	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	1217 - 1224
53	RN_2 (Y comp.)	RN_3 (Y comp.)	
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	1225 - 1232
56	VN_2 (Y comp.)	VN_3 (Y comp.)	
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	**ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	$DOWNTORK_2$ (POSTORKU)	$DOWNTORK_3$ (NEGTORKU)	3115, 3116
66	$DOWNTORK_4$ (POSTORKV)	$DOWNTORK_5$ (NEGTORKV)	3117, 3120
67	SPARE	SPARE	
68	TCDH	TCDH	1776-7
69	$DELVEET2_0$ (X comp.)	$DELVEET2_1$ (X comp.)	2274 - 2301
70	$DELVEET2_2$ (Y comp.)	$DELVEET2_3$ (Y comp.)	
71	$DELVEET2_4$ (Z comp.)	$DELVEET2_5$ (Z comp.)	
72	TTPI	TTPI	3635-6
73	$DELVEET3_0$	$DELVEET3_1$	2365 - 2372
74	$DELVEET3_2$	$DELVEET3_3$	
75	$DELVEET3_4$	$DELVEET3_5$	
76	ELEV	ELEV	2256-7

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
77	CDUS	PIPA (X)	0036, 0037
78	PIPA (Y)	PIPA (Z)	0040, 0041
79	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	TIG	TIG	3441-2
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	*DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	$DOWNTORK_0$ (POSTORKP)	$DOWNTORK_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	
95	SPARE	SPARE	
96	CENTANG	CENTANG	3620-1
97	NN	NN	3466-7
98	DIFFALT	DIFFALT	3577 - 3600
99	DELVTPF	DELVTPF	2347 - 2350
100	SPARE	SPARE	

ORBITAL MANEUVERS LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77774_8)	SYNC PATTERN (77340_8)	
2	R-OTHER$_0$ (X comp.)	R-OTHER$_1$ (X comp.)	
3	R-OTHER$_2$ (Y comp.)	R-OTHER$_3$ (Y comp.)	1717 - 1724
4	R-OTHER$_4$ (Z comp.)	R-OTHER$_5$ (Z comp.)	
5	V-OTHER$_0$ (X comp.)	V-OTHER$_1$ (X comp.)	
6	V-OTHER$_2$ (Y comp.)	V-OTHER$_3$ (Y comp.)	1725 - 1732
7	V-OTHER$_4$ (Z comp.)	V-OTHER$_5$ (Z comp.)	
8	TETCSM (T-OTHER)	TETCSM (T-OTHER)	1570-1
9	DELLT4	DELLT4	3451-2
10	RTARG$_0$ (X comp.)	RTARG$_1$ (X comp.)	
11	RTARG$_2$ (Y comp.)	RTARG$_3$ (Y comp.)	3443 - 3450
12	RTARG$_4$ (Z comp.)	RTARG$_5$ (Z comp.)	
13	ELEV	ELEV	2256-7
14	TEVENT	TEVENT	1341-2
15	REFSMMAT$_0$ (R_1C_1)	REFSMMAT$_1$ (R_1C_1)	
16	REFSMMAT$_2$ (R_1C_2)	REFSMMAT$_3$ (R_1C_2)	
17	REFSMMAT$_4$ (R_1C_3)	REFSMMAT$_5$ (R_1C_3)	1733 - 1746
18	REFSMMAT$_6$ (R_2C_1)	REFSMMAT$_7$ (R_2C_1)	
19	REFSMMAT$_8$ (R_2C_2)	REFSMMAT$_9$ (R_2C_2)	
20	REFSMMAT$_{10}$ (R_2C_3)	REFSMMAT$_{11}$ (R_2C_3)	
21	TCSI	TCSI	3633-4
22	DELVEET1$_0$ (X comp.)	DELVEET1$_1$ (X comp.)	
23	DELVEET1$_2$ (Y comp.)	DELVEET1$_3$ (Y comp.)	2266 - 2273
24	DELVEET1$_4$ (Z comp.)	DELVEET1$_5$ (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
25	$VGPREV_0$ (VGTIG-X comp.)	$VGPREV_1$ (VGTIG-X comp.)	3700 - 3705
26	$VGPREV_2$ (VGTIG-Y comp.)	$VGPREV_3$ (VGTIG-Y comp.)	
27	$VGPREV_4$ (VGTIG-Z comp.)	$VGPREV_5$ (VGTIG-Z comp.)	
28	$DNRADATA_6$ (DNLRVELZ)	$DNRADATA_7$ (DNLRALT)	1335, 1336
29	TPASS4 (TPF time)	TPASS4 (TPF time)	3630-1
30	REDOCTR	$THETAD_0$ (X-angle)	0320, 0321
31	$THETAD_1$ (Y-angle)	$THETAD_2$ (Z-angle)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	0074 - 0107
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	
45	$DSPTAB_0$	$DSPTAB_1$	1022 - 1035
46	$DSPTAB_2$	$DSPTAB_3$	
47	$DSPTAB_4$	$DSPTAB_5$	
48	$DSPTAB_6$	$DSPTAB_7$	
49	$DSPTAB_8$	$DSPTAB_9$	
50	$DSPTAB_{10}$	$DSPTAB_{11}$	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	
53	RN_2 (Y comp.)	RN_3 (Y comp.)	1217 - 1224
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	
56	VN_2 (Y comp.)	VN_3 (Y comp.)	1225 - 1232
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	**ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	$DOWNTORK_2$ (POSTORKU)	$DOWNTORK_3$ (NEGTORKU)	3115, 3116
66	$DOWNTORK_4$ (POSTORKV)	$DOWNTORK_5$ (NEGTORKV)	3117, 3120
67	SPARE	SPARE	
68	TCDH	TCDH	1776-7
69	$DELVEET2_0$ (X comp.)	$DELVEET2_1$ (X comp.)	
70	$DELVEET2_2$ (Y comp.)	$DELVEET2_3$ (Y comp.)	2274 - 2301
71	$DELVEET2_4$ (Z comp.)	$DELVEET2_5$ (Z comp.)	
72	TTPI	TTPI	3635-6
73	$DELVEET3_0$ (X comp.)	$DELVEET3_1$ (X comp.)	
74	$DELVEET3_2$ (Y comp.)	$DELVEET3_3$ (Y comp.)	2365 - 2372
75	$DELVEET3_4$ (Z comp.)	$DELVEET3_5$ (Z comp.)	
76	$DNRADATA_1$ (DNRRANGE)	$DNRADATA_2$ (DNRRDOT)	1330, 1331

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
77	$DNRADATA_4$ (DNLRVELX)	$DNRADATA_5$ (DNLRVELY)	1333, 1334
78	$DNRADATA_6$ (DNLRVELZ)	$DNRADATA_7$ (DNLRALT)	1335, 1336
79	DIFFALT	DIFFALT	3577 - 3600
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	TIG	TIG	3441-2
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	* DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	$DOWNTORK_0$ (POSTORKP)	$DOWNTORK_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	
95	PIPTIME1	PIPTIME1	3560-1
96	$DELV_0$ (X comp.)	$DELV_1$ (X comp.)	0324 - 0331 (words 96–98)
97	$DELV_2$ (Y comp.)	$DELV_3$ (Y comp.)	
98	$DELV_4$ (Z comp.)	$DELV_5$ (Z comp.)	
99	SPARE	SPARE	
100	TGO	TGO	3516-7

DESCENT AND ASCENT LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77773_8)	SYNC PATTERN (77340_8)	
2	$LRCDUDL_0$ (LRXCDUDL)	$LRCDUDL_1$ (LRYCDUDL)	2334, 2335
3	$LRCDUDL_2$ (LRZCDUDL)	LRVTIMDL (MSB)	2336, 2337
4	VSELECT	*VMEAS (MSB)	3651, 3652
5	LRVTIMDL	LRVTIMDL	2337 - 2340
6	VMEAS (LR Velocity)	VMEAS (LR Velocity)	3652-3
7	MKTIME	MKTIME	3754-5
8	HMEAS (LR Range)	HMEAS (LR Range)	3654-5
9	RM_0 (RR Range)	RM_1 (RR Range rate)	3756, 3757
10	AIG (Y-angle)	AMG (Z-angle)	3457, 3460
11	AOG (X-angle)	TRKMKCNT	3461, 3462
12	$TANGNB_0$ (RR Trunnion)	$TANGNB_1$ (RR Shaft)	3752, 3753
13	MKTIME	MKTIME	3754-5
14	TEVENT	TEVENT	1341-2
15	$UNFC_0$ (X comp.)	$UNFC_1$ (X comp.)	
16	$UNFC_2$ (Y comp.)	$UNFC_3$ (Y comp.)	3252 - 3257
17	$UNFC_4$ (Z comp.)	$UNFC_5$ (Z comp.)	
18	$VGVECT_0$ (X comp.)	$VGVECT_1$ (X comp.)	
19	$VGVECT_2$ (Y comp.)	$VGVECT_3$ (Y comp.)	3645-3652
20	$VGVECT_4$ (Z comp.)	$VGVECT_5$ (Z comp.)	
21	TTF	TTF	3642-3
22	DELTAH	DELTAH	3664-5

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
23	RLS_0 (X comp.)	RLS_1 (X comp.)	2022 - 2027
24	RLS_2 (Y comp.)	RLS_3 (Y comp.)	
25	RLS_4 (Z comp.)	RLS_5 (Z comp.)	
26	ZDOTD	ZDOTD	2276-7
27	$X789_x$ (RR Shaft bias)	$X789_x$ (RR Shaft bias)	1700-1
28	$X789_y$ (RR Trunnion bias)	$X789_y$ (RR Trunnion bias)	1702-3
29	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
30	REDOCTR	$THETAD_0$ (X-angle)	0320, 0321
31	$THETAD_1$ (Y-angle)	$THETAD_2$ (Z-angle)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	0074 - 0107
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	
45	$DSPTAB_0$	$DSPTAB_1$	1022 - 1035
46	$DSPTAB_2$	$DSPTAB_3$	
47	$DSPTAB_4$	$DSPTAB_5$	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
48	$DSPTAB_6$	$DSPTAB_7$	
49	$DSPTAB_8$	$DSPTAB_9$	1022 - 1035
50	$DSPTAB_{10}$	$DSPTAB_{11}$	
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	
53	RN_2 (Y comp.)	RN_3 (Y comp.)	1217 - 1224
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	
56	VN_2 (Y comp.)	VN_3 (Y comp.)	1225 - 1232
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	*ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	$DOWNTORK_2$ (POSTORKU)	$DOWNTORK_3$ (NEGTORKU)	3115, 3116
66	$DOWNTORK_4$ (POSTORKV)	$DOWNTORK_5$ (NEGTORKV)	3117, 3120
67	RGU_0 (X comp.)	RGU_1 (X comp.)	
68	RGU_2 (Y comp.)	RGU_3 (Y comp.)	2626 - 2633
69	RGU_4 (Z comp.)	RGU_5 (Z comp.)	
70	VGU_0 (X comp.)	VGU_1 (X comp.)	
71	VGU_2 (Y comp.)	VGU_3 (Y comp.)	3626 - 3633
72	VGU_4 (Z comp.)	VGU_5 (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
73	LAND$_0$ (X comp.)	LAND$_1$ (X comp.)	
74	LAND$_2$ (Y comp.)	LAND$_3$ (Y comp.)	3634 - 3641
75	LAND$_4$ (Z comp.)	LAND$_5$ (Z comp.)	
76	AT	AT	2262-3
77	TLAND	TLAND	2400-1
78	FC	*RDOTV	3615, 3616
79	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	TIG	TIG	3441-2
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	*DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	DOWNTORK$_0$ (POSTORKP)	DOWNTORK$_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	
95	PIPTIME1	PIPTIME1	3560-1
96	DELV$_0$ (X comp.)	DELV$_1$ (X comp.)	
97	DELV$_2$ (Y comp.)	DELV$_3$ (Y comp.)	0324 - 0331
98	DELV$_4$ (Z comp.)	DELV$_5$ (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
99	PSEUDO55	FC	3614, 3615
100	TTOGO	TTOGO	3453-4

LUNAR SURFACE ALIGNMENT LIST

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
1	LIST ID (77772$_8$)	SYNC PATTERN (77340$_8$)	
2	R-OTHER$_0$ (X comp.)	R-OTHER$_1$ (X comp.)	
3	R-OTHER$_2$ (Y comp.)	R-OTHER$_3$ (Y comp.)	1717 - 1724
4	R-OTHER$_4$ (Z comp.)	R-OTHER$_5$ (Z comp.)	
5	V-OTHER$_0$ (X comp.)	V-OTHER$_1$ (X comp.)	
6	V-OTHER$_2$ (Y comp.)	V-OTHER$_3$ (Y comp.)	1725 - 1732
7	V-OTHER$_4$ (Z comp.)	V-OTHER$_5$ (Z comp.)	
8	TETCSM (T-OTHER)	TETCSM (T-OTHER)	1570-1
9	RANGRDOT	RANGRDOT+1	3760, 3761
10	AIG	AMG	3457, 3460
11	AOG	TRKMKCNT	3461, 3462
12	TANGNB$_0$ (RR trunnion)	TANGNB$_1$ (RR shaft)	3752, 3753
13	MKTIME	MKTIME	3754-5
14	TALIGN	TALIGN	2774-5
15	REFSMMAT$_0$ (R_1C_1)	REFSMMAT$_1$ (R_1C_1)	
16	REFSMMAT$_2$ (R_1C_2)	REFSMMAT$_3$ (R_1C_2)	
17	REFSMMAT$_4$ (R_1C_3)	REFSMMAT$_5$ (R_1C_3)	1733 - 1746
18	REFSMMAT$_6$ (R_2C_1)	REFSMMAT$_7$ (R_2C_1)	
19	REFSMMAT$_8$ (R_2C_2)	REFSMMAT$_9$ (R_2C_2)	
20	REFSMMAT$_{10}$ (R_2C_3)	REFSMMAT$_{11}$ (R_2C_3)	
21	YNBSAV$_0$ (X comp.)	YNBSAV$_1$ (X comp.)	
22	YNBSAV$_2$ (Y comp.)	YNBSAV$_3$ (Y comp.)	2236 - 2243
23	YNBSAV$_4$ (Z comp.)	YNBSAV$_5$ (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
24	$ZNBSAV_0$ (X comp.)	$ZNBSAV_1$ (X comp.)	
25	$ZNBSAV_2$ (Y comp.)	$ZNBSAV_3$ (Y comp.)	2244 - 2251
26	$ZNBSAV_4$ (Z comp.)	$ZNBSAV_5$ (Z comp.)	
27	$X789_x$ (RR shaft bias)	$X789_x$ (RR shaft bias)	1700-1
28	$X789_y$ (RR trunnion bias)	$X789_y$ (RR trunnion bias)	1702-3
29	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
30	REDOCTR	$THETAD_0$ (X-angle)	0320, 0321
31	$THETAD_1$ (Y-angle)	$THETAD_2$ (Z-angle)	0322, 0323
32	RSBBQ	RSBBQ+1	1432, 1433
33	OMEGAP	OMEGAQ	3021, 3022
34	OMEGAR	ALPHAQ	3023, 3024
35	CDUXD	CDUYD	3234, 3235
36	CDUZD	*DELCDUX	3236, 3237
37	CDUX	CDUY	0032, 0033
38	CDUZ	CDUT	0034, 0035
39	FLAGWRD0	FLAGWRD1	
40	FLAGWRD2	FLAGWRD3	
41	FLAGWRD4	FLAGWRD5	0074 - 0107
42	FLAGWRD6	FLAGWRD7	
43	FLAGWRD8	FLAGWRD9	
44	FLGWRD10	FLGWRD11	
45	$DSPTAB_0$	$DSPTAB_1$	
46	$DSPTAB_2$	$DSPTAB_3$	
47	$DSPTAB_4$	$DSPTAB_5$	1022 - 1035
48	$DSPTAB_6$	$DSPTAB_7$	
49	$DSPTAB_8$	$DSPTAB_9$	
50	$DSPTAB_{10}$	$DSPTAB_{11}$	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
51	TIMENOW (TIME2)	TIMENOW (TIME1)	0024-5
52	RN_0 (X comp.)	RN_1 (X comp.)	
53	RN_2 (Y comp.)	RN_3 (Y comp.)	1217 - 1224
54	RN_4 (Z comp.)	RN_5 (Z comp.)	
55	VN_0 (X comp.)	VN_1 (X comp.)	
56	VN_2 (Y comp.)	VN_3 (Y comp.)	1225 - 1232
57	VN_4 (Z comp.)	VN_5 (Z comp.)	
58	PIPTIME	PIPTIME	1233-4
59	OMEGAPD	OMEGAQD	3242, 3243
60	OMEGARD	*ECDUW	3244, 3245
61	$CADRFLSH_0$	$CADRFLSH_1$	0372, 0373
62	$CADRFLSH_2$	$FAILREG_0$	0374, 0375
63	$FAILREG_1$	$FAILREG_2$	0376, 0377
64	RADMODES	DAPBOOLS	0110, 0111
65	OGC	OGC	2737 - 2740
66	IGC	IGC	2741-2
67	MGC	MGC	2743-4
68	BESTI (STAR ID1)	BESTJ (STAR ID2)	2755, 2756
69	$STARSAV1_0$ (X comp.)	$STARSAV1_1$ (X comp.)	
70	$STARSAV1_2$ (Y comp.)	$STARSAV1_3$ (Y comp.)	2760 - 2765
71	$STARSAV1_4$ (Z comp.)	$STARSAV1_5$ (Z comp.)	
72	$STARSAV2_0$ (X comp.)	$STARSAV2_1$ (X comp.)	
73	$STARSAV2_2$ (Y comp.)	$STARSAV2_3$ (Y comp.)	2766 - 2773
74	$STARSAV2_4$ (Z comp.)	$STARSAV2_5$ (Z comp.)	
75	$GSAV_0$ (X comp.)	$GSAV_1$ (X comp.)	
76	$GSAV_2$ (Y comp.)	$GSAV_3$ (Y comp.)	2230 - 2235
77	$GSAV_4$ (Z comp.)	$GSAV_5$ (Z comp.)	

WORD #	FIRST REGISTER	SECOND REGISTER	ERASABLE ADDRESS
78	AGSK (K-FACTOR)	AGSK (K-FACTOR)	2020-1
79	LASTTCMD (LASTYCMD)	LASTSCMD (LASTXCMD)	0112, 0113
80	LEMMASS	CSMMASS	1326, 1327
81	IMODES30	IMODES33	1277, 1300
82	TIG	TIG	3441-2
83	OMEGAP	OMEGAQ	3021, 3022
84	OMEGAR	ALPHAQ	3023, 3024
85	CDUXD	CDUYD	3234, 3235
86	CDUZD	*DELCDUX	3236, 3237
87	CDUX	CDUY	0032, 0033
88	CDUZ	CDUT	0034, 0035
89	ALPHAQ	ALPHAR	3024, 3025
90	$DOWNTORK_0$ (POSTORKP)	$DOWNTORK_1$ (NEGTORKP)	3113, 3114
91	CHANNEL11	CHANNEL12	
92	CHANNEL13	CHANNEL14	
93	CHANNEL30	CHANNEL31	
94	CHANNEL32	CHANNEL33	
95	PIPTIME1	PIPTIME1	3560-1
96	$DELV_0$ (X comp.)	$DELV_1$ (X comp.)	0324 - 0331 (words 96–98)
97	$DELV_2$ (Y comp.)	$DELV_3$ (Y comp.)	
98	$DELV_4$ (Z comp.)	$DELV_5$ (Z comp.)	
99	SPARE	SPARE	
100	SPARE	SPARE	

The following table gives the downlink parameters, basic scale factors, units, and sections referenced for definitions. The pages following this table contain a list of definitions compiled from the listed references.

Mnemonic	Scale Factor	Units	Definition Reference
$AGSBUFF_{0-13}$ (12SP) (2DP)	Position - B25 (EARTH) B23 (MOON) Velocity - B15 (EARTH) B13 (MOON) Time - B18	feet feet/second seconds	EXVB
AGSK (DP)	B28	centiseconds	EXVB
AIG (SP)	B-1 (2's comp)	revolutions	RNAV SERV
ALPHAQ (SP)	B-2	revolutions/second squared	DAPA
ALPHAR (SP)	B-2	revolutions/second squared	DAPA
AMG (SP)	B-1 (2's comp)	revolutions	RNAV SERV
AOG (SP)	B-1 (2's comp)	revolutions	RNAV SERV
AOTCODE (SP)	OCTAL	optics code	ALIN
AT (DP)	B-9	meters/centisecond squared	ASCT
BESTI (SP)	B14	six times the decimal equivalent of the star IDs	ALIN
BESTJ (SP)	B14		ALIN
$CADRFLSH_{0-2}$ (3SP)	OCTAL	address	DINT
CDUS (SP)	B-1 (2's comp)	revolutions	RADR
CDUT (SP)	B-1 (2's comp)	revolutions	RADR
CDUX (SP)	B-1 (2's comp)	revolutions	IMUC COOR DAPA
CDUY (SP)	B-1 (2's comp)	revolutions	IMUC COOR DAPA

Mnemonic	Scale Factor	Units	Definition Reference
CDUZ (SP)	B-1 (2's comp)	revolutions	IMUC COOR DAPA
CDUXD (SP)	B-1 (2's comp)	revolutions	DAPA
CDUYD (SP)	B-1 (2's comp)	revolutions	DAPA
CDUZD (SP)	B-1 (2's comp)	revolutions	DAPA
CHANNELS (see CHANNEL tables)			
CENTANG (DP)	B0	revolutions	TRGL
COMPNUMB (SP)	B14	counts	EXVB
CSMMASS (SP)	B16	kilograms	DAPB
DAPBOOLS (see FLAGWORD tables or DAPA)			
DELCDUX (SP)	B-1 (2's comp)	revolutions	DAPA
DELLT4 (DP)	B28	centiseconds	TRGL
DELTAH (DP)	B24	meters	SERV
$\underline{D}$ELV (3DP)	B14	centimeters/second	SERV
$\underline{D}$ELVEET1 (3DP)	B7	meters/centisecond	TRGX
$\underline{D}$ELVEET2 (3DP)	B7	meters/centisecond	TRGX
$\underline{D}$ELVEET3 (3DP)	B7	meters/centisecond	TRGL
$\underline{D}$ELVLVC (3DP)	B7	meters/centisecond	TRGX
DELVTPF (DP)	B7	meters/centisecond	TRGL
DIFFALT (DP)	B29	meters	TRGX
$DNRADATA_1$ (SP)	B14	counts	RADR
$DNRADATA_2$ (SP)	B14	counts	RADR
$DNRADATA_4$ (SP)	B14	counts	RADR

Mnemonic	Scale Factor	Units	Definition Reference
$DNRADATA_5$ (SP)	B14	counts	RADR
$DNRADATA_6$ (SP)	B14	counts	RADR
$DNRADATA_7$ (SP)	B14	counts	RADR
$DOWNTORK_{0-5}$ (6SP)	B5	jet-seconds	DAPB
$DSPTAB_{0-11}$ (see DSKY)			
ELEV (DP)	B0	revolutions	TRGL TRGX
$FAILREG_{0-2}$ (3SP)	OCTAL	alarm code	PGSR
FC (SP)	B14	throttle pulses	DESC
$FLAGWRD_{0-11}$ (see FLAGWORD tables)			
GSAV (3DP)	B1	unit vector	ALIN
HMEAS (DP)	B28	counts	SERV
IMODES30, 33 (see IMUC and INTR Sections)			
LAND (3DP)	B24	meters	DESC
LASTSCMD (SP)	B14	pulses	RADR
LASTTCMD (SP)	B14	pulses	RADR
LEMMASS (SP)	B16	kilograms	DAPB
LRVCDUDL (3DP)	B-1 (2's comp)	revolutions	SERV
LRVTIMDL (DP)	B28	centiseconds	SERV
MKTIME (DP)	B28	centiseconds	SERV RNAV
NN (DP)	B14	counts	TRGX
OGC, IGC, MGC (3DP)	B0	revolutions	COOR
OMEGAP (SP)	B-3	revolutions/sec	DAPA
OMEGAQ (SP)	B-3	revolutions/sec	DAPA
OMEGAR (SP)	B-3	revolutions/sec	DAPA

Mnemonic	Scale Factor	Units	Definition Reference
OMEGAPD (SP)	B-3	revolutions/sec	DAPA
OMEGAQD (SP)	B-3	revolutions/sec	DAPA
OMEGARD (SP)	B-3	revolutions/sec	DAPA
PIPA (3SP)	B14	counts	SERV
PIPTIME (DP)	B28	centiseconds	SERV
PIPTIME1 (DP)	B28	centiseconds	SERV
PSEUDO55 (SP)	B14	throttle pulses	DESC
RADMODES (see FLAGWORD tables or RADR)			
RANGRDOT (2SP) (Same as $DNRADATA_{1,2}$)			
R-OTHER (3DP)	B29	meters	TELE
REDOCTR (SP)	B14	counts	PGSR
[$REFSMMAT_{0-11}$] (6DP)	B1	unit vector	COOR
RGU (3DP)	B24	meters	DESC
RLS (3DP)	B27	meters	DESC
RM (DP)	B29	meters	RNAV
RN (3DP)	B29	meters	SERV
RSBBQ (2SP)	OCTAL	address	PGSR
RTARG (3DP)	B29	meters	TRGL
STARSAV1 (3DP)	B1	unit vector	ALIN
STARSAV2 (3DP)	B1	unit vector	ALIN
TALIGN (DP)	B28	centiseconds	ALIN
$TANGNB_0$ (SP)	B-1	revolutions	RADR
$TANGNB_1$ (SP)	B-1	revolutions	RADR

Mnemonic	Scale Factor	Units	Definition Reference
TCDH (DP)	B28	centiseconds	TRGX
TCSI (DP)	B28	centiseconds	TRGX
TEVENT (DP)	B28	centiseconds	BURN
TETCSM (DP)	B28	centiseconds	ORBI
TGO (DP)	B28	centiseconds	BURN
THETAD (3SP)	B-1 (2's comp)	revolutions	ATTM IMUC
TIG (DP)	B28	centiseconds	BURN
TIMENOW (DP)	B28	centiseconds	EXVB
TLAND (DP)	B28	centiseconds	DESC
TPASS4 (DP)	B28	centiseconds	TRGL
TRKMKCNT (SP)	B14	counts	RNAV
TTF 8 (DP)	B17	centiseconds	DESC
TTOGO (DP)	B28	centiseconds	BURN
TTPI (DP)	B28	centiseconds	TRGL
UNFC (3DP)	variable	variable	BURN DESC
$UPBUFF_{0-19}$ (20SP)	OCTAL	variable	EXVB
UPCOUNT (SP)	B14	counts	EXVB
UPOLDMOD (SP)	B14	program number	EXVB
UPVERB (SP)	B14	last digit of verb	EXVB
V-OTHER (3DP)	B7	meters/centisecond	TELE
VGPREV (3DP)	B7	meters/centisecond	BURN
VGU (3DP)	B10	meters/centisecond	DESC
VGVECT (3DP)	B7	meters/centisecond	ASCT

Mnemonic	Scale Factor	Units	Definition Reference
VMEAS (DP)	B28	counts	SERV
$\underline{V}$N (3DP)	B7	meters/centisecond	SERV
VSELECT (SP)	B14	LR velocity code	SERV
$X789_{0-1}$ (DP)	B5 (EARTH) B3 (MOON)	radians	RNAV
$X789_{2-3}$ (DP)	B5 (EARTH) B3 (MOON)	radians	RNAV
$\underline{Y}$NBSAV (3DP)	B1	unit vector	ALIN
ZDOTD (DP)	B7	meters/centisecond	ASCT
$\underline{Z}$NBSAV (3DP)	B1	unit vector	ALIN

$AGSBUFF_{0,2,4}$: Single precision X, Y, and Z components of the LM position vector, scaled B25 (earth) and B23 (moon) in units of feet and in stable member coordinates.

$AGSBUFF_{1,3,5}$: Single precision X, Y, and Z components of the LM velocity vector, scaled B13 (moon) or B15 (earth) in units of feet/second and in stable member coordinates.

$AGSBUFF_{6,8,10}$: Single precision X, Y, and Z components of the CSM position vector, scaled B25 (earth) or B23 (moon) in units of feet and in stable member coordinates.

$AGSBUFF_{7,9,11}$: Single precision X, Y, and Z components of the CSM velocity vector, scaled B15 (earth) or B13 (moon) in units of feet/second and in stable member coordinates.

$AGSBUFF_{12,13}$: Double precision difference between the timetag of the state vectors in $AGSBUFF_{0-11}$ and the time stored in AGSK, scaled B18 in units of seconds.

AGSK: Double precision time of AGS initialization, scaled B28 in units of centiseconds.

ALPHAQ, ALPHAR: Single precision storage for the most significant halves of AOSQ and AOSR for down telemetry, scaled B-2 in units of revolutions per second squared.

AOSQ, AOSR: Double precision disturbing acceleration due to thrust vector/ c.g. offset or other external torques, scaled B-2 in units of revolutions per second squared.

AOG, AIG, AMG: Single precision storage for ICDU angles, scaled B-1 in units of revolutions and stored in two's complement form.

AOTCODE: A single precision scalar containing the star selection code in bits 6-1 (an octal number from 1 to 45_8 for stars, 0 for a planet, and 46_8-50_8 for sun, earth, and moon), and the AOT detent code in bits 9-7 (1,2,3,4,5 or 6 for AOT detents; 0 for COAS calibration; 7 for COAS position to be specified.)

AT: Double precision LM thrust acceleration magnitude, scaled B-9 in units of meters per centisecond squared.

BESTI: Single precision value, scaled B14, of the index parameter for star #1 of the "best" star pair as determined by "R56." It is the star farthest from the AOT center detent position and will be zero if no star pairs are found that are satisfactory. In "R59" it is the value of the index parameter of the first celestial body used for marking (if two bodies are to be used). It is equal to six times the decimal equivalent of the "star selection code."

BESTJ: See BESTI. In "R56" it is the index parameter for star #2 which is the closest star to the AOT center detent position. In "R59" it is the index parameter of the second celestial body used for marking (if two are to be used) or the index parameter of the single body being used (Technique 3 alignment).

$CADRFLSH_i$ (i = 0,1,2): Three single precision cells for storage of return address information required by priority, mark and normal display requests. In the program CADRFLSH is used for storage of the address of the step after that at which the display interface routine is called.

CDUT, CDUS: LGC input counters incremented directly from the Coupling Data Unit to maintain LGC knowledge of the RR trunnion and shaft angles, respectively. Single precision angles stored in two's complement form and scaled B-1 in units of revolutions.

$\underline{C}DU$ (CDU_x, CDU_y, CDU_z): Single precision vector containing the measured values of the IMU gimbal angles (outer, inner, and middle gimbal in X, Y, and Z components, respectively), scaled B-1 in units of revolutions and stored in two's complement form. Each component is an LGC input counter incremented directly from the Coupling Data Unit in response to changes in the IMU gimbal angles.

$\underline{C}DUD$: Single precision vector interface with steering and attitude maneuver routines containing the desired values for the IMU gimbal angles (outer, inner, and middle gimbal angles in X, Y, and Z components, respectively), scaled B-1 in units of revolutions and stored in two's complement form.

CENTANG: Double precision central angle between the passive vehicle's position at TIG and at intercept, scaled B0 in units of revolutions.

COMPNUMB: Single precision number of components (each single precision octal) in a program 27 update, scaled B14 and unitless.

CSMMASS: Single precision astronaut input of the mass of the CSM, scaled B16 in units of kilograms.

DELCDU: Interface with steering and attitude maneuver routines, minus desired change in gimbal angles per 100 millisecond period, scaled B-1 in units of revolutions, stored in two's complement form.

DELLT4: Double precision maneuver transfer time, scaled B28 in units of centiseconds.

DELTAH: Double precision difference between the calculated altitude and that measured by the Landing Radar, scaled B24 in units of meters.

DELV: Double precision sensed-change-in-velocity vector, scaled B14 in units of centimeters per second (one PIPA pulse represents one centimeter per second on the LM) and expressed in Platform coordinates.

DELVEET1: Double precision vector corresponding to the velocity-to-be-gained vector for the CSI burn, scaled B7 in units of meters per centi-second. Parallel to the orbital plane of the passive vehicle and perpendicular to the active vehicle position vector at TCSI.

DELVEET2: Double precision vector corresponding to the velocity-to-be-gained vector for the CDH burn, scaled B7 in units of meters per centisecond. Parallel to the orbital plane of the passive vehicle.

DELVEET3: Double precision velocity to be gained vector, scaled B7 in units of meters per centisecond. Calculated by the "INITVEL" routine.

DELVLVC: Double precision velocity vector expressed in local vertical coordinates, scaled B7 in units of meters per centisecond. In the local vertical coordinate system, X is along the horizontal component of velocity, Z points toward the center of attraction, and Y completes a right-handed, orthogonal system.

DELVTPF: Double precision magnitude of the velocity to be gained in the final rendezvous maneuvers of the terminal phase, scaled B7 in units of meters per centisecond.

DIFFALT: Double precision difference of passive and active vehicle altitudes at the time of CDH, scaled B29 in units of meters; negative if the passive vehicle is below the active vehicle at CDH, and displayed to the astronaut during P32-72 and P33-73.

$DNRADATA_i$: Special storage for downlink of radar data. i = 1,2,4,5,6, and 7 to index six single precision cells (consecutive except between i = 2 and 4) alternately labeled DNRRANGE, DNRRDOT, DNLRVELX, DNLRVELY, DNLRVELZ, and DNLRALT, respectively. RNRAD is a single precision LGC counter advanced directly by whichever radar circuit is enabled for sampling, scaled B14 in units of counts.

Sample Type	Value of 1 count	
* DNRRDOT	-0.19135344 meters/second	-0.6278 fps

Sample Type	Value of 1 count	
DNRRANGE		
Low scale	2.859024 meters	9.38 feet
High scale	22.872192 meters	75.04 feet
** DNLRVELX	-0.1962912 meters/second	-0.6440 fps
** DNLRVELY	0.3694176 meters/second	1.212 fps
** DNLRVELZ	0.2642006 meters/second	0.8668 fps
DNLRALT		
Low scale	0.3288792 meters	1.0790 feet
High scale	1.644796 meters	5.3950 feet

* DNRRDOT has a bias to be subtracted from the raw counts.

K:RDOTBIAS: Double precision constant stored as 17000 x 2^{-28}, scaled B28 in units of radar counts (same as RNRAD). Equation value: 17000.

**$DNLRVEL_S$: These have a bias to be added to the raw counts.

K:LVELBIAS: Single precision constant stored as -12288 x 2^{-14}, scaled B14 in units like those of RNRAD. Equation value: -12288.

$DOWNTORK_i$ (i = 0-5): Single precision table of quantities for downlink which give cumulative jet on times for the various axes; the correspondence is (0, +P; 1, -P; 2, +U; 3, -U; 4, +V; 5, -V); Scaled B5 in units of seconds.

ELEV: Double precision elevation angle of the line-of-sight to the passive vehicle; measured from the vector which is perpendicular to the active vehicle position vector, perpendicular to $\underline{R}ACT * \underline{R}PASS$, and whose dot product with the active vehicle velocity vector is positive. An angle between 0 and 1 (0 and 360 degrees) scaled B0 in units of revolutions. ELEV is greater than $\frac{1}{2}$ (180 degrees) if the passive vehicle is below the active vehicle's local horizontal. ELEV is an astronaut input in P32-72 and an optional input in P34-74.

$FAILREG_i$ (i = 0,1,2): Three single precision registers used for storage of the alarm code information. $FAILREG_{0,1}$ are zeroed via an "ERROR RESET"; $FAILREG_2$ is unaltered. All three registers are zeroed by a Verb 36. $FAILREG_0$ contains the first alarm after the "ERROR RESET"; $FAILREG_1$ contains the second; and $FAILREG_2$ always contains the most recent.

FC: Single precision storage for the magnitude of desired thrust, scaled B14 in units of DPS throttle pulses.

K:SCALEFAC: Double precision constant stored as 797.959872 x 2^{-16}, scaled B16 in units of DPS throttle pulses/kilogram meter per centisecond squared. Equation value: 797.959872. (Equivalent to 12.532 newtons or 2.8173 pounds force per pulse.)

G̲SAV: Double precision storage for unit gravity vector determined in previous pass through "P57," scaled B1 and expressed in navigation base coordinates.

HMEAS: Double precision Landing Radar measurement of altitude, scaled B28 in units of low scale landing radar altitude bits.

K:HSCAL: Double precision constant stored as -0.3288792, scaled B0 in units of meters per bit. Equation value: -0.3288792. (Equivalent to 1.0790 feet per bit.)

L̲AND: Double precision position vector of the landing site, scaled B24 in units of meters, measured from the center of the moon and expressed in the Platform coordinate system.

LASTSCMD, LASTTCMD: Storage for the previous value of total RR shaft and trunnion CDU error counters; scaled B14 in units of RR drive pulses. Used to convert present position deviation into a desired rate command to be inserted into CDUSCMD or CDUTCMD.

K:RDESGAIN: Single precision constant stored as 0.53624, scaled B12 in units of RR drive pulses per radian of error. Equation value 2196.5. (Equivalent to 0.5 x $2sec^{-1}$ x 360 deg/rev x $(10/384)^{-1}$ pulses per degree per second. The first two terms null 0.5 of the error in ½ second, and the fourth is derived from the fact that a saturated error counter causes a drive rate of 10 degrees per second.)

K:RRSPGAIN: Single precision constant stored as 0.59062, scaled B15 in units of RR drive pulses per revolution of error. Equation value: 19353. (Equivalent to 0.7 x $2sec^{-1}$ x 360 deg/rev x $(10/384)^{-1}$ pulses per degree per second. The first two terms "null 0.7 of the error in ½ second," and the fourth is derived from the fact that a saturated error counter causes a drive rate of 10 degrees per second.)

LEMMASS: Single precision astronaut input of the mass of the LM, scaled B16 in units of kilograms.

LRCDUDL: Single precision vector storage for the value of the three ICDU angles at the estimated midpoint of an LR velocity reading, scaled B-1 in units of revolutions and stored in two's complement form.

LRVTIMDL: Double precision time at the estimated midpoint of the LR velocity sample, scaled B28 in units of centiseconds.

MKTIME: Double precision time of PIPA readings which are associated with the LR altitude measurement for downlink purposes, scaled B28 in units of centiseconds. Also the time of RR range rate measurement, considered to be RR time of mark.

NN: Double precision number designating the apsidal crossing after CSI at which the CDH burn will be executed, scaled B14 and unitless. (NN = 1 indicates that the CDH burn will be executed at the first apsidal crossing after CSI.) NN is used in P34, 35, 74 and 75 as a flag to specify precision or conic integration. In "S3435.2 ," it is used to set VTARGTAG.

OGC, IGC, MGC: Double precision commanded gimbal angles scaled B0 in units of revolutions or (equivalently) scaled B21 in units of gyro torque pulses of 2^{-21} revolutions each.

OMEGAP, OMEGAQ, OMEGAR: Single precision estimated vehicle rate, calculated using commanded accelerations and times, scaled B-3 in units of revolutions per second. Limited to ±0.12499 (±44.997 degrees/second) by overflow checks.

OMEGAPD, OMEGAQD, OMEGARD: Single precision rate biases generated in the attitude maneuver and steering routines, scaled B-3 in units of revolutions per second.

PIPA: Single precision sensed-change-in-velocity vector, scaled B14 in units of centimeters per second and expressed in the Platform coordinate system. The three components are incremented directly from the Pulse-Integrating, Pendulous Accelerometers on the stable member of the Inertial Measurement Unit.

PIPTIME: Double precision time of the most recent PIPA read cycle, scaled B28 in units of centiseconds; time at which the average-g and free-flight state vector is valid.

PIPTIME1: Temporary storage for PIPTIME to avoid changing the downlink state vector until it is updated homogeneously.

PSEUDO55: Single precision storage for telemetry of the throttle command sent to the descent engine, scaled B14 in units of throttle pulses.

R-OTHER: Double precision navigation position vector of the CSM, scaled B29 in units of meters. See RCSM in SERV section.

REDOCTR: Single precision counter set to zero in a fresh start and incremented whenever a hardware restart occurs; scaled B14 and unitless.

[REFSMMAT]: Double precision, 3x3 transformation matrix, scaled B1 and unitless. Defined such that Asm = [REFSMMAT] Arf where A is a vector expressed in stable member and reference coordinates, respectively.

RGU: Double precision position vector of the LM, scaled B24 in units of meters, measured from the landing site on the moon's surface and expressed in the Descent Guidance coordinate system.

RLS: Double precision vector position of the landing site relative to the center of the moon, scaled B27 in units of meters and expressed in the Selenographic (moon-fixed) coordinate system; part of the erasable load.

RM: Double precision magnitude of measured range, scaled B29 in units of meters. Also used in routine 29 as two single precision storage cells (RM_0 and RM_1) for downlink. They are identical to $DNRADATA_1$, and $DNRADATA_2$, respectively.

RN: Double precision vector position of the LM measured from the center of the earth or moon, scaled B29 in units of meters and expressed in the Reference coordinate system.

RSBBQ: Storage for the value of the address where a hardware restart occurred. The most significant part contains the BBANK and SUPERBNK information; the least significant part contains the Q-register information.

RTARG: Target position vector input to "INITVEL". Scaled B29 in units of meters. Upon exit, "INITVEL" loads RTARG with the biased target position vector, if such a biased vector is calculated.

STARSAV1, STARSAV2: Double precision vectors scaled B1 and unitless. Used to store the two "measurement" vectors for comparison with two "reference" vectors to determine IMU alignment. Expressed in stable member coordinates.

TETCSM: Double precision state vector time for CSM scaled B28 in units of centiseconds.

TALIGN: Double precision time for determination of IMU alignment, scaled B28 in units of centiseconds.

$TANGNB_0$, $TANGNB_1$: Temporary two's complement storage (astronaut desired or radar marked) radar position angles (trunnion and shaft, respectively), scaled B-1 in unit of revolutions.

TCDH: Double precision time of ignition of the CDH burn, scaled B28 in units of centiseconds; an astronaut input in P33-P73.

TCSI: Double precision time of ignition of the CSI burn, scaled B28 in units of centiseconds. It may be either an astronaut input or computed by the program.

TEVENT: Double precision time-of-event for downlink information, scaled B28 in units of centiseconds.

TGO: Double precision predicted length of burn, scaled B28 in units of centiseconds.

THETAD: Single precision vector containing the gimbal angles that define the desired orientation to which the attitude maneuver routines are to maneuver; scaled B-1 in units of revolutions and stored in two's complement form. Also called CPHI in program.

TIG: Double precision predicted time of ignition input to the burn routines, or predicted cutoff time, scaled B28 in units of centiseconds.

TIMENOW: Double precision computer clock, incremented every centisecond (one hundredth of a second) by the LGC oscillator; scaled B28 in units of centiseconds.

TLAND: Double precision nominal time of lunar landing, scaled B28 in units of centiseconds; part of the erasable load.

TPASS4: Double precision scheduled time of target intercept, scaled B28 in units of centiseconds.

TRKMKCNT: Single precision count of number of navigation updates made during P20 or P22, scaled B14 and unitless. Cell also used in R29 to indicate data storage for downlink; 1 - data stored, 0 - data not stored.

TTF: Double precision negative time from now until achiement of target conditions of the present quidance phase, scaled B17 in units of centiseconds.

TTOGO: Double precision time until engine ignition (or cutoff), scaled B28 in units of centiseconds.

TTPI: Double precision time of terminal phase initiation, scaled B28 in units of centiseconds; an astronaut input in P32-P72 and P34-P74.

UNFC: Double precision desired thrust acceleration vector, with variable scaling in units of meters per centisecond squared and expressed in the Platform coordinate **system.** During the pre-ignition phase computations for the powered descent maneuver (P63), UNFC represents the Delta-V vector for the pre-full throttle thrust, scaled B7 in units of meters per centisecond.

$UPBUFF_{0-19}$: Single precision buffer cells for P27 updates.

UPCOUNT: Single precision number of components received in a P27 update, scaled B14 and unitless.

UPOLDMOD: Single precision storage for the value of MODREG at the initialization of a P27 update , scaled B14.

UPVERB: Single precision indication of the verb that initiated a P27 update, scaled B14 and unitless.

V-OTHER: Double precisions navigation velocity vector of the CSM scaled B7 in units of meters/centisecond.

$\underline{V}$GPREV: Double precision previous value of $\underline{V}$G, program notation also VGTIG, scaled B7 in units of meters per centisecond and expressed in the reference coordinate system.

$\underline{V}$GU: Double precision velocity vector of the LM relative to the rotating moon, scaled B10 in units of meters per centisecond and expressed in the Descent Guidance coordinate system.

$\underline{V}$GVECT: Double precision velocity-to-be-gained vector in Platform coordinates, scaled B7 in units of meters per centisecond.

VMEAS: Double precision velocity measurement (sum of 5 samples) from the LR sampling, scaled B28 in units of Landing Radar velocity bits.

$K:VSCAL_0$: Double precision constant stored as 0.5410829105, program notation VZSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: 0.002642006/5. (Equivalent to 0.8668 fps per bit; the "5" averages the sum of five samples.)

K:VSCAL$_2$: Double precision constant stored as 0.7565672446, program notation VYSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: 0.003694176/5. (Equivalent to 1.212 fps per bit; the "5" averages the sum of five samples.)

K:VSCAL$_4$: Double precision constant stored as -0.4020043770, program notation VXSCAL, scaled B-10 in units of meters per centisecond per bit. Equation value: -0.001962912/5. (Equivalent to -0.644 fps per bit; the "5" averages the sum of five samples.)

$\underline{V}$N: Double precision inertial velocity vector of the LM, scaled B7 in units of meters per centisecond and expressed in the Reference coordinate system.

VSELECT: Single precision index used to distinguish among the Z (0), Y (1) and X (2) axes of the Landing Radar coordinate system, scaled B14 and unitless.

$\underline{X}$789: Double precision vector containing the best estimate of bias necessary to offset RR position error, scaled B5 (earth) or B3 (moon) in units of radians.

$\underline{Y}$NBSAV, $\underline{Z}$NBSAV: Double precision unit vectors in the directions of the Y and Z navigation base axes, scaled B1 and expressed in moon-fixed coordinates.

ZDOTD: Double precision desired downrange velocity scaled B7 in units of meters/centisecond.

TEST

VBTSTLTS Inhibit interrupts

If MODREG ≠ 0, proceed to "ALM/END"

Switch bit 1 of IMODES33 to 1 (indicate lamp test in progress)

Switch bits 7, 6, 5, 4, 3 and 1 of channel 11 to 1
(Illuminate operator error lamp; initiate verb-noun flash; illuminate key release, temperature caution, uplink activity, and ISS warning lamps)

Switch $DSPTAB_{11}$ to $100\ 000\ 110\ 111\ 100_2$
(Signal "T4RUPT" to light the program alarm, tracker fail, gimbal lock warning, no attitude, and LR fail lamps)

Perform "C13STALL"

Switch bit 10 of channel 13 to 1 (Light restart, standby and computer warning lamps)

Switch $DSPTAB_i$ to -05675_8 for i = 10 thru 0 (display all 8's)

Switch $DSPTAB_i$ to -07675_8 for i = 1, 4 and 6 (display + signs)

NOUT = 11

Release interrupt inhibit

Call "TSTLTS2" in 5 seconds

End job

TSTLTS2 Establish "TSTLTS3" (pr30)

End task

TSTLTS3 Inhibit interrupts

Switch bits 7, 4, 3 and 1 of channel 11 to 0
(Leave verb-noun and key release flashing)

Perform "C13STALL"

Switch bit 10 of channel 13 to 0

TS = bit 4 of channel 12

Switch $DSPTAB_{11}$ to 40000_8 + TS (leave no attitude lamp lit if in coarse align)

Switch bit 1 of IMODES33 to 0

Switch bits 13, 12 and 11 of IMODES33 to 1
(Reset PIPA fail, Downlink fail and Uplink fail bits)

Switch bits 13, 12 and 10 of IMODES30 to 1
(Reset IMU fail, ICDU fail and PIPA fail bits)

Switch bit 15 of IMODES30 to 0 (Reset IMU temperature bit)

Switch RADMODES bit 7 (RCDUFAIL) to 1

Switch RADMODES bits 8 (LRVELFLG), 5 (LRALTFLG) and 4 (RRDATAFL) to 0

Release interrupt inhibit

Establish "DSPMMJOB" (pr30)

MONSAVE1 = 40000_8

Switch bit 6 of channel 11 to 0 (verb-noun flash off)

Perform "RELDSP"

If CADRSTOR ≠ +0, proceed to "PINBRNCH"

End job

SHOWSUM2 Perform "PRIOCHNG" with A = 07000_8 (change priority to 07_8)

SKEEP6 = 1

SMODE = +0

SELFRET = "SELFCHK"

Proceed to "STSHOSUM"

SDISPLAY SKEEP3 = SKEEP2

SKEEP2 = TSbank

$MPAC_2$ = "SKEEP1"

Proceed to "GOXDSPF" with TS = K:V05N01 (SKEEP1, SKEEP2, SKEEP3)
(If terminate, continue at next step; if proceed, proceed to "NXTBNK"; if other response, repeat at previous step.)

SKEEP1 = "SELFCHK"

Proceed to "ENDEXT"

<u>SELFCHK</u> SKEEP1 = "ERASCHK"

Perform "CHECKNJ"

If SMODE = +0:

    Proceed to second step of "SELFCHK" (idle loop)

If SMODE = -0:

    SCOUNT = SCOUNT + 1

    Proceed to address specified in SKEEP1

If |SMODE| > 10_8:

    SMODE = +0

    Proceed to "SELFCHK"

SCOUNT = SCOUNT + 1

If |SMODE| = 1, 2, 3, 6, 7 or 10_8:

    Proceed to address specified in SKEEP1

If |SMODE| = 4, proceed to "ERASCHK"

If |SMODE| = 5, proceed to "ROPECHK"

<u>CHECKNJ</u> SELFRET = return address

Proceed to "ADVAN"

(Returns to caller of "CHECKNJ" via SELFRET if or when no jobs require processing.)

<u>ERASCHK</u> SKEEP2 = 1

EBANK = 0

SKEEP7 = 01461_8 (address of first non-special cell in bank 0)

$SKEEP3 = 01777_8$ (last address in bank 0)

ERASLOOP Inhibit interrupts

SKEEP4 = EBANK

$SKEEP5 = E_{SKEEP7}$

i = SKEEP7 + 1

$SKEEP6 = E_i$

ERESTORE = SKEEP7

$E_{SKEEP7} = SKEEP7$

$E_i = SKEEP7 + 1$

$TS = E_{SKEEP7} - E_i$

If TS $\neq$ -1, perform "PRERRORS"

If ERESTORE $\neq$ +0: (did not perform "PRERRORS")

$E_{SKEEP7} = - E_{SKEEP7}$

$E_i = - E_i$

$TS = E_i - E_{SKEEP7}$

If TS $\neq$ -1, perform "PRERRORS"

If ERESTORE $\neq$ +0:

$E_{SKEEP7} = SKEEP5$ (restore original contents)

$E_i = SKEEP6$

ERESTORE = +0

Release interrupt inhibit

Perform "CHECKNJ"

EBANK = SKEEP4 (in case it was changed by another job)

SKEEP7 = SKEEP7 + 1

If SKEEP7 $\neq$ SKEEP3, proceed to "ERASLOOP"

If SKEEP2 > 0:

    SKEEP2 = SKEEP2 - 1

    SKEEP7 = 00061_8 (first non-special cell in unswitched erasable)

    SKEEP3 = 01373_8

    Proceed to "ERASLOOP"

(Otherwise SKEEP2 = 0)

SKEEP2 = 1

EBANK = EBANK + 1

If EBANK = 2:

    SKEEP7 = 01400_8

    SKEEP3 = 01773_8

    Proceed to "ERASLOOP"

If EBANK < 10_8:

    SKEEP7 = 01400_8

    SKEEP3 = 01777_8

    Proceed to "ERASLOOP"

EBANK = 3

SKEEP2 = 50_8

<u>CNTRLOOP</u> TS = - contents of cell specified by (SKEEP2 + 10_8)

If SKEEP2 > 0:

    SKEEP2 = SKEEP2 - 1

    Proceed to "CNTRLOOP"

CYR = 25252_8 (should cycle right and become 12525_8)

CYL = 25252_8 (should cycle left and become 52524_8)

SR = 25252_8 (should shift right and become 12525_8)

EDOP = 25252_8 (should shift right 7 and become 00125_8)

TS = 25252_8 + CYR + CYL + SR + EDOP + 52400_8

If TS $\neq$ -1, perform "PRERRORS"

TS = CYR + CYL + SR + EDOP + 1

If TS $\neq$ -1, perform "PRERRORS"

$SCOUNT_1 = SCOUNT_1 + 1$

SKEEP1 = "ROPECHK"

Proceed to second step of "SELFCHK"

PRERRORS If ERESTORE $\neq$ +0:

E_{SKEEP7} = SKEEP5 (restore original contents)

E_i = SKEEP6

ERESTORE = +0

Inhibit interrupts

SFAIL = return address

$ALMCADR_0$ = SFAIL

ERCOUNT = ERCOUNT + 1

TS = 01102_8

Perform "ALARM2"

If SMODE $\geq$ +0:

SMODE = +0

Proceed to "SELFCHK"

If SMODE < 0, proceed to "SELFCHK"

Return via SFAIL

ROPECHK SKEEP6 = −0

STSHOSUM SKEEP4 = 0

SKEEP7 = 1

SKEEP1 = 0

SKEEP3 = 00000_8

SKEEP5 = 1

COMADRS L = SKEEP4

FCADR = bits 15-11 of SKEEP4 + SKEEP3

Perform "SUPDACAL"

SKEEP2 = A (contents of fixed memory cell specified by FCADR and L)

SKEEP1 = SKEEP1 + SKEEP2 with end around carry of ± overflow

TS = 02000_8 + SKEEP3 − SKEEP2

Proceed to "ADRSCHK"

FXADRS SKEEP2 = contents of cell specified in SKEEP3 (fixed-fixed banks 2,3)

SKEEP1 = SKEEP1 + SKEEP2 with end around carry of ± overflow

TS = SKEEP3 − SKEEP2

ADRSCHK TSadr = bits 10-1 of SKEEP3

If TSadr = 01777_8, proceed to "SOPTION" (end of bank)

If SKEEP5 < 0, proceed to "SOPTION" (2 consecutive TC SELF instructions have been encountered, indicating that the rest of the bank is unused)

If TS = −0: (contents of cell equals its address)

If SKEEP5 = 0, SKEEP5 = −1 (2nd consecutive TC SELF)

If SKEEP5 > 0, SKEEP5 = SKEEP5 − 1 (1st TC SELF)

If TS ≠ −0, SKEEP5 = 1

(SKEEP6 = 1 or -0)

If SKEEP6 > 0 and NEWJOB > 0:

Proceed to "CHANG1"

When jobs of higher priority are finished (if any), "CHANG1" will return here

Proceed to "ADRS+1"

If SKEEP6 = -0, perform "CHECKNJ"

ADRS+1 SKEEP3 = SKEEP3 + 1 (increment address)

If SKEEP7 ≥ +0, proceed to "COMADRS"

Proceed to "FXADRS"

SOPTION TSbank = integral part of SKEEP4 (rescaled from B4 to B14)

TS = 8^3 (fractional part of SKEEP4) (rescaled from B4 to B14)

If TS ≠ 0: (equals 30_8 or 40_8)

TSbank = TS + TSbank - 30_8

If SKEEP6 ≠ 0, proceed to "SDISPLAY"

SKEEP1 = |SKEEP1|

If SKEEP1 ≠ TSbank, perform "PRERRORS"

NXTBNK If SKEEP4 = 33.04_8:

If SKEEP6 ≠ 0, proceed to "STSHOSUM"

Proceed to "SELFCHK"

SKEEP4 = SKEEP4 + 1

If SKEEP4 = 30_8, SKEEP4 = SKEEP4 + 0.03_8 (011_2 in bits 7-5 for "SUPDACAL")

If SKEEP4 ≥ 40_8, SKEEP4 = 30.04_8 (100_2 in bits 7-5 for "SUPDACAL")

(Only difference between address in banks 30_8-33_8 and 40_8-43_8 is the SUPERBNK setting.)

If SKEEP7 > 0:

SKEEP7 = SKEEP7 - 1 (+0 if 0)

SKEEP1 = 0

SKEEP3 = 00000_8

SKEEP5 = 1

Proceed to "COMADRS"

If SKEEP7 = +0: (banks 2 and 3 are addressed using the "fixed-fixed" address scheme instead of the normal bank address scheme)

SKEEP7 = -1

SKEEP3 = 04000_8

SKEEP1 = 0

SKEEP5 = 1

Proceed to "FXADRS"

If SKEEP7 = -1: (bank 3)

SKEEP7 = -0

SKEEP3 = 06000_8

SKEEP1 = 0

SKEEP5 = 1

Proceed to "FXADRS"

SKEEP7 = 64

SKEEP1 = 0

SKEEP3 = 00000_8

SKEEP5 = 1

Proceed to "COMADRS"

REDO MODREG = 07

Establish "DSPMMJOB" (pr30)

Perform "IMUZERO"

Perform "IMUSTALL"

If ISSGOOD = 0, proceed to "SOMERR2"

NDXCTR = 0

TORQNDX = 0

OVFLOWCK = 0

Set matrix [XNB] = 0

DSPTEM1 $+1_{dp}$ = LATITUDE rescaled B-2 revs

DSPTEM1 = AZIMUTH twos complement, B-1 in revs

Proceed to "GOFLASH" with TS = K:V06N41
(If terminate, proceed to "ENDTEST1"; If proceed, continue below; if other response, repeat this step)

AZIMUTH = DSPTEM1 ones complement, B0 revs

LATITUDE = DSPTEM1 +1 rescaled to B0 revs

WANGI = -cos(LATITUDE)

WANGO = sin(LATITUDE)

$$[XNB] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sin(AZIMUTH) & \cos(AZIMUTH) \\ 0 & -\cos(AZIMUTH) & \sin(AZIMUTH) \end{bmatrix}$$

Perform "CALCGA"

Perform "IMUCOARS"

If FLAGWRD3 bit 14 (GLOKFAIL) = 1:

NDXCTR = NDXCTR + 1

Switch FLAGWRD3 bit 14 (GLOKFAIL) to 0

Perform "IMUSTALL"

If ISSGOOD = 0, proceed to"SOMERR2"

If NDXCTR > 0:

    Proceed to "PIPACHK"

Perform "IMUFINE"

Perform "IMUSTALL"

If ISSGOOD = 0, proceed to "SOMERR2"

Call "GOESTIMS" in $PERFDLAY_{dp}$ centi-seconds

A = "ESTIMS"

Proceed to "JOBSLEEP"

GOESTIMS   Awaken job with starting address "ESTIMS"

End task

ESTIMS   Inhibit interrupts

Call "ALLOOP" in 1SECXT Centi-seconds

PIPA = 0

Release interrupt inhibit

Zero 77 erasable memory cells starting at location "ALXIS -1"

GCOMPSW = 0

ALXIS = 144

CMPX1 = -1

ALK = K:soupy0

ALK_2 = K:soupy2

DELV = 0

GCOMP = 0

If TORQNDX $\leq$ 0:

  $\underline{E}$RVECTOR = K:omegms (sinLATITUDE, -cosLATITUDE, 0)

  TStmark = TIMENOW

  $\underline{E}$RCOMP = $\underline{0}$

Proceed to "SLEEPIE"

SLEEPIE If TORQNDX > 0, perform "EARTHR*"

End job

TORQUE DSPTEM2 = 0

DSPTEM2 +1 = $DRIFTI_{sp}$

TS = POSITON - 1

$SOUTHDR_{TS}$ = $DRIFTI_{sp}$

Perform "SHOW"

Proceed to "PIPACHK"

PIPACHK If NDXCTR = 0, perform "EARTHR*"

DATAPL +4 = 17

LENGTHOT = 58

RESULTCT = 1

$PIPA_{PIPINDEX}$ = 0

$DATAPL_{sp}$ = 0

Perform "CHECKG"

Call "PIPATASK" in .02 seconds

End job

PIPATASK LENGTHOT = LENGTHOT - 1

If LENGTHOT > 0:

  Call "PIPATASK" in 5.12 seconds

Establish "PIPJOBB" (pr20)

End task

PIPJOBB If NDXCTR = 0, perform "EARTHR*"

If LENGTHOT > 0, end job.

RESULTCT = 5

Perform "CHECKG"

If $DATAPL+1_{dp} < 0$, $DATAPL+4_{dp} = - DATAPL+4_{dp}$

If $DATAPL+1_{dp} = 0$, perform "CCSHOLE"

$DATAPL+4_{dp} = DATAPL+4_{dp} - DATAPL+0_{dp}$

$TS = DATAPL+6_{dp} - DATAPL+2_{dp}$

If TS < 0:

  $TS = TS + 2^{28}$ (the 2^{28} corresponds to 2^{23} centiseconds

$DSPTEM2_{dp}$ = K:dc585 $DATAPL+4_{dp}$ / TS (with forced sign agreement)

If NDXCTR > 0:

  THETAD = 0

  Perform "IMUCOARS"

  Perform "IMUSTALL"

  If ISSGOOD = 0, proceed to "SOMERR2"

Perform "SHOW"

LENGTHOT = 3990

TS = POSITON - 2

$DRIFTT_{sp} = - SOUTHDR_{TS}$

If PIPINDEX > 0:

  $ERCOMP_x = ERCOMP_x$ + K:bt5

  $ERCOMP_y = ERCOMP_y$ - K:bt5

If PIPINDEX = 0:

  $ERCOMP_y = ERCOMP_y$ - K:bt5

  $ERCOMP_z = ERCOMP_z$ + K:bt5

Perform "EARTHR*"

$ERVECTOR_x = 0$

$TORQNDX = (1 - 2^{-28})$

$LOSVEC = CDU_x$

Proceed to "ESTIMS"

VALMIS DSPTEM2 +1 = $DRIFTO_{sp}$

DSPTEM2 = 0

Perform "SHOW"

ENDTEST1 Switch FLAGWRD0 bit 8 (IMUSE) to 0

MODREG = -0

Establish "DSPMMJOB" (pr30)

Proceed to "ENDEXT"

CHECKG QPLACE = return address

Inhibit interrupts

ZERONDX = - $PIPA_{PIPINDEX}$

If $PIPA_{PIPINDEX}$ + ZERONDX = 0:

Release interrupt inhibit

Check for new job and perform it if required; if a new job is performed, proceed to 2nd line of "CHECKG"

If a new job is not performed:

Inhibit interrupts

Proceed to 4th line of "CHECKG"

$DATAPL_{RESULTCT}$ = $PIPA_{PIPINDEX}$

$DATAPL_{RESULTCT + 1}$ = (Channel 3, Channel 4) (sample with special precaution to assure that the two halves are consistent)

Release interrupt inhibit

Proceed to address specified by QPLACE

```
SHOW        DSPTEM2 +2 = POSITON

            Proceed to "GOFLASH" with TS = K:V06N98
                (if terminate, proceed to "ENDTEST1"; if proceed,
                 proceed below; if other response, proceed to "SHOW")

            Return

EARTHR*     TSt = TIMENOW

            TSt1 = TSt - TStmark

            If TSt1 < 0:

                TSt1 = TSt1 + 2^28            centi-seconds

            ERCOMP = ERCOMP + [XSM] TSt1 ERVECTOR

            TStmark = TSt

            TS = "ERCOMP"

            Perform "IMUPULSE"

            Perform "IMUSTALL"

            If ISSGOOD = 0, proceed to "SOMERR2"

            Return

ALLOOP      If OVFLOWCK > 0, end task

            TS = ALTIM

            If TS = +0:

                ALTIMS = +0

                ALTIM = -0

            If TS = -0:

                ALTIM = +0

            If TS < 0:             (should not be positive)

                ALTIM = -( |ALTIM| - 1)

            If GEOCOMPS - 1 = 0  or  LENGTHOT > 0:

                Call "ALLOOP" in 1SECXT centi-seconds
```

$\underline{D}ELV = \underline{P}IPA$ (most significant half only)

$\underline{P}IPA = \underline{0}$

Establish "ALFLT" (pr20)

End task

ALFLT If GEOCOMPS > 0, perform "1/PIPA"

$\underline{T}S = [XSM]^T \underline{D}ELV$

$DPIPAY = - TS_y$

$DPIPAZ = TS_z$

If GEOCOMPS > 0, proceed to "PERFERAS"

If ALTIMS ≥ 0:

$TS = 144 - ALXIS$

$ALTIM = ALFDK_{TS}$

$ALTIMS = ALFDK_{TS + 1}$

$ALDK = ALFDK_{TS + 2}$

$ALDK_2 = ALFDK_{TS + 4}$

$ALDK_4 = ALFDK_{TS + 6}$

$ALDK_6 = ALFDK_{TS + 8}$

$ALDK_8 = ALFDK_{TS + 10}$

$ALXIS = ALXIS - 12$

INTY = INTY - K:pipasc DPIPAY

$DELM_y$ = K:vesc $VLAUN_y$ - INTY

INTZ = INTZ - K:pipasc DPIPAZ

$DELM_z$ = K:vesc $VLAUN_z$ - INTZ

ALK = ALDK ALK

$ALK_2 = ALDK_2\ ALK_2$

$INTY = INTY + ALK\ DELM_y$

$ALK_4 = ALK_4 + ALDK_4$

$ANGX = ANGX + 4\ ALK_4\ DELM_y$

$VLAUN_y = VLAUN_y + K{:}ask0\ DELM_y$

$ANGZ = ANGZ + ALK_2\ DELM_y$

$ALK_6 = ALK_6 + ALDK_6$

$DRIFTO = DRIFTO + 4\ ALK_6\ DELM_y$

$ACCWD_y = ACCWD_y + K{:}ask2\ DELM_y$

$INTZ = INTZ + ALK\ DELM_z$

$ALK_8 = ALK_8 + ALDK_8$

$DRIFTI = DRIFTI + 4\ ALK_8\ DELM_z$

$VLAUN_z = VLAUN_z + K{:}ask0\ DELM_z$

$ANGY = ANGY + ALK_2\ DELM_z$

$ACCWD_z = ACCWD_z + K{:}ask2\ DELM_z$

$\underline{T}S = [TRANSM1]\ (POSNV_y, VLAUN_y, ACCWD_y)$

$(POSNV_y, VLAUN_y, ACCWD_y) = \underline{T}S$

$\underline{T}S = [TRANSM1]\ (POSNV_z, VLAUN_z, ACCWD_z)$

$(POSNV_z, VLAUN_z, ACCWD_z) = \underline{T}S$

SNANGi = sin(K:georgj ANGi) (i = x, y, z)

CSANGi = cos(K:georgj ANGi) (i = x, y, z)

PERFERAS Proceed to erasable memory cell 3400 (E7, 1400)

This is the point where the program apparently returns from erasable memory.

If LENGTHOT > 0:

LENGTHOT = LENGTHOT - 1

Proceed to "SLEEPIE"

If TORQNDX > 0, $LOSVEC_1 = CDU_x$

OGC = [XSM] (-K:georgj)(ANGX, ANGY, ANGZ)

TS = "OGC"

Perform "IMUPULSE"

Perform "IMUSTALL"

If ISSGOOD = 0, proceed to "SOMERR2"

If TORQNDX > 0, proceed to "VALMIS"

ERVECTOR = K:omegms (sinLATITUDE, -cosLATITUDE, 0)

TStmark = TIMENOW

ERCOMP = 0

Proceed to "TORQUE"

SOMEERRR OVFLOWCK = 1

Perform "ALARM" with TS = 01600_8

Proceed to "ENDTEST1"

SOMERR2 Perform "ALARM" with TS = 01601_8

Switch FLAGWRDO bit 8 (IMUSE) to 0

End job

1SECXT: Single precision quantity, scale factor B14, units centi-seconds, giving required period of computations for "ALLOOP".

A: See MATX section.

$ACCWD_y$, $ACCWD_z$: Double precision value of horizontal acceleration of launch vehicle (due to sway) in north-south and east-west directions respectively, scale factor B9, units cm/sec^2.

ALDK, $ALDK_2$: Set of double precision buffer cells used to contain the values of the time constants for the erection angles (PIPA outputs and east axis leveling angle), scale factor B0, as read from the ALFDK table set.

$ALDK_4$, $ALDK_6$, $ALDK_8$: Set of double precision buffer cells used to contain the values of the slopes of the gains for azimuth angle, vertical drift, and north-south drift respectively; scale factor B0, read from ALFDK table set.

$ALFDK_i$: Table of erasable memory quantities used in "ALFLT" to update values of parameters to be used for filtering in gyro drift computations. The table consists of five double precision constants, one single precision constant (the setting for ALTIM), and a reset value of ALTIMS (which could be e.g. -1 for all tables). Values must be initialized by an erasable memory load (with the first value at "ALFDK", with settings for ALTIM, ALTIMS, ALDK, $ALDK_2$, $ALDK_4$, $ALDK_6$, and $ALDK_8$ stored in that order (first two single precision, remainder double precision). Scale factor of first two assumed B14, and the remainder assumed B0, in this writeup.

ALK, ALK_2, ALK_4, ALK_6, ALK_8: Values of gains updated each cycle in gyro drift determination computations. ALK and ALK_2 are initialized to non-zero values in "ESTIMS" and multiplied by time constants for PIPA outputs and erection angles respectively, with scale factors due to initialization of B0 (ALK) and B2. The others (ALK_4, ALK_6, and ALK_8) are initialized to 0 values in "ESTIMS", and are incremented each cycle to achieve varying gains for azimuth angle, vertical drift, and north-south drift respectively: all are considered to have scale factor B0 (see ALFDK).

ALMCADR: See PGSR section.

ALTIM: Single precision value of time remaining prior to change in filter constants for drift measurements, scale factor B14, units seconds. To cause a set of gains to be used for T seconds, ALTIM is set to-(T - 2).

ALTIMS: Single precision flag cell set to 0 when a gain change should be made (see ALTIM), and then reset (e.g. to -1) when the gain change has been done, scale factor B14.

ALXIS: Single precision cell, scale factor B14, used to control selection of values from $ALFDK_i$ erasable memory table (set to 144 in "ESTIMS").

ANGX, ANGY, ANGZ: Values of determined angle changes about vertical, south, and east axes respectively, scale factor B0, units revolutions: they are azimuth alignment angle, south axis leveling angle, and east axis leveling angle respectively.

AZIMUTH: Double precision erasable memory constant, scale factor B0, units revolutions. It gives the azimuth of the vehicle Z-axis east of north.

CADRSTOR: See DINT section.

CDU: See COOR section.

CMPX1: Single precision cell, scale factor B14, used to set proper contents of index register X1 to permit use of an index loop (X1 is set sucessively to $\pm$ 1) to perform calculations in "ALFLT".

$CSANG_i$ (i = X, Y, Z): Values of cosine of ANGX, ANGY, and ANGZ, scale factor B1, stored in push-down list locations 16D, 18D, and 20D respectively.

CYR, CYL, SR, EDOP: Cycle right, cycle left, shift right and shift right 7 registers.

DATAPL: Set of cells used to retain "prelaunch data", loaded in "CHECKG" with sampled accelerometer value and corresponding value of time information in (channel 3, channel 4) scaling (B23 cs).

$DELM_y$, $DELM_z$: Value of measurement quantity in south and easterly directions used in drift test, scale factor B-2, units radians.

$\underline{DELV}$: See SERV section.

DPIPAY, DPIPAZ: Value of accelerometer output modified for use in gyro parameter calculations. The y axis of this system is south and and the z axis is east (from [XSM]). Scale factor is B14, units accelerometer counts.

DRIFTI: Value of gyro drift measurement output displayed in "TORQUE", scale factor (assumed) B0, units radians, giving the south gyro drift.

DRIFTO: Value of gyro drift measurement output displayed in "VALMIS", scale factor (assumed) B0, units radians, giving the vertical gyro drift.

DRIFTT: Input drift to gyro drift determination routine (to separate east gyro drift from azimuth error), scale factor B0, units radians. It has only its most significant half loaded by calling routines, with the least significant half set to 0.

$DSPTAB_i$ (i = 0 - 10): See DSKY section.

$DSPTAB_{11}$: See INTR section.

DSPTEM1, etc: See DATA section.

E_{ADR}: Contents of single precision erasable memory register whose address is ADR.

EBANK: See MATX section.

ERCOMP: Value of gyro compensation to be sent to gyros, scale factor B21, units pulses (or scale factor B0, units revolutions, since one pulse is 2^{-21} revolution).

ERCOUNT: Single precision count of errors encountered in the erasable memory self-check, scale factor B14, initialized at 0 by a fresh start.

ERESTORE: Single precision storage for the address of the first of two erasable memory cells currently being tested by the "ERASCHK" routine. Set to +0 when the "ERASCHK" is complete or not functioning.

ERVECTOR: Earth rotation vector initialized in "ESTIMS", scale factor B1, units gyro pulses / centi-second.

FCADR: See MATX section.

GCOMP: Value of required gyro compensation command, computed with a scale factor B14, but used in "IMUPULSE" with a scale factor B21 (or, alternatively, with a scale factor B0 revolutions rather than B21 gyro pulses, since there are 2^{21} gyro pulses / revolution).

GCOMPSW: Single precision control cell used to bypass the performance of "1/PIPA" and "NBDONLY" if it is negative.

GEOCOMPS: Single precision control cell which when positive will cause the calculations performed in "ALFLT" to be skipped and control transferred to the erasable memory programs. Normally set to 0.

IMODES30: See IMUC section.

IMODES33: See INTR section.

INTY, INTZ: Value of filtered accelerometer output (corrected for vehicle sway etc.) used in gyro drift test, scale factor B-2, units radians. Could also be considered to be "south" and "east" velocity increments expressed in units of g's (see K:pipasc).

ISSGOOD: See IMUC section.

K:ask0: Constant, program notation "ALSK", scale factor B12, stored as $05427\ 12577_8$, equation value 709.833965. Value corresponds to $0.72402338 \times 980.402 \times 2^{-12}$ where first term is wind-induced sway velocity gain, second converts DELM to units of cm/sec (i.e. units of VLAUN), and third term is scale factor.

K:ask2: Constant, program notation "ALSK +2", scale factor B12, stored as $77567\ 44202_8$, equation value -34.2167470. Value corresponds to $0.03490074 \times (-1) \times 980.402 \times 2^{-12}$, where first term is wind-induced accelerometer gain, second is an equation factor, third converts to units of cm/sec^2, and fourth term is scale factor.

K:bt5: Constant, program notation "BIT5", scale factor B0, units revolutions, stored as 00020_8, equation value 0.00098. Value is 2^{-10} revolution corresponding to about $360/1024 = 0.35^\circ$, serving to offset platform to account for accelerometer dead zones. Could also be considered to have value of 2^{11} gyro torquing pulses; there are 2^{21} pulses/rev.

K:dc585: Constant, program notation "DEC585", scale factor B9, stored as 06200_8, equation value 100.0. The 100 corresponds to 1.0×100, where first term is accelerometer nominal scale factor (cm/sec per count) and second converts denominator in "PIPJOBB" from units of centi-seconds to seconds. Result has scale factor B14, units cm/sec^2 (measured gravity, with integral part in R1 of N98 and fractional part in R2).

K:georgj: Constant, program notation "GEORGEJ", scale factor B-2, stored as $24276\ 14066_8$, equation value 0.159154942. Value corresponds to $(1/2\pi) \times 2^2$, to convert between radians and revolutions (the interpretive language trig functions require angle measurements in revs).

K:omegms: Constant, program notation "OMEG/MS", scale factor B0, units gyro pulses/centi-second, stored as $07623\ 26552_8$, equation value 0.243390478. Value corresponds approximately to $(1/86164.0932) \times 10^{-2} \times 2^{21}$, where first term is earth rotation period in seconds (used to derive constant), second converts to centi-seconds, and third is number of gyro torquing pulses in one revolution.

K:pipasc: Constant, program notation "PIPASC", scale factor B-7, stored as $04133\ 02265_8$, equation value 0.001019989. Value corresponds to $1.0 \times (1/980.402) \times 2^7$ where first term is nominal accelerometer scale factor (cm/sec per count), second is normalization factor (acceleration due to gravity), and third is scale factor. For convenience in description, a fourth factor of "1/second" has been assumed reflected in this constant, giving for units of result (in INTY etc.) radians.

K:soupy0: Constant, program notation "SOUPLY", scale factor B0, stored as $35730\ 00035_8$, equation value 0.935058704. Used in "ESTIMS" to initialize ALK.

K:soupy2: Constant, program notation "SOUPLY +2", scale factor B2, stored as $10317\ 17550_8$, equation value 1.05065691. Used in "ESTIMS" to initialize ALK_2.

K:V06N98: See list of major variables.

K:vesc: Constant, program notation "VELSC", scale factor B-9, stored as $57223\ 66451_8$, equation value -0.001019989. Value corresponds to $(-1) \times (1/980.402) \times 2^9$, where first term is an equation factor, second converts for acceleration due to gravity, and third is scale factor.

L: See MATX section.

LATITUDE: Erasable memory (double precision) constant, with scale factor B0, units revolutions. It gives the local vertical astronomical latitude of the pad.

LENGTHOT: Single precision cell, scale factor B14, used to contain time duration information. It is loaded in "PIPACHK" and "PIPJOBB" and decremented in "PIPATASK" and "PERFERAS".

LOSVEC, $LOSVEC_1$: Single precision cells, scale factor B-1, units revolutions, used to contain the value of CDU_x in "PIPJOBB" and when the program returns from erasable memory, for use in DSKY monitoring of performance (by an address-to-be-specified noun).

MODREG: See DATA section.

MONSAVE1: See DATA section.

$MPAC_2$: See DINT section.

NDXCTR: Single precision cell, scale factor B14, initialized to 0 in "REDO" and incremented to 1 for a "gimbal lock" return from "CALCGA" (angle of 60 degrees or more).

NEWJOB: See MATX section.

NOUT: See INTR section.

OGC: See COOR section.

OVFLOWCK: Single precision flag which will terminate the IMU performance test if set. It is initialized to zero in "REDO" and set in "SOMEERRR" to indicate overflow has occurred somewhere in the erasable memory program calculations.

PERFDLAY: Communication cell with routine calling "GOESTIMS", apparently not loaded by program control. Scale factor B28, units centi-seconds. (the "LONGCALL" entrance to the waitlist system is used).

PIPA: See IMUC section.

PIPINDEX: Single precision cell, scale factor B14, used to select accelerometer axis under test (0,1, and 2 for X, Y, Z respectively). Must be loaded manually.

POSITON: Single precision cell, scale factor B14, used for indexing and display purposes (used in previous programs to select desired stable member orientation from fixed memory information). Must be loaded manually (inputs in "SHOW" do not change it.).

$POSNV_y$, $POSNV_z$: Values of horizontal displacement of launch vehicle in south and east directions respectively, assumed scale factor B9, units cm (assumed since scaling of [TRANSM1] elements not known, but treated as B1)

QPLACE: Single precision cell used to retain return address information.

RADMODES: See RADR section.

RESULTCT: Single precision cell, scale factor B14, used to select the proper storage locations in "CHECKG"; it is set to 1 at the start of the accelerometer sampling interval and to 5 at its end, to cause storage in appropriate DATAPL locations.

SCOUNT, $SCOUNT_1$: Single precision counters, scale factor B14, used to count cycles through the self-check and erasable memory test routines, respectively.

SELFRET: Single precision storage for the address of the current position in the self-check routine, for return after other jobs are completed.

SFAIL: Single precision address in the self-check routine where an error was detected.

SKEEP1: Single precision storage for the branching address in the self-test routine, or for an octal bank sum (the sum of the contents of all the cells in a bank of fixed memory), accumulated with end around carry of a + or - overflow (+1 for + overflow; -1 for - overflow). (Example of end around carry of overflow: $25701_8 + 32405_8 = 20307_8$)

SKEEP2: Single precision cell used in the bank sum display as temporary storage for the bank number. Used in the erasable-memory self-check to indicate whether the bank being checked is a normal bank or the unswitched bank. Used in the fixed-memory self-check as temporary storage for the value in the cell being checked.

SKEEP3: Single precision cell used as temporary storage for the "bugger word" during a bank sum display. (The bugger word is the last word in a fixed memory bank which adjusts the sum to make it equivalent to the bank number.) Used in the erasable-memory self-check as storage for the last address in the bank to be checked. Used in the fixed-memory self-check as storage for the address of the next fixed-memory cell whose contents are to be added to the bank sum.

SKEEP4: Single precision storage for the value of the EBANK in the erasable-memory self-test routines, scaled B6 and expressed as an octal number. Single precision storage for the value of the FBANK number in the fixed-memory self-test routines, scaled B4 and expressed as an octal number with the SUPERBNK setting in bits 7-5, the 64ths octal digit.

SKEEP5: Single precision storage for the contents of the first of two consecutive E-memory cells being checked; or single precision index indication (by being set to +0 and then to -1) that two consecutive fixed-memory cells contain their own addresses (TC SELF), thus signifying that the remainder of a bank contains no information.

SKEEP6: Single precision storage for the contents of the second of two consecutive E-memory cells being checked; or a single precision flag set to -0 to indicate a standard fixed-memory check or 1 to indicate a verb 91 bank sum display.

SKEEP7: Single precision address of the first of two consecutive E-memory cells being checked; or a single precision index counted down to signal banks 2 and 3 so that they may be addressed directly instead of through the FBANK setting.

SMODE: Single precision index set equal to +0 to stop the LGC self-test; set to -0, ± 1, ± 2, ± 3, ± 6, ± 7, or ± 10_8 to cause the self-test routine to alternate between "ERASCHK" and "ROPECHK"; set to ± 4 to cause the self-test routine to perform only "ERASCHK"; and set ± 5 to cause the self-test routine to perform only "ROPECHK". If an error is encountered, the self-test routine will return to idle unless SMODE is negative.

$SNANG_i$: (i = X,Y,Z), value of sine of ANGX, ANGY, and ANGZ, scale factor B-2, stored in push-down list locations 10D, 12D, and 14D respectively.

$SOUTHDR_i$: Indexed cell used in "TORQUE" to retain the value of $DRIFTI_{sp}$ for subsequent initialization of $DRIFTT_{sp}$ in "PIPJOBB", for proper initialization (e.g. to 1) and subsequent incrementing (e.g. to 2) of POSITON.

THETAD: See ATTM section.

TIMENOW: See EXVB section.

TORQNDX: Quantity set to 0 if no torquing is done and to 1 (scale factor B0) if torquing is to be done via "EARTHR*" at the start of "SLEEPIE".

[TRANSM1]: Transformation matrix used as a sway transition matrix, contained in erasable memory (must be initialized to values as part of an erasable memory load before running test). Assumed scaling in this writeup for all elements is B1 (after being used to perform multiplication, a left shift of 1 is done).

TStmark: Value of time when previous earth-rate compensation was made, scale factor B28, units centi-seconds.

$VLAUN_y$, $VLAUN_z$: Value of horizontal velocity of launch vehicle (due to sway) in north-south and east-west directions respectively, scale factor B9, units cm/sec.

WANGI: Value of (-cos LATITUDE) loaded in "REDO", scale factor B0.

WANGO: Value of (sin LATITUDE) loaded in "REDO", scale factor B0.

[XNB] : See COOR section.

[XSM] : See COOR section.

ZERONDX: Single precision cell, used as an input parameter to an erasable memory zeroing routine, (not shown in this writeup). It is loaded in "CHECKG" with accelerometer information whenever checks for accelerometer output are made after an interruption.

P34 Switch FLAGWRD2 bit 5 (AVFLAG) to 1

Skip next step

P74 Switch FLAGWRD2 bit 5 (AVFLAG) to 0

Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 1

Proceed to "GOFLASH" with TS = K:V06N37 (TTPI)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

CENTANG = K:130DEG

NN = 0 (most significant half only)

Perform "DISPLAYE"

Switch FLAGWRD2 bit 7 (ETPIFLAG) to 0

TIG = TTPI

If ELEV ≠ 0, switch FLAGWRD2 bit 7 (ETPIFLAG) to 1

Perform "SELECTMU"

Perform "VN1645"

P34/P74C Switch FLAGWRD7 bit 15 (ITSWICH) to 1

If FLAGWRD2 bit 7 (ETPIFLAG) = 0:
Switch FLAGWRD7 bit 15 (ITSWICH) to 0

NOMTPI = 0

INTLOOP TDEC1 = TTPI + NOMTPI

Perform "PRECSET" (get $\underline{R}$ACT3, $\underline{V}$ACT3, $\underline{R}$PASS3, $\underline{V}$PASS3)

Perform "S33/34.1" (get ELEV or TPI time)

If TSnosol ≠ 0: (no solution)

(If TSnosol ≠ 0)

Perform "ALARM" with TS = 00611_8

Proceed to "GOFLASH" with TS = K:V05N09
(If terminate, proceed to "GOTOPOOH"; if proceed, proceed to the second step of "P74"; other response, proceed to previous step.)

If FLAGWRD7 bit 15 (ITSWICH) = 1:

Switch FLAGWRD7 bit 15 (ITSWICH) to 0

Proceed to "INTLOOP"

If FLAGWRD2 bit 7 (ETPIFLAG) = 0, perform "DISPLAYE"

If FLAGWRD2 bit 7 (ETPIFLAG) = 1:

Proceed to "GOFLASH" with TS = K:V06N37 (TTPI)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

CSTH = cosCENTANG

SNTH = sinCENTANG

RVEC = RPASS3 (both shifted left two if necessary to make the scaling B29 and B7 (earth) or B27 and B5)

VVEC = VPASS3

Switch FLAGWRD7 bit 9 (RVSW) to 1

Perform "TIMETHET"

INTIME = TTPI

TPASS4 = TTPI + T

Perform "S34/35.2" (get DELVEET3 and DELVLVC)

DELVTPI = |DELVEET3|

DELVTPF = |VPASS4 - VTPRIME|

RVEC = RACT3

VVEC = VIPRIME

Perform "PERIAPO"

POSTTPI = TShp

TIG = TTPI

Proceed to "GOFLASH" with TS = K:V06N58 (POSTTPI, DELVTPI, DELVTPF)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Perform "S34/35.5" (get DELVSIN)

Perform "VN1645" (astronaut recycle or finalize options)

Proceed to "P34/P74C"

DISPLAYE NORMEX = return address

Proceed to "GOFLASH" with TS = K:V06N55 (NN, ELEV, and CENTANG)

(If terminate, proceed to "GOTOPOOH"; if proceed, return via NORMEX; other response, repeat this step.)

S34/35.5 SUBEXIT = return address

If FLAGWRD2 bit 6 (FINALFLG) = 0:

Switch FLAGWRD1 bit 7 (UPDATFLG) to 1

Perform "S34/35.4"

Proceed to "GOFLASH" with TS = K:V06N59 (DVLOS)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Return via SUBEXIT

Switch FLAGWRD6 bit 3 (NTARGFLG) to 0

GDTd2 = DELVLVC

Proceed to "GOFLASH" with TS = K:V06N81 (DELVLVC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

TSsum = 0

i = 5

NTARGCHK TS = $GDTd2_i - DELVLVC_i$ (check for astronaut overwrite of DELVLVC)

TSsum = TSsum + TS

If i > 0:
  i = i - 1
  Proceed to "NTARGCHK"

If TSsum ≠ 0:
  Switch FLAGWRD6 bit 3 (NTARGFLG) to 1
  Perform "S34/35.3"

DELVSIN = DELVEET3

Perform "S34/35.4"

Proceed to "GOFLASH" with TS = K:V06N59 (DVLOS)
  (If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Return via SUBEXIT

P35 Switch FLAGWRD2 bit 5 (AVFLAG) to 1

TK = ATIGINC

Skip next two steps

P75 Switch FLAGWRD2 bit 5 (AVFLAG) to 0

TK = PTIGINC

Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 1

Perform "SELECTMU"

Perform "VN1645"

P35/P75B TSTRT = TIMENOW

TIG = TSTRT + TK

INTIME = TIG

TDEC1 = TIG

Perform "PRECSET"

ULOS = unit(RPASS3 - RACT3)

UNRM = unit(RACT3 * VACT3)

Perform "S34/35.2" (Lambert solution)

Perform "S34/35.5" (get DELVSIN)

Perform "VN1645" (astronaut recycle or finalize)

Proceed to "P35/P75B"

S33/34.1 NORMEX = return address

TITER = -K:posmaxsp

SECMAX = K:MAX250

RAPREC = RACT3

VAPREC = VACT3

RPPREC = RPASS3

VPPREC = VPASS3

ELCALC ULOS = unit(RPASS3 - RACT3)

UNRM = unit(RACT3 * VACT3)

UP = unit(ULOS - (ULOS · unitRACT3) unitRACT3)

TSelev = arccos(UP · ULOS sign(UNRM * RACT3 · UP))

(TSelev is positive, between 0 and 180 degrees - 0 and ½ rev)

If ULOS · RACT3 < 0:

TSelev = K:posmaxdp - TSelev

If FLAGWRD7 bit 15 (ITSWICH) = 0:

    TTPI = TTPI + NOMTPI

    If FLAGWRD2 bit 7 (ETPIFLAG) = 0

        ELEV = TSelev

    TSnosol = 0

    Return via NORMEX

DELELO = DELEL

DELEL = TSelev - ELEV

If |DELEL| < K:ELEPS:

    TSnosol = 0

    Return via NORMEX

If TITER = 1:

    TSnosol = 1

    Return via NORMEX

TITER = TITER - 1

TSrdif = $|\underline{R}PASS3| - |\underline{R}ACT3|$

TS = $(\frac{1}{2}$ - ELEV) signTSrdif

If TS < 0: (desired ELEV impossible)

    TSnosol = TS ($\neq$ 0)

    Return via NORMEX

TScsd = $-\cos(\frac{1}{2}$ - ELEV) $|\underline{R}ACT3| / |\underline{R}PASS3|$

TS = 1 - |TScsd|

If TS < 0:

    TSnosol = TS

    Return via NORMEX

```
TS16 = |RPASS3| unit(UNRM * unitRACT3) · VACT3

TS = unitRPASS3 * VPASS3

TS16 = TS16 - |RACT3| unit(TS * unitRPASS3) · VPASS3

TSs = unitRACT3 * unitRPASS3 · UNRM

TSc = arccos(unitRACT3 · unitRPASS3) signTSs

TSca = (½ - arccosTScsd) signTSrdif + ELEV - ½ + TSc

TSt = K:TWOPI TSca |RACT3||RPASS3| / TS16

If |TSt| ≥ SECMAX, TSt = SECMAX signTSt

If TITER < 0:    (first pass)

     TITER = 14

     DELTEEO = TSt


     DELTEE = TSt

     Proceed to "ADTIME"

If DELEL DELELO < 0:    (solution is surrounded)

     SECMAX = SECMAX / 3

     DELTEEO = - |TSt| signDELTEEO / 2

     DELTEE = DELTEEO

     Proceed to "ADTIME"

If |DELELO| < |DELEL|:

     DELTEEO = -DELTEEO / 2

     DELTEE = 3 DELTEEO

     Proceed to "ADTIME"

DELTEEO = |TSt| signDELTEEO

DELTEE = DELTEEO
```

ADTIME NOMTPI = NOMTPI + DELTEE

Perform "INTINT" with TSv = VAPREC, TSr = RAPREC, TSo = 0,
TSt = NOMTPI, and TSintyp = NOMTPI

RACT3 = RATT

VACT3 = VATT

Perform "INTINT" with TSv = VPPREC, TSr = RPPREC, TSo =0,
TSt = NOMTPI, and TSintyp = NOMTPI

RPASS3 = RATT

VPASS3 = VATT

Proceed to "ELCALC"

S34/35.2 SUBEXIT = return address

TSv = VPASS3, TSr = RPASS3, TSo = INTIME, and TSt = TPASS4

If most significant half of NN = 0:
- TSintyp = K:TWOPI
- Perform "INTINT"

If most significant half of NN ≠ 0:
- TSintyp = 0
- Perform "INTINT"

RTARG = RATT

VPASS4 = VATT

ACTCENT = arccos(unitRACT3 · unitRTARG) sign(unitRACT3 * unitRTARG · UNRM)

If ACTCENT < 0, ACTCENT = K:posmaxdp + ACTCENT

DELLT4 = TPASS4 - INTIME

VTARGTAG = NN

CNANGL = K:EPSFOUR

RINIT = RACT3

VINIT = VACT3

Perform "INITVEL"

$$\text{DELVLVC} = \begin{bmatrix} -\text{unitRACT3} * \text{UNRM} \\ -\text{UNRM} \\ -\text{unitRACT3} \end{bmatrix} \text{DELVEET3}$$

Return via SUBEXIT

S34/35.3 NORMEX = return address

$$\text{DELVEET3} = \text{DELVLVC} \begin{bmatrix} -\text{unitRACT3} * \text{UNRM} \\ -\text{UNRM} \\ -\text{unitRACT3} \end{bmatrix}$$

Perform "INTINT" with TSv = DELVEET3 + VACT3, TSr = RACT3, TSo = TIG, TSt = TPASS4, and TSintyp = K:posmaxdp

RTARG = RATT

$$\text{DVLOS} = \begin{bmatrix} \text{ULOS} \\ -\text{unit}(\text{ULOS} * \text{UNRM}) * \text{ULOS} \\ -\text{unit}(\text{ULOS} * \text{UNRM}) \end{bmatrix} \text{DELVEET3}$$

Return via NORMEX

S34/35.4 NORMEX = return address

$$\text{DVLOS} = \begin{bmatrix} \text{ULOS} \\ -\text{unit}(\text{ULOS} * \text{UNRM}) * \text{ULOS} \\ -\text{unit}(\text{ULOS} * \text{UNRM}) \end{bmatrix} \text{DELVEET3}$$

Return via NORMEX

INITVEL Switch FLAGWRD1 bit 2 (GUESSW) to 1

HAVEGUES NORMEX = return address

RTARG1 = RTARG

If MUDEX $\neq$ 0, rescale RINIT, VINIT, and RTARG1

RTMAG = $|\text{RTARG1}|$

ITCTR = -1

COZY4 = cosCNANGL

R1VEC = RINIT

R2VEC = RTARG1

TDESIRED = DELLT4

UN = unit(unitRINIT * VINIT)

COZY4 = (unitRTARG1 $\cdot$ unitRINIT) + COZY4

Switch FLAGWRD7 bit 10 (NORMSW) to 0

<u>INITVEL2</u> If COZY4 < 0:

Switch FLAGWRD7 bit 10 (NORMSW) to 1

$\underline{R}2VEC = |\underline{R}2VEC|$ unit($\underline{R}2VEC$ - ($\underline{R}2VEC \cdot \underline{U}N$) $\underline{U}N$)

If ITCTR < 0, $\underline{R}TARG1 = \underline{R}2VEC$

X1 = -MUDEX - 2 (-2 for earth sphere, -10 for lunar sphere)

$\underline{T}S$ = unit$\underline{R}$1VEC * unit$\underline{R}$2VEC

TSz = Z component of $\underline{T}S$.

If TSz > 0, X1 = X1 + 10

If TSz ≤ 0, X1 = X1 + 2

If X1 = 0, $\underline{T}S = -\underline{T}S$

TS = ($\underline{T}S$ * unit$\underline{R}$1VEC) · unit $\underline{R}$2VEC

If TS ≥ 0, GEOMSGN = $K{:}MU_0$ (only most significant half of K:MUo is used for setting GEOMSGN)

If TS < 0, GEOMSGN = $-K{:}MU_0$

ITERCTR = 20

Perform "LAMBERT"

Switch FLAGWRD1 bit 2 (GUESSW) to 0

$\underline{V}IPRIME = \underline{V}VEC$

If VTARGTAG = 0, proceed to "INITVEL7"

Perform "INTSTALL"

Switch FLAGWRD0 bit 12 (MOONFLAG) to 0

If MUDEX ≠ 0, switch FLAGWRD0 bit 12 (MOONFLAG) to 1

$\underline{R}1VEC = \underline{R}INIT$

$\underline{R}CV = \underline{R}INIT$

$\underline{V}CV = \underline{V}IPRIME$

TET = INTIME

TDEC1 = INTIME + DELLT4

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0

Perform "INTEGRVS"

VTARGET = VATT

ITCTR = ITCTR + 1

If ITCTR ≠ VTARGTAG:

    R2VEC = R2VEC + RTARG1 - RATT   (bias target vector)

    Proceed to "INITVEL2"

RTARG1 = R2VEC

INITVEL7 DELVEET3 = VIPRIME - VINIT

VTPRIME = VTARGET

If MUDEX ≠ 0, rescale VTPRIME, VIPRIME, DELVEET3, and RTARG1

MUE = $K{:}MUTABLE_{MUDEX}$

MUdA = (MUE) (RdA) / R1

Rescale MUE

MUASTEER = MUE

RTARG = RTARG1

Switch FLAGWRD2 bit 8 (XDELVFLG) to 0

Return via NORMEX

ACTCENT: Double precision central angle between active and passive vehicles, scaled B0 in units of revolutions.

ATIGINC: Double precision time between midcourse burn targeting by the active vehicle and TIG, scaled B28 in units of centiseconds. Part of the erasable load.

CENTANG: Double precision central angle between the passive vehicle's position at TIG and at intercept, scaled B0 in units of revolutions.

CNANGL: Double precision central angle of a cone around -RINIT, scaled B0 in units of revolutions. Target vectors within this cone are projected into the orbital plane of the active vehicle because of the sensitivity of the transfer plane orientation to a change in RTARG when RTARG is close to -RINIT.

COZY4: Value used by "INITVEL" to determine if the original target position vector falls within the cone specified by CNANGL. Scaled B2 and unitless.

CSTH: See CONC section.

DELEL, DELELO: Double precision present and previous increments to ELEV during the TPI-time/elevation-angle iteration, scaled B0 in units of revolutions.

DELLT4: Double precision maneuver transfer time, scaled B28 in units of centiseconds.

DELTEE, DELTEEO: Double precision increment to NOMTPI during the TPI-time/elevation-angle iteration, scaled B28 in units of centiseconds.

DELVEET3: Double precision velocity-to-be-gained vector, scaled B7 in units of meters per centisecond. Calculated by the "INITVEL" routine.

DELVLVC: See TRGX section.

DELVSIN: See TRGX section.

DELVTPF: Double precision magnitude of the velocity to be gained in the final rendezvous maneuvers of the terminal phase, scaled B7 in units of meters per centisecond.

DELVTPI: Double precision magnitude of velocity to be gained, as calculated by P34-P74. Scaled B7 in units of meters per centisecond.

DVLOS: Double precision velocity-to-be-gained vector, scaled B7 in units of meters per centisecond and expressed in "line-of-sight" coordinates. (See "S34/35.3" for definition of "line-of-sight" coordinates.)

ELEV: Double precision elevation angle of the line-of-sight to the passive vehicle; measured from the vector which is perpendicular to the active vehicle position vector, perpendicular to RACT * RPASS, and whose dot product with the active vehicle velocity vector is positive. An angle between 0 and 1 (0 and 360 degrees) scaled B0 in units of revolutions. ELEV is greater than $\frac{1}{2}$ (180 degrees) if the passive vehicle is below the active vehicle's local horizontal. ELEV is an astronaut input in P32-72 and an optional input in P34-74.

GDTd2: A temporary storage location for DELVLVC. If DELVLVC is overwritten by the astronaut, the previous value of DELVLVC will still be in GDTd2, thus making possible a program comparison of the two values, and detection of the astronaut overwrite.

GEOMSGN: Single precision flag to assure that a unit normal vector computed in "GEOM" will have the same relation to the orbital plane of the active vehicle as UN will have when computed in "INITVEL".

INTIME: Double precision time-tag associated with RINIT and VINIT, scaled B28 in units of centiseconds.

ITCTR: Single precision counter measuring the number of iterations through the loop which biased a Lambert target vector to achieve a more accurate estimate of velocity required, based on precision integration of the biased conic solution.

ITERCTR: See CONC section.

K:130DEG: Double precision constant stored as 0.3611111111, scaled B0 in units of revolutions. Equation value: 0.3611111111.

K:ELEPS: Double precision constant stored as $0.27777777 \times 10^{-3}$, scaled B0 in units of revolutions. Equation value: 0.00027777777. (Equivalent to 0.1 degrees.)

K:EPSFOUR: Double precision constant stored as 0.0416666666, scaled B0 in units of revolutions. Equation value: 0.0416666666. (Equivalent to 15 degrees.)

K:MAX250: Double precision constant stored as 25000×2^{-28}, scaled B28 in units of centiseconds. Equation value: 25000.

K:MU_0: See ORBI section.

K:MUTABLE: See CONC section.

K:TWOPI: Double precision constant stored as 6.283185307 x 2^{-4}, scaled B4 and unitless. Equation value: 6.283185307.

MUASTEER: Equal to MUE, but rescaled to B42 (earth) or B36 (moon).

MUdA: Ratio of gravitational constant (MUE) to semi-major axis. Scaled B14 (earth) or B10 (moon).

MUDEX: See CONC section.

MUE: See TRGX section.

NN: See TRGX section.

NOMTPI: Double precision iterative addition to TTPI, scaled B28 in units of centiseconds.

NORMEX: Single precision octal return address storage.

POSTTPI: Double precision perigee altitude, calculated in P34-74, scaled B29 in units of meters.

PTIGINC: Double precision time between midcourse burn targeting by the passive vehicle and TIG, scaled B28 in units of centiseconds. Part of the erasable load.

R1: See CONC section.

R1VEC, R2VEC: See CONC section.

RACT3, VACT3: Double precision position and velocity vectors of the active vehicle prior to a particular burn, scaled B29 and B7 respectively in units of meters and meters per centisecond. Both vectors are also used as temporary storage for intermediate active vehicle position and velocity vectors.

RAPREC, VAPREC, RPPREC, VPPREC: Double precision vector storage for RACT3, VACT3 and RPASS3, VPASS3 as they were at entry to "S33/34.1".

RATT, VATT: See ORBI section.

RCV, VCV: See CONC section.

RdA: See CONC section.

RINIT, VINIT: Double precision active vehicle position and velocity vectors, scaled B29 and B7 respectively in units of meters and meters per centisecond. Rescaled at the beginning of "INITVEL" to B27 and B5 respectively if the CSM is within the moon's sphere of influence.

RPASS3, VPASS3 : Double precision position and velocity vectors of the passive vehicle prior to a particular burn, scaled B29 and B7 in units of meters and meters per centisecond respectively. Both vectors are also used as temporary storage for intermediate passive vehicle position and velocity vectors.

RTARG: Target position vector input to "INITVEL". Scaled B29 in units of meters. Upon exit, "INITVEL" loads RTARG with the biased target position vector, if such a biased vector is calculated.

RTARG1: Value of RTARG used within "INITVEL", scaled B29 (earth) or B27 (moon), in units of meters.

RTMAG: Magnitude of RTARG1, scaled B29 (earth) or B27 (moon), in units of meters.

RVEC, VVEC: See CONC section.

SECMAX: Double precision maximum limit on changes to NOMTPI, scaled B28 in units of centiseconds.

SNTH: See CONC section.

SUBEXIT: Single precision octal return address storage.

T: See CONC section.

TDEC1: See ORBI section.

TDESIRED: See CONC section.

TET: See ORBI section.

TIG: See BURN section.

TIMENOW: See EXVB section.

TITER: Single precision iteration counter.

TK: Double precision time between the initiation of P35-75 and the ignition of a midcourse correction burn, scaled B28 in units of centiseconds. **Program notation is "KT".**

TPASS4: Double precision scheduled time of target intercept, scaled B28 in units of centiseconds.

TSTRT: Double precision time of initiation of P35-75, scaled B28 in units of centiseconds.

TTPI: Double precision time of terminal phase initiation, scaled B28 in units of centiseconds; an astronaut input in P32-72 and P34-74.

ULOS: Double precision unit vector along the line-of-sight vector, scaled B1 and unitless.

UN: Double precision unit vector perpendicular to the active vehicle orbital plane, scaled B1 and unitless.

UNRM: Double precision unit vector perpendicular to the active vehicle orbital plane, scaled B1 and unitless.

UP: A unit vector perpendicular to RACT3 and perpendicular to RACT3 * ULOS, whose dot product with ULOS is positive, scaled B1 and unitless.

VIPRIME: Double precision velocity vector computed by the Lambert routine at the time INTIME. Scaled B7 in units of meters per centisecond.

VPASS4: Double precision velocity vector of the passive vehicle at the time of target intercept. Scaled B7 in units of meters per centisecond.

VTARGET: See CONC section.

VTARGTAG: Single precision cell, scaled B14, which specifies the number of iterations through the LAMBERT/INTEGRVS routines. If VTARGTAG = 0, "LAMBERT" is performed to obtain initial and final velocity vectors, and "INTEGRVS" is not entered. If VTARGTAG > 0, "INTEGRVS" output is used to bias the target position vector in order to obtain a more accurate "LAMBERT" solution.

VTPRIME: Double precision velocity vector, equal to VTARGET, which is calculated in the "INITVEL" routine. Scaled B7 in units of meters per centisecond.

X1: Index register 1.

TRGX

P30 Switch FLAGWRD1 bits 7 (UPDATFLG) and 5 (TRACKFLG) to 1

Proceed to "GOFLASH" with TS = K:V06N33 (TIG)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Proceed to "GOFLASH" with TS = K:V06N81 (DELVLVC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Switch FLAGWRD1 bit 7 (UPDATFLG) to 0

Perform "S30.1"

Switch FLAGWRD1 bit 7 (UPDATFLG) to 1

Switch FLAGWRD2 bit 8 (XDELVFLG) to 1

Proceed to "GOFLASH" with TS = K:V06N42 (HAPO, HPER, DELVSAB)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Switch FLAGWRD2 bit 6 (FINALFLG) to 1

Perform "VN1645"

Go back one step

P32 Switch FLAGWRD2 bit 5 (AVFLAG) to 1

Skip next step

P72 Switch FLAGWRD2 bit 5 (AVFLAG) to 0

CENTANG = 0

Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 1

NN = 0 (least significant half only)

TCSI = 0

VN0611 Proceed to "GOFLASH" with TS = K:V06N11 (TCSI)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

If - (TCSI) < 0:
Proceed to "VN0655"

TDEC1 = TETLEM

Perform "PRECSET"

RVEC = RACT3

VVEC = VACT3

Switch FLAGWRD7 bit 9 (RVSW) to 1

RDESIRED = K:posmaxdp

Perform "TIMERAD" (compute time to apogee)

TCSI = T + TDEC2

Proceed to "VN0611"

VN0655 Proceed to "GOFLASH" with TS = K:V06N55 (NN, ELEV, CENTANG)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

Proceed to "GOFLASH" with TS = K:V06N37 (TTPI)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

TIG = TCSI

Perform "SELECTMU" (switches FINALFLG to 0)

Perform "VN1645" (switches UPDATFLG to 0)

P32/P72B Perform "ADVANCE" (advances LM and CSM vectors to TIG, sets XDELVFLG)

Perform "INTINT" with TSv = VPASS1, TSr = RPASS1, TSo = TCSI, TSt = TTPI, and TSintyp = K:TWOPI

RPASS3 = RATT

VPASS3 = VATT

Proceed to "CSI/A"

P32/P72C If FLAGWRD2 bit 6 (FINALFLG) = 0:
Switch FLAGWRD1 bit 7 (UPDATFLG) to 1

Proceed to "P32/P72E"

P32/P72E If T1TOT2 ≥ K:60MIN:

T1TOT2 = T1TOT2 - K:60MIN

Proceed to "P32/P72E"

P32/P72F If T2TOT3 ≥ K:60MIN:

T2TOT3 = T2TOT3 - K:60MIN

Proceed to "P32/P72F"

Proceed to "GOFLASH" with TS = K:V06N75 (DIFFALT, T1TOT2, T2TOT3)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

$$[LVCMAT] = \begin{bmatrix} -\ unit\underline{R}ACT1 * \underline{U}P1 \\ -\ \underline{U}P1 \\ -\ unit\underline{R}ACT1 \end{bmatrix}$$

$\underline{D}ELVLVC = [LVCMAT]\ \underline{D}ELVEET1$

Proceed to "GOFLASH" with TS = K:V06N81 ($\underline{D}$ELVLVC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

$\underline{D}ELVSIN = [LVCMAT]^T\ \underline{D}ELVLVC$

$DELVSAB = |\underline{D}ELVSIN|$

$\underline{D}ELVEET1 = \underline{D}ELVSIN$

$$[LVCMAT] = \begin{bmatrix} -\ unit\underline{R}ACT2 * \underline{U}P1 \\ -\ \underline{U}P1 \\ -\ unit\underline{R}ACT2 \end{bmatrix}$$ ($\underline{R}$ACT1 set equal to $\underline{R}$ACT2)

$\underline{D}ELVLVC = [LVCMAT]\ \underline{D}ELVEET2$

Proceed to "GOFLASH" with TS = K:V06N82 ($\underline{D}$ELVLVC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

TTPIO = TTPI

Perform "VN1645" (astronaut recycle or finalize)

Proceed to "P32/P72B"

P33 Switch FLAGWRD2 bit 5 (AVFLAG) to 1

Skip next step

P73 Switch FLAGWRD2 bit 5 (AVFLAG) to 0

Switch FLAGWRD1 bits 5 (TRACKFLG) and 7 (UPDATFLG) to 1

Proceed to "GOFLASH" with TS = K:V06N13 (TCDH)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

TTPI = TTPIO

TIG = TCDH

Perform "SELECTMU" (switches FINALFLG to 0)

Perform "VN1645" (switches UPDATFLG to 0)

P33/P73B Perform "ADVANCE" (LM and CSM vectors to TIG, set XDELVFLG)

Perform "CDHMVR"

Perform "INTINT" with TSv = VACT3, TSr = RACT2, TSo = TCDH,
TSt = TTPI, and TSintyp = 0

RACT3 = RATT

VACT3 = VATT

Perform "INTINT" with TSv = VPASS2, TSr = RPASS2, TSo = TCDH,
TSt = TTPI, and TSintyp = 0

RPASS3 = RATT

VPASS3 = VATT

Switch FLAGWRD7 bit 15 (ITSWICH) to 1

NOMTPI = 0

Perform "S33/34.1" (get transfer time to TPI)

If TSnosol ≠ 0: (no solution)

Perform "ALARM" with TS = 00611_8

(If no solution)

Proceed to "GOFLASH" with TS = K:V05N09
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, proceed to second step of "P73".)

NOMTPI = 0

If FLAGWRD2 bit 6 (FINALFLG) = 0:

Switch FLAGWRD1 bit 7 (UPDATFLG) to 1

TTPI = TTPI + NOMTPI

T1TOT2 = TTPI - TCDH

P33/P73E If T1TOT2 $\geq$ K:60MIN:

T1TOT2 = T1TOT2 - K:60MIN

Proceed to "P33/P73E"

T2TOT3 = TTPI - TTPIO

P33/P73F If |T2TOT3| $\geq$ K:60MIN:

T2TOT3 = T2TOT3 - K:60MIN signT2TOT3

Proceed to "P33/P73F"

Proceed to "GOFLASH" with TS = K:V06N75 (DIFFALT, T1TOT2, T2TOT3)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

$$[\text{LVCMAT}] = \begin{bmatrix} -\text{unit}\underline{\text{R}}\text{ACT1} * \underline{\text{U}}\text{P1} \\ -\underline{\text{U}}\text{P1} \\ -\text{unit}\underline{\text{R}}\text{ACT1} \end{bmatrix}$$

$\underline{\text{D}}\text{ELVLVC} = [\text{LVCMAT}]\ \underline{\text{D}}\text{ELVEET2}$

Proceed to "GOFLASH" with TS = K:V06N81 ($\underline{\text{D}}$ELVLVC)
(If terminate, proceed to "GOTOPOOH"; if proceed, continue at next step; other response, repeat this step.)

$\underline{\text{D}}\text{ELVSIN} = [\text{LVCMAT}]^T\ \underline{\text{D}}\text{ELVLVC}$

$\text{DELVSAB} = |\underline{\text{D}}\text{ELVSIN}|$

$\underline{\text{D}}\text{ELVEET2} = \underline{\text{D}}\text{ELVSIN}$

Perform "VN1645" (astronaut recycle or finalize)

Proceed to "P33/P73B"

SELECTMU MUDEX = 0

If FLAGWRD8 bit 12 (CMOONFLG) = 1, MUDEX = 8

i = MUDEX + 6

RP1dMU = K:MUTABLE$_i$

MUE = K:MUTABLE$_{MUDEX}$

If FLAGWRD8 bit 12 (CMOONFLG) = 1, rescale MUE (sr6)

Switch FLAGWRD2 bit 6 (FINALFLG) to 0

Return

VN1645 SUBEXIT = return address

pMGA = K:DPmp01

If FLAGWRD2 bit 6 (FINALFLG) = 1:

  pMGA = 2 K:DPmp01

  If FLAGWRD3 bit 13 (REFSMFLG) = 1 and MODREG < 64:

    Perform "GET+MGA" with TS = DELVSIN

Perform "COMPTGO"

Delay 1 second (via "DELAYJOB")

Proceed to "GOFLASH" with TS = K:V16N45 (TRKMKCNT, TTOGO, pMGA)
(If terminate, contine at next step; if proceed, proceed to "N45PROC"; other response, proceed to "CLUPDATE".)

DISPDEX = Z (to stop computation of TTOGO)

Proceed to "GOTOPOOH"

N45PROC If FLAGWRD2 bit 6 (FINALFLG) = 1:

  DISPDEX = Z

  Proceed to "GOTOPOOH"

Switch FLAGWRD2 bit 6 (FINALFLG) to 1

CLUPDATE DISPDEX = Z

Switch FLAGWRD1 bit 7 (UPDATFLG) to 0

Return via SUBEXIT

S30.1 QTEMP = return address

TDEC1 = TIG

Perform "LEMPREC"

RTIG = RATT

VTIG = VATT

$$\lfloor LVCMAT \rfloor = \begin{bmatrix} -\text{unit}(\underline{VTIG} * \underline{RTIG}) * \text{unit}\underline{RTIG} \\ \text{unit}(\underline{VTIG} * \underline{RTIG}) \\ -\text{unit}\underline{RTIG} \end{bmatrix}$$

DELVSIN = DELVLVC [LVCMAT]

DELVSAB = |DELVSIN|

TSr = RTIG

TSv = VTIG + DELVSIN

Perform "PERIAPO1"

HPER = TShp (pericenter altitude)

If HPER ≥ K:MAXNM:

HPER = K:MAXNM

HAPO = TSha (apocenter altitude)

If HAPO ≥ K:MAXNM:

HAPO = K:MAXNM

Return via QTEMP

ADVANCE SUBEXIT = return address

TDEC1 = TIG

Perform "PRECSET"

Switch FLAGWRD2 bit 8 (XDELVFLG) to 1

VPASS2 = VPASS3

VPASS1 = VPASS3

RPASS2 = RPASS3

RPASS1 = RPASS3

UP1 = unit (unitRPASS1 * VPASS1)

RTIG = RACT3

RACT2 = |RACT3| unit(RACT3 - (RACT3 · UP1) UP1)

RACT1 = RACT2

VTIG = VACT3

VACT2 = |VACT3| unit(VACT3 - (VACT3 · UP1) UP1)

VACT1 = VACT2

Return via SUBEXIT

PRECSET NORMEX = return address

TDEC2 = TDEC1

Perform "LEMPREC"

If FLAGWRD2 bit 5 (AVFLAG) = 1:

    RACT3 = RATT

    VACT3 = VATT

If FLAGWRD2 bit 5 (AVFLAG) = 0:

    RPASS3 = RATT

    VPASS3 = VATT

TDEC1 = TDEC2

Perform "CSMPREC"

If FLAGWRD2 bit 5 (AVFLAG) = 1:

RPASS3 = RATT

VPASS3 = VATT

If FLAGWRD2 bit 5 (AVFLAG) = 0:

RACT3 = RATT

VACT3 = VATT

Return via NORMEX

INTINT RTRN = return address

Perform "INTSTALL"

Switch FLAGWRD3 bit 4 (INTYPFLG) to 0

If TSintyp $\neq$ 0, Switch FLAGWRD3 bit 4 to 1 (conic integration)

TDEC1 = TSt

Switch FLAGWRD0 bit 12 (MOONFLAG) to 1

If FLAGWRD8 bit 12 (CMOONFLG) = 0, Switch FLAGWRD0 bit 12 to 0

TET = TSo

RCV = TSr

VCV = TSv

Perform "INTEGRVS"

Return via RTRN

CDHMVR SUBEXIT = return address

UNVEC = unitRACT2

CSTH = unitRPASS2 · UNVEC

TS = - RACT2 * RPASS2 · UP1

SNTH = $\sqrt{1 - CSTH^2}$ signTS

VVEC = VPASS2

RVEC = RPASS2

Switch FLAGWRD7 bit 9 (RVSW) to 0

Perform "TIMETHET"

DIFFALT = |TSr| - |RACT2|

SMALLA = R1 / RdA

$$TSa = (TSv \cdot UNVEC)\left[\frac{SMALLA}{SMALLA - DIFFALT}\right]^{3/2}$$

$$TSb = \sqrt{\frac{2\ MUE}{|RACT2|} - \frac{MUE}{SMALLA - DIFFALT} - TSa^2}$$

VACT3 = TSa UNVEC + TSb unit(UP1 * UNVEC)

DELVEET2 = VACT3 - VACT2

Return via SUBEXIT

CSI/A Switch FLAGWRD6 bits 12 (S32.1F3B) and 14 (S32.1F2) to 1

Switch FLAGWRD6 bits 13 (S32.1F3A) and 15 (S32.1F1) to 0

LOOPCT = 0

CSIALRM = 0

CSI/B TS = |RACT1| (1 + |RACT1| / |RPASS3|)

$TS = \sqrt{2\ MUE / TS}$

DELVCSI = TS - unit(UP1 * unitRACT1) · VACT1

DELDV = K:INITST

CSI/B1 LOOPCT = LOOPCT + 1

If LOOPCT ≥ K:LOOPMX, proceed to "SCNDSOL" with TSsp = 6

CSI/B2 If |DELVCSI| ≥ K:DVMAX1:

If FLAGWRD6 bit 15 (S32.1F1) = 1 or if FLAGWRD6 bits 12 (S32.1F3B) and 13 (S32.1F3A) both equal 1, proceed to "SCNDSOL" with TSsp = 7

Switch FLAGWRD6 bit 15 (S32.1F1) to 1

(If |DELVCSI| ≥ K:DVMAX1)

DELVCSI = K:DVMAX2 signDELVCSI

DELVEET1 = DELVCSI unit(UP1 * unitRACT1)

VACT4 = VACT1 + DELVEET1 (ignoring overflow if any)

VVEC = VACT4

RVEC = RACT1

Switch FLAGWRD7 bit 9 (RVSW) to 1

SNTH = K:SN359+

CSTH = K:CS359+

Perform "TIMETHET"

HAFPA1 = T/ 2

Perform "PERIAPO"

POSTCSI = TShp

If CENTANG ≠ 0, proceed to "CIRCL"

If ECC < K:ONETHTH, proceed to "CIRCL"

If (|RACT1 · VACT4| / R1) < K:NICKELDP, proceed to "CIRCL"

TScs = P - 1

TS = $\sqrt{P \ R1}$ RT1dMU / R1

If FLAGWRD8 bit 12 (CMOONFLG) = 1, rescale TS (s13)

RDOTV = RACT1 · VACT4

TSsn = |RDOTV| TS

TSden = $\sqrt{TScs^2 + TSsn^2}$

SNTH = TSsn / TSden

CSTH = TScs / TSden

VVEC = -VACT4 signRDOTV

RVEC = RACT1

Switch FLAGWRD7 bit 9 (RVSW) to 1

Perform "TIMETHET"

If RDOTV < 0; TCDH = NN HAFPA1 - HAFPA1 + T + TCSI

If RDOTV ≥ 0, TCDH = TCSI + NN HAFPA1 - T

Skip next step

CIRCL TCDH = TCSI + NN HAFPA1

If TTPI < TCDH, proceed to "SCNDSOL" with TS$_{sp}$ = 5

Perform "INTINT" with TSv = VACT4, TSr = RACT1, TSo = TCSI,
TSt = TCDH, and TSintyp = K:TWOPI

RACT2 = RATT

VACT2 = VATT

Perform "INTINT" with TSv = VPASS1, TSr = RPASS1, TSo = TCSI,
TSt = TCDH, and TSintyp = K:TWOPI

RPASS2 = RATT

VPASS2 = VATT

Perform "CDHMVR"

TSr = RACT2

TSv = VACT3

Perform "PERIAPO1"

POSTCDH = TShp

Perform "INTINT" with TSv = VACT3, TSr = RACT2, TSo = TCDH,
TSt = TTPI, and TSintyp = K:TWOPI

RACT3 = RATT

VACT3 = VATT

TSu = cosELEV unit(UP1 * unitRACT3) + sinELEV unitRACT3

TSrsin = TSu · RACT3

TS = RPASS3 · RPASS3 - RACT3 · RACT3 + TSrsin^2

```
If TS < 0:
    If LOOPCT = 1:
        CSIALRM = 1
        Proceed to "ALMXIT"
(If TS < 0)
    DELDV = DELDV / 2
    DELVCSI = DVPREV - DELDV
    Proceed to "CSI/B1"
TSk2 = -TSrsin - √TS
TSk1 = -TSrsin + √TS
TS = TSk2
If |TSk2| ≥ |TSk1|, TS = TSk1
URPESTIM = unit(RACT3 + TS TSu)
TS = (unitRPASS3 * URPESTIM) · (unitVPASS3 * unitRPASS3)
GAMP32 = arccos(URPESTIM · unitRPASS3) signTS
If FLAGWRD6 bit 14 (S32.1F2) = 1, proceed to "FRSTPAS"
TSslope = (GAMP32 - GAMPREV) / (DELVCSI - DVPREV)
DVPREV = DELVCSI
If FLAGWRD6 bits 12 (S32.1F3B) and 13 (S32.1F3A) are both 1:
    TS = GAMPREV (GAMP32 - GAMPREV)
    If TS ≥ 0, proceed to "FIFTYFPS"
    DELDV = K:INITST sign DELDV
    Switch FLAGWRD6 bit 13 (S32.1F3A) to 1
    Switch FLAGWRD6 bit 12 (S32.1F3B) to 0
    Proceed to "FRSTPAS"
```

If FLAGWRD6 bits 12 (S32.1F3B) and 13 (S32.1F3A) are both 0, proceed to "FIFTYFPS"

DELDV = GAMP32 / TSslope

GAMPREV = GAMP32

If |DELDV| < K:EPSILN1, proceed to "CSI/SOL"

If |DELDV| ≥ K:DELMAX1, DELDV = K:DELMAX1 signDELDV

DELVCSI = DELVCSI - DELDV

Proceed to "CSI/B1"

FRSTPAS GAMPREV = GAMP32

DVPREV = DELVCSI

DELVCSI = DELVCSI - DELDV

Switch FLAGWRD6 bit 14 (S32.1F2) to 0

Proceed to "CSI/B1"

FIFTYFPS DELDV = K:FIFPSDP signTSslope signGAMPREV

DELVCSI = DELVCSI - DELDV

GAMPREV = GAMP32

Switch FLAGWRD6 bits 12 (S32.1F3B) and 13 (S32.1F3A) to 1

Proceed to "CSI/B2"

SCNDSOL If FLAGWRD6 bit 12 (S32.1F3B) = 0 or bit 13 (S32.1F3A) = 1:

Proceed to "ALMXIT"

CSIALRM = TS_{sp}

Switch FLAGWRD6 bits 12 (S32.1F3B), 13 (S32.1F3A), and 15 (S32.1F1) to 0

Switch FLAGWRD6 bit 14 (S32.1F2) to 1

LOOPCT = 0

Proceed to "CSI/B"

ALMXIT

TS = 00600_8 + CSIALRM - 1 (i.e., 00605_8 if CSIALRM = 6)

Perform "VARALARM"

Proceed to "GOFLASH" with TS = K:V05N09
(If terminate, proceed to "GOTOPOOH"; if proceed, repeat this step; other response, continue at next step.)

Proceed to third step of "P72"

CSI/SOL

i = MUDEX

If POSTCSI < K:$PMIN_i$, proceed to "SCNDSOL" with TSsp = 2

If POSTCDH < K:$PMIN_i$, proceed to "SCNDSOL" with TSsp = 3

T1TOT2 = TCDH - TCSI

If T1TOT2 < K:TMIN, proceed to "SCNDSOL" with TSsp = 4

T2TOT3 = TTPI - TCDH

If T2TOT3 < K:TMIN, proceed to "SCNDSOL" with TSsp = 5

Proceed to "P32/P72C"

GET.LVC

$$\underline{D}\text{ELVLVC} = \begin{bmatrix} \text{unit}((\underline{R}\text{INIT} * \underline{V}\text{INIT}) * \underline{R}\text{INIT}) \\ -\text{unit}(\underline{R}\text{INIT} * \underline{V}\text{INIT}) \\ -\text{unit}\underline{R}\text{INIT} \end{bmatrix} \underline{T}\text{S}$$

Switch FLAGWRD5 bit 2 (MGLVFLAG) to 1

Return

GET+MGA

$\underline{I}$GAX = (REFSMMAT$_{21}$, REFSMMAT$_{22}$, REFSMMAT$_{23}$)

pMGA = arcsin(unit$\underline{T}$S · $\underline{I}$GAX)

If pMGA < 0, pMGA = 1 + pMGA

Switch FLAGWRD5 bit 2 (MGLVFLAG) to 0

Return

CENTANG: See TRGL section.

CSIALRM: Single precision decimal number that is converted to octal and added to 00577_8 to be displayed to indicate any failure in the CSI targeting iteration.

CSTH: See CONC section.

DELDV: Double precision increment to DELVCSI in one CSI iteration, scaled B7 in units of meters per centisecond.

DELVCSI: Double precision magnitude of velocity to be gained during the CSI burn, scaled B7 in units of meters per centisecond.

DELVEET1: Double precision vector corresponding to the velocity-to-be-gained vector for the CSI burn, scaled B7 in units of meters per centi-second. Parallel to the orbital plane of the passive vehicle and perpendicular to the active vehicle position vector at TCSI.

DELVEET2: Double precision vector corresponding to the velocity-to-be-gained vector for the CDH burn, scaled B7 in units of meters per centi-second. Parallel to the orbital plane of the passive vehicle.

DELVLVC: Double precision velocity vector expressed in local vertical coordinates, scaled B7 in units of meters per centisecond. In the local vertical coordinate system, X is along the horizontal component of velocity, Z points toward the center of attraction, and Y completes a right-handed, orthogonal system.

DELVSAB: Double precision magnitude of velocity to be gained for input to the thrusting programs, scaled B7 in units of meters per centisecond.

DELVSIN: Double precision velocity vector for input to the thrusting programs, scaled B7 in units of meters per centisecond and expressed in reference coordinates.

DIFFALT: Double precision difference of passive and active vehicle altitudes at the time of CDH, scaled B29 in units of meters; negative if the passive vehicle is below the active vehicle at CDH, and displayed to the astronaut during P32-72 and P33-73.

DISPDEX: See BURN section.

DVPREV: Previous value of DELVCSI.

ECC: Double precision eccentricity, scaled B3 and unitless.

ELEV: See TRGL section.

GAMP32, GAMPREV: Double precision error angle (and previous value of that angle) between projected rendezvous point and desired rendezvous point, scaled B0 in units of revolutions.

HAFPA1: Double precision time corresponding to half of a period in the post-CSI, pre-CDH orbit, scaled B28 in units of centiseconds.

HAPO, HPER: Double precision heights above the earth or moon at apogee and at perigee respectively, scaled B29 in units of meters.

$\underline{I}$GAX: Double precision unit vector along the inner gimbal axis, scaled B1 and expressed in reference coordinates.

K:60MIN: Double precision constant stored as 360000 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 360000.

K:CS359+: Double precision constant stored as 0.499999992, scaled B1 and unitless. Equation value: 0.999999984.

K:DELMAX1: Double precision constant stored as 0.6096000 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.6096. (Equivalent to 200 feet per second.)

K:DPmp01: Double precision constant stored as 77777_8 61337_8, scaled B0 in units of revolutions. Equation value: -0.455078125 x 2^{-14}. (Equivalent to - 0.01 degrees.)

K:DVMAX1: Double precision constant stored as 3.0480 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 3.0480. (Equivalent to 1000 fps.)

K:DVMAX2: Double precision constant stored as 3.014472 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 3.014472. (Equivalent to 989 fps.)

K:EPSILN1: Double precision constant stored as 0.0003048 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.0003048. (Equivalent to 0.1 fps.)

K:FIFPSDP: Double precision constant stored as -0.152400 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: -0.1524. (Equivalent to - 50 fps.)

K:INITST: Double precision constant stored as 0.03048 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.03048. (Equivalent to 10 fps.)

K:LOOPMX: Double precision constant stored as 16 x 2^{-28}, scaled B28 and unitless. Equation value: 16.

K:MAXNM: See EXVB section.

K:MUTABLE$_{MUDEX}$: See CONC section.

K:NICKELDP: Double precision constant stored as 0.021336 x 2^{-7}, scaled B7 in units of meters per centisecond. Equation value: 0.021336. (Equivalent to 7.0 fps)

K:ONETHTH: Double precision constant stored as 0.0001 x 2^{-3}, scaled B3 and unitless. Equation value: 0.0001.

K:PMIN$_0$: Double precision constant stored as 157420 x 2^{-29}, scaled B29 in units of meters; program notation PMINE. Equation value: 157420.

K:PMIN$_8$: Double precision constant stored as 10668 x 2^{-29}, scaled B29 in units of meters; program notation PMINM. Equation value: 10668.

K:SN359+: Double precision constant stored as -0.000086601, scaled B1 and unitless. Equation value: -0.000173202.

K:TMIN: Double precision constant stored as 60000 x 2^{-28}, scaled B28 in units of centiseconds. Equation value: 60000.

K:TWOPI: Double precision constant stored as 6.283185307 x 2^{-4}, scaled B4 and unitless. Equation value: 6.283185307.

LOOPCT: Double precision iteration counter, scaled B28 and unitless.

[LVCMAT]: Double precision, 3x3 transformation matrix defined such that $\underline{A}$lv = [LVCMAT] $\underline{A}$ref, where $\underline{A}$ is a vector expressed in local vertical and reference coordinates respectively.

MODREG: See DATA section.

MUDEX: See CONC section.

MUE: Double precision gravitational constant, scaled B36 in units of meters cubed per centisecond squared.

NN: Double precision number designating the apsidal crossing after CSI at which the CDH burn will be executed, scaled B14 and unitless. (NN = 1 indicates that the CDH burn will be executed at the first apsidal crossing after CSI.) NN is used in P34-74 and P35-75 as a flag to specify precision or conic integration. In "S34/35.2", it is used to set VTARGTAG.

NOMTPI: See TRGL section.

NORMEX: Single precision octal return address storage.

P: See CONC section.

pMGA: Double precision middle gimbal associated with a desired thrust direction, scaled B0 in units of revolutions.

POSTCDH: Double precision height above the earth or the moon at the perigee of the orbit of the active vehicle after the CDH burn, scaled B29 in units of meters.

POSTCSI: Double precision height above the earth or the moon at the perigee of the orbit of the active vehicle after the CSI burn, scaled B29 in units of meters.

QTEMP: Single precision octal return address storage.

R1: See CONC section.

RACT1, VACT1: Double precision position and velocity vectors of the active vehicle at TCSI, prior to the CSI burn, scaled B29 and B7 in units of meters and meters per centisecond respectively; rotated into the orbital plane of the passive vehicle.

RACT2, VACT2: Double precision position and velocity vectors of the active vehicle at TCDH, prior to the CDH burn, scaled B29 and B7 in units of meters and meters per centisecond respectively; rotated into the orbital plane of the passive vehicle.

RACT3, VACT3: See TRGL section.

RATT, VATT: See ORBI section.

RCV, VCV: See CONC section.

RdA: See CONC section.

RDESIRED: See CONC section.

RDOTV: Double precision dot product (RACT1 · VACT4) scaled B36 in units of meters squared per centisecond.

[REFSMMAT] : See COOR section.

RINIT, VINIT: See TRGL section.

RPASS1, VPASS1: Double precision position and velocity vectors of the passive vehicle at TCSI, scaled B29 and B7 in units of meters and meters per centisecond respectively.

RPASS2, VPASS2: Double precision position and velocity vectors of the passive vehicle at TCDH, scaled B29 and B7 in units of meters and meters per centisecond respectively.

RPASS3, VPASS3: See TRGL section.

RT1dMU: Double precision storage for the inverse of the square root of MUE, scaled B-17 (earth) or B-14 (moon) in units of centiseconds/meters to the three-halves power.

RTIG, VTIG: Double precision position and velocity vectors for input to the thrusting programs, scaled B29 and B7 in units of meters and meters per centisecond.

RTRN: Single precision octal return address storage.

RVEC, VVEC: See CONC section.

SMALLA: Double precision semi-major axis, in units of meters.

SNTH: See CONC section.

SUBEXIT: Single precision octal return address storage.

T: See CONC section.

T1TOT2: Double precision transfer time between CSI and CDH, scaled B28 in units of centiseconds; displayed to the astronaut in P32-P72. Transfer time between CDH and TPI when displayed to the astronaut in P33-P73.

T2TOT3: Double precision transfer time between CDH and TPI (P32-P72) or time difference between TTPI in P33 and TTPI used in P32 (displayed in P33-P73); scaled B28 in units of centiseconds.

TCDH: Double precision time of ignition of the CDH burn, scaled B28 in units of centiseconds; an astronaut input in P33-P73.

TCSI: Double precision time of ignition of the CSI burn, scaled B28 in units of centiseconds. It may be either an astronaut input or computed by the program.

TDEC1: See ORBI section.

TDEC2: Temporary storage for TDEC1 to assure that both active and passive states are advanced to the same time.

TET: See ORBI section.

TETLEM: "Permanent" time value for the LM state vector, scaled B28 in units of centiseconds.

TIG: See BURN section.

TRKMKCNT: See RNAV section.

TTOGO: See BURN section.

TTPI: See TRGL section.

TTPIO: Double precision storage for TPI time used in P32-72 for information in P33-73, scaled B28 in units of centiseconds.

UP1: Double precision unit vector perpendicular to the orbital plane of the passive vehicle, scaled B1 and unitless.

UNVEC: Double precision unit vector along RACT2, scaled B1 and unitless.

URPESTIM: A unit vector in the direction of the passive vehicle position vector that would satisfy the required TPI conditions, derived from the estimate of DELVCSI.

VACT4: Double precision velocity vector of the active vehicle at TCSI after the addition of the velocity gained in the CSI burn, scaled B7 in units of meters per centisecond.

Z: Z register, or program counter. Contains address of the next step.

TABLES

TABLES

Program and Routine List

Programs

Number (GSOP)	Title in this document	Page
00	POOH	PGSR-6
06	P06	PGSR-13
07	SYSTEST	EXVB-10
12	P12LM	ASCT-1
20	PROG20	RNAV-1
21	PROG21	RNAV-6
22	PROG22	RNAV-1
25	PROG25	RNAV-7
27	V7xUPDAT	EXVB-8
30	P30	TRGX-1
32	P32	TRGX-1
33	P33	TRGX-4
34	P34	TRGL-1
35	P35	TRGL-4
40	P40LM	BURN-1
41	P41LM	BURN-2
42	P42LM	BURN-1
47	P47LM	BURN-13
51	P51	ALIN-1
52	PROG52	ALIN-14
57	P57	ALIN-25
63	P63LM	DESC-1
64	STARTP64	DESC-4
65	P65START	DESC-5
66	STARTP66	DESC-11
68	LANDJUNK	DESC-18
70	P70 or P70A	ASCT-3
71	P71 or P71A	ASCT-3
72	P72	TRGX-1
73	P73	TRGX-4
74	P74	TRGL-1
75	P75	TRGL-4
76	P76	ORBI-23

Routines

Number (GSOP)	Title in this document	Page
00	GOTOPOOH	PGSR-11
01	ABCLOAD	DATA-19
02	R02BOTH	IMUC-19
03	DAPDATA1	DAPB-19
04	R04	EXVB-7
05	SBANDANT	EXVB-27
09	R10,R11	SERV-14
10	LANDISP	SERV-15
11	R10,R11	SERV-14
12	MUNRETRN	SERV-7
13	LUNLAND	DESC-4
20	RADAREAD	RADR-19
21	R21LEM	RNAV-13
22	R22LEM	RNAV-17
23	R23LEM	RNAV-19
24	R24LEM	RNAV-20
25	RRAUTCHK	RADR-1
26	R21LEM4	RNAV-14
29	R29	RNAV-31
30	V82CALL	EXVB-17
31	V83CALL	EXVB-24
33	ALINTIME	EXVB-6
36	R36	EXVB-26
40	DVMON	SERV-3
41	MIDTOAV1	ORBI-19
47	AGSINIT	EXVB-12
50	CAL53A	ALIN-18
51	R51	ALIN-19
52	R52	ALIN-21
53	AOTMARK	ALIN-3
54	CHKSDATA	ALIN-13
55	R55	ALIN-22
56	TRMTRACK	EXVB-6
57	MARKRUPT	ALIN-6
58	PLANET	ALIN-12
59	R59	ALIN-30
60	R60LEM	ATTM-1
61	R61LEM	RNAV-10
62	R62DISP	ATTM-11
63	V89CALL	ATTM-11
65	R65LEM	RNAV-10
76	TESTXACT	EXVB-2
77	R77	EXVB-9

Noun List

The following is an interpretation of the list of nouns that are used or defined in the LUMINARY program (See DATA section). Other nouns are either illegal or meaningless when used with the LUMINARY program. The list includes the tag assigned to the components of each noun (in this document); the number of components in each noun; the magnitude and units used in a decimal display of each noun; the section in which each component of each noun is defined; and an indication if the noun is a "no-load" or "decimal only" noun. Routines making use of a noun are listed in parentheses under the above information.

Special Considerations:

1. The X, Y and Z components of a vector quantity are displayed in R1-R3 respectively with the same scaling and units for each component.

2. Single component nouns appear in R1 only.

Noun	Tag	Comp.	Def.	Decimal Display	Comment
00	spare				
01	---	3	---	.XXXXX without regard to scaling or units	Address supplied
	(OHWELL1, OHWELL2, SDISPLAY)				
02	---	3	---	XXXXX. without regard to scaling or units	Address supplied
	(UPVERIFY)				
03	---	3	---	XXX.XX degrees	Address supplied
	(none)				
04	DSPTEM1	1	DATA	XXX.XX degrees	
	(GVBETER)				
05	DSPTEM1	1	DATA	XXX.XX degrees	
	(R22LEM, CHKSDATA)				
06	OPTION1	3	DATA	Octal only	
	OPTION2		DATA	Octal only	
	OPTION3		DATA	Octal only	
	(DSPOPTN, GOPERF4, GOPERF4R) GOPERF_R				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
07	XREG	3	DATA	Octal only	When used with verb 25, noun 7 is used to switch a specified bit or bits of a flagword or channel. In that case, ECADR of the flagword or channel is loaded into XREG, bits to be changed are set in YREG (01004_8 indicates that bits 10 and 3 are to be switched), ZREG is set to zero if bits are to be reset, non-zero if bits are to be set.
	YREG		DATA	Octal only	
	ZREG		DATA	Octal only	
	(none)				
			(See "ABCLOAD" routine)		
08	ALMCADR$_0$	3	PGSR	Octal only	
	ALMCADR$_1$		PGSR	Octal only	
	ERCOUNT		TEST	Octal only	
	(none)				
09	FAILREG$_0$	3	PGSR	Octal only	
	FAILREG$_1$		PGSR	Octal only	
	FAILREG$_2$		PGSR	Octal only	
	(used whenever alarm codes are displayed)				
10	---	1	---	Octal only	Channel to be specified
	(none)				
11	TCSI	3	TRGX	00XXX. hours	Decimal only- three components must be supplied
				000XX. minutes	
				0XX.XX seconds	
	(VN0611)				
12	OPTIONX$_0$	2	EXVB	Octal only	
	OPTIONX$_1$		EXVB	Octal only	
	(VBCOARK, V82CALL, R04Z, V89CALL)				
13	TCDH	3	TRGX	00XXX. hours	Decimal only- three components must be supplied
				000XX. minutes	
				0XX.XX seconds	
	(P33, P73)				
14	DSPTEMX$_0$	3	DATA	XXXXX. without regard	
	DSPTEMX$_1$		DATA	XXXXX. to scaling	
	DSPTEMX$_2$		DATA	XXXXX. or units	
	(none)				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
15	--- (none)	1	---	Octal only	Increment machine address
16	$DSPTEMX_{dp}$ (AGSDISPK, R36, AGSVCALC)	3	DATA	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
17	spare				
18	FDAI (V89RECL, TOBALLA, REDOMANC)	3	ATTM	XXX.XX degrees	
19	spare				
20	CDU (GYCOARS, VBZERO, VBCOARK)	3	IMUC	XXX.XX degrees	
21	PIPA (none)	3	IMUC	XXXXX. pulses for each	
22	THETAD (VBCOARK, IMUATTCK, P51, R62DISP, P52D, INITBY)	3	IMUC	XXX.XX degrees	
23	spare				
24	$DSPTEMX_{dp}$ (ALINTIME)	3	DATA	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only- three components must be supplied
25	$DSPTEM1_0$ $DSPTEM1_1$ $DSPTEM1_2$ (GOPERF1, GOPERF1R, R04X)	3	DATA DATA DATA	XXXXX. without regard XXXXX. to scaling XXXXX. or units	
26	$DSPTEM1_0$ $DSPTEM1_1$ $DSPTEM1_2$ (none)	3	DATA DATA DATA	Octal only Octal only Octal only	Used to load address, bank and priority or delay information before a verb 30 or verb 31
27	SMODE (none)	1	TEST	XXXXX. unitless	
28	spare				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
29	spare				
30	spare				
31	spare				
32	mTPER (none)	3	EXVB	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only- three components must be supplied
33	TIG (P76, P30, ORBCHGO, P12LM)	3	BURN	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
34	DSPTEM1$_{dp}$ (PROG2, P52B, P57OPT)	3	DATA	00XXX. hours 000XX. minutes 0XX.XX seconds	(=TALIGN) Decimal only three components must be supplied
35	TTOGO (none)	3	BURN	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
36	TIMENOW (none)	3	EXVB	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
37	TTPI (VN0655, P34, P74, INTLOOP)	3	TRGL	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
38	TET (none)	3	ORBI	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
39	spare				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
40	TTOGO	3	BURN	XX XX min-sec	No load - decimal only
	DELVSAB		TRGX	XXXX.X fps	
	DVTOTAL		SERV	XXXX.X fps	
	(POSTBURN, CLOKJOB, V99RECYC)				
41	$DSPTEM1_0$	2	DATA	XXX.XX degrees	
	$DSPTEM1_1$		DATA	XX.XXX degrees	
	(REDO)				
42	HAPO	3	TRGX	XXXX.X nautical miles	Decimal only
	HPER		TRGX	XXXX.X nautical miles	
	DELVSAB		TRGX	XXXX.X fps	
	(P30)				
43	LAT	3	COOR	XXX.XX degrees	Decimal only
	LONG		COOR	XXX.XX degrees	
	ALT		COOR	XXXX.X nautical miles	
	(P21VSAVE, LANDJUNK)				
44	HAPOX	3	EXVB	XXXX.X nautical miles	No load - decimal only
	HPERX		EXVB	XXXX.X nautical miles	
	TFF		EXVB	XX XX min - sec	
	(V82GOFLP, V82GON)				
45	TRKMKCNT	3	RNAV	XXXXX. unitless	No load - decimal only
	TTOGO		BURN	XX XX min-sec	
	pMGA		TRGX	XXX.XX degrees	
	(VN1645)				
46	DAPDATR1	1	DAPB	Octal only	
	(DAPDATA1)				
47	LEMMASS	2	DAPB	XXXXX. pounds mass	Decimal only
	CSMMASS		DAPB	XXXXX. pounds mass	
	(DAPDATA2)				
48	PITTIME	2	DAPB	XXX.XX degrees	Decimal only
	ROLLTIME		DAPB	XXX.XX degrees	
	(DAPDATA2, TRIMDONE)				
49	R22DISPR	3	RNAV	XXXX.X nautical miles	Decimal only
	R22DISPV		RNAV	XXXX.X fps	
	WHCHREAD		RNAV	XXXXX.	
	(N49DSP (R22LEM96))				
50	spare				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
51	PITCHANG	2	EXVB	XXX.XX degrees	Decimal only
	YAWANG		EXVB	XXX.XX degrees	
	(SBANDEX)				
52	ACTCENT	1	TRGL	XXX.XX degrees	
	(none)				
53	spare				
54	RANGE	3	EXVB	XXX.XX nautical miles	Decimal only
	RRATE		EXVB	XXXX.X fps	
	RTHETA		EXVB	XXX.XX degrees	
	(DISPN5X)				
55	NN	3	TRGX	XXXXX. unitless	Decimal only
	ELEV		TRGL	XXX.XX degrees	
	CENTANG		TRGL	XXX.XX degrees	
	(DISPLAYE, VN0655)				
56	RR-AZ	2	RNAV	XXX.XX degrees	
	RR-ELEV		RNAV	XXX.XX degrees	
	(DSPRRLOS)				
57	spare				
58	POSTTPI	3	TRGL	XXXX.X nautical miles	Decimal only
	DELVTPI		TRGL	XXXX.X fps	
	DELVTPF		TRGL	XXXX.X fps	
	(INTLOOP)				
59	DVLOS	3	TRGL	XXXX.X fps	Decimal only
	(S34/35.5, NTARGCHK)				
60	FORVEL	3	SERV	XXXX.X fps	Decimal only
	HDOTDISP		SERV	XXXX.X fps	
	HCALC1		DESC	XXXXX. feet	
	(VERTDISP)				
61	TTFDISP	3	DESC	XX XX min-sec	No load - Decimal only
	TTOGO		BURN	XX XX min-sec	
	OUTOFPLN		DESC	XXXX.X nautical miles	
	(CLOKJOB)				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
62	ABVEL TTOGO DVTOTAL (CLOKJOB, V99RECYC)	3	SERV BURN SERV	XXXX.X fps XX XX min-sec XXXX.X fps	No load - decimal only
63	ABVEL HDOTDISP HCALC1 (CLOKJOB, P63DISPS, CUTOFF, V99RECYC , ASCTERM1)	3	SERV SERV DESC	XXXX.X fps XXXX.X fps XXXXX. feet	Decimal only
64	FUNNYDSP HDOTDISP HCALC (P64DISPS)	3	DESC SERV SERV	XX XX XXXX.X fps XXXXX. feet	No load - decimal only
65	SAMPTIME (none)	3	DSKY	00XXX. hours 000XX. minutes 0XX.XX seconds	Decimal only - three components must be supplied
66	$RSTACK_6$ Bits 7-6 of Channel 33 (RO4X)	2	RADR ---	XXXXX. feet 0000X. unitless	No load - decimal only
67	$RSTACK_0$ $RSTACK_2$ $RSTACK_4$ (RO4X)	3	RADR RADR RADR	XXXXX. fps XXXXX. fps XXXXX. fps	
68	RANGEDSP TTFDISP DELTAH (DSP68, WAIT68)	3	DESC DESC SERV	XXXX.X nautical miles XX XX min - sec XXXXX. feet	No load - decimal only
69	DLAND (none)	3	DESC	XXXXX. feet	Decimal only. Components in z,y,x order.
70	$AOTCODE_0$ $AOTCODE_1$ $AOTCODE_2$ (R52, R59)	3	ALIN --- ---	Octal only Octal only Octal only	Note that noun 70 is displayed with verb 01. The second and third components are meaning-less.

Noun	Tag	Comp.	Def.	Decimal Display	Comment
71	AOTCODE$_0$	3	ALIN	Octal only	Like noun 70
	AOTCODE$_1$		---	Octal only	
	AOTCODE$_2$		---	Octal only	
	(GETDAT, CHANGEVB, GETMKS)				
72	CDU$_t$	2	RADR	XXX.XX degrees	(360 degrees - CDU$_t$ displayed)
	CDU$_s$		RADR	XXX.XX degrees	
	(RO4X, VBZERO, VBCOARK, R21DISP)				
73	TANG$_0$	2	RADR	XXX.XX degrees	(display of TANG$_0$ is the comp-
	TANG$_1$		RADR	XXX.XX degrees	lement of the value stored)
	(VBCOARK)				
74	TTOGO	3	BURN	XX XX min-sec	No load - decimal only
	YAW		ASCT	XXX.XX degrees	
	PITCH		ASCT	XXX.XX degrees	
	(CLOKJOB, V99RECYC)				
75	DIFFALT	3	TRGX	XXXX.X nautical miles	No load - decimal only
	T1TOT2		TRGX	XX XX min-sec	
	T2TOT3		TRGX	XX XX min-sec	
	(P32/P72F, P33/P73F)				
76	ZDOTD	3	ASCT	XXXX.X ft/sec	Decimal only
	RDOTD		ASCT	XXXX.X ft/sec	
	XRANGE		ASCT	XXXX.X nautical miles	
	(P12LM)				
77	TTOGO	2	ASCT	XX XX min-sec	No load - decimal only
	YDOT		ASCT	XXXX.X fps	
	(none)				
78	DNRRANGE	3	RADR	XXX.XX nautical miles	No load - decimal only
	DNRRDOT		RADR	XXXXX. fps	
	TTOTIG		RADR	XX XX min-sec	
	(RO4X)				
79	CURSOR	3	ALIN	XXX.XX degrees	Decimal only
	SPIRAL		ALIN	XXX.XX degrees	
	POSCODE		ALIN	XXXXX. unitless	
	(DSPV6N79, 79DISP)				
80	DATAGOOD	2	RNAV	XXXXX. unitless	
	OMEGDISP		RNAV	XXX.XX degrees	
	(R24LEM)				
81	DELVLVC	3	TRGX	XXXX.X fps	Decimal only
	(P30, S34/35.5, P32/P72F, P33/P73F)				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
82	DELVLVC (P32/P72F)	3	TRGX	XXXX.X fps	Decimal only
83	DELVIMU (P47BODY)	3	BURN	XXXX.X fps	Decimal only
84	DELVOV (P76)	3	ORBI	XXXX.X fps	Decimal only
85	VGBODY (TIGNOW, CUTOFF, P41LM, TIG-30A)	3	BURN	XXXX.X fps	Decimal only
86	DELVLVC (none)	3	TRGX	XXXX.X fps	Decimal only
87	AZ	2	ALIN	XXX.XX degrees	
	EL		ALIN	XXX.XX degrees	
	(GETDAT, GETAZEL)				
88	STARAD (PLANET)	3	ALIN	.XXXXX unitless	Decimal only
89	LANDLAT	3	ALIN	XX.XXX degrees	Decimal only
	LANDLONG		ALIN	XX.XXX degrees	
	LANDALT		ALIN	XXX.XX nautical miles	
	(N89DISP)				
90	RANGE	3	EXVB	XXX.XX nautical miles	Decimal only
	RRATE		EXVB	XXXX.X fps	
	RTHETA		EXVB	XXX.XX degrees	
	(R36)				
91	P21ALT	3	RNAV	XXXXXB. nautical miles	
	P21VEL		RNAV	XXXXX. ft/sec	
	P21GAM		RNAV	XXX.XX degrees	
	(none)				
92	THRDISP	3	DESC	XXXXX. percent	
	HDOTDISP		SERV	XXXX.X fps	
	HCALC1		DESC	XXXXX. feet	
	(none)				
93	OGC	3	COOR	XX.XXX degrees	
	IGC		COOR	XX.XXX degrees	
	MGC		COOR	XX.XXX degrees	
	(IMUFINEK, R55, INITBY)				
94	spare				
95	spare				
96	spare				

Noun	Tag	Comp.	Def.	Decimal Display	Comment
97	$DSPTEM1_0$	3	DATA	XXXXX. unitless	
	$DSPTEM1_1$		DATA	XXXXX. unitless	
	$DSPTEM1_2$		DATA	XXXXX. unitless	
	(none)				
98	$DSPTEM2_0$	3	DATA	XXXXX. unitless	
	$DSPTEM2_1$		DATA	.XXXXX unitless	
	$DSPTEM2_2$		DATA	XXXXX. unitless	
	(SHOW)				
99	WWPOS	3	RNAV	XXXXX. feet	Decimal only
	WWVEL		RNAV	XXXX.X fps	
	WWBIAS		RNAV	XX.XXX radians	
	(V67CALL, GOTOPOOH - in GOTOPOOH, the noun is not processed)				

Alarm Codes

All alarm codes are listed in octal. The names of the "Routines" where each code is generated refer to the titles as they appear in this document.

Alarm		Routines	Significance
00107	L	YMKRUPT	Five or more mark pairs already recorded
	L	SURFAGAN	
00111	L	MARKCHEX	Mark missing
00112	L	MARKRUPT	Mark or mark reject attempted with the respective routine inoperative
00113	L	SOMEKEY	No inbits in channel 16
00114	L	YMKRUPT	Wrong mark
00115	L	MARKRUPT	Mark reject when no marks taken to reject
00206	L	IMUZERO	ICDU zero attempted while in coarse align because of gimbal lock warning
00207	L	IMUMON	ISS turn-on request not present for 90 seconds
00210	L	IMUCHK	IMU not operating when required
	L	RO2BOTH	
	L	IMUZERO	
00211	L	COARS2	Coarse align error
00212	L	PIPFREE	PIPA fail when primary PIPA fail monitor has been disabled (PIPA not in use)
	L	C33TEST	
00213	L	TNONTEST	"Turn-on delay initiate" signal from IMU present without "IMU operate" signal
00214	L	IMUMON	Program using IMU when it was turned off
00217	L	CURTAINS	Bad return from stall routine (e.g., switching fail)
00220	L	AGSINIT	IMU not aligned - no REFSMMAT
	L	RO2BOTH	
00401	L	KALCMAN3	Desired gimbal angles yield gimbal lock
	L	CALCGA	
	L	DCMCL	
00402	L	NOATTCNT	FINDCDUW not controlling attitude
00404	D	R59ALM	Desired star not available in any detent position
00405	D	R51	"R56" did not find star pair for optical sighting
00421	L	WMATEND	W-matrix overflow
00501	P	R23LEM	Manual RR acquisition not within limits
00502	L	RRDESNB	Bad RR gimbal angle input (verb 41)
00503	L	RRDESK2	RR designate failure
	P	R21LEM	
	P	R21LEM1	
00510	L	RRZERO	RR zero requested (verb 40) with RR auto discrete absent
00511	L	MUNRETRN	LR not in proper position
00514	P	P20LEMB7	RR auto discrete removed while RR in use
00515	L	RRCDUCHK	RR CDU fail discrete detected
00520	L	RADAREAD	RR read interrupt initiated when not called for

Alarm		Routines	Significance
00521	L	RADAREAD	No data good while reading radar
	L	RESAMPLE	
00522	L	DORSAMP	LR position change during LR reading
00523	L	LRP2COMM	LR position 2 not achieved in 23 seconds
	P	POSALARM	
00525	P	R22LEM	Actual RR LOS differs from computed LOS by more than 3°
00526	P	P20LEM1	Range greater than 400 nautical miles
	P	CSMINT	
00527	L	LRS24.1	LOS not in mode limits
00530	P	60TIMES	LOS not in coverage (P22) less than 10 minutes from now
00600	D	CIRCL	Imaginary roots on first iteration of CSI solution
00601	D	CSI/SOL	Projected perigee altitude after CSI less than minimum
00602	D	CSI/SOL	Projected perigee altitude after CDH less than minimum
00603	D	CSI/SOL	Time between CSI and CDH less than minimum
00604	D	CSI/SOL	Time between CDH and TPI less than minimum
	D	CIRCL	
00605	D	CSI/B1	Too many iterations
00606	D	CSI/B2	Projected DELVCSI exceeds maximum
00611	D	INTLOOP	No TPI ignition time solution for given elevation angle
	D	P33/P73B	
00701	D	DSPOPTN	Illegal option code was selected
00777	L	SETISSW	ISS warning caused by PIPA fail
01102	L	PRERRORS	AGC self test error
01105	L	DNTMFAST	Downlink too fast
01106	L	UPTMFAST	Uplink too fast
01107	L	GOPROG3	Phase table discrepancy; restart failure; causes fresh start.
01301	L	ARCSUB	Arcsin or arccosine input too large
01406	L	1406ALM	Bad return from ROOTPSRS routine during descent guidance
01407	L	VGAIN*	VG increasing
01410	L	EXVERT	Unintentional overflow in descent (P63 or P64) guidance
01412	L	EXGSUB	Calculation of descent ignition time not converging
01520	L	V37	Verb 37 not permitted at this time
01600	L	SOMEERRR	Overflow in drift test
01601	L	SOMERR2	Bad IMU torque
01703	L	MIDTOAV1	Insufficient time remaining before scheduled ignition
	L	CKMID2	
01706	D	P40ALM	Incorrect program requested for vehicle configuration
02001	L	TRYUORV	Jet failures have disabled Y-Z translation
02002	L	+XORULGE	Jet failures have disabled X translation
02003	L	SELECTP	Jet failures have disabled P rotation
02004	L	FAILOOP	Jet failures have disabled U-V rotation
03777	L	SETISSW	ISS warning caused by ICDU fail
04777	L	SETISSW	ISS warning caused by ICDU and PIPA fail
07777	L	SETISSW	ISS warning caused by IMU fail

Alarm		Routines	Significance
10777	L	SETISSW	ISS warning caused by IMU fail and PIPA fail
13777	L	SETISSW	ISS warning caused by IMU and ICDU fail
14777	L	SETISSW	ISS warning caused by IMU, ICDU and PIPA fail
20105		AOTMARK	Optical mark routine called when already busy
20430		GOBAQUE	Acceleration overflow in integration
20607		TIMERAD,TIMETHET COMMNOUT	No solution
21103		CCSHOLE	Theoretically impossible branch taken
21204		DLY2 LONGCALL	Zero or negative time applied to waitlist, delay or LONGCALL
21302		SQRT	Negative argument input to square root routine
21406		1406POO	Bad return from ROOTPSRS routine during descent pre-ignition phase
21501		DSPALARM	Invalid display interface information from internal program
31104		DELLOOP	Too many users of delay routine
31201		FINDVAC2	Too many jobs requiring working storage
31202		NOVAC3	Too many jobs
31203		DLY2 WTLST5	Too many tasks
31206		NV50DSP FLASHSUB	More than one program active in display interface routines at one time
31207		AOTMARK	No working storage available for marks
31210		GOMANUR IMUSTALL AOTSTALL RADSTALL	Two users of the same device or routine (radar, AOT, IMU, KALCMAN3)
31211		AOTMARK	Illegal interrupt of extended verb
31502		MAKEPLAY MAKEPRIO FLASHSUB	Illegal flashing display
32000		PAXIS	DAP still in progress from last TIME5 interrupt when another TIME5 interrupt is initiated

L Alarm lights "Program Caution" light and the code is stored in noun 09 (which is on telemetry)

D Alarm does all of "L" plus the alarm code is displayed as part of program logic

P Alarm does all of "D" functions except with a priority display.

If the first number is a 3 in the code, the alarm does all of "L" functions plus a "BAILOUT" or software restart.

If the first number is a 2 in the code, the alarm does all of "L" functions plus a "POODOO" which terminates all program activity.

Checklist Codes

Checklist Code	Routine	Meaning
13	P52D	Key in Normal Coarse Align or Gyro Torque Coarse Align Terminate: Proceed to "GOTOPOOH" Proceed: Do Normal Coarse Align; "REGCOARS" Enter or Resequence: Do Gyro Torque Coarse Align
14	P57POST	Key in Fine Alignment Option Terminate: Proceed to "GOTOPOOH" Proceed: Align LM based on keyed in option ("ATTCHK") Enter or Resequence: Determine LM's position from gravity vector if option 2 alignment
	R51K	Key in Fine Alignment Option Terminate: Proceed to "GOTOPOOH" Proceed: Check alignment by recycling through Routine 51 Enter or Resequence: Exit Routine 51
	ASTNRET	Key in Fine Alignment Option Terminate: Proceed to "GOTOPOOH" Proceed: Check alignment by performing Routine 51 and continuing Enter or Resequence: Continue without check
15	P51D	Perform Celestial Body Acquisition; Key in Proceed Option Terminate: Proceed to "GOTOPOOH" Proceed: Continue to marking proceed Enter or Resequence: Perform Coarse Align to zero IMU; Then redisplay code 15
	R51	Perform Celestial Body Acquisition; Key in Proceed Option Terminate: Proceed to "GOTOPOOH" Proceed: Continue with "R51" Enter or Resequence: Proceed to "R51E" (maneuver LM and mark)
62	P06	Switch AGC Power down; Enter standby mode

Checklist Code	Routine	Meaning
201	R04X	Switch RR to automatic tracking mode Terminate: Exit Routine 04 Proceed: Continue with Routine 04 (after checking that RR is switched to auto mode) Enter or Resequence: Repeat checklist request
	P20LEMB7	Switch RR to automatic tracking mode Terminate: Exit Program 20 or 22 Proceed: Continue Program 20 or 22 Enter or Resequence: Perform "R23LEM" (manual acquisition monitor) if not on lunar surface; then continue Program 20
203	P40AUTO	Switch Control to PGNCS; switch DAP to AUTO; switch Throttle Control to Auto (if descent) Terminate: Proceed to "GOTOPOOH" Proceed: Continue present program if switches are properly set Enter or Resequence: Continue present program regardless of switch setting
205	R23LEM	Perform Manual Acquisition of CSM with RR Terminate: Exit Program 20 Proceed: Continue in Routine 23 Enter or Resequence: Perform Automatic Maneuver; then continue in Routine 23
500	ASTNRET	Switch LR antenna to position 1 Terminate: Proceed to "GOTOPOOH" Proceed: Initialize Landing Radar Control ("SETPOS1"), if LR is in position 1 and continue Enter or Resequence: Initialize Landing Radar Control ("SETPOS1") and continue regardless of Landing Radar Position

Option Codes

This is a list of the option codes displayed in R1 in conjunction with a V04N06 (OPTION1) or V04N12 (OPTIONX$_0$) to request the astronaut to load into R2 the option he desires for program LUMINARY (OPTION2 or OPTIONX$_1$ respectively). In the case of option code 10 in R1, an OPTION3 is also defined as a flagword indicator which is set by the program and loaded into R3. V05 is used in this case.

Option Code R1	Purpose	Input of R2	Routine	Option Code Types
00001	Specify IMU Orientation	1- Preferred Attitude 2- Nominal Attitude 3- Attitude specified by present REFSMMAT 4- Landing Site Attitude	P52B P570PT	R1-OPTION1 R2-OPTION2
00002	Specify Vehicle	1- This Vehicle (LM) 2- Other Vehicle (CSM)	PROG21	R1-OPTION1 R2-OPTION2
			V82CALL	R1-OPTIONX$_0$ R2-OPTIONX$_1$
00003	Specify Tracking Attitude	1- Point Z-axis (preferred) 2- Point X-axis	V89CALL	R1-OPTIONX$_0$ R2-OPTIONX$_1$
00004	Specify Radar	1- RR (Rendezvous Radar) 2- LR (Landing Radar)	R04Z	R1-OPTIONX$_0$ R2-OPTIONX$_1$
00006	Specify RR Coarse Align Option	1- Lock-on 2- Continuous Designate	VBCOARK	R1-OPTIONX$_0$ R2-OPTIONX$_1$

Option Code R1	Purpose	Input of R2	Routine	Option Code Types
00010	Specify Alignment Option	0- Time critical - Stored LM Attitude or Present REFSMMAT 1- REFSMMAT + Lunar Gravity Vector 2- Two Bodies 3- One Body + Lunar Gravity Vector	DSPOPTN	R1-OPTION1 R2-OPTION2 R3-OPTION3

R3-OPTION3 Definition

Bit 7 = 1: REFSMMAT is defined
= 0: REFSMMAT is not defined
Bit 4 = 1: Stored LM attitude available
= 0: Stored LM attitude not available

All other bits are zero

Option Code R1	Purpose	Input of R2	Routine	Option Code Types
00012	Specify CSM Orbit Option	1- No Orbit Change 2- Change Orbit to pass over the LM	ORBCHGO	R1-OPTION1 R2-OPTION2

Job Priorities

Priority	Job	Routine Where Priority Established (or changed)	
35	NBDONLY	SVCT3	
33	MAKEPLAY	MAKEPLAY	(change)
32	HIGATJOB	MUNRETRN	
	JAMTERM	NVDSP	
	LRVJOB	VALTCHK	
	LRHJOB	R10,R11	
31	TTFINCR	TTFINCR	(change)
30	TSTLTS3	TSTLTS2	
	MONDO	MONREQ	
	CHARIN	KEYRUPT1 UPRUPT	
	UPJOB	UPSTORE	
	PROCKEY	PROCEEDE	
	DSPMMJOB	TSTLTS3 REDO UPUPDATE UPOUT4 V37XEQ GOPROG3 ENDTEST1 P65START STARTP66 STARTP64	
27	1/ACCSET	DAPIDLER	
	1/ACCJOB	VOPENED RCSMONIT SETMINDB SETMAXDB PFLITEDB	
	N49DSP	R22LEM96	
	CLOKJOB	CLOKTASK	
26	DODES	MOREDES	
	PROG20	PROG20	(change)
	P20LEMB	P20LEMB	(change)
	P20LEMC3	P20LEMD1	
	R22LEM42	P20LEMC1	
	PROG25	PROG25	(change)
	R61C+L01	R61C+L06	
	R29RDJOB	R29READ	
	NEWDELHI	UPDTCALL	
	R21LEM10	R21LEM9	
	RELINUS	RELINUS	(change)
25	ENDEXT	TICKTEST	
	DORSAMP	RADSAMP	
	DATGDCHK	CALLDGCH	
	COMFAIL	DVMON	
23	POSGOOD	POSGOOD	(change)
22	REDO	SYSTEST	
	NORMLIZE	PREREAD	
	GETRVN	GETRVN	(change)
	RODCOMP	RODTASK	

Priority	Job	Routine Where Priority Established (or changed)
21	R29REMOJ	R29
	R29DODES	BEGDES29
	1/GYRO	1/PIPA
	LASTBIAS	PREREAD
20	ALFLT	ALLOOP
	PIPJOBB	PIPATASK
	RRDESK2	VBCOARK
	S40.13	P40SJUNK
	P47BODY	STARTP47
	SERVICER	READACCS
	TTFINCR	TTFINCR (change)
	POSTAND	P06
17	CUTOFF	GOCUTOFF ENGOFF1
	P41BLANK	TIG-35
	TIG-30A	TIG-30.1
	UPTHROT	UPTHROT (change)
	[illegible]	[illegible]
16	TIGNOW	TIGTASK
15	PLAYJUM1	NORMRET
	CHANGEVB	REMARK
	GETDAT	AOTMARK
14	P25LEM1	RELINUS P25LEM1 V37RET
13	ADDGRAV	GRABGRAV
	ASTNRET	CLOKJOB
	P76	V37XEQ
	P75	V37XEQ
	P74	V37XEQ
	P73	V37XEQ
	P72	V37XEQ
	P35	V37XEQ
	P34	V37XEQ
	P33	V37XEQ
	P32	V37XEQ
	P30	V37XEQ
	LANDJUNK	V37XEQ
	P63LM	V37XEQ
	P57	V37XEQ
	PROG52	V37XEQ
	P51	V37XEQ

Priority	Job	Routine Where Priority Established	(or changed)
13 (cont.)			
	P47LM	V37XEQ	
	P42LM	V37XEQ	
	P41LM	V37XEQ	
	P40LM	V37XEQ	
	PROG25	V37XEQ	
	PROG22	V37XEQ	
	PROG21	V37XEQ	
	PROG20	V37XEQ	
	P12LM	V37XEQ	
	P06	V37XEQ	
12	POSTBURN	GOPOST ENGINOFF	
10	TRIMDONE	PITCHOFF	
	R62DISP	CREWMANU	
	ATTACHIT	ATTACHED	
	V89CALL	V89PERF	
	RELINUS	CHKLINUS	
	S40.9	UPDATEVG	
7	DAPDATA1	DAPDISP	(change)
	ALINTIME	ALINTIME	(change)
	V82CALL	V82PERF	(change)
	R36	V90PERF	
	V82GOFF1	V82GOFLP	
	V82GON1	V82GON	
	SHOWSUM2	SHOWSUM2	(change)
5	RRLOSDSP	DSPRRLOS	
	V67CALL	V67	
	DISPN5X	R31CALL	
	SBANDEX	SBANDEX	(change)
	STATINT1	STATEINT POOH	
	DYNMDISP	P41LM	
4	DSPRRLOS	DSPRRLOS	(change)
	AGSINIT	V47TXACT	
	SBANDANT	VB64	
	SBANDEX	SBANDEX	(change)
3	V83CALL	R31CALL	
	GETRVN	GETRVN	(change)
0	DUMMYJB2	(not established - proceeded to by EJSCAN)	
same as user	MAKEPLAY	GODSPRS1	
	ENDRET	ENDRET	(change)
	NV50DSP	NV50DSP	(change)
	(caller)	GOODMANU	

INDEX

Index of Routines

This index includes, in addition to the list of all routines described and their locations in this document, a list of each place the routine is "referenced." This includes any reference to a routine by name, for example, by means of "Perform", "Proceed to", "Establish", checking of routine address, etc. It does not include, as a reference, routines that flow into (are followed directly by) another routine.

+ON	DSKY-10	2INTOUT DSPSIGN
+XORULGE	DAPA-19	RCS
-ON	DSKY-10	DSPSIGN
*ENTER	BURN-10	V99RECYC
*PROCEEDE	BURN-5	V99RECYC
1/ACCJOB	DAPB-1	PFLITEDB RCSMONIT SETMAXDB SETMINDB VOPENED
1/ACCONT	DAPB-4	1/ACCS
1/ACCRET	DAPB-11	DOCKTEST
1/ACCS	DAPB-1	1/ACCJOB SERVOUT
1/ACCSET	DAPB-1	DAPIDLER
1/GYRO	IMUC-1	1/PIPA NBD2
1/PIPA	IMUC-2	ADDGRAV ALFLT SERVICER
1/WLOOP	CONC-11	1/WLOOP
11DSPIN	DSKY-13	-ON DSP2DEC SGNCOM +ON
1406ALM	DESC-17	TTF/8CL
1406P00	DESC-17	TTF/8CL
1STAR	ALIN-30	ASTAR P57OPT3
1STO2S	MATX-8	INCAZ

1TO2SUB	MATX-8	1STO2S 2V1STO2S V1STO2S
2BLANK	DSKY-13	5BLANK DSPMMJOB MMCHANG NOUN REQMM
2INTOUT	DATA-14	DECDSP3
2STARS	ALIN-30	P57OPT2
2V1STO2S	MATX-8	
360-CDUO	DATA-14	DECDSP3
5BLANK	DSKY-12	2INTOUT BLANKSUB CLEAR REQDATZ
60TIMES	RNAV-15	60TIMES
79DISP	ALIN-32	INCAZ
8192AUG	IMUC-12	8192AUG STRTGYR2

A-PCHK	ORBI-16	CKMID2 TESTLOOP
ABCLOAD	DATA-19	VERBFAN
ABLOAD	DATA-19	VERBFAN
ABORT	PGSR-13	
ABORTALM	ASCT-7	P70 P71
ABRTIGN	BURN-5	IGNITION
ABRTJASK	ASCT-3	P71A
ACCOMP	ORBI-8	NBRANCH NEXTCOL
ACCTHERE	DAPB-7	BOTHAXES
ACDT+C12	DAPA-31	DONEYET2 PAXFILT
ADDGRAV	ALIN-35	GRABGRAV
ADRS+1	TEST-8	ADRSCHK
ADRSCHK	TEST-7	COMADRS

ADTIME	TRGL-8	ELCALC
ADVAN	MATX-10	CHECKNJ
ADVANCE	TRGX-7	P32/P72B P33/P73B
AFCCALC1	DESC-7	VERTGUID
AFTERTJ	DAPA-24	BACKHAND MINRTN TJLAW4
AGSDISPK	EXVB-12	AGSDISPK
AGSINIT	EXVB-12	V47TXACT
AGSVCALC	EXVB-12	AGSDISPK
ALARM	PGSR-11	INTLOOP P33/P73B CALCGA WMATEND MIDTOAV1 CKMID2 UPTMFAST DNTMFAST PRIOLARM RRDESK2 LRP2COMM V37 GOPROG3 RADAREAD LRS24.1 RRCDUCHK RRDESNB RRZERO RESAMPLE SOMEKEY YMKRUPT MARKCHEX R51 DSPOPTN R59ALM SETISSW IMUZERO COARS2 IMUCHK MARKRUPT KALCMAN3 AGSINIT PIPFREE IMUMON TNONTEST C33TEST RO2BOTH VGAIN* 1406ALM MUNRETRN P40ALM DCMCL NOATTCNT EXGSUB STEER? TRYUORV SELECTP +XORULGE P66 FAILOOP SOMEERRR SOMERR2 SURFAGAN DORSAMP
ALARM1	PGSR-14	ARCSUB
ALARM2	PGSR-11	BAILOUT BAILOUT1 CURTAINS POODOO POODOO1 VARALARM ALARM1 PRERRORS
ALFLT	TEST-16	ALLOOP
ALGORTHM	DAPA-28	NEGUSUM
ALINTIME	EXVB-6	GOEXTVB
ALLCOAST	DAPA-32	SEUDOPOO COASTSET GOPOST GOCUTOFF
ALLOOP	TEST-15	ESTIMS ALLOOP
ALM/END	EXVB-2	GOEXTVB TESTACT VBZERO IMUFINEK IMUATTCK CREWMANU R77 V89PERF GOSHOSUM VBCOARK SYSTEST RDRUSECK ATTACHED VBTSTLTS

ALMCYCLE	DATA-17	TESTNN ABLOAD ABCLOAD PUTCOM PUTNORM PUTDCSF2 BINROUND DPINSF+2 DEGINSF HMSIN MMCHANG
ALMXIT	TRGX-15	CIRCL SCNDSOL
ALOAD	DATA-18	VERBFAN
ALOADED	ORBI-4	INTEGRV
ALTDSPLY	DAPA-3	CHEKBITS
ALTOUT1	SERV-18	LANDISP
ALTROUT1	SERV-17	LANDISP
AOTMARK	ALIN-3	P51C R51E R59OUT
AOTSTALL	ALIN-3	P51C R51E R59OUT
APSIDES	CONC-2	PERIAPO
ARCCOS	MATX-3	
ARCSIN	MATX-3	
ARCSUB	MATX-3	ARCCOS ARCSIN
ARCTAN	COOR-5	LAT-LONG RANGEBQ BALLANGS
ARCTRGSP	BURN-24	NB2CDUSP
ARCTRIG	COOR-6	CALCGA CALCGTA RRANGLES RRLOSDSP
AROUT1SF	DATA-27	DECDSP3
ARTOUTSF	DATA-10	DECDSP3 LRPOSOUT
ASCENT	ASCT-7	ATMAG P12LM
ASCTERM	ASCT-11	CMPONENT
ASCTERM1	ASCT-11	ATMAG
ASTAR	ALIN-33	R59RET
ASTNRET	DESC-3	CLOKJOB
ASTOK	RNAV-37	R22LEM96
ATMAG	ASCT-7	UPTHROT ABRTIGN

ATTACHED	EXVB-8	GOEXTVB
ATTACHIT	EXVB-28	ATTACHED
ATTCHK	ALIN-26	GVDETER P57POST SURFDISP
ATTCK2	IMUC-19	IMUATTCK
ATTSTEER	DAPA-23	CHEKSTIK RHCACTIV TSNEXTS
AVEIT	ALIN-10	JUSTOA
AVERAGEG	SERV-3	
AVESTAR	ALIN-10	AVEIT
AVETOMID	ORBI-18	AVGEND
AVGEND	SERV-5	READACCS
AXISGEN	ALIN-14	INITBY R51E P51C
AZEL	ALIN-21	R52
BACKHAND	DAPA-23	BACKHAND FAILOOP FEEDBACK DOROTAT TJLAW4
BACKP	DAPA-7	
BADRAD	RADR-22	RADAREAD RENDRAD RESAMPLE
BAILOUT	PGSR-12	PAXIS MAKEPLAY MAKEPRIO NV50DSP FLASHSUB
BAILOUT1	PGSR-12	FINDVAC2 NOVAC3 GOMANUR RADSTALL DLY2 IMUSTALL AOTMARK AOTSTALL DELLOOP WTLST5
BALLANGS	ATTM-2	R60LEM R61C+LO2 REDOMANC V89RECL
BANKCALL	MATX-25	
BANKJUMP	MATX-25	
BEGDES	RADR-9	DORREPOS RR1AX2 STARTDES 60TIMES
BEGDES29	RNAV-34	BEGDES29 R29.LOS
BIGIQ	DAPB-22	COMMEQS

BINROUND	DATA-24	PUTDCSF2
BITSOFF1	DATA-21	CHANBITS
BITSOFF2	DATA-21	CHANBITS
BLANKDSP	DATA-2	NVSUB
BLANKET	DINT-6	CHKLINUS R04Z DSPRRLOS R21DISP VBCOARK SBANDEX DAPDATA1 DAPDATA2 TRIMDONE
BLANKSUB	DSKY-11	PASTEVB NV50DSP
BLOAD	DATA-18	VERBFAN
BOTHAXES	DAPB-5	STMIN-
BOTHPAD	EXVB-18	V82GOFF1
BOTHSGN	DSKY-5	NEGSGN POSGN
BRNCHCTR	CONC-5	KEPLOOP
BURNBABY	BURN-3	P12RET P40IN P41LM ASTNRET
BYLMATT	ALIN-26	DSPOPTN
C13STALL	RADR-25	WOZERO T6JOBCHK JTLST ZEROENBL STARTP64 ERROR DODOWNTM REDESMON C13STALL TSTLTS3 VBTSTLTS
C33TEST	IMUC-7	TNONTEST
CA+ECE	IMUC-9	GLOCKMON
CAGESUB	IMUC-8	TNONTEST
CAGESUB1	IMUC-8	IMUMON
CAGESUB2	IMUC-8	TNONTEST
CAL53A	ALIN-18	REGCOARS
CALCGA	COOR-6	REDO S52.2 INITBY GVDETER
CALCGRAV	SERV-6	NORMLIZE CALCRVG
CALCGTA	COOR-7	GYCOARS R55 INITBY
CALCN83	BURN-13	STARTP47

CALCN85	BURN-12	ULLGNOT POSTBURN
CALCPERR	DAPA-16	MOREIDLE PURGENCY
CALCRGVG	DESC-6	EXGSUB TTFINCR
CALCRVG	SERV-5	AVERAGEG
CALCSMSC	COOR-1	INITBY P57OPTO P57OPT1 R56 REFMF S52.2
CALCTFF	EXVB-22	SR30.1
CALCTPER	EXVB-22	SR30.1
CALLDGCH	RNAV-30	R24END R24LEM3 LRS24.1
CANV37	PGSR-5	V37 V37RET
CCSHOLE	PGSR-15	EJSCAN SPECTEST T6JOBCHK PIPJOBB
CD*TR*GS	COOR-1	RANGEBQ READRDOT AVESTAR INITBY JUSTOA BALLANGS SETPOS
CDHMVR	TRGX-9	P33/P73B CIRCL
CDULOGIC	MATX-7	
CDUTODCM	ATTM-9	KALCMAN3 VECPOINT
CDUTRIG	COOR-1	COMPDISP DODES R61C+L02 RRDESSM SBANDANT INITBY ADDGRAV P57OPTO P57OPT1 R56 R59 S52.2 S41.1 REFMF P57POST RODCOMP P57OPT3
CGCALC	DESC-8	AFCCALC1 [illegible]
CHANBITS	DATA-20	ABCLOAD
CHANG1	MATX-13	DORROUT ADRSCHK V82GON
CHANG2	MATX-14	
CHANGEVB	ALIN-9	REMARK
CHANJOB4	MATX-14	ADVAN CHANG1 EJSCAN
CHARALRM	DSKY-2	CHARIN PROCKEY
CHARIN	DSKY-1	KEYRUPT1 UPRUPT
CHECKG	TEST-14	PIPACHK PIPJOBB CHECKG

CHECKNJ	TEST-3	SELFCHK ERASLOOP ADRSCHK
CHEKBITS	DAPA-3	DAPIDLER PAXIS
CHEKSTIK	DAPA-21	TSNEXTS
CHKLINUS	ATTM-12	TOBALLA REDOMANC
CHKSDATA	ALIN-13	P51C R51E SURFLINE
CHKVISFZ	DAPA-11	SUPERJOB
CIRCL	TRGX-12	CSI/B2
CKMID2	ORBI-21	ENDSTATE
CLEAR	DSKY-6	CHARIN
CLEANDSP	DINT-3	P41BLANK V99RECYC
CLOAD	DATA-18	VERBFAN
CLOCPLAY	DINT-18	CLOKJOB V99RECYC
CLOKJOB	BURN-9	CLOKTASK
CLOKTASK	BURN-9	CLOKTASK COMPTGO STCLOK3
CLOSEOUT	DAPA-30	BACKHAND TJLAW4 FEEDBACK XTRANS FAILOOP DOROTAT
CLUPDATE	TRGX-7	VN1645
CMPONENT	ASCT-9	MAINENG
CNTRLOOP	TEST-5	CNTRLOOP
COARS	IMUC-15	IMUCOARS
COARS2	IMUC-15	COARS2
COARSE	ALIN-2	CAL53A INITBY LUNG P51 COARSE
COASTSET	BURN-8	ENGINOF2
COMADRS	TEST-7	ADRS+1 NXTBNK
COMFAIL	BURN-11	DVMON
COMFAIL2	BURN-12	CLOKJOB

COMFAIL3	BURN-12	CLOKJOB
COMFAIL4	BURN-12	CLOKJOB
COMMEQS	DAPB-22	1/ACCS
COMMINIT	ASCT-6	GOABORT
COMMNOUT	CONC-2	TIMERAD
COMPDISP	EXVB-25	GETRVN
COMPTGO	BURN-9	VN1645
CONST	ASCT-9	MAINENG
CONTMANU	ATTM-7	NEWANGL
CONTSERV	SERV-11	UPDATCHK VALTCHK MUNRETRN
COPYCYC	SERV-4	NORMLIZE AVERAGEG
COPYCYC1	SERV-11	MUNRETRN
CORFOUND	MATX-12	NOVAC3
COSINE	MATX-1	
CREWMANU	EXVB-6	GOEXTVB
CSI/A	TRGX-10	P32/P72B
CSI/B	TRGX-10	SCNDSOL
CSI/B1	TRGX-10	CIRCL FRSTPAS
CSI/B2	TRGX-10	FIFTYFPS
CSI/SOL	TRGX-15	CIRCL
CSMCONIC	ORBI-3	V89RECL LRS24.1 CSMINT
CSMINT	RNAV-8	LPS20.1
CSMPREC	ORBI-2	P41SPOT AGSVCALC V82GOFF1 DOCMBASE R36 P76 PRECSET
CSMVEC	EXVB-10	GOEXTVB

CTRCHECK	DAPB-18	POSTHRST
CURTAINS	PGSR-12	P51C COARSE GYCOARS R51E INITBY R59OUT R55
CUTOFF	ASCT-12	ENGOFF1 GOCUTOFF

DAPATTER	EXVB-7	GOEXTVB
DAPDATA1	DAPB-19	DAPDATA1 DAPDISP
DAPDATA2	DAPB-20	DAPDATA2
DAPDISP	EXVB-6	GOEXTVB
DAPIDLER	DAPA-3	STARTSB1 MOREIDLE
DAPT4S	DAPA-32	PROCEEDE
DATGDCHK	RNAV-30	CALLDGCH
DCMCL	BURN-21	FINDCDUW
DCMTOCDU	ATTM-9	FINDGIMB NEWANGL
DECDSP	DATA- 9	VERBFAN
DECDSP3	DATA-10	
DECEND	DSKY-3	NUM
DEGINSF	DATA-24	PUTDCSF2
DEGOUTSF	DATA-11	DECDSP3 360-CDUO
DELAYJOB	MATX-18	
DELCOMP	ATTM-9	VECPOINT WCALC
DELLOOP	MATX-18	DELLOOP
DELTIME	CONC-12	COMMNOUT KEPLOOP LAMBLOOP
DERCLOOP	DESC-17	DERCLOOP
DESCBITS	DESC-17	SOMEKEY
DESRETRN	RADR-7	LUNDESCH RRDESSM

DETENTCK	DAPA-13	TSNEXTP
DGCHECK	RADR-21	LRHEIGHT RADAREAD RENDRAD
DIFEQ+2	ORBI-12	NBRANCH
DISPCHNG	BURN-5	TIG-5 P40SJUNK
DISPEXIT	DESC-9	P66 P66HZ STEER? 1406ALM RODCOMP LUNLAND
DISPLAYE	TRGL-3	P34 P74 INTLOOP
DISPN5X	EXVB-24	R31CALL
DISPRSET	SERV-18	LANDISP SPEEDRUN
DLY2	MATX-20	DELLOOP WAITLIST VARDELAY
DNDUMP	TELE- 5	DNDUMP1
DNDUMP1	TELE-5	DNDUMPI3
DNDUMP2	TELE-	DNDUMP1 DNDUMP
DNDUMPI	TELE-5	DNDUMP DNEDUMP
DNDUMPI3	TELE-5	DNDUMPI
DNEDUMP	EXVB-9	GOEXTVB
DNPHASE1	TELE-2	STARTSUB DNDUMP DNPHASE2
DNPHASE2	TELE-2	DNPHASE1
DNTMFAST	TELE-1	C33TEST
DOCKED	DAPB-2	1/ACCS
DOCKTEST	DAPB-4	SPSCONT
DOCMBASE	EXVB-24	V83CALL
DODES	RADR-10	MOREDES
DODNCHAN	TELE-2	NEXTINSL DNPHASE2
DODNPTR	TELE-3	NEXTINSL DNPHASE2

DODOWNTM	TELE-1	Called via program interrupt #8
DOFSTART	PGSR-1	LIGHTSET
DOFSTRT1	PGSR-1	GOPROG3 GOPROG
DOIT	DESC-15	THROTTLE
DONEYET2	DAPA-32	TIMQGMBL
DOROTAT	DAPA-24	AFTERTJ
DORREPOS	RADR-3	RRGIMON
DORROUT	RADR-12	DODES
DORSAMP	RADR-16	RADSAMP
DOT6RUPT	DAPA-1	Called via program interrupt #1
DOW..	ORBI-14	INTGRATE NBRANCH
DPDAT1	DAPB-19	DAPDATA1
DPINSF+2	DATA-24	PUTDCSF2
DPOUT	DATA-11	DECDSP3
DSP2DEC	DSKY-11	DSPDPDEC
DSP68	EXVB-30	WAIT68 LRON
DSPA	DATA-7	DSPAB VERBFAN
DSPAB	DATA-8	VERBFAN DSPABC
DSPABC	DATA-8	VERBFAN
DSPALARM	DATA-5	ENTER NVSUB TESTNN MIXNOUN DSPA DSPB DSPC DSPAB DSPABC DECDSP3 DSPDPDEC BLOAD ABLOAD ABCLOAD PUTDCSF2 VERBFAN CLOAD
DSPB	DATA-7	VERBFAN
DSPC	DATA-7	VERBFAN
DSPCOM2	DATA-8	DSPA DSPC DSPB
DSPDC2NR	DSKY-11	M/SOUT

DSPDCEND	DATA-14	DECDSP3 ARTOUTSF DEGOUTSF M/SOUT 2INTOUT RRANGOUT RRDOTOUT AROUT1SF
DSPDCPUT	DATA-9	DSPDCEND
DSPDCWD1	DSKY-9	DSPDCWD1 DSPDC2NR DSPDECVN DSP2DEC
DSPDECVN	DSKY-11	2INTOUT UPDATNN UPDATVB DSPMMJOB
DSPDECWD	DSKY-9	HMSOUT DSPDCEND
DSPDPDEC	DATA-15	VERBFAN
DSPFMEM	DATA-26	VERBFAN
DSPIN	DSKY-13	NUM WDAGAIN DSPDCWD1 5BLANK M/SOUT
DSPIN1	DSKY-13	11DSPIN DSPIN
DSPMMJOB	DATA-17	STARTP66 UPOUT4 STARTP64 P65START REDO TSTLTS3 ENDTEST1 UPUPDATE V37XEQ GOPROG3
DSPOCTWD	DSKY-9	TESTNN DSPCOM2 DSPFMEM
DSPOPTN	ALIN-26	DSPOPTN
DSPRRLOS	RNAV-43	VERB85
DSPSCAN	INTR-4	T4RUPT QUIKDSP DSPSCAN
DSPSIGN	DSKY-10	DSPDECWD DSPDC2NR DSP2DEC
DSPV6N79	ALIN-11	YMKRUPT CHANGEVB DSPV6N79 SURFAGAN
DUMMYJB2	MATX-10	DOFSTRT1 ENDRSTRT EJSCAN
DVMON	SERV-3	AVERAGEG
DXCOMP	CONC-4	PERIODCH
DYNMDISP	BURN-12	P41LM DYNMDISP

EARTHMX	COOR-2	LAT-LONG LALOTORV
EARTHR*	TEST-15	PIPACHK PIPJOBB SLEEPIE
EJSCAN	MATX-16	PRIOCHNG JOBSLEEP EJSCAN
ELCALC	TRGL-5	ADTIME
ENDEXT	DINT-17	R62DISP V89CALL V89RECL AVEIT R04END V67CALL RRLOSDSP VBCOARK IMUFINEK IMUATTCK ALINTIME V73UPDAT AGSINIT AGSDISPK AGSVCALC UPOUT4 COMPDISP DAPDATA2 DAPDATA1 TRIMDONE SDISPLAY WAIT68 SBANDEX ENDTEST1 ENDR03 V82CALL TICKTEST V82GON2 R36
ENDIMU	IMUC-18	IMUZERO2 COARS2 IMUFINED
ENDINT	ORBI-2	STATINT1
ENDLRV	SERV-13	LRVJOB
ENDMANU1	ATTM-2	TOBALLA R61TEST
ENDMANUV	ATTM-2	
ENDOFJOB	MATX-16	DORROUT LRS24.1 OMEGCALC DATGDCHK TESTNN DSPCOM2 HMSOUT DSPDPDEC MONITOR ALMCYCLE OKTOPLAY MAKEMARK OKTOCOPY XCHSLEEP NOUN ENTER R61C+L06 DSPDCEND
ENDP76	ORBI-25	P76
ENDPRCHG	MATX-15	EJSCAN
ENDRADAR	RADR-18	STDESIG RRZ2
ENDRET	DINT-16	NORMWAKE MARKWAKE
ENDR29RD	RNAV-33	R29RDJOB
ENDRRD29	RNAV-33	R29RDJOB R29RANGE
ENDR03	DAPB-21	DAPDATA2
ENDRSTRT	PGSR-11	GOPROG3
ENDSTATE	ORBI-13	DIFEQ+2

ENDTEST1	TEST-14	REDO SHOW SOMEERRR
ENDTFF	EXVB-23	CALCTFF
ENDTNON	IMUC-10	TNONTEST ENDTNON
ENEMA	PGSR-10	WHIMPER ABRTJASK
ENGINOF1	BURN-8	SEUDOPOO
ENGINOF2	BURN-8	ENGOFF1
ENGINOF3	BURN-8	BURNBABY
ENGINOF4	BURN-8	COMFAIL2
ENGINOFF	BURN-7	ENGOFTSK
ENGOFF1	ASCT-11	MAINENG
ENGOFTSK	BURN-7	P42IGN STEERING
ENTER	DSKY-6	CHARIN
ENTERUV	DAPA-22	RHCACTIV
ENTPASO	DATA-3	ENTER NVSUB ALMCYCLE
ERASCHK	TEST-3	SELFCHK
ERASLOOP	TEST-4	ERASLOOP
ERROR	DSKY-7	CHARIN
ESTIMS	TEST-11	PIPJOBB GOESTIMS REDO
EXBRAK	DESC-8	CGCALC
EXDSPRET	DINT-4	VBCOARK IMUFINEK IMUATTCK AGSVCALC
EXGSUB	DESC-2	CGCALC
EXNORM	DESC-8	CGCALC

EXTLOGIC	DESC-8	AFCCALC1
FAILOOP	DAPA-27	SELCTSUB FAILOOP
FAZAB3	RNAV-40	FAZC
FAZC	RNAV-39	INCORP2
FEEDBACK	DAPA-26	DOROTAT
FETCH2WD	TELE-3	DNPHASE2 NEXTINSL
FIFTYFPS	TRGX-14	CIRCL
FINDCDUW	BURN-20	ASCTERM STEER? VGAIN* P66HZ
FINDGIMB	ATTM-4	VECPOINT
FINDVAC	MATX-11	
FINDVAC2	MATX-11	SPVAC
FIRSTTME	BURN-18	RASTEER1
FIXDELAY	MATX-24	
FLASHSUB	DINT-13	NVDSP NV50DSP
FLATOUT	DESC-15	P63ZOOM
FRSTPAS	TRGX-14	CIRCL
FXADRS	TEST-7	ADRS+1 NXTBNK
GAMCOMP	ORBI-10	ACCOMP
GEOM	CONC-9	LAMBERT PARAM
GET.LVC	TRGX-15	S40.1 S40.1B

GET+MGA	TRGX-15	VN1645
GETAZEL	ALIN-21	R52
GETCADR	MATX-24	LONGCYCL
GETCOMP	DATA-21	BLOAD CLOAD ABLOAD ABCLOAD PUTNORM PUTCOM
GETDAT	ALIN-4	AOTMARK GETDAT PASTIT
GETI	DATA-22	PUTCOM DEGINSF
GETINREL	DSKY-4	5BLANK NUM POSGN CLEAR +ON -ON NEGSGN
GETLMATT	ALIN-26	DSPOPTN
GETMKS	ALIN-6	MARKCHEX SURFAGAN OPTAXIS
GETRANS	BURN-14	UPDATEVG
GETRVN	EXVB-26	V83CALL REV83
GETX	CONC-10	TIMERAD TIMETHET LAMBLOOP
GLOCKMON	IMUC-8	PROCEEDE
GOABORT	ASCT-4	ABRTJASK
GOBAQUE	ORBI-12	GAMCOMP OBLATE ENDSTATE
GOCUTOFF	BURN-11	*ENTER
GODSP	DINT-1	ASCTERM1
GODSPR	DINT-1	REDOMANC GYCOARS
GODSPRET	DINT-1	P51 P41LM
GODSPRS1	DINT-5	GODSPR GOFLASHR GOPERF1R GOPERF2R GOPERF4R REGODSPR REFLASHR GOXDSPR GOXDSPFR GOMARK2R PRIODSPR
GOESTIMS	TEST-11	REDO
GOEXTVB	EXVB-1	VERBFAN

GOFLASH	DINT-1	R62DISP V89CALL V89RECL PLANET CHKSDATA DISPLAYE P52B R59 GVDETER R59ALM 79DISP INITBY DSPOPTN VN0611 P34 R55 R52 P52D REDO SHOW ORBCHGO PROG21 REP40ALM P12LM CUTOFF LANDJUNK GOTOPOOH P76 N89DISP GETAZEL P74 INTLOOP S34/35.5 R51 P21VSAVE P57OPT P30 VN0611 P32/P72F P33 P73 P33/P73B P33/P73F VN1645 ALMXIT NTARGCHK VN0655
GOFLASH2	DINT-5	GODSP GODSPRET GOFLASH GOPERF1 GOPERF2 GOPERF4 REGODSP REFLASH CLEANDSP GOXDSP EXDSPRET GOXDSPF GOMARK2 GOMARK3 GOMARK4 KLEENEX PRIODSP CLOCPLAY
GOFLASHR	DINT-1	POSTBURN P47BODY
GOLOADV	EXVB-6	GOEXTVB
GOMANUR	ATTM-1	REDOMANC
GOMARK2	DINT-4	
GOMARK2R	DINT-4	
GOMARK3	DINT-4	AGSVCALC
GOMARK3R	DINT-4	TRIMDONE WAIT68
GOMARK4	DINT-5	PASTIT R04X
GOODMANU	ATTM-8	NOGO
GOODRAD	RADR-22	LRHEIGHT RADAREAD RENDRAD
GOPERF1	DINT-1	P51 P52D R51 R51K P57POST P20LEMB7 R23LEM P40AUTO ASTNRET P06
GOPERF1R	DINT-2	
GOPERF2	DINT-2	
GOPERF2R	DINT-2	TOBALLA R21DSP
GOPERF4	DINT-2	ORBCHGO PROG21
GOPERF4R	DINT-2	P52B P57OPT

GOPOST	BURN-11	*ENTER
GOPROG	PGSR-9	Called via program interrupt #11 VERB69
GOPROG2	PGSR-10	TRMTRACK SEUDOPOO V37RET RESET22
GOPROG2A	PGSR-10	
GOPROG3	PGSR-10	GOPROG
GOSHOSUM	EXVB-10	GOEXTVB
GOTOPOOH	PGSR-11	TERMASC P12LM TERM40 STOPCLOK P40AUTO P21VSAVE P47BODY SERVIDLE ASTNRET P64DISPS LANDJUNK N89DISP ENDP76 P34 P74 INTLOOP DISPLAYE S34/35.5 GETAZEL VERTDISP P30 P57OPT KILLAOT P32/P72F P33 P73 P33/P73B P33/P73F VN1645 VN0611 N45PROC ALMXIT NTARGCHK GOPROG3 POSTAND ORBCHGO PROG21 GVDETER DSPOPTN INITBY REP40ALM P57POST 79DISP R59ALM R59 R55 R52 R51K R51 REGCOARS P52D P52B CHKSDATA VN0655 R61TEST R02BOTH P51 P51C IMUCHK
GOXDSP	DINT-3	
GOXDSPF	DINT-4	SDISPLAY VBCOARK IMUFINEK IMUATTCK AGSDISPK OHWELL1 OHWELL2 UPVERIFY V82CALL V82GOFLP V82GON DISPN5X R36 ALINTIME R04X DSPV6N79 GETDAT V67CALL
GOXDSPFR	DINT-4	DAPDATA1 DAPDATA2 VBCOARK SBANDEX DSPRRLOS R04Z DSP68
GOXDSPR	DINT-3	
GRABGRAV	ALIN-35	GREED
GREED	ALIN-35	ADDGRAV
GTS	DAPA-28	QRAXIS TRYGTS
GTSQRT	DAPA-34	RSTOFGTS
GUILDRET	DESC-4	IGNALOOP
GVDETER	ALIN-33	BYLMATT GVDETER
GYCOARS	ALIN-17	P52D
GYROEXIT	IMUC-13	STRTGYR2 8192AUG

HAVEGUES	TRGL-9	S40.9
HIENERGY	CONC-8	LAMBLOOP
HIGATJOB	SERV-7	MUNRETRN
HMSIN	DATA-25	PUTDCSF2
HMSOUT	DATA-11	DECDSP3
IDLERET3	DINT-15	RECALTST
IFAILOK	IMUC-17	IMUFINE
IGNALOOP	DESC-2	EXGSUB
IGNITION	BURN-5	*PROCEED TIG-0
IMUATTCK	EXVB-5	GOEXTVB
IMUBAD	IMUC-18	ENDTNON STRTGYRO 8192AUG COARS COARS2 IMUZERO2 ENDIMU IMUFINED
IMUCHK	ALIN-36	P51 P57
IMUCOARS	IMUC-14	REDO PIPJOBB VBCOARK COARSE
IMUFINE	IMUC-17	REDO IMUFINEK COARSE
IMUFINED	IMUC-17	STRTGYR2 IMUFINE
IMUFINEK	EXVB-4	GOEXTVB
IMUGOOD	IMUC-18	
IMUMON	IMUC-4	PROCEEDE
IMUPULSE	IMUC-10	EARTHR* PERFERAS IMUFINEK 1/GYRO STRTGYR2 GYCOARS R55 INITBY
IMUSTALL	IMUC-17	REDO PIPJOBB PERFERAS VBCOARK IMUFINEK AGSVCALC VBZERO 1/GYRO COARSE R55 INITBY GYCOARS EARTHR*

IMUZERO	IMUC-13	REDO AGSVCALC VBZERO
IMUZERO2	IMUC-14	IMUZERO
INCAZ	ALIN-31	INCAZ
INCOR2-3	RNAV-37	NEWZCOMP
INCORP1	RNAV-37	LGCUPDTE
INCORP2	RNAV-38	ASTOK
INFINITY	CONC-12	WLOOP 1/WLOOP POLYCOEF
INITBY	ALIN-28	SURFLINE
INITCDUW	BURN-20	GOABORT P41SPOT
INITREAD	RADR-17	LRALT LRVELX LRVELY LRVELZ RRRDOT RRRANGE
INITV	CONC-8	LAMBLOOP SUFFCHEK
INITVEL	TRGL-9	S40.1B S34/35.2
INITVEL2	TRGL-10	INITVEL2
INITVEL7	TRGL-11	INITVEL2
INJTARG	ASCT-5	GOABORT
INTEGRV	ORBI-3	STATINT1 CSMPREC LEMPREC CSMCONIC LEMCONIC DIFEQ+2 AVETOMID MIDTOAV2 LSR22.3 LSR22.4 ORBCHGO UPPSV PROG21 UPPSV4
INTEGRVS	ORBI-4	EXGSUB P76 INITVEL2 INTINT P21CONT OTHINT REV83
INTEXIT	ORBI-5	TESTLOOP
INTGRATE	ORBI-8	TIMESTEP LUNSPH
INTINT	TRGX-9	ADTIME S34/35.2 S34/35.3 P32/P72B P33/P73B CIRCL
INTLOOP	TRGL-1	INTLOOP
INTLZE	SERV-15	LANDISP
INTSTALL	ORBI-21	EXGSUB STATINT1 CSMPREC LEMPREC CSMCONIC LEMCONIC AVETOMID MIDTOAV2 P76 INITVEL2 INTINT UPJOB GETRVN OTHCONIC ATTACHIT CANV37 INCORP2 LSR22.3 LSR22.4 UPPSV V67CALL ORBCHGO INTSTALL TRMTRACK REV83 PROG21

INTWAKE	ORBI-21	INTEXIT AVETOMID ATTACHIT FAZAB3 V67CALL
INTWAKE1	ORBI-22	INTWAKEU P76 ORBCHGO
INTWAKEU	ORBI-22	UPEND70 UPEND71 UPEND72 UPERROUT
INVRSEQN	CONC-11	GETX WLOOP
ISSZERO	IMUC-7	IMUMON TNONTEST
ITERATOR	CONC-12	LAMBLOOP
ITURNON2	IMUC-10	IMUMON

JAMTERM	DINT-18	NVDSP
JETSOFF	DAPA-17	TSNEXTP DETENTCK PURGENCY PJETSLEC SELECTP
JOBSLEEP	MATX-15	REDO TCGETCAD
JOBWAKE	MATX-17	WAKER
JOBWAKE4	MATX-17	JOBWAKE4
JOBXCHS	DINT-9	MAKEMARK MAKEPRIO
JTLST	DAPA-2	PJETSLEC DOROTAT

JUSTOA	ALIN-12	SURFSTAR

KALCMAN3	ATTM-4	GOMANUR
KEPCONVG	CONC-5	KEPLOOP BRNCHCTR
KEPLERN	CONC-3	KEPPREP
KEPLOOP	CONC-4	BRNCHCTR
KEPPREP	ORBI-15	RVCON GOBAQUE NBRANCH
KEYRUPT1	DSKY-1	Called via program interrupt #5
KILLAOT	ALIN-36	GETDAT PASTIT DSPV6N79
KLEENEX	DINT-5	

LALOTORV	COOR-4	N89DISP
LAMBERT	CONC-6	INITVEL2
LAMBLOOP	CONC-7	HIENERGY LAMBLOOP
LAMENTER	CONC-12	INITV
LANDISP	SERV-15	R10,R11
LANDJUNK	DESC-18	
LASTBIAS	IMUC-3	PREREAD
LAT-LONG	COOR-3	LANDJUNK N89DISP P21VSAVE
LEMCONIC	ORBI-3	S52.3 V89RECL LPS20.1 LRS24.1 SBANDANT
LEMPREC	ORBI-2	LOCSAM P57D AGSVCALC V82GOFF1 V83CALL REV83 R36 S30.1 PRECSET P12LM S40.1B P63LM
LEMVEC	EXVB-9	GOEXTVB
LGCUPDTE	RNAV-36	RANGEBQ
LIGHTSET	PGSR-10	GOPROG GOPROG2A

LITIT	RADR-25	ONLITES
LOADLV	DINT-14	ALOAD BLOAD CLOAD ABLOAD ABCLOAD PUTNORM HMSIN GOLOADLV BITSOFF2
LOCSAM	ALIN-13	PLANET R51
LODNNTAB	DATA-4	TESTNN ALOAD BLOAD CLOAD ABLOAD ABCLOAD UPDATNN
LOENERGY	CONC-8	LAMBLOOP
LOGSUB	MATX-6	
LONGCALL	MATX-23	
LONGCYCL	MATX-23	LONGCALL LONGCYCL
LPS20.1	RNAV-8	P20LEM1 R21LEM1 R61C+L02 READRDOT 60TIMES
LRALT	RADR-17	DORSAMP LRHJOB
LRHEIGHT	RADR-20	RADAREAD
LRHJOB	SERV-12	R10,R11
LROFF	EXVB-7	GOEXTVB
LRON	EXVB-7	GOEXTVB
LRP2COMM	EXVB-30	LRPOS2K1
LRPOS2	RADR-23	LRP2COMM HIGATJOB
LRPOS2K	EXVB-7	GOEXTVB
LRPOS2K1	EXVB-7	LRPOS2K
LRPOSCAN	RADR-23	LRPOS2 LRPOSCAN
LRPOSOUT	DATA-11	DECDSP3
LRS22.1	RNAV-21	R22LEM
LRS24.1	RNAV-28	DATGDCHK R24LEM

LRVELX	RADR-17	DORSAMP LRVJOB
LRVELY	RADR-17	DORSAMP LRVJOB
LRVELZ	RADR-17	DORSAMP LRVJOB
LRVJOB	SERV-12	VALTCHK
LSORIENT	ALIN-18	P52LS P57D
LSPOS	COOR-7	LOCSAM SBANDANT TIMESTEP LUNSPH ACCOMP
LSR22.3	RNAV-23	R22LEM
LSR22.4	RNAV-24	LSR22.3
LUNDESCH	RADR-24	RRANGLES
LUNG	ALIN-34	GVDETER
LUNLAND	DESC-4	P63IGN
LUNSPH	ORBI-7	TIMESTEP

M/SOUT	DATA-13	DECDSP3
MAINENG	ASCT-8	
MAKECADR	MATX-26	
MAKEMARK	DINT-7	MAKEPLAY
MAKEPLAY	DINT-6	GOFLASH2 GODSPRS1 BLANKET
MAKEPRIO	DINT-8	MAKEPLAY
MANUSTOP	ATTM-8	NEWANGL

MARKCHEX	ALIN-9	PASTIT SURFAGAN
MARKPLAY	DINT-8	MAKEMARK JOBXCHS XCHSLEEP NV50DSP MARKWAKE PINBRNCH
MARKRET	DINT-16	TERMATE
MARKRUPT	ALIN-6	Called via program interrupt #6
MARKWAKE	DINT-17	NORMRET
MFREF	COOR-8	SAMETYP P570PT3
MIDTOAV1	ORBI-19	P41SPOT
MIDTOAV2	ORBI-20	P47LM
MINIMP	EXVB-9	GOEXTVB
MINQR	DAPA-21	TSNEXTS
MINRTN	DAPA-21	FAILOOP FEEDBACK DOROTAT TJLAW4
MIXNOUN	DATA-5	TESTNN
MKRELEAS	ALIN-4	AVEIT
MMCHANG	DATA-26	ENTER VERBFAN
MODE2CHK	RADR-7	RRLIMNB
MONDO	DATA-16	MONREQ
MONITOR	DATA-15	VERBFAN
MONREQ	DATA-15	MONITOR MONREQ
MOONMX	COOR-3	P12LM P63LM LANDJUNK OBLATE USEPIOS LAT-LONG LALOTORV MFREF REFMF P52LS P57POST
MOREDES	RADR-10	STDESIG
MOREIDLE	DAPA-3	DAPIDLER PAXIS CHEKBITS
MOVEACSM	ORBI-17	ENDSTATE A-PCHK INTWAKEU P76 ORBCHGO FAZAB3
MOVEALEM	ORBI-18	ENDSTATE A-PCHK AVETOMID INTWAKEU FAZAB3

MOVEPCSM	ORBI-16	INTEGRV FAZC
MOVEPLEM	ORBI-17	INTEGRV FAZC ATTACHIT
MUNGRAV	SERV-7	P12LM P41SPOT NORMLIZE RVBOTH NOREASON IGNALOOP
MUNRETRN	SERV-7	RVBOTH
N45PROC	TRGX-6	VN1645
N49DSP	RNAV-19	R22LEM96
N89DISP	ALIN-16	P52LS P57POST
NB2CDUSP	BURN-23	DCMCL
NBD2	IMUC-2	LASTBIAS
NBDONLY	IMUC-1	SVCT3
NBRANCH	ORBI-12	ACCOMP DOW..
NBTOSM	COOR-1	VMEASCHK COMPDISP DODES RANGEBQ AVESTAR JUSTOA P57OPT1 P57OPT3 INITBY GVDETER P57POST RODCOMP
NEEDLER	DAPA-4	ALTDSPLY
NEEDLES	DAPA-5	
NEGPROD	BURN-18	RASTEER1
NEGSGN	DSKY-4	CHARIN
NEGUSUM	DAPA-29	GTS ALGORTHM NEGUSUM
NEWANGL	ATTM-6	NEWDELHI
NEWDELHI	ATTM-8	UPDTCALL
NEWSTATE	CONC-12	COMMNOUT
NEWZCOMP	RNAV-38	INCORP1
NEXTCOL	ORBI-14	DIFEQ+2

NEXTINSL	TELE- 3	DNPHASE2
NOATTCNT	BURN-23	FINDCDUW DCMCL
NOGO	ATTM-8	KALCMAN3 NEWDELHI
NOMINIMP	EXVB-9	GOEXTVB
NOR29NOW	SERV-11	COPYCYC1 R29 R29.LOS
NOREASON	SERV-9	UPDATCHK
NORMBNCH	DINT-18	PINBRNCH
NORMLIZE	SERV-2	PREREAD
NORMRET	DINT-16	TERMATE ENDEXT
NORMWAKE	DINT-17	NORMRET
NORRGMON	RADR-2	RRCDUCHK RRGIMON RRAUTCHK
NOTMUCH	DAPB-6	BOTHAXES
NOUN	DSKY-2	CHARIN
NOVAC	MATX-12	
NOVAC2	MATX-12	FINDVAC2
NOVAC3	MATX-12	NOVAC3
NTARGCHK	TRGL-4	NTARGCHK
NUM	DSKY-2	CHARIN
NV50DSP	DINT-11	NV50DSP
NVDSP	DINT-11	NV50DSP TERMATE NORMBNCH PLAYJUM1 MARKPLAY REDOPRIO
NVSUB	DATA-1	NV50DSP ENDRET PINBRNCH TESTXACT
NVSUBEND	DATA-2	NVSUB DSPALARM
NXTBNK	TEST-8	SDISPLAY

OANB	ALIN-5	AZEL INGAZ OPTAXIS
OBLATE	ORBI-10	ACCOMP
OCCULT	ALIN-24	PIC3
OHWELL1	EXVB-14	UPUPDATE OHWELL1
OHWELL2	EXVB-14	UPUPDATE OHWELL2 UPVERIFY
OKTOCOPY	DINT-9	JOBXCHS
OKTOPLAY	DINT-6	MAKEPLAY
OMEGCALC	RNAV-30	
ONLITES	RADR-25	RADLITES
OPTAXIS	ALIN-5	GETDAT
ORBCHGO	RNAV-1	
ORIGCHNG	ORBI-7	TIMESTEP LUNSPH
OTHCONIC	EXVB-25	REV83
OTHINT	EXVB-25	GETRVN
OUTSNUFF	EXVB-9	GOEXTVB

P06	PGSR-13	P06
P12IGN	BURN-5	IGNITION
P12INIT	ASCT-6	P12LM GOABORT
P12LM	ASCT-1	
P12RET	ASCT-2	CMPONENT
P1CHK	SERV-8	POSALARM
P20LEM1	RNAV-3	R24END CSMINT P20LEM1
P20LEMA	RNAV-3	R21LEM1 R22LEM
P20LEMB	RNAV-4	P20LEMB1 P20LEMB7
P20LEMB1	RNAV-4	R23LEM
P20LEMB3	RNAV-4	P20LEMB7 P20LEMB3
P20LEMB7	RNAV-4	R22LEM
P20LEMC	RNAV-5	R22LEM R22RSTRT UPPSV4
P20LEMC1	RNAV-5	RELINUS V37RET P20LEMWT R22WAIT P20LEMC1
P20LEMC3	RNAV-5	P20LEMD1 R21LEM1 R21LEM
P20LEMD1	RNAV-5	P20LEMC P20LEMD1
P20LEMF	RNAV-5	P20LEMC
P20LEMWT	RNAV-5	P20LEMB3 R21DISP R22WAIT P20LEMB R61C+L02 R21LEM1
P21CONT	RNAV-7	PROG21
P21VSAVE	RNAV-6	P21CONT
P25LEM1	RNAV-6	V37RET P25LEM1 RELINUS
P2CHK	SERV-8	POSALARM
P30	TRGX-1	
	TRGL-11	
P32	TRGX-1	
P32/P72B	TRGX-2	P32/P72F
P32/P72C	TRGX-2	CSI/SOL
P32/P72E	TRGX-3	P32/P72E P32/P72C
P32/P72F	TRGX-3	P32/P72F

P33	TRGX-4	
P33/P73B	TRGX-4	P33/P73F
P33/P73E	TRGX-5	P33/P73E
P33/P73F	TRGX-5	P33/P73F
P34	TRGL-1	
P34/P74C	TRGL-1	INTLOOP
P35	TRGL-4	
P35/P75B	TRGL-4	P35/P75B

P40ALM	BURN-2	P40LM P42LM
P40AUTO	BURN-13	UPTHROT BURNBABY P40AUTO
P40IGN	BURN-6	IGNITION
P40IN	BURN-1	P40LM
P40LM	BURN-1	
P40SJUNK	BURN-4	TIG-5
P40SPOT	BURN-3	BURNBABY
P40ZOOM	BURN-7	
P41BLANK	BURN-4	TIG-35
P41LM	BURN-2	
P41SPOT	BURN-3	BURNBABY
P42IGN	BURN-6	P63IGN ABRTIGN IGNITION
P42LM	BURN-1	
P42STAGE	BURN-1	REP40ALM

P47BODY	BURN-13	STARTP47 P47BODY
P47LM	BURN-13	
P51	ALIN-1	P51 P51C
P51B	ALIN-1	P51
P51C	ALIN-1	P51C
P52B	ALIN-15	
P52D	ALIN-16	P52B P52D P52LS
P52LS	ALIN-15	
P57	ALIN-25	
P57D	ALIN-26	P57OPT
P57OPT	ALIN-25	P57OPT
P57OPT0	ALIN-27	ATTCHK
P57OPT1	ALIN-27	ATTCHK
P57OPT2	ALIN-28	ATTCHK
P57OPT3	ALIN-28	ATTCHK
P57POST	ALIN-36	SURFLINE INITBY SURFDISP
P63DISPS	DESC-9	DISPEXIT
P63IGN	BURN-6	IGNITION
P63LM	DESC-1	
P63ZOOM	BURN-7	ZOOM
P64DISPS	DESC-9	DISPEXIT P64DISPS
P65START	DESC-5	
P66	DESC-11	LUNLAND
P66HZ	DESC-11a	P66
P66VERT	DESC-12	VERTGUID P66HZ

P70	ASCT-3	V37
P70A	ASCT-3	R10,R11
P71	ASCT-3	V37
P71A	ASCT-3	P70A R10,R10
P72	TRGX-2	
P73	TRGX-4	
P74	TRGL-1	
P75	TRGL-4	
P76	ORBI-23	

PACKOPTN	ALIN-26	P57OPT
PARAM	CONC-9	TIMERAD APSIDES TIMETHET
PASTEVB	DATA-16	MONDO DSPALARM
PASTIT	ALIN-6	CHANGEVB
PAXFILT	DAPA-10	PAXIS
PAXIS	DAPA-6	STARTDAP SPSCONT
PERFERAS	TEST-17	ALFLT
PEGI	DAPA-15	RATERROR
PERIAPO	CONC-13	INTLOOP CSI/B2
PERIAPO1	CONC-13	S30.1 CIRCL
PERIODCH	CONC-3	PERIODCH
PFAILOK	IMUC-10	UNZ2
PFLITEDB	DAPB-18	P12RET P40IN ASTNRET

PIC1	ALIN-22	PIC1 PIC3
PIC3	ALIN-22	PIC3
PICEND	ALIN-24	PIC1
PINBRNCH	DINT-17	TSTLTS3 ABORTALM VBRELDSP IDLERET3 ALM/END VBZERO RRDESEND TRMTRACK LROFF LRP2COMM DAPATTER TOTATTER SNUFFOUT CSMWEG DNEDUMP OUTSNUFF MINIMP NOMINIMP R77END WMATRXNG UPDATOFF ATTACHIT V37BAD R04Z RATEDSP V59GP63
PIPACHK	TEST-12	REDO TORQUE
PIPASR	SERV-1	PREREAD READACCS NBDONLY LUNG GRABGRAV
PIPATASK	TEST-12	PIPACHK PIPATASK
PIPFREE	IMUC-3	AVGEND
PIPJOBB	TEST-13	PIPATASK
PIPUSE	IMUC-3	LASTBIAS
PITCHOFF	DAPB-21	TRIMGIMB
PITFALL	DESC-16	Called via program interrupt #10
PJETSLEC	DAPA-16	TSNEXTP PEGI PURGENCY
PLANET	ALIN-12	P51C R51E AZEL R59RETASSTAR
PLAYJUM1	DINT-7	OKTOPLAY XCHSLEEP NV50DSP NORMRET NORMWAKE NORMBNCH
POLYCOEF	CONC-11	WLOOP
POODOO	PGSR-12	[illegible]MANUR AOTMARK T4D6P00 DSPALARM [illegible] [illegible] FLASHSUB POODOO1 CCSHOLE GOBAQUE COMMNOUT TIMETHET TIMERAD
POODOO1	PGSR-12	SQRT DLY2 LONGCALL
POOH	PGSR-6	CANV37
POSALARM	SERV-8	P2CHK
POSGN	DSKY-4	CHARIN
POSGOOD	SERV-8	P1CHK P2CHK
POSTAND	PGSR-13	POSTAND P06

POSTBURN	BURN-8	ENGINOFF GOPOST POSTBURN
POSTHRST	DAPB-17	SPSRCS SPSSTART
POSTJUMP	MATX-25	
PRECSET	TRGX-8	INTLOOP P35/P75B ADVANCE VNO611
PREPOS29	RNAV-32	R29
PREREAD	SERV-1	STARTP47 NEIBNOT
PRERRORS	TEST-6	ERASLOOP CNTRLOOP SOPTION
PRIOCHNG	MATX-15	RELINUS SHOWSUM2
PRIODSP	DINT-5	R22LEM N49DSP
PRIODSPR	DINT-5	PRIOLARM R24LEM
PRIOLARM	DINT-18	POSALARM P20LEM1 R21LEM1 R22LEM R23LEM P20LEMB7 R21LEM CSMINT 60TIMES
PROCEED	DINT-15	RECALTST
PROCEEDE	INTR-3	T4RUPT
PROCKEY	INTR-3	PROCEEDE
PROG20	RNAV-1	
PROG20A	RNAV-3	PROG20 ORBCHGO
PROG21	RNAV-6	P21VSAVE
PROG22	RNAV-1	
PROG25	RNAV-7	
PROG52	ALIN-14	
PROGLARM	PGSR-11	ALARM2
PURGENCY	DAPA-16	TSNEXTP DETENTCK RATERROR
PUTCOM	DATA-21	ALOAD BLOAD CLOAD ABLOAD ABCLOAD
PUTCOM2	DATA-25	PUTCOM PUTNORM BINROUND DPINSF+2 DEGINSF

PUTDCSF2	DATA-23	PUTCOM PUTNORM
PUTNORM	DATA-22	PUTCOM
QERRCALC	DAPA-18	MOREIDLE QRAXIS
QRAXIS	DAPA-18	SKIPPAXS PJETSLEC JETSOFF
QRTIME	DAPA-23	ENTERUV
QUADGUID	DESC-7	
QUICTRIG	BURN-23	FINDCDUW SERVICER VMEASCHK R29DODES
QUIKDSP	INTR-4	T4RUPT
R02BOTH	IMUC-19	V89CALL PROG52 P12LM P40LM P47LM P63LM P42STAGE P41LM PROG20A PROG25
R04	EXVB-7	GOEXTVB
R04END	RADR-15	R04Z R04X
R04X	RADR-13	R04X
R04Z	RADR-13	R04 R77
R10,R11	SERV-14	R10,R11 READACCS ABRTJASK
R21DISP	RNAV-16	R21LEM1
R21LEM	RNAV-13	P20LEMF
R21LEM1	RNAV-13	
R21LEM2	RNAV-13	DORROUT
R21LEM4	RNAV-14	R21LEM1
R21LEM8	RNAV-16	60TIMES
R21LEM9	RNAV-16	R21LEM8
R21LEM10	RNAV-13	R21LEM R21LEM9
R22LEM	RNAV-17	R22LEM96 R22RSTRT R22LEM R22LEM42
R22LEM42	RNAV-16	P20LEMC1 R22LEM
R22LEM96	RNAV-18	LGCUPDTE
R22RSTRT	RNAV-19	
R22WAIT	RNAV-18	R22LEM

R23LEM	RNAV-19	P20LEMB7 R23LEM3
R23LEM2	RNAV-20	R23LEM
R23LEM3	RNAV-20	R23LEM
R24END	RNAV-20	R24LEM
R24LEM	RNAV-20	R24LEM3 R21LEM R21LEM1
R24LEM3	RNAV-20	R24LEM
R29	RNAV-36	COPYCYC1
R29.LOS	RNAV-36	R29
R29DODES	RNAV-34	BEGDES29 R29DODES
R29DPAS2	RNAV-35	R29DODES
R29RANGE	RNAV-33	R29RANGE
R29RDJOB	RNAV-32	R29READ
R29READ	RNAV-32	R29DPAS2 R29READ
R29REMOJ	RNAV-32	R29
R31CALL	EXVB-23	V83PERF R31CALL
R36	EXVB-26	V90PERF R36
R51	ALIN-19	P52B REGCOARS R51 R51K ASTNRET
R51E	ALIN-19	R51 R51E
R51K	ALIN-20	GYCOARS
R52	ALIN-21	R51E AZEL
R55	ALIN-22	R51E
R56	ALIN-22	R51
R59	ALIN-30	79DISP R59ALM R59
R59ALM	ALIN-33	STOREOS
R59OUT	ALIN-33	R59 R59ALM

R59RET	ALIN-33	R59OUT
R60LEM	ATTM-1	R62DISP V89RECL R61C+L02 P40IN P41LM ASTNRET AZEL
R61C+L01	RNAV-10	R61C+L06
R61C+L02	RNAV-11	R61C+L01 R61LEM
R61C+L06	RNAV-12	R61C+L02
R61LEM	RNAV-10	P20LEMA R24LEM3 R22LEM R23LEM3
R61TEST	ATTM-2	TOBALLA
R62DISP	ATTM-11	CREWMANU
R65LEM	RNAV-10	P25LEM1 R22LEM R22LEM42 RANGEBQ
R77	EXVB-9	GOEXTVB
R77CHECK	RADR-24	RADAREAD
R77END	EXVB-9	GOEXTVB
RADAREAD	RADR-19	Called via program interrupt #9
RADLITES	RADR-25	GOODRAD RESAMPLE
RADSAMP	RADR-15	R04X R04Z RADSAMP
RADSTALL	RADR-18	R29RDJOB DORSAMP R29RANGE R22RSTRT R21LEM1 R29REMOJ READRDOT R21LEM VBZERO RRDESK2 LRP2COMM HIGATJOB LRHJOB LRVJOB R61C+L01
RADSTART	RADR-24	INITREAD RESAMPLE RADSTART
RANGEBQ	RNAV-26	LSR22.3 LSR22.4
RASTEER1	BURN-17	S40.8
RATEDAMP	DAPA-15	DETENTCK
RATEDISP	EXVB-7	GOEXTVB
RATELOOP	DAPA-6	RATELOOP SUPERJOB
RATERROR	DAPA-14	DETENTCK RATEDAMP
RCS	DAPA-18	QRAXIS TRYGTS
RCSMONIT	DAPA-33	PROCEEDE
RDBADEND	RADR-19	BADRAD ENDRADAR LRPOSCAN STDESIG
RDGIMS	SERV-13	LRVJOB
RDRUSECK	EXVB-29	VBZERO VBCOARK LRPOS2K1 R04 R77

READACCS	SERV-1	READACCS
READRDOT	RNAV-21	READRDOT
RECALTST	DINT-15	VBRESEQ LOADLV
RECTEST	ORBI-8	TIMESTEP LUNSPH
RECTIFY	ORBI-5	INTEGRVS RVCON RECTOUT ORIGCHNG RECTEST GOBAQUE FAZC
RECTOUT	ORBI-4	A-PCHK
REDESIG	DESC-5	TTFINCR
REDESMON	DESC-16	PITFALL REDESMON
REDO	TEST-10	SYSTEST
REDOMANC	ATTM-1	TOBALLA
REDOPRIO	DINT-9	NV50DSP
REFLASH	DINT-3	CLOKJOB P64DISPS VERTDISP
REFLASHR	DINT-3	TIGNOW
REFMF	COOR-8	LANDJUNK GETLMATT SURFDISP
REGCOARS	ALIN-5	P52D
REGODSP	DINT-2	TIG-30A CLOKJOB P63DISPS P64DISPS
REGODSPR	DINT-3	
REJECT	ALIN-9	MARKRUPT
RELDSP	DINT-13	TSTLTS3 ABORTALM VBRELDSP VERBFAN VBRESEQ VBRQEXEC LOADLV RECALTST V37BAD POOH V37XEQ VBRQWAIT
RELDSP1	DINT-13	VBRELDSP MONITOR
RELINUS	ATTM-12	CHKLINUS
REMARK	ALIN-9	VACSTOR
REMODE	RADR-9	R29REMOJ

RENDRAD	RADR-21	RADAREAD
REP40ALM	BURN-2	REP40ALM
REQDATX	DSKY-5	ALOAD ABLOAD ABCLOAD
REQDATY	DSKY-5	BLOAD ABLOAD ABCLOAD
REQDATZ	DSKY-5	TESTNN CLOAD ABCLOAD
REQMM	DSKY-5	MMCHANG
RESAMPLE	RADR-22	RADAREAD
RESET22	PGSR-7	RESET22
RESET57	EXVB-30	WAIT68
[illegible]	[illegible]	[illegible]
RESTORDB	DAPB-18	ALLCOAST DAPDATA2 ENDRO3 TERMASC TERM40 TRMTRACK R23LEM R61C+LO2
RETURNTJ	DAPB-16	Z123COMP ZONE1 ZONE2 ZONE3 ZONE4 RUFLAW2 RUFLAW12 ZONE5
REV83	EXVB-24	COMPDISP
RGOODEND	RADR-18	GOODRAD LRPOS2 LRPOSCAN R77CHECK REMODE RRZERO
RGVGCALC	DESC-6	TTFINCR REDESIG
RHCACTIV	DAPA-22	CHEKSTIK
RMODINV	RADR-8	LUNDESCH REMODE RRDESNB RRDESSM
RNDREFDR	IMUC-15	P06 IMUMON
RODCOMP	DESC-12	P66VERT RODTASK
RODTASK	DESC-12	P66VERT
ROOTLOOP	DESC-17	ROOTLOOP
ROOTPSRS	DESC-17	TTF/8CL
ROPECHK	TEST-7	SELFCHK CNTRLOOP
RPCOMP2	ASCT-12	ASCENT CMPONENT
RR1AX2	RADR-4	RR1AX2 RRTONLY
RRANGLES	RADR-7	RRDESSM
RRANGOUT	DATA-26	DECDSP3
RRAUTCHK	RADR-1	PROCEEDE
RRCDUCHK	RADR-1	RRAUTCHK
RRDESDUN	RADR-11	DODES
RRDESEND	EXVB-5	GOEXTVB
RRDESK2	EXVB-5	VBCOARK
RRDESNB	RADR-6	RRDESK2 R21LEM R21LEM8

RRDESSM	RADR-5	LRS24.1 R21LEM1 60TIMES
RRDOTOUT	DATA-27	DECDSP3
RRGIMON	RADR-1	RRCDUCHK
RRLIMCHK	RADR-7	RRGIMON RRDESSM R23LEM
RRLIMNB	RADR-23	RRDESNB
RRLOSDSP	RNAV-43	DSPRRLOS RRLOSDSP
RRNB	RADR-8	RRDESNB OMEGCALC READRDOT
RRNBMPAC	RADR-8	RRLOSDSP
RROUT	RADR-4	DORROUT RR1AX2 R29DPAS2
RRRANGE	RADR-16	DORSAMP R29RANGE READRDOT R61C+L01
RRRDOT	RADR-17	DORSAMP R22RSTRT R29RDJOB READRDOT R61C+L01
RRSONLY	RADR-4	DORREPOS REMODE
RRTONLY	RADR-3	DORREPOS REMODE PREPOS29
RRTURNON	RADR-2	RRAUTCHK
RRZ2	RADR-12	RRZERO
RRZERO	RADR-12	VBZERO
RRZEROSB	RADR-2	RRTURNON RRZ2
RSTOFGTS	DAPA-29	
RUFLAW1	DAPB-15	TJETLAW
RUFLAW12	DAPB-16	RUFLAW1
RUFLAW2	DAPB-15	TJETLAW
RUFLAW3	DAPB-16	TJETLAW
RVBOTH	SERV-6	AVERAGEG
RVCON	ORBI-4	
S30.1	TRGX-7	P30
S33/34.1	TRGL-5	INTLOOP P33/P73B
S34/35.2	TRGL-8	INTLOOP P35/P75B

S34/35.3	TRGL-9	NTARGCHK
S34/35.4	TRGL-9	S34/35.5 NTARGCHK
S34/35.5	TRGL-3	INTLOOP P35/P75B
S40.1	BURN-15	P40IN P41LM
S40.1B	BURN-16	S40.1
S40.2,3	BURN-16	P40IN P41LM
S40.8	BURN- 14	UPDATEVG
S40.9	BURN-17	UPDATEVG
S40.13	BURN-18	P40SJUNK
S40.132	BURN-19	S40.13 S40.13D
S40.13D	BURN-19	S40.13
S41.1	BURN-12	P41LM DYNMDISP CALCN83 CALCN85
S52.2	ALIN-18	P52D CAL53A
S52.3	ALIN-18	P52B
SAMETYP	ALIN-27	P57OPT1
SBANDANT	EXVB-27	SBANDEX VB64
SBANDEX	EXVB-28	SBANDANT
SCALCHNG	RADR-22	LRHEIGHT RENDRAD
SCALLOOP	DAPA-34	SCALLOOP
SCNDSOL	TRGX-14	CSI/B1 CSI/B2 CIRCL CSI/SOL
SDISPLAY	TEST-2	SOPTION
SELCTSUB	DAPA-27	+XORULGE DOROTAT
SELECTMU	TRGX-6	P34 P74 P35 P75 P33 P73 VN0655
SELECTP	DAPA-17	TRYUORV PJETSLEC SELECTP

SELFCHK	TEST-3	SHOWSUM2 SELFCHK CNTRLOOP PRERRORS NXTBNK STARTSB2 SDISPLAY
SERVEXIT	SERV-4	TERM40 STEERING CALCN85 CALCN83 SERVIDLE ABRTJASK ULLGNOT
SERVICER	SERV-2	READACCS ABRTJASK
SERVIDLE	SERV-5	POODOO
SERVOUT	SERV-4	AVERAGEG DVMON
SET57	EXVB-30	DSP68
SETCOARS	IMUC-15	GLOCKMON IMUCOARS
SETGLOCK	IMUC-9	GLOCKMON
SETIFLGS	ORBI-2	STATINT1 AVETOMID LSR22.3 LSR22.4 UPPSV UPPSV4
SETISSW	IMUC-8	PIPUSE PIPFREE IMUMON C33TEST UNZ2 PFAILOK IFAILOK IMUZERO2
SETLOC	MATX-13	CORFOUND SPECTEST JOBWAKE4
SETMAXDB	DAPB-18	RESTORDB]
SETMINDB	DAPB-18	CUTOFF P41LM TIGNOW R61C+LO2 R23LEM RESTORDB
SETPOS	SERV-19	SETPOS1 SETPOS2
SETPOS1	SERV-19	ASTNRET
SETPOS2	SERV-19	POSGOOD
SETRRECR	RADR-3	STARTDES R29 DORREPOS
SETTRKF	RADR-2	RRCDUCHK RADLITES RRZEROSB
SEUDOPOO	PGSR-6	CANV37
SGNCOM	DSKY-4	POSGN NEGSGN
SHOW	TEST-15	TORQUE PIPJOBB VALMIS SHOW
SHOWSUM2	TEST-2	GOSHOSUM
SIGNTEST	DSKY-5	POSGN NEGSGN
SINE	MATX-1	

SKIPPAXS	DAPA-11	
SLAP1	PGSR-1	VERBFAN
SLEEPIE	TEST-12	ESTIMS PERFERAS
SMTONB	COOR-2	SBANDANT S41.1 FINDCDUW SETPOS R59 ADDGRAV DODES RRDESSM R29DODES R61C+LO2 READRDOT
SNAPLOOP	TELE-4	SNAPLOOP DODNPTR
SNUFFOUT	EXVB-8	GOEXTVB
SOMEERRR	TEST-18	
SOMEKEY	ALIN-7	MARKRUPT
SOMERR2	TEST-18	REDO PIPJOBB EARTHR* PERFERAS
SOPTION	TEST-8	ADRSCHK
SPARCSIN	MATX-9	
SPCOS	MATX-2	
SPECTEST	MATX-13	JOBWAKE4
SPEEDRUN	SERV-15	LANDISP
SPSCONT	DAPB-3	1/ACCS
SPSIN	MATX-2	
SPSRCS	DAPB-17	PURGENCY TJLAW4
SPSSTART	DAPB-17	SPSRCS
SPVAC	MATX-11	
SQRT	MATX-5	
SR30.1	EXVB-20	BOTHPAD V82GON2
STARTDAP	DAPA-5	DAPIDLER
STARTDES	RADR-7	RRDESSM
STARTP47	BURN-13	P47LM
STARTP64	DESC-4	GUILDRET
STARTP66	DESC-11	LUNLAND GUILDRET

STARTSB1	PGSR-2	ENEMA
STARTSB2	PGSR-3	GOPROG2 STARTSB1
STARTSUB	PGSR-2	SLAP1 GOPROG LIGHTSET
STATEINT	ORBI-1	ENDINT
STATINT1	ORBI-1	POOH STATEINT SEUDOPOO
STCLOK3	BURN-9	P40SPOT COMFAIL EXGSUB
STDESIG	RADR-9	BEGDES MOREDES DORROUT CSMINT R21LEM9 LRS24.1
STEER?	DESC-9	AFCCALC1 EXBRAK
STEERING	BURN-7	ULLGNOT
STIKLOAD	DAPB-19	
STMIN-	DAPB-8	ACCTHERE
STOPCLOK	BURN-10	CLOKJOB V99RECYC
STOPRATE	DAPA-32	TRMTRACK ALLCOAST STEERING NOATTCNT DVMON STRTP66A P66HZ NOGO T406ALM STEER? ATMAG
STRTGYRO	IMUC-11	IMUPULSE STRTGYR2 8192AUG
STRTGYR2	IMUC-11	STRTGYR2
STRTP66A	DESC-11	LUNLAND
STSHOSUM	TEST-7	SHOWSUM2 NXTBNK
SUFFCHEK	CONC-8	LAMBLOOP HIENERGY
SUPDACAL	MATX-26	C[illegible]ADRS
SUPERJOB	DAPA-10	PAXFILT BACKP
SURFAGAN	ALIN-11	DSPV6N79
SURFDISP	ALIN-30	
SURFEND	ALIN-11	DSPV6N79
SURFLINE	ALIN-28	SAMETYP R59RET

SURFSTAR	ALIN-11	AVESTAR
SURFSTOR	ALIN-8	YMKRUPT
SVCT3	IMUC-1	SVCT3 T3RUPT STARTSB2 DLY2 WTLST5
SVDWN1	TELE-6	ENDSTATE A-PCHK INTWAKEU P76 ATTACHIT ORBCHGO FAZAB3
SVDWN2	TELE-6	ENDSTATE A-PCHK INTWAKEU FAZAB3
SWCALL	MATX-25	
SWRETURN	MATX-25	
SYNCT4	INTR-5	QUIKDSP SYNCT4
SYSTEST	EXVB-10	GOEXTVB
T3RUPT	MATX-22	TASKOVER Called via program interrupt #3
T4RUPT	INTR-2	QUIKDSP SYNCT4 T4RUPT Called via program interrupt #4
T5RUPT	DAPA-1	Called via program interrupt #2
T6JOBCHK	DAPA-1	DOT6RUPT
TASKOVER	MATX-22	WAKER LONGCYCL VARDELAY
TCGETCAD	MATX-19	DELLOOP
TDISPSET	DESC-10	TTFINCR TTF/8CL
TERMASC	ASCT-12	CUTOFF
TERMATE	DINT-15	RECALTST
TERM40	BURN-9	POSTBURN TIGNOW
TESTLOOP	ORBI-5	ALOADED GOBAQUE ENDSTATE WMATEND CKMID2
TESTNN	DATA-3	MONDO
TESTXACT	EXVB-2	VBCOARK IMUATTCK V47XACT DAPDISP CREWMANU ALINTIME R04 VB64 V67 V73UPDAT V82PERF V83PERF VERB85 V90PERF GOSHOSUM SYSTEST IMUFINEK LRON V89PERF

TFFEL1	EXVB-23	CALCTFF	
THROTTLE	DESC-14	STEER? RODCOMP	
THROTUP	ASCT-6	UPTHROT	
TICKTEST	EXVB-19	V82CALL TICKTEST	
TIG-0	BURN-5	TIG-5	
TIG-5	BURN-4	TIG-30 COMFAIL2	
TIG-30	BURN-4	TIG-35	
TIG-30.1	BURN-4	TIG-35	
TIG-30A	BURN-4	TIG-30.1	
TIG-35	BURN-3	P41SPOT	
TIGNOW	BURN-8	TIGTASK POSTBURN	
TIGTASK	BURN-5	TIG-0	
TIMEDIDL	EXVB-15	UPEND73 UPEND70	
TIMEGMBL	DAPA-31	TRYGTS SPSCONT	
TIMERAD	CONC-1	VN0611	
TIMESTEP	ORBI-6		
TIMETHET	CONC-2	INTLOOP	CDHMVR CSI/B2 ORBCHGO
TIMQGMBL	DAPA-31	TIMEGMBL	
TJETLAW	DAPB-12	PURGENCY TJLAW4	
TJLAW	DAPA-24		
TJLAW4	DAPA-24	TJLAW	
TNONTEST	IMUC-6	IMUMON	
TOBALLA	ATTM-1	REDOMANC ENDMANUV	
TOPSEUDO	DAPA-23	ENTERUV	
TORQUE	TEST-12	PERFERAS	
TOTATTER	EXVB-7	GOEXTVB	

TRIMDONE	DAPB-21	PITCHOFF
TRIMGIMB	DAPB-21	DAPDATA2
TRMTRACK	EXVB-6	R61TEST GOEXTVB R22LEM R23LEM2 P20LEM1 60TIMES P20LEMB7 CSMINT R21LEM R21LEM1 R24LEM R21DISP
TRYGTS	DAPA-27	QRAXIS
TRYUORV	DAPA-11	TRYUORV
TSNEXTP	DAPA-12	CHKVISFZ TRYUORV
TSNEXTS	DAPA-20	RCS +XORULGE
TSTLTS2	TEST-1	VBTSTLTS
TSTLTS3	TEST-1	TSTLTS2
TTFINCR	DESC-5	GUILDRET STARTP64
TTF/8CL	DESC-6	RGVGCALC
TWIDDLE	MATX-19	

ULLGNOT	BURN-4	TIG-30
ULLGTASK	BURN-4	TIG-30 STOPCLOK GOPOST GOCUTOFF
UNZ2	IMUC-9	ISSZERO ENDTNON
UPDATCHK	SERV-8	MUNRETRN
UPDATEVG	BURN-14	STEERING CALCN85
UPDATOFF	EXVB-11	GOEXTVB
UPDATNN	DATA-17	MONDO NVSUB
UPDATVB	DATA-17	CLEAR NVSUB PASTEVB ALMCYCLE ABLOAD ABCLOAD
UPDTCALL	ATTM-8	CONTMANU
UPEND70	EXVB-16	UPJOB
UPEND71	EXVB-16	UPJOB
UPEND72	EXVB-17	UPJOB
UPEND73	EXVB-15	UPSTORE

UPERROUT	EXVB-17	UPEND71 UPEND72
UPJOB	EXVB-16	UPSTORE
UPOUT4	EXVB-17	OHWELL1 OHWELL2 UPVERIFY UPEND73 UPEND70 UPEND71 UPERROUT
UPPSV	RNAV-9	P20LEMC3 R21LEMS
UPPSV4	RNAV-10	UPPSV
UPRUPT	TELE-1	Called via program interrupt #7
UPSTORE	EXVB-15	UPVERIFY
UPTHROT	ASCT-6	GOABORT
UPTMFAST	TELE-1	C33TEST
UPUPDATE	EXVB-14	V73UPDAT
UPVERIFY	EXVB-15	OHWELL2 UPVERIFY
USEPIOS	ORBI-15	INTEGRV ATTACHIT
V1STO2S	MATX-8	
V37	PGSR-5	MMCHANG VERB96
V37BAD	PGSR-5	V37
V37RET	PGSR-14	AVGEND
V37XEQ	PGSR-8	RESET22
V47TXACT	EXVB-6	GOEXTVB
V59GP63	EXVB-30	LRPOS2K
V67	EXVB-8	GOEXTVB
V67CALL	RNAV-49	V67
V70UPDAT	EXVB-8	GOEXTVB
V71UPDAT	EXVB-8	GOEXTVB
V72UPDAT	EXVB-8	GOEXTVB
V73UPDAT	EXVB-8	GOEXTVB
V82CALL	EXVB-17	V82PERF

V82GOFF1	EXVB-18	V82GOFLP
V82GOFLP	EXVB-17	V82GOFLP
V82GON	EXVB-19	V82CALL
V82GON1	EXVB-19	V82GON V82GON2
V82GON2	EXVB-20	V82GON1
V82PERF	EXVB-10	GOEXTVB
V83CALL	EXVB-24	R31CALL
V33PERF	EXVB-10	GOEXTVB
V89CALL	ATTM-11	V89PERF
V89PERF	EXVB-10	GOEXTVB
V89RECL	ATTM-11	V89RECL
V90PERF	EXVB-10	GOEXTVB
V99RECYC	BURN-10	*ENTER
VACSTOR	ALIN-8	YMKRUPT
VALMIS	TEST-14	PERFERAS
VALTCHK	SERV-10	VMEASCHK WSTOR
VARALARM	PGSR-14	ALMXIT
VARDELAY	MATX-24	
VB64	EXVB-8	GOEXTVB
VBCOARK	EXVB-3	GOEXTVB
VBPROC	DINT-14	PROCKEY VERBFAN
VBRELDSP	DSKY-8	CHARIN
VBRESEQ	DINT-14	VERBFAN
VBRQEXEC	PGSR-8	VERBFAN
VBRQWAIT	PGSR-8	VERBFAN

VBTERM	DINT-14	VERBFAN JAMTERM
VBTSTLTS	TEST-1	VERBFAN
VBZERO	EXVB-3	GOEXTVB
VECPNT1	ATTM-3	R61C+L02
VECPOINT	ATTM-3	R60LEM REDOMANC V89RECL
VERB	DSKY-2	CHARIN
VERB69	EXVB-8	GOEXTVB
VERB85	EXVB-10	GOEXTVB
VERB96	EXVB-11	GOEXTVB
VERBFAN	DATA-5	ENTPASO TESTNN MIXNOUN
VERTDISP	DESC-10	DISPEXIT
[illegible]	[illegible]	[illegible]
VGAIN*	BURN-14	FIRSTTME
VMEASCHK	SERV-9	UPDATCHK NOREASON
VN0611	TRGX-1	VN0611
VN0655	TRGX-2	VN0611
VN1645	TRGX-6	P34 P74 INTLOOP P35 P75 P35/P75B P30 VN0655 P32/P72F P33 P73 P33/P73F
VOPENED	DAPA-33	RCSMONIT
[illegible]	[illegible]	
WAIT68	EXVB-30	WAIT68
WAITLIST	MATX-19	
WAKER	MATX-19	DELLOOP
WANTAPS	BURN-4	TIG-30
WCALC	ATTM-6	KALCMAN3
WDAGAIN	DSKY-9	WDAGAIN
WHIMPER	PGSR-12	SERVIDLE POODOO BAILOUT1 ABORT
WITCHONE	DINT-10	OKTOPLAY OKTOCOPY JOBXCHS

WLINIT	RNAV-36	LSR22.3 LSR22.4
WLOOP	CONC-10	WLOOP
WMATEND	ORBI-15	DIFEQ+2
WMATRXNG	EXVB-10	GOEXTVB
WOZERO	TELE-6	DNPHASE1 DNPHASE2 DNDUMPI
WRITEP	DAPA-17	T6JOBCHK PJETSLEC JETSOFF
WRITEU	DAPA-26	T6JOBCHK DOROTAT XTRANS FAILOOP
WRITEV	DAPA-26	T6JOBCHK DOROTAT XTRANS FAILOOP
WSTOR	SERV-10	VMEASCHK
WTLST5	MATX-21	DLY2 WTLST5
XCHSLEEP	DINT-10	JOBXCHS
XMKRUPT	ALIN-7	MARKRUPT
XTRANS	DAPA-26	TSNEXTS AFTERTJ
YMKRUPT	ALIN-7	MARKRUPT
Z123COMP	DAPB-13	TJETLAW ZONE4
ZATTEROR	DAPA-32	NEWDELHI R61C+L02 NOMINIMP CUTOFF P41LM TIGNOW LANDJUNK STARTDAP DETENTCK TSNEXTS ENTERUV PFLITEDB VERTDISP
ZDOTDCMP	ASCT-12	ASCENT
ZEROENBL	DAPA-13	DETENTCK RATERROR
ZONE1	DAPB-13	TJETLAW ZONE4
ZONE2	DAPB-14	TJETLAW ZONE4
ZONE3	DAPB-14	TJETLAW ZONE4
ZONE4	DAPB-14	TJETLAW
ZONE5	DAPB-14	TJETLAW
ZOOM	BURN-7	P40IGN TIG-0 COMFAIL2